

# Dora

Guo Xionghui  
steve@dora.network

Tyler Kot  
tyler@dora.network

Li Xing  
star@dora.network

Version 1.0.9  
August 1, 2018

## Abstract

With the emergence of the new generation of smart contract platform, such as Ethereum, the application of Blockchain technology is greatly promoted, but these platforms can't meet the wider application requirements due to the lack of performance. Dora solves this problem from 3 aspects: vertical expansion, horizontal expansion and consensus algorithm. Vertical expansion works on the parallelism between contracts and contract itself; horizontal expansion improves performance using the technology of Hub-Zone chain. The consensus algorithm tries to find the balance between decentralization and performance based on DPoS, VRF, and BFT. In order to facilitate the migration of existing applications, Dora's virtual machine is fully compatible with EVM. At the same time, Dora can promote the development of the upper-layer applications by free transaction fees. In general, Dora<sup>1</sup> is a highly parallelized, high performance public chain without transaction fees.

## 1 The Current Landscape and Prospect of Blockchain

### 1.1 Review of Blockchain Technology

In September 2008, Satoshi Nakamoto published the whitepaper[1] of BitCoin. In January 2009, the main chain of Bitcoin was formally launched, and it was running safely, creating a new era of encrypt digital asset and pulling the new technology into the public view. In July 2015, the Ethereum[2] was launched. EVM of Ethereum could execute Turing-complete smart contract, it marks the second generation of Blockchain technology. Among these massive applications of Ethereum, encrypt digital token is undoubtedly the most important and popular one. The ERC-20 standard is introduced in the Ethereum community, and the ERC-721 standard is proposed, which builds up the foundation for the issuance and circulation of cryptocurrency, token and non-fungible token. In 2017, the scale of cryptocurrency markets increased by more than fifty times, and the issuance and appreciation of thousands of new cryptocurrencies and Tokens were the main factors. At present, the innovative technology of Blockchain is mainly embodied in the consensus mechanism, block structure and network structure.

The first is the consensus mechanism. PoW consensus algorithm is adopted in Bitcoin at the very beginning. PoW requires a large number of miners to perform hash operations to compete for block accounting, which will lead to a large amount of electricity consumption. In order to overcome this shortcoming of PoW, the new PoS consensus algorithm is proposed, which determines the accounting right of the block based on both the number of the users holding the token and the time of the user's possession of the token, which greatly reduces the power consumed due to the competition for the right to account and improves the efficiency. Both PoW and PoS need miners as well to fight for the right to account. Delegated Proof of Stake (DPoS) refers to the company's operating mechanism, and selects a limited number of agent accounting nodes by the decentralization of voting. These selected agency nodes take turns to generate the blocks according to the rules to avoid the competition to account, thus further improves the efficiency. Regarding these three

---

<sup>1</sup>Invested by the Loopring foundation,an incubator project from Hongkong Loopnest accelerator

consensus algorithms, theoretically, as long as no more than 50% of the nodes are erroneous, the whole network is safe. All of them belong to the algorithm family of indirect consensus, which first must compete for the right to account, then generate blocks, and finally solve the fork problem through a deterministic method (such as the maximum difficulty). In the most of Consortium blockchains, BFT algorithm is adopted, which belongs to the direct consensus algorithm. This algorithm can form a consensus immediately among the participants after finishing a round of operation, and does not have to compete for the right of account in advance and has no forks either, however, it can guarantee the whole network is safe only if no more than 1/3 accounting nodes in the network are erroneous.

The next is the organization structure of the block. A tree structure is adopted in the traditional Blockchain. In such case, a block has one and only one parent block, a global ordered linear ledger of all blocks is formed through the paternity relationship between blocks. In the latest research, multiple parent blocks are allowed, such as the IOTA[3], it mandatorily requires that the new block must point to two parent blocks, and then organize into a DAG, which stipulates a linear ledger with the global order of multiple transactions as a nonlinear ledger that records only partial order relations. Thus the confirmation speed of transaction is improved.

The last is the network structure. The large network is sharded into small networks, and the structure of the Hub-Zone chain is adopted. For example, Ardor[4], Cosmos[5], Asch[6] and PChain[7]. In Cosmos, main chain is called as Hub, and the other sub chains are Zone. Interaction between Hub and Zone is done via IBC(Inter Blockchain Communication) protocol. When a currency is transferred from one Zone to another Zone through Hub, Hub is responsible for maintaining the invariance of its total amount, but Hub is not responsible for verifying the transactions on a single Zone, the Tendermint[8] consensus algorithms is used on both Hub and Zone. PChain is also similar to Cosmos and tries to form a sub chain according to each dApp, but the consensus algorithm is modified, PoS is adopted for both main chain and sub chain. Because there is no record and inspection mechanism for the transaction of the parent chain to the sub chain, the security of the subchain is fully maintained by the miners of the subchain, and it will be a potential security problem in the case of less subchain miners. Considering about this risk, a new miner role is designed in Asch and Ardor to generate and transfer the block records from the subchain to the parent chain. In the design of Asch, it requires the creator of the subchain to upload subchain blocks, while the design of the Ardor does not do the mandatory requirements of the role. By comparison the Ardor is more considerate. After a period of time, snapshots and clipping of the sub chain block data can be done by parent chain to solve its data expansion problem.

## 1.2 Two Problems to be Solved in the Current Blockchain: Performance and Storage

With the increase of demand, a very urgent problem faced by Blockchain is scalability. At present, the low throughput capacity of Blockchain (about 7 transactions per second in Bitcoin, about 15 transactions per second in Ethereum) is not enough to meet the needs of global financial transactions. By comparison, 56000 TPS transactions can be handled in Visa, while Alipay has achieved peak 200000+TPS transactions in November 2017. Currently, the scalability problem based on Blockchain network has brought great limitations to its applications. For example, since the smart contract function is supported by the Ethereum, many projects run directly in the Ethereum as dApp, but with more and more projects running on the chain, the load becomes increasingly heavier, and just only one project like "Cryptokitties" made the transfer function difficult to succeed in the entire Ethereum. How to improve Blockchain throughput without affecting security and decentralization is still to be explored.

As time goes on, another problem faced by Blockchain is data explosion. As shown in figure 1 (data source, <http://bc.daniel.net.nz>), By the end of Feb. 2018, the current total block size is more than 150G in Bitcoin, and the new node needs to take 14 days to synchronize with all the blocks; in Ethereum, the size of the block is more than 650G, and the new node needs to take 8 days to synchronize with all the blocks, and is still increasing by 145M per day.

