Credits Decision Consensus Algorithm

Version 4.1/ 20.02.2019

(Some details might be added or changed)

List of content

General definition

A distributed network consensus is a method of independent nodes reaching a common solution. The main goal of consensus is a stable operation of the network of nodes where rounds reproduce either with a certain frequency or an algorithm themselves cyclically measuring continuous life of ordered interactions of the nodes like a logical core of the electronic system. The consensus is capable of performing additional tasks, such as processing transactions, verifying smart contracts' execution, being a basis for the operation of the distributed system for launching applications or any other payload which requires reaching consensus.

Formation of transactions and their distribution over the network

During the operation of a blockchain platform, two main network processes are performed in parallel:

- Formation of the transactions and their distribution between network nodes;
- Cyclical rounds execution where Trusted Nodes (TN) are granted authority to conduct consensus, which results forma common resolution regarding including transactions in a block and its generation is made.

Formation of transactions packages

The lifecycle of a transaction starts from a wallet where fields required for further processing are filled, such as :

- Sender's address
- Recipient's address
- Transaction amount
- Max fee that a user is willing to pay for a transaction

Once a user fills all fields, the transaction is signed with ED25519, an asymmetric encryption algorithm.

User enters his\her private key to sign the transaction. Private key storage in the wallet is not provided due to security reasons.

Synchronization of transactions packages

A node collects signed transactions from wallets and forms a package (Fig. 1)

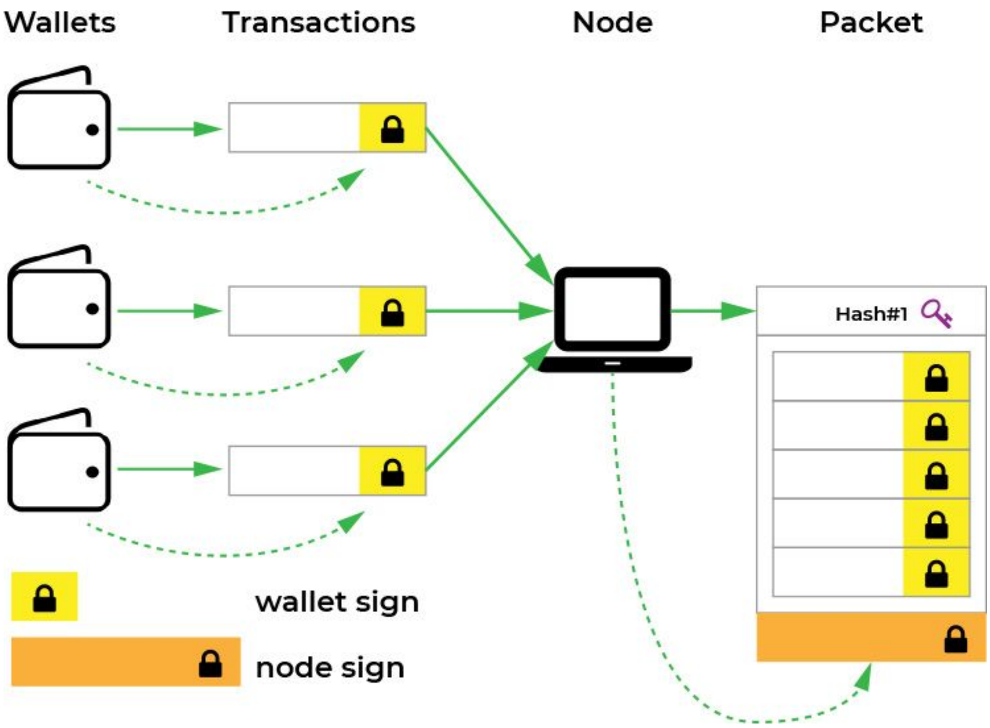


Figure 1. Formation of transactions package

Transactions package includes an ordered number of transactions (up to 500) and is given a header that contains hash calculated with Blake2s algorithm. The formed package is signed with the node's private key generated by Ed25519 and then sent to the network through the node's neighbors

Credits platform p2p network provides the ability to search for the transactions package by its hash. When a node receives transactions package from its neighbour, the node copies package's hash to the local buffer and sends it further to the network.

Synchronization of transactions on Trusted Nodes

Once the nodes received the list of transactions packages contained in the round table, they begin to check the availability of these packages in the local buffer. If some packages are missing, the synchronization process is initiated. A node requests missing packages from its neighbor nodes and if the packages are missing on those nodes as well, those neighbors request the packages from their neighbors and the process is repeated until missing packages are found. Availability of all transactions packages contained in the round table on all Trusted Nodes is the condition for a round to begin.

Formation of the ordered list of transactions

After the transactions packages synchronization between the Trusted Nodes of the current round completes, each TN has all transactions packages and can form a full list of transactions of the round (Fig. 2). List of round transactions is stored exclusively in a node's local buffer and is not shared over the network which is why it doesn't require security measures like hashing and encryption to prevent its changes.

The list of transactions packages is sorted in the round table, transactions are sorted in the transactions packages, consequently, a final list of transactions of the round is also sorted due to transitivity. Thus, for consensus to be reached, each TN must have the same ordered list of transactions to be processed in the round with the list corresponding to the list of hashes of the round table with transactions packages.

