MCV: Mutual Chain Voting

Asynchronous BFT Protocol

as a distributed registry for ULTRANET infrastructure

(continuously updated)

www.ultranet.org

**Abstract.** There are several existing ABFT protocols that offer the highest throughput and minimal transaction fees, but they all have their own drawbacks and vulnerabilities. The most effective of them in terms of performance are those based on the Directional Acyclic Graph (DAG). However, classic DAG protocols require master nodes to prevent forks, double-spends and other attacks having a negative impact on network decentralization and reliability. Other approaches, such as Hashgraph, do not rely on master nodes but their confirmation time grows dramatically if the number of nodes is relatively high while network activity is relatively low. MCV is designed to achieve a throughput and cost close to those of DAG protocols but without sacrificing decentralization and scalability. It is based on a special data structure called Roundchain. Roundchain is similar to Blockchain but instead of chaining single blocks it chains sets of blocks called rounds. This allows all, or almost all, “members” to get their blocks added at each chain iteration, in contrast to a traditional blockchain where only one “winner” can place their block onto a chain at a time. MCV protocol, in turn, provides a simple voting mechanism that allows for reaching consensus for every round. Together, MCV and Roundchain make the technology inherently multithreaded, providing performance limited only by network throughput and without sacrificing decentralization and reliability.

***Disclaimer:*** *This document is for information purposes only. Ultranet Organization does not guarantee the accuracy of, or the conclusions reached in, this document, and this document is provided “as is”. Ultranet Organization does not make, and expressly disclaims, all representations and warranties, whether express, implied, statutory or otherwise, whatsoever, including, but not limited to (i) warranties of merchantability, fitness for a particular purpose, suitability, usage, title, or non-infringement; (ii) that the contents of this document are free from error; and (iii) that such contents will not infringe third-party rights. Ultranet Organization and its affiliates shall have no liability for damages of any kind arising out of the use of, reference to, or reliance on this document or any of the content contained herein, even if advised of the possibility of such damages. In no event will Ultranet Organization or its affiliates be liable to any person or entity for any damages, losses, liabilities, costs, or expenses of any kind, whether direct or indirect, consequential, compensatory, incidental, actual, exemplary, punitive, or special for the use of, reference to, or reliance on this document or any of the content contained herein, including, without limitation, any loss of business, revenues, profits, data, use, goodwill, or other intangible losses.*

# Contents

[Contents 3](#_Toc70068030)

[Glossary 4](#_Toc70068031)

[Consensus Algorithm 6](#_Toc70068032)

[Ultranet Specifics: Emission 10](#_Toc70068033)

[Ultranet Specifics: Fees and Costs 11](#_Toc70068034)

[Conclusion 12](#_Toc70068035)

# Glossary

**Account** – a cryptographic public/private key pair

**Member** – an account who declared itself as a potential block generator by placing a special transaction and notifying others that it is online. Members are eligible to generate blocks and participate in a consensus algorithm

**Block or Vote** – an element of a round that contains a data to add to database and hash reference to parent round

**Transaction** – a cryptographically signed data structure that describes some write operation on a database

**Round** – a set of blocks that are sorted by their hashes. Each round has its own sequence index. Each member can place only one block per round.

**Parent round** – a round Ri is a parent of round Rj when i = j- Lag, where Lag is a constant. The rounds in the range [0… Lag] are called *genesis rounds* and have a parent round reference set to zero.

**Roundchain** – a sequence of rounds where each block in each round has a reference to a parent round.

**Lag** – the number of rounds between a round and its parent round

**Node** – a network computer with special software that can receive and validate blocks from other nodes, send its own, and relay received blocks to other nodes and participate in the MCV consensus algorithm

Figure 1. The chain of block rounds

A

B

Members

C

D

E

F

Lag

Rounds

- Vote with transactions

- Vote

Time

Lag

Lag

Ri

Ri+1

Ri+2

Ri+3

Ri+4

Ri+5

Ri+6

Ri+7

Ri+7

Ri+8

Hash references to Ri round with Lag=4

Partially filled rounds

# Consensus Algorithm

The idea behind MCV is to create a distributed ledger algorithm with the following requirements:

* No Proof-of-Work, as it has lowest speed and highest hardware requirements among other algorithms
* No Proof-of-Stake, as it makes the richest miners take maximal profits so the rich get richer. It is also relatively slow without sharding
* No traditional blockchain, due to the principle of “one winner – one block”, which makes it slow and expensive
* No DAG, as despite its high speed and cheapness, it requires masternodes to prevent forks, which have a negative effect on the decentralization
* No Hashgraph and similar, as it can operate fast only with a very limited number of members, above which the confirmation time grows dramatically
* Minimal effort to create blocks
* Minimal possible transaction fees
* More than one block can be accepted at a time.