## 1.3 The Prospect of Dora

Dora introduces a brand new Blockchain architecture designed to meet the needs of global business activities with Blockchain technology. Inspired by the pipeline model and branch prediction systems that are currently applied in many

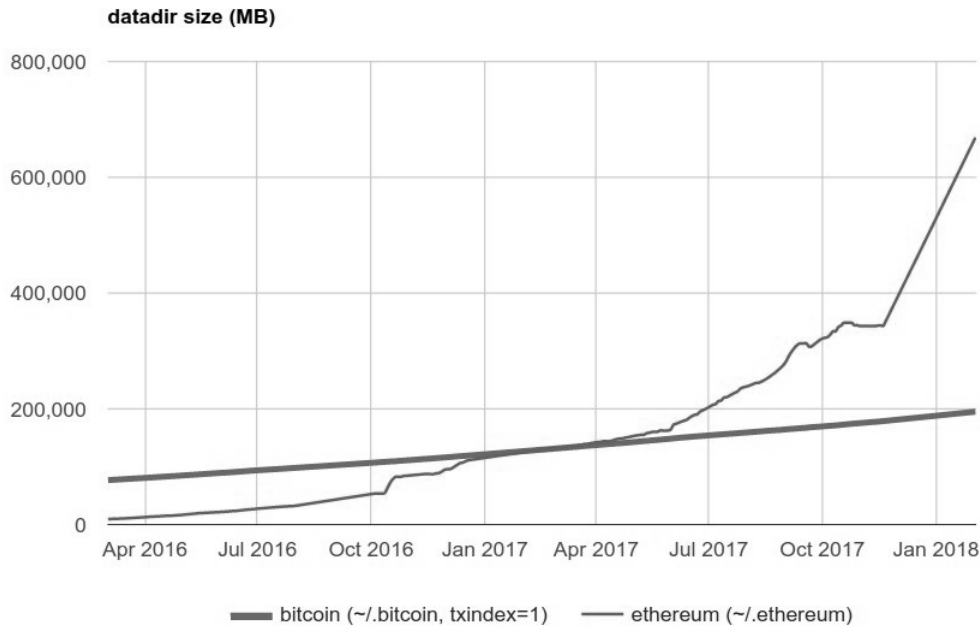


Figure 1: Blockchain Data Explosion

applications in CPU, our core members are trying to apply these technologies and ideas into Blockchain area, creating a unique solution dedicated for the scalability problem of Blockchain. The solution can substantially increase the applicable scope of the Blockchain without compromising its security and decentralization characteristics.

Our design will drive the Blockchain into the next generation, increasing the current TPS by thousands or even tens of thousands of times, while maintaining security and decentralization. The network we are building now will have no congestion and will support node snapshots to facilitate new nodes to join the network quickly. Ultimately, the goal of Dora is to be a public chain that everyone can use and everyone can use easily.

## 2 Dora's Vertical Expansion: Parallelization

There are two types of expansion solution in traditional Blockchain: state channel and Multi Chain sharding. Lightning network [9] deploys state channel technology to alleviate the scalability problem of Blockchain. The basic idea is that for the frequent transactions between a fixed group of parties, once all parties complete the transactions, one of them will just publish the final result without having to generate multiple transaction records on the chain (essentially, it decreases the storage of the intermediate results). However, the lightning network is only suitable for frequent transactions among a fixed group of the parties, and if the transaction goals of the user are random and the transaction occurs occasionally, then it will lead to poor efficiency. Multi chain sharding is a horizontal expansion technology, it increases the number of chains or segments, so that transactions are dispersed to achieve final expansion, but this is usually accompanied by security problems on subchains or subsections.

Serialization mechanism is taken as the assumption by all of these solutions when in transaction, because the serialization of the transaction enables each miner node to perform and verify independently. The same output result obviously can be obtained after the same serial operation starting from the same genesis block.

The TPS of the system is always constrained by the performance of a single node definitely because the serialization of transactions. However, we need to think about it carefully, is serial execution the only option for transactions? Can we do parallel processing at the transaction level in order to design a vertical expansion technology?

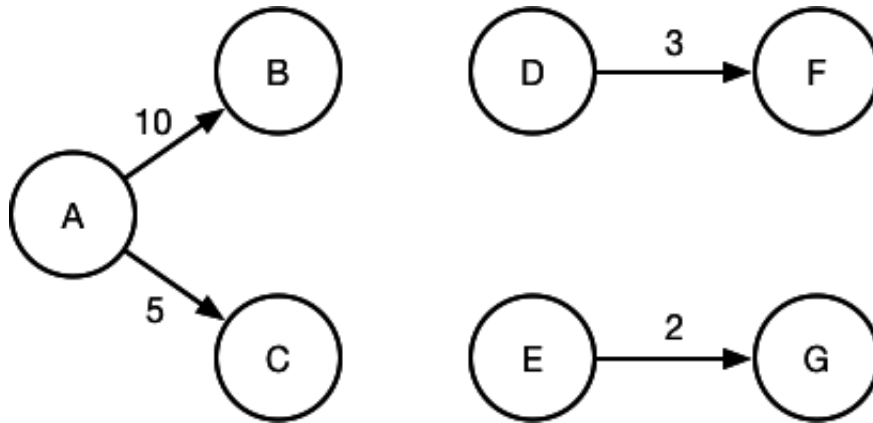


Figure 2: Common Transfer Transaction

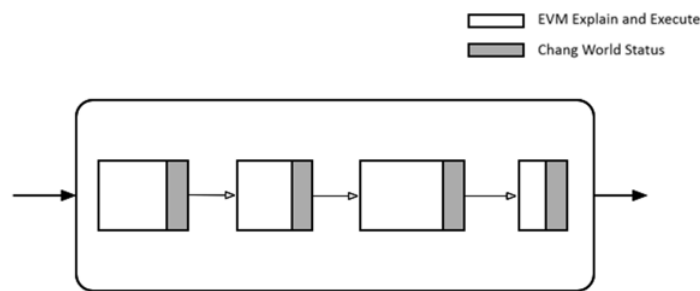


Figure 3: Serialization of Smart Contract

## 2.1 Parallel Transactions for Common Transfer

The account is regarded as a node. When transaction occurs from account A to account B, an arrow is added between node A and node B, the number next to the arrow indicates the transaction amount. Thus, a list of transactions within a block can form a diagram. Let's look at an example, assuming that a block contains the following transactions. {A transfers 10 tokens to B, and transfers 5 tokens to C, D transfers 3 tokens to F, E transfers 2 tokens to G }, As shown in figure 2.