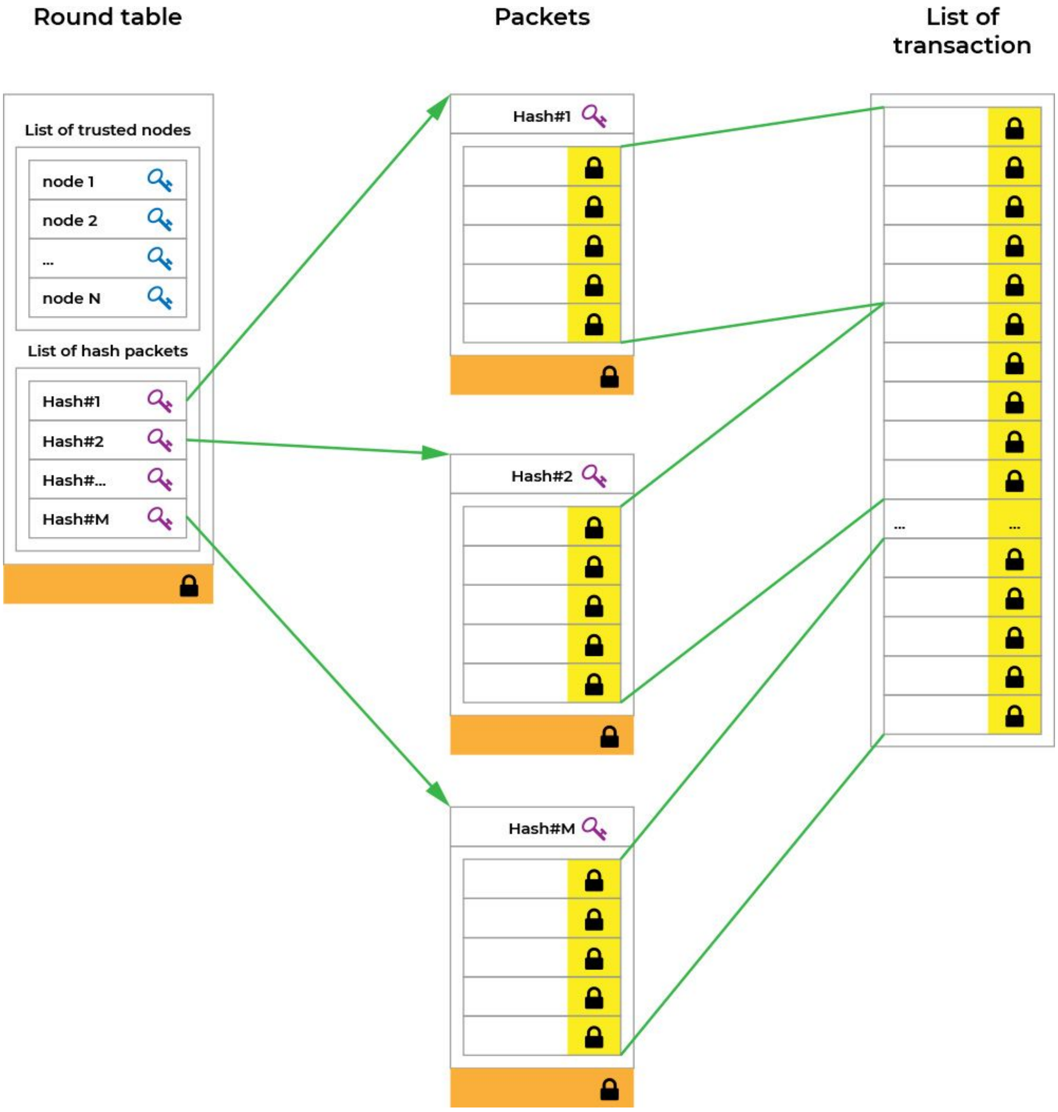


Figure 2. Formation of the list of transactions

After the list of transactions formation, a node can begin their validation.

Consensus Reaching

After all network nodes released transactions packages and exchanged them with each other, nodes responsible for consensus execution need to be selected and those nodes will have to validate transactions and generate a block. In the very beginning, network launch starts from Genesis-block formation and Big Bang.

Afterward, the network operates in cycles due to the rotation of rounds and constant selection of new Trusted Nodes for consensus execution.

Stage I

Formation of the list of Trusted Nodes.

The algorithm of operations of all nodes implies that if a node has an up-to-date blockchain, then during generation of the last block it has to calculate the hash of the block and send it to all Trusted Nodes of the next round whose public keys are contained in the meta block.

The trusted node of the current round in turn upon receiving the last block's hash forms a preliminary list of the candidates to become a TN in the next round. Then, based on this information final list of next round TNs will be determined and included in the new metablock. If the hash is not correct, the node is marked as "incorrect".

After a block is generated and recorded in the blockchain, nodes send a request to the TNs of the next round to include them in the list of candidates for becoming a TN.

For this purpose, they form a package containing last block's hash signature and node's public key, after which the package is signed by node's digital signature.

Each Trusted Node of the current round checks the validity of the signature in the