The core of MCV is a data structure called Roundchain, which represents a 2D grid of blocks where the column is a timeline of a particular member (like the blockchain of a single member) and the row (round) is an obligation for each member to place its block there.

Each block refers to a whole parent round with some delta called Lag. This guaranties cryptographic integrity for all data stored in all rounds similar to the ordinary blockchains.

In order to become eligible to add blocks, a particular account must satisfy some age and bail requirements. This measure limits the number of members, which is important for the voting mechanism. Any particular member can add only one block per round. Both these restrictions prevent the grid from growing infinitely in a horizontal direction. Vertical growth for each member, in turn, is limited by accepting only those blocks whose round lies in a specific range. This range is needed because the order in which a node receives blocks is not predictable for recent ones due to the nature of peer-to-peer networks. The advancement of this range is driven by a voting mechanism.

The following steps are required for an account to enable block production:

1. Have a non-zero balance.
2. Run a node and synchronize with the network.
3. Send and await a confirmation of a special transaction called “Candidacy Declaration”, which costs some amount of tokens and tells the others that the account is going to act as a member

Unlike known single‑chain technologies, there is no competition between members as block-creation efforts are negligible and there is no difference for senders between all available members. This will force members to compete with each other and so to keep fees as low as possible, and also means that transaction pool overflow is not possible.

Each member can generate only one block per round – which prevents a chain from forking. If a member creates two different blocks with the same round index, then the network treats them as a cheater, seizes its bail, and distributes it between round members.

To prevent members from adding the same transaction to more than one block in the same round each transaction hash is generated from data that includes the chosen member account (its public key). Another, constrain “Maximum Round” prevents a member from deferring a transaction placement indefinitely long.

To be eligible to generate blocks, an account must “declare candidacy” by adding a special transaction to the ledger that locks some amount of tokens used to determine the member’s rank and to punish it in the event of malicious activity. If there are more than Mmax members present in the network then they are sorted by its age-bail rank (account age divided by accession bail) and only top Mmax members are eligible to create blocks and vote for rounds.

For each round, the algorithm expects all or top Mmax members to either create a normal block or send a vote block – either of them acting as a vote for a parent round. A vote block has no transactions and is not stored in the Roundchain. In this way, members can vote for a particular candidate (a subset of blocks) of a parent round if more than one exists. The parent round with the most votes is considered to be the winner. There must be M\*2/3 of blocks collected by any round Ri to elect round Ri-Lag. As soon as Ri+Lag round voted, which means Ri is elected, and all [R0 … Ri-‍1] rounds are confirmed, then round Ri is also considered as confirmed. In other words, for any round elected by a corresponding child round, if all previous rounds are confirmed then this round is also flagged as confirmed.

Figure 2. Voting and Confirmation

Ri

Ri+1

Ri+2

Ri+3

Ri+4

Ri+5

Ri+6

Ri+7

Ri+8

Ri+9

Ri+10

Ri+11

Ri+12

Ri+13



Lag = 8

Voted for Ri-Lag

Confirmed by Ri+Lag

Voted for Ri+1-Lag

Voted for Ri+3-Lag

Collecting votes

Collecting votes

Collecting votes

Voted for Ri+2-Lag

Confirmed by Ri+1+Lag

Confirmed by Ri+2+Lag

Voted for Ri+4-Lag

Voted for Ri+5-Lag

Voted for Ri+6-Lag

Voted for Ri+7-Lag

Voted for Ri+8-Lag

Voted for Ri+9-Lag

Voted for Ri+10-Lag

Round voting has two further purposes. The second one determines how the chain grows. There are no time intervals used in this algorithm, so collecting a required number of votes means that a round is over and the chain can advance to the next one. The speed of MCV is thus limited only by the network speed. Since blocks can be received in a non-historical order, some “window” is required in a round sequence where blocks can be still accepted. For a first non-voted round Ri, this window lies in the range [Rlast\_confirmed + 1 … Ri+Lag\*2].

The third purpose is to determine whether a member is offline. This is done at the confirmation stage for the round. Having a list of current members and their blocks received, the algorithm can determine whether any member has not sent a block or vote and remove this member from the list, and thus from participation in any further processing until the member signals its readiness to continue by sending a special “online” block again.

Figure 2. The Example of Block Fork caused by its Issuer

w

Block 7

Accepted

A

Members

R7

R8

Rounds

Time

A fork detected:

Block issuer is punished

Block 8

Rejected

Block 8

Rejected

# Ultranet Specifics: Emission

# Conclusion

MCV is the first voting-based protocol that is designed to be as high-throughput as DAG-based asynchronous BFT algorithms but without a need of centralized masternodes and so to be able to deploy true homogenous decentralized networks for distributed registries.