From the diagram, It is obvious that the whole transaction set is divided into three sub connected graphs. {A,B,C}, {D,F}{E,G}. Transactions in the same connected graph can only be executed sequentially because of dependencies. But in fact, transactions in different connected graphs can be executed in parallel. For example, the execution sequence between D to F and E to G is different only in the middle temporary state, but it does not affect the state that the block eventually reaches, and in the real Blockchain world, we can easily find a set of transactions without dependency.

## 2.2 Parallel among Smart Contracts

For common transfer transactions, it is easy to divide the subconnected graph according to the account involved, but what if a transaction is a call to a smart contract? First let's take a look at the example shown in the figure3 how smart contract runs on chain. The execution logic of a smart contract can be considered as two parts: the first part is EVM explanation and execution marked by white, and the second part is what truly change the world state and marked by gray. The grey part is the key component which influence the final world status. A miner will collect the world state of the external account that the smart contract relies on during an execution process of the smart contract, and can also collect the world state of the external account that is affected ultimately.

The two smart contracts are considered as connected as long as there are intersections in account sets which rely on

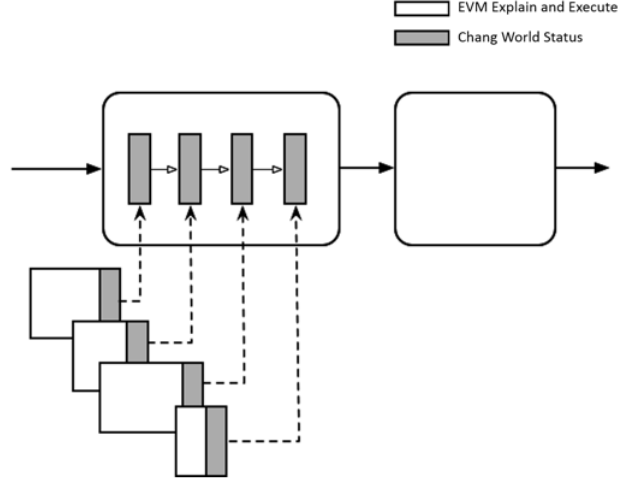


Figure 4: Parallelization of Smart Contract

and change by these two smart contracts. In this way, all the smart contract transactions in a block can be divided into different connected subgraphs, and each subgraph can be executed in parallel. As shown in figure 4, Assuming that the calls of these four smart contracts are all in different connected subgraphs, once the parallelization is executed, the generation speed of the block can be greatly enhanced, thus improving the TPS of the entire system. This parallelization can be achieved either by single machine multithread or by parallel process with multiple machines.

By analyzing the block data between block height 5592867 and 5609843 of Ethereum, the following conclusions can be drawn: 53.7% of the total transaction volume are smart contract transactions. The average execution time of each smart contract transaction is 1.29ms, the maximum execution time is 51.1ms, and the shortest is 0.14ms. That is to say, if the smart contract call is executed serially, the maximum TPS of the system can only reach 775. When executed in parallel, theoretically, there is no limitation.

The vertical expansion technology is essentially similar to the branch prediction in the CPU system, that is, the branch is executed in advance by pipelining. If the prediction is correct, then the subsequent pipeline only needs to be executed in sequence, thus greatly improving the parallelism of the system. Applying this idea in the Blockchain system is proposed originally by Dora. Theoretically, this idea can be applied to any current public chain, thus increasing the TPS of the whole system.

A formal definition of the algorithm is given below.

### 2.2.1 Symbol Definition

**Definition 2.1.** First, let's define some of the concepts which will be used later.

*Space of Smart Contract:*  $\mathbb{T}$

*Block:*  $B_t = \{T_1..T_{n_t}\} \in 2^{\mathbb{T}}$ , it is a set of smart contracts.

*Address Space of Account:*  $\mathbb{A}$

*Status Space of Account:*  $\mathbb{L}$

*Space of World Status:*  $\mathbb{L}^{\mathbb{A}}$ , It is a set of mapping from account address space to state space.

*Function of World State Transition:*  $\Gamma : \mathbb{L}^{\mathbb{A}} \times 2^{\mathbb{T}} \rightarrow \mathbb{L}^{\mathbb{A}}$ , e.g.  $\sigma_t = \Gamma(\sigma_{t-1}, B_{t-1})$  and  $\sigma_t, \sigma_{t-1} \in \mathbb{L}^{\mathbb{A}}$  It implements a set of smart contracts based on the current world state and updates the world status.

*Serial Execution:* For any pair of contracts in a group of contract, if all operations of one of the contracts are prior to another one, then the execution of this pair of contracts is called serial execution. i.e, the atomic serialization.

*Security:* For a parallel execution of a pair of contracts, if the change result of the world status is equivalent to the result when in some of the serial execution, it is said that the parallel execution is secure. i.e, given the parallel state transition

function  $\Gamma_p$  and the set  $\mathbb{S}$  of all serial transition functions, it is necessary to meet  $(\exists S)(S \in \mathbb{S}) \wedge (S(\sigma, B_{t-1}) = \Gamma_p(\sigma, B_{t-1}))$   
 Set of reading account of smart contract:  $T_r \in 2^{\mathbb{A}}$   
 Set of writing account of smart contract:  $T_w \in 2^{\mathbb{A}}$   
 Status of account:  $L(T_r) \in 2^{\mathbb{L}}$

### 2.2.2 Algorithm Flow

The following is the flow of the SSP algorithm 1, which can improve the performance of the system and reduce the execution time on the premise of ensuring security. It should be noted that common transfer can be regarded as a simplified smart contract, so SSP is also applicable.

