Chapter 3. Consensus Scheme of Locus Chain

▪ The core idea of Locus Chain consensus algorithm

A blockchain system should have a 'consensus algorithm,' which is an algorithmic method to ensure the reliability and integrity of proposed information, over whole participants, in an open manner.

The consensus algorithm of Locus Chain is based on the following ideas:

- ✓ By the design of Account-Wise Transaction Chain (AWTC), which is essentially a Directed-Acyclic-Graph structure, every transaction in Locus Chain is explicitly ordered and grouped by its issuer account. In AWTC, the transactions could be easily verified using its issue order and issuer account.
- ✓ By the property of gossip protocol network, every node in the network group will eventually receive the same piece of information, in reasonably near future.

By this, we can assume this; "When some reasonable time is passed, the majority of network nodes will share the same transactions received. If then, the majority of honest network nodes may reach to an agreement on the contents of received transactions."

Locus Chain implements this idea by grouping transactions by a certain time intervals ("Rounds"). Then transactions included in each Round are identified, ordered and digested to a hash value. Then a consensus algorithm is executed to agree on an identified hash value, for each Round.

▪ Components of Locus Chain consensus algorithm

Here we define important concepts of Locus Chain consensus algorithm.

- Shards and Nodes

  As described in the previous chapter, transactions belong to a certain account are primarily managed by a certain shard. A Shard is physically a subset of whole Locus Chain network, and logically manages a subset of total accounts and its transactions. When new transactions are issued, first the transactions are spread over its issuer's primary shard, then spread over the whole network. Locus Chain's consensus algorithm

is also executed in shard basis, because Shards are consist of a roughly same number of nodes. The number of nodes in a shard affects times to spread transactions fully. Therefore, the count of nodes in a shard should be kept in a certain number to keep communication time in a reasonable range.

Nodes should have reasonably accurate internal clock to measure time lapses. We assume nodes can generally know when a round should start and end.

- Transactions

  A transaction is a signed chunk of a data unit, added to Locus Chain network. Transactions are signed with its issuer account's secret key. Nodes check transaction's validity using the issuer's public key.

  An account must set an integer index number to issued transactions. The first transaction of every account must have the index number of 0, and subsequent transactions by the same account must have index numbers increased one by one. So all transactions of a certain account have unique index numbers. Also, every transaction (excluding the first transaction) must include a hashed value of the preceding transaction and a round number.

  The order of transactions of a certain account is verified using the index number and the hash value of the preceding transaction. Nodes should verify the index number, round number and preceding hash when an unknown transaction is received. Nodes participating gossip network (and the consensus algorithm) must use verified transactions only. For this, nodes should keep information required for transaction verification, such as a history of transactions.

  If a node receives an invalid transaction, the transaction must be ignored. For example, if a node receives a transaction containing a round number not matching to the current round.

- Rounds

  A Round is a unit of time to gather and process new transactions. Currently, each round is around 2 minutes. Rounds have a "Round Number," an integer index number increased by one from the previous round. Transactions with a round number same to current round are gathered and processed with consensus algorithm.

  A consensus algorithm for a round starts at a reasonable time after the end of the round, to ensure new transactions to be fully propagated over the shard. Currently, the consensus algorithm begins after about 1 minute after the end of the round.

- Round State Propose Committee and Voting Committee

  Locus Chain's consensus algorithm is executed for each round. The nodes participating in each consensus are elected in a fair random fashion, to save computing power and network resources by limiting the number of nodes required.

  Two groups of nodes "committee nodes" are elected for each round. A "Round State Proposer Committee" and a "Round State Voting Committee."

  "Round State Proposer Committee" is a group of nodes ("RS proposers") capable of proposing a Round State. When the consensus algorithm starts, each proposer generates a hash digest value of transactions received for the round, then propose it as a "Round State Value Candidate" by signing the hash value. "Round State Voting Committee" is a group of nodes ("RS voters") capable of verifying Round State Value Candidates. Each RS voters verify and compare the proposed RS value candidates, then vote for a value closest to the voter's own transactions received.

▪ Rough Steps of Locus Chain's Consensus Algorithm

Locus Chain's consensus algorithm is executed in several independent steps like below.

1. Committed Election: In the middle of a round end, each shards elects RS proposers and RS voters in random. Currently, the number of proposers is adjusted to about 5, and the number of voters is adjusted to about 50.
2. Round State Propose: After a certain time from the end of the round, each node of RS proposers calculates a hash value according to received transactions, then propose the value as a Round State value candidate.
3. Round State Vote: Each node of RS voters collects Round State Value Candidates from step 2, then cast a vote for the value which closes to own transactions.
4. Round State Confirm: Each RS voters collects votes from step 3, then find a Round State value candidate that gains enough votes, typically over 2/3 of all votes. If a voter found such a value, then the voter signs the proposed value and broadcasts the result. The signed results of RS voters are permanently recorded and used as proofs of the transactions included in the round.

Details of each step are explained in the next chapter.