---

#### Algorithm 1: Safe Speculative Parallelization(SSP)

---

```

Procedure proposer.Propose()
  repeat
    Receive  $T \in \mathbb{T}$ ;
    Execute smart contract  $\sigma = \Gamma((T_r, L(T_r)), T)$ ;
    Send the tuple  $(\sigma, T_r, T_w)$  to validator;
  until exit;
  return;

Procedure validator.Validate()
  Set of valid contracts  $\mathbb{V} \leftarrow \emptyset$ ;
  Delta world state  $\Delta\sigma \leftarrow \text{null}$ ;
  repeat
    Receive the tuple  $(\sigma, T_r, T_w)$  from proposer;
    for each valid contract  $V \in \mathbb{V}$  do
      if  $V_r \cap T_w \neq \emptyset$  or  $V_w \cap T_r \neq \emptyset$  then
        Abort  $T$ ;
      end
    end
    if  $T$  is not aborted then
       $\mathbb{V} \leftarrow \mathbb{V} \cup T$ ;
       $\Delta\sigma \leftarrow \Delta\sigma + \sigma$ ;
    end
  until timeout or the  $|\mathbb{V}| > n$ ;
  Update world state  $\sigma_{t+1} \leftarrow \sigma_t + \Delta\sigma$ ;
  return;

```

---

## 2.3 Instruction Level Parallelism in Smart Contracts

Dora is fully compatible with the Ethereum EVM instruction set and performs performance optimization at the implementation level. First, Dora performs optimization when compiling, including loop unrolling, instruction merging, etc. Second, Dora performs optimization when in runtime, including JIT and so on. At the same time, it allows programmers to mark the code (Annotation) to communicate with compiler which parts can be optimized in parallel.

## 2.4 Adjusting the Size of the Block Dynamically and Block Compaction

In order to improve the TPS of the entire network, Dora will adjust the size of the block dynamically according to the current network congestion, and we will also introduce a block compaction technology, As shown in figure 5:

In a traditional Blockchain, a block usually consists of a block header and a series of transaction lists. E.g, in Ethereum, a transaction takes about more than 200 bytes. When a node broadcasts a block, it will pack the original transactions into the block and send it out, which is a complete waste of network bandwidth resources, since each original transaction

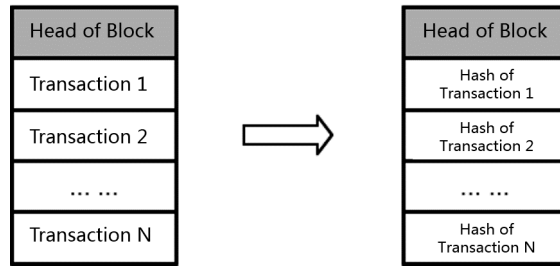


Figure 5: Block Compaction

has been broadcast to every node of the network before that, in theory we can use a compression mode, that is, put only the Hash value of the transaction in the block (only 32 bytes). When the node receives the Hash list of block transactions, only if the corresponding transaction data is not found locally, then go to other nodes to ask for the data. In this way, the bandwidth of the network can be saved about 7 times, which is almost as much as that of TPS.

### 3 Dora Horizontal Expansion: Hub-Zone Chain

Dora also supports horizontal expansion of the Blockchain in the way of Hub-Zone chains (Figure 6). By dividing the set of smart contracts into separate subsets (basically divided by dApp), each of these subsets runs on a subchain independently, allowing the transaction of the master chain to be diverted, which is equal to performing sharding processing on the input set.

The Hub-Zone chain uses an account model, account can be shared on both Hub and Zone chain. However, for the same account, the world status will be recorded on the different chains separately. E.g., a user can create an account with a wallet, which can be used to receive and send the transfer transactions of the Hub chain and Zone chain, but the number of tokens owned by the Hub chain is recorded on the Hub chain, Zone chain is same as well. At the same time, a special cross chain transaction(CrossChainTransaction) is supported to transfer tokens between Hub chain and Zone chain. The Hub chain supports multiple currency transfers directly.

#### 3.1 Zone Chain Security

As shown in figure 7, the Dora master chain supports a special transaction. The miners that generate blocks in the Zone chain are responsible for packing the sub block information (ChildChainBlock) that have not been recorded on the master chain and transferring it to the master chain, but it should be noted that the complete Zone block info will not be recorded on the Hub chain, only head info of Zone block will be recorded. These data are only records. The miners of the Hub chain are not responsible for verification when they perform packaging. In order to ensure the security of Zone chain data, Dora designed a new supervisor role.

The supervisor will monitor all transactions of the Hub chain and Zone chain simultaneously. When a Zone chain block package message(ChildChainBlock) is detected, it will perform verification referring to the transaction data previously received from Zone chain. Once the problem is found for Zone chain, the evidence will be submitted to the Hub chain, and the Hub chain miners will check the evidence. If the evidence is valid, the miners of this particular Zone chain will be punished.

#### 3.2 Data Clipping and Snapshot

To solve the problem of data explosion, the Dora Hub chain supports the clipping of the Zone chain block information, and the data recorded on the Hub chain only retain the latest block data(proposed 2048 blocks in the early stage) of Zone chain , and the data prior to that can be removed from the Hub chain.

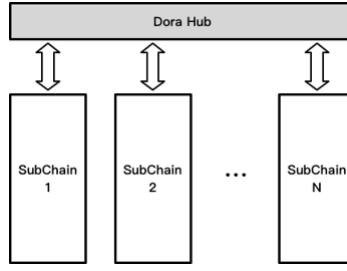


Figure 6: Hub-Zone Chain

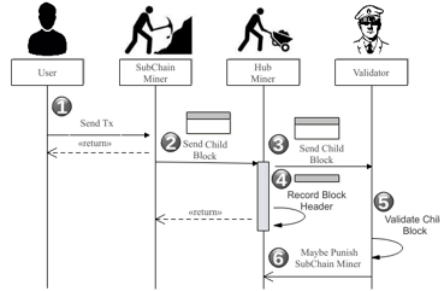


Figure 7: Zone Chain Security

The Hub chain also supports the snapshot, and a snapshot contains the world status information of all the accounts of the current block, so that a new node can be synchronized quickly from a snapshot block, and even a secondary node can delete all the block data before the snapshot.

### 3.3 Cross Chain Token Transaction Between Hub and Zone Chain in Dora

Generally, a token will circulate only within a particular subchain, but Dora designs a special CrossChainTransaction that allows tokens in the same account to be transferred to each other between the Zone Chain and the Hub chain.

```

CrossChainTransaction: {
    blockno: block number, rawtransaction in which block in which specific block of this transaction
    blockhash:Block Hash corresponding to blockno
    merklepath:Merkel tree path to verify if the rawtransaction is in block( blockno).
    rawtransaction:{
        owner: "0xfbc2a4...ed",
        symbol: UT,
        value: 100,
        direction: 0 or 1 0 for from Zone Chain to Hub Chain 1 for from Hub Chain to Zone Chain
    }
}

```

As shown in figure 8, how the token UT is transferred from the Zone chain to the Hub chain.

Suppose that account A transfers 100 UT to the Hub chain, when cross chain transactions occur:

- 1.First A launches a special transfer transaction on the UT Zone chain, which requesting to transfer the 100 UT from the Zone account A to the Hub account A.
- 2.When the transaction is confirmed on the Zone chain, then awaiting miner of the Zone chain records the block information containing the transaction into the Hub chain.
- 3.Then on the Hub chain, A initiates a CrossChainTransaction from the Zone chain to the Hub chain, with transfer transaction in step 1, the block height and the Merkle path for Hub chain to verify;



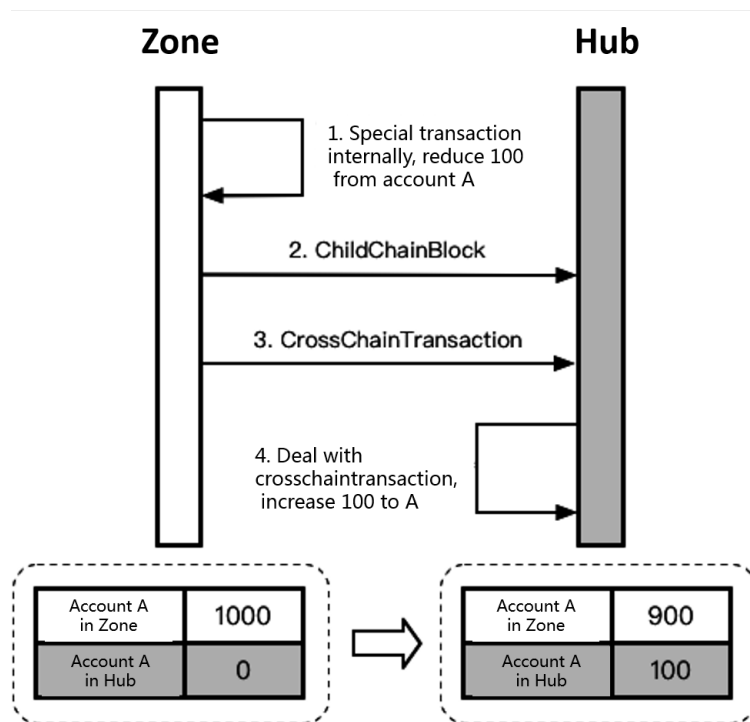


Figure 8: Token transferring between Hub Chain and Zone Chain

4. The miner of the Hub chain verifies the CrossChainTransaction. If verified, it will be packaged into the Hub chain block, and the state record of the Hub chain account A is updated to transferred in 100 UT.

On the contrary, the token can be transferred back to the Zone chain from the Hub chain, the advantage of which is that the tokens are retained in the same account during the entire transfer process and do not require additional controlled accounts to be introduced as other cross chain solutions.

At the same time, a special TokenSwapAction is designed on the Dora Hub chain, which allows to perform atomic exchange between two types of tokens, and the signature information of the two parties is required to carry, and determine whether the exchange can be performed after checks the balance of the two parties.

```
TokenSwapAction: {
  taker: "LRC",
  takerAddr: 0xfbc2a4...ed
  takerValue: 1000,
  maker: "RDN",
  makerAddr: 0xabc35e...ef
  makerValue: 500,
  takerOrderWithSignature
  makerOrderWithSignature
}
```

Through this atomic token exchange TokenSwapAction and cross chain operation CrossChainTransaction, it is easy to implement cross chain transactions between the Hub chains and Zone chain (Figure 9). Dora will create some special bridging sub chains which is able to insert other existing public chains, so that the cross chain transactions of existing chains can be supported as well.

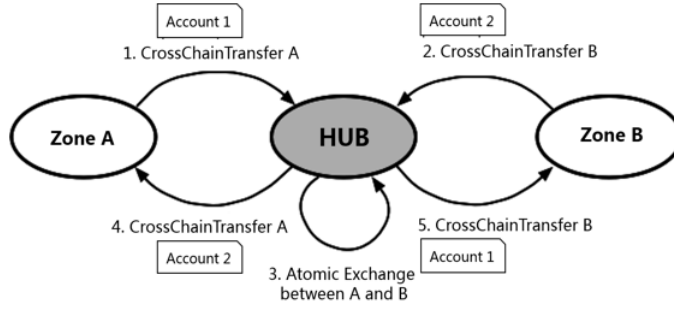


Figure 9: CrossChain Transaction

## 4 Consensus Algorithm DVBC

### 4.1 The Idea of Algorithm

Considering the balance between high performance and high security requirements, Dora proposed a hybrid consensus algorithm DVBC (Delegate Verifiable BFT Consensus). DVBC is based on DPoS[10], VRF[11] and BFT[8]. Assuming that there are  $M$  nodes,  $N + 1$  nodes should be selected in total, and the initial value of  $N$  is 21. At the beginning of each round of consensus, Dora first uses the first layer algorithm DPoS to select the number of top  $N/3$  nodes from the number of  $M$  nodes according to the size of the vote, and these nodes will automatically become the bookkeeper of the current round. In second layer algorithm, the remaining nodes  $M - N/3$  are selected as candidates. In these nodes, the VRF algorithm is utilized, the  $2*N/3+1$  candidate nodes are selected randomly as the bookkeepers. In the last layer, we use the BFT[8] algorithm to reach the consensus of the blocks among these selected  $N + 1$  bookkeepers. This ensures that as long as there are no more than  $N/3$  fraud nodes in the  $N + 1$  nodes, the correctness of the blocks is guaranteed and there is no fork at all. Obviously, this algorithm can prevent super-nodes from defrauding jointly. Additionally, because the candidate nodes also have the probability to be selected as the bookkeeper, this economic incentive mechanism motivates more nodes to be selected as candidate nodes, thereby improving the security and robustness of the network.

Considering each dApp has the different requirements for security and performance, it is allowed in Dora to specify which consensus algorithm for subchains to be used when creating subchains: DPoS+VRF, BFT, or both.

### 4.2 Formal Description

Algorithm 2 describes the overall process of DVBC. Among them, DPoS, VRF and BFT are respectively described by algorithm 3, 4 and 5.

---

#### Algorithm 2: DVBC

---

```

Function DVBC()
  while true do
    list of delegates  $L = \emptyset$ ;
     $L = \text{DPoS}(N/3, \text{candidate\_votes})$ ;
     $L.append(\text{VRF}(2 * N/3 + 1, \text{candidates}))$ ;
     $\text{BFT}(L)$ ;
  end

```

---

DPoS provides a sampling algorithm based on the voting ratio as the probability of candidate nodes being selected.

The VRF algorithm first uses the public key of each candidate node and the Hash value of the previous block as the source information. The information is given to the VRF Hash function to calculate the new Hash value, and the  $2*N/3+1$  nodes are sorted by the value. The advantage of using VRF is that it can quickly form consensus, find candidate nodes,

---

**Algorithm 3: DPoS**

---

**Function** DPoS( $m, candidate\_votes$ )  
list of delegates  $L = \emptyset$ ;  
 $status = candidate\_votes$ ;  
 $sum = summation(candidate\_votes)$ ;  
**for**  $i = 0; i < m; ++i$  **do**  
     $status += candidate\_votes$ ;  
     $top = sort(status).top\_index()$ ;  
     $L.append(top)$ ;  
     $status(top) -= sum$ ;  
**end**  
return  $L$ ;

---

---

**Algorithm 4: VRF**

---

**Function** VRF( $m, candidates$ )  
list of delegates  $L = \emptyset$ ;  
 $info = candidates.pub\_key + previous\_block\_hash$ ;  
 $candidates\_hash = VRF\_hash(sec\_key, info)$ ;  
 $topn = sort(candidates\_hash).topn(m)$ ;  
 $L.append(topn)$ ;  
return  $L$ ;

---

and be able to be verified.

The BFT algorithm is described by calculus  $\pi$  [12][13][14]. The basic syntax of calculus  $\pi$  is shown in figure 10. The advantage of using calculus  $\pi$  is that it is easy to analyze the feature of the algorithm, such as security and flexibility, and can also use software to do formal verification for the algorithm. The algorithm consists of 4 states: propose, prevote, precommit and commit.  $P$ ,  $V$  and  $C$  are the abbreviations of propose, prevote and precommit, which correspond to the 3 main functions in the algorithm respectively. For convenience to describe recursion relations, we also define subfunctions PrevoteSub and PrecommitSub, which are simplified as VS and CS. For getProposal function, it will return  $p$  directly when  $p$  is not empty, otherwise integrate the latest transaction information into a proposal and return it. The function  $round\_robin(k) = k \bmod N$  indicates selecting a process responsible for propose according to the current turn, similar to the role of leader. The consensus protocol between  $N$  nodes can be expressed as  $Consensus ::= \prod_{i=1}^N Y_i$ . The status of each node is recorded as  $s = \{k, P, v\}$ .  $k$  indicates the round.  $p$  represents the block information of the proposed round, and  $v$  represents the voting information of all rounds.  $v'_k$  and  $v''_k$  respectively represents prevote and precommit voting information in the  $k$  round. The process transfers information through a communication channel, where  $cp_i$ ,  $cv_i$ ,  $cc_i$ , and  $cw_i$  represent the communication channels for used for  $i$ process for propose, prevote, precommit, and commit.

The function Propose first judges whether it is the current proposer or not, if so, package the latest transaction to the propose PR to send PR through the  $cp$  channel to other processes; and enter the V state. If not, and there is propose already, it will also enter the V state; otherwise, accept the propose from the  $cp$  channel or go into the V state after the timeout.

The function Prevote will broadcast the received proposals to other processes through the  $cv$  channel and accept the proposal from the  $cv$  channel of other processes; then create a shared channel  $C$ , and send the received proposals through  $C$ ; and enter the PrevoteSub subfunction. In the PrevoteSub function, if the same proposal is received by more than  $2/3$  processes, it enters the C state. Otherwise, if the received proposal of the process is over  $2/3$ , but not consistent, it will enter C state carrying the empty proposal; otherwise, if the proposal of the  $2/3$  process has not been received yet, it will be handled according to concrete situation. After read the information of the channel  $C$ , it will keep waiting if the message is old; if it is the current round information, the voting information is merged and it will keeps waiting; if its round is backward, it returns to the P state and overtakes the current round.

---

**Algorithm 5:** BFT

---

**Procedure Propose()**

$P_i^{k,p,v} ::=$  if  $i == \text{round\_robin}(k)$  then  
     $\overline{cp}_i < pr > |V_i^{k,pr,v}$ , where  $pr = \text{getProposal}(p)$   
    else if  $p \neq \emptyset$  then  
         $V_i^{k,p,v}$   
    else  
         $c_{\text{round\_robin}(k)}(pr) \cdot V_i^{k,pr,v} + \text{timeout}_{\text{round\_robin}(k)} \cdot V_i^{k,\emptyset,v}$

**Procedure Prevote()**

$V_i^{k,p,v} ::= \overline{cv}_i < p > |(\text{new } c)(\prod_{j=1}^N cv_j(pr) \cdot \overline{c} < cv_j, pr > |VS_i^{k,p,v}(c))$

**Procedure PrevoteSub()**

$VS_i^{k,p,v}(c) ::=$  if  $\max_b(|\{pr \in v_k : pr.block == b\}|) > \frac{2}{3}N$  then

$C_i^{k,b,v}$

    else if  $|v_k| > \frac{2}{3}N$  then

$C_i^{k,\emptyset,v}$

    else

$c(cv, vote)$ . if  $vote.round < k$  then

$cv(pr) \cdot \overline{c} < cv, pr > |VS_i^{k,p,v}(c)$

        else if  $vote.round == k$  then

$VS_i^{k,p,vote \cup v}(c)$

        else

$P_i^{vote.round,p,vote \cup v}$

**Procedure Precommit()**

$C_i^{k,p,v} ::= \overline{cc}_i < p > |(\text{new } c)(\prod_{j=1}^N cc_j(pr) \cdot \overline{c} < cc_j, pr > |CS_i^{k,p,v}(c))$

**Procedure PrecommitSub()**

$CS_i^{k,p,v}(c) ::=$  if  $\max_b(|\{com \in v_k : com.block = b\}|) > \frac{2}{3}N$  then

$\overline{cw}_i < b >$

    else if  $|v_k| > \frac{2}{3}N$  then

$P_i^{k+1,\emptyset,v}$

    else

$c(pc, vote)$ . if  $vote.round < k$  then

$pc(com) \cdot \overline{c} < pc, com > |CS_i^{k,p,v}(c)$

        else if  $vote.round == k$  then

$CS_i^{k,p,vote \cup v}(c)$

        else

$P_i^{vote.round,p,vote \cup v}$

---

$P ::=$	$0$	<i>empty summation</i>
	$P_1 P_2$	<i>composition(parallelization)</i>
	$P_1 + P_2$	<i>summation(nondeterministic choice)</i>
	$(new\ x)P$	<i>channel creation</i>
	$\alpha.P$	<i>prefix form</i>
	$\tau.P$	<i>silent action</i>
	$x(y).P$	<i>input</i>
	$\bar{x} < y > .P$	<i>output</i>
	$(x).P$	<i>restriction</i>
	$timeout_i.P$	<i>timeout of process i</i>
	$P^s(y)$	<i>function to which pass variable s and y</i>
$\sum_{i=1}^n P_i ::=$	$P_1 + P_2 + .. + P_n$	<i>multiple summation</i>
$\prod_{i=1}^n P_i ::=$	$P_1 P_2 .. P_n$	<i>multiple composition</i>

Figure 10: Description of Calculus Symbol  $\pi$

The function Precommit broadcasts proposal to other processes from the  $cc$  channel and accepts a proposal from the  $cc$  channel of other processes; meanwhile, the channel  $C$  is created to send the received information . Similar to function Prevote , if the same proposal is received from more than 2/3 processes, the proposal is formally submitted; otherwise, if the proposal is inconsistent, go to the state of  $P$  and start the next round; otherwise, continue to wait, if you find that your round is backward, then back to the state of  $P$ .

### 4.3 Parallelization of BFT algorithm

For four states of a block, propose, prevote, precommit, commit, assuming their corresponding time are  $T_1, T_2, T_3, T_4$ , the time needed to generate and confirm  $N$  blocks by serial execution is  $N * (T_1 + T_2 + T_3 + T_4)$ .

From the point of view of pipeline, the four states can be executed in an orderly and parallel manner. Ideally, as shown in Figure 11, the time required to generate and confirm  $N$  blocks in pipeline mode is  $N * T_1 + T_2 + T_3 + T_4$ . When  $N$  is large enough, after this pipelining operation, the time is approximately  $N * T_1$ .

## 5 The Feature of Dora

### 5.1 Horizontal and Vertical Expansion

Dora supports not only conventional Multi Chain horizontal expansion, but also unique vertical expansion mechanism based on branch prediction , so it can achieve higher TPS than other systems.

### 5.2 Consensus Algorithm Parallelization

Dora also fully utilized the concept of pipelining in the design of the consensus algorithm, so that the block generation and confirmation process are parallelized, therefore it can also enhance the TPS.



## 6 Economic Model of Dora

Dora introduces DNT (DoraNetworkToken) tokens, it is used to activate the internal ecology, and is the currency circulating in its own system. The initial offering of DNT is 1 billion.

Referring to the GAS cost model of Ethereum, Dora will internally calculate the GAS consumed by the smart contract and the ordinary transfer, and then multiply it by the GAS price that the user is willing to pay, then calculate out the DNT which will be used for the final transfer. However, Dora users will not actually consume these DNT. The DNT will be locked first and then released back to the user account in 3 days. When the tokens are locked, they can not be used for trading.

It is free for ordinary users to initiate transactions, but in order to avoid zero cost trading attacks, the system will calculate the GAS costs that the account can use free of charge based on the number of tokens locked in the account, and the GAS costs in a transaction will be actually deducted when its cost exceeds the free quota. Taking 3 days as a cycle, the maximum free quotas for account  $U$  in this cycle are as follows:

$$F_u = \frac{U's \text{ locked tokens}}{\text{the locked tokens of all accounts at present}} * \text{The total GAS cost accumulated to the current block in the cycle}$$

The more tokens are locked up, the more free quota there will be. However, when user's free quota is consumed up, the user may immediately transfers the balance to another account to get additional free quotas quickly. In order to avoid this problem, Dora designs a set of slow-start mechanisms, for each account in a cycle the free quota will increase over time on the basis of the average value of previous  $k$  transactions until the maximum value is reached  $F_u$ .

$$C_u = \frac{\sum_{n=1}^k Gas_n}{k} * (\text{Height of current block} - \text{The block height of user's previous } k \text{ times of transaction} + 1)$$

A transaction has to meet the above two conditions at the same time and can be free, that is less than  $\min(F_u, C_u)$ , otherwise the transaction GAS costs will be destroyed immediately, Dora bookkeeper does not rely on GAS fee to gain profit, Dora will issue additional 1% to 5% of DNT annually, the bookkeeper could earn rewards through the increase of DNT.

The specific annual increase is determined by the ecological status of the whole public chain. If the economy is overheated, it will consider decreasing the volume of additional sales, otherwise increasing. The economic situation is represented by the dApp index. It is determined by factors such as price changes, total transaction volume and per capita transaction volume of main dApp in the chain,, and will continue to add more factors. The function relationship between the exponential  $I$  and the increment  $A$  is (shown as in figure 12):  $A = 0.01 + 0.04 * \frac{1}{1 + (\frac{1}{1-I})^\alpha}$ , Among them,  $I, A \in [0, 1], \alpha = 3$ . Dora divides the economic situation into 3 types: overheating, overcooling and moderate. When the economy changes from moderate to overheating or overcooling, the economy is pulled back to a moderate state through a faster response than linear mechanism. When the economy changes from overcooling to moderate, Dora will maintain a higher volume of issuance to prevent economic retreat. Similarly, when the economy changes from overheating to moderate, it will prevent the return to overheating by maintaining a lower volume of increase.  $\alpha$  is a super parameter, which controls the speed of reaction.

Dora allows nodes to set up their own share proportions, and motivates token-holders to vote and select bookkeeping nodes through market mechanism. Suppose that the number of rewarded tokens in each block is  $W$ , and a node is willing to take proportion  $p$  of tokens as a voting reward, then when the node generates a block, all the users who vote for the node will share the  $W * p$  award in proportion to the number of votes. Specific to each voting user is  $I_u = \frac{V_u}{\sum_{k=1}^N V_k} * W * p$ . The award for the node consists of two parts, one is voting award because as a common user it voted itself, and the other is a reward obtained by the bookkeeper. The specific calculation formula is:

$$I_u = \frac{V_u}{\sum_{k=1}^N V_k} * W * p + W * (1 - p)$$

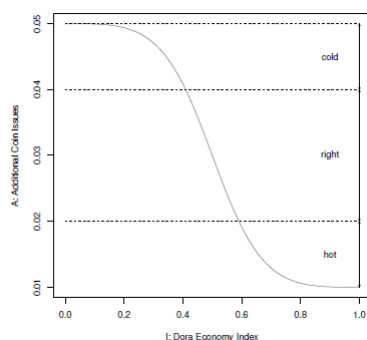


Figure 12: The relationship between index  $I$  and the increment  $A$

## 7 Summary and Further Work

Dora proposes a SSP algorithm to parallelize smart contracts. It fully utilizes the parallelism at the instruction level, and designs a DVBC consensus algorithm that takes both security and efficiency into account, thus making the whole system capable of achieving a performance of more than a million TPS. In addition, Dora is fully compatible with Ethereum EVM, which reduces the transplant costs of dApp developers, and the free trading model can allow the general public to enter the Blockchain area at low cost as well.

Technology has always been progressing, and Dora will continue to focus on the latest research directions in the area of the Blockchain, absorbing and improving existing research results, such as more efficient and secure consensus algorithms. In addition, some trusted smart contracts will be introduced to accelerate the efficiency of parallel execution by specifying preconditions in advance. Dora will also consider to dynamically extend the type of transactions supported on the chain based on the need for upper level applications.

In addition to supporting EVM, Dora also plans to support WebAssembly, so that more developers can quickly develop dApp on Dora.

## Appreciation

We sincerely appreciate the investment by Loopring foundation and the Hongkong Loopnest incubator into our project. Many of the ideas in this article come from Mr. Wang Dong, the CEO of Loopring[15]. He also provided a lot of help and guidance in this thesis writing, and we would like to express our heartfelt thanks to him.

## References

- [1] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. <https://bitcoin.org/bitcoin.pdf>, Dec 2008.
- [2] Vitalik Buterin. Ethereum: A next-generation smart contract and decentralized application platform. <https://github.com/ethereum/wiki/wiki/White-Paper>, 2014.
- [3] Serguei Popov. The tangle. [https://www.iotatoken.com/IOTA\\_Whitepaper.pdf](https://www.iotatoken.com/IOTA_Whitepaper.pdf).
- [4] Ardor whitepaper. <https://www.jelurida.com/sites/default/files/JeluridaWhitepaper.pdf>, 2016.
- [5] Jae Kwon and Ethan Buchman. Cosmos, a network of distributed ledgers. <https://cosmos.network/resources/whitepaper>.
- [6] Asch whitepaper. <https://github.com/AschPlatform/asch/blob/master/docs/whitepaper/index.md>.



- [7] PCHAIN Foundation. Pchain whitepaper. <https://pchain.org/#whitepaper>.
- [8] Ethan Buchman. Tendermint: Byzantine fault tolerance in the age of blockchains. [http://atrium.lib.uoguelph.ca/xmlui/bitstream/handle/10214/9769/Buchman\\_Ethan\\_201606\\_MAsc.pdf](http://atrium.lib.uoguelph.ca/xmlui/bitstream/handle/10214/9769/Buchman_Ethan_201606_MAsc.pdf), Jun 2016.
- [9] Joseph Poon and Thaddeus Dryja. The bitcoin lightning network. <https://lightning.network/lightning-network-paper.pdf>, 2016.
- [10] dantheman. Dpos consensus algorithm - the missing white paper. <https://github.com/liuchengxu/blockchain-tutorial/blob/master/content/misc/dpos-consensus-algorithm-this-missing-white-paper.md>, 2017.
- [11] Silvio Micali, Michael Rabin, and Salil Vadhan. Verifiable random functions. In *Proceedings of the 40th Annual Symposium on the Foundations of Computer Science*, pages 120–130, New York, NY, October 1999. IEEE.
- [12] Robin Milner, Joachim Parrow, and David Walker. A calculus of mobile processes, I. *Inf. Comput.*, 100(1):1–40, 1992.
- [13] Robin Milner, Joachim Parrow, and David Walker. A calculus of mobile processes, II. *Inf. Comput.*, 100(1):41–77, 1992.
- [14] Robin Milner. *Communicating and mobile systems - the Pi-calculus*. Cambridge University Press, 1999.
- [15] Daniel Wang, Jay Zhou, and Alex Wang. Loopring(loopring): A decentralized token exchange protocol. [https://github.com/Loopring/whitepaper/raw/master/zh\\_whitepaper.pdf](https://github.com/Loopring/whitepaper/raw/master/zh_whitepaper.pdf), 2018.

## Disclaimer

This document is the conceptual document of the project (white paper), not to sell or collect the shares, securities or other controlled products of the Dora products and related companies. This document can not be used as a prospectus or any other form of standard contract documents, nor is it a proposal for the advice or collection of securities or any other regulated products in any jurisdiction. This document can not be any sales, subscription, or invitation to any other person to buy and subscribe to any securities, as well as the form of contact, contract or commitment based on this basis. The white paper has not been examined by any judicial regulatory body in any country or region. Any information or analysis presented in this document does not constitute any proposal to participate in the decision of the investment of the token, and does not make any specific recommendation with a tendency. You must listen to all necessary professional advice, such as taxation and accounting. Once the investor participates in the investment, it means to understand and accept the risk of the project. If the investor's country or region has the relevant laws and regulations to restrict or prohibit the participation of the investment, then please do not participate in the investment. Otherwise, it is considered that the individual will bear all the corresponding results and consequences for this purpose. No declarations and guarantees are given to the accuracy or completeness of any content described in this document, or in other ways that are related to the project; in the absence of preconditions, any statement and assurance can not be given to any achievement or reasonable content of a forward-looking, conceptual statement; Nothing in this document shall serve as a basis for any future commitment or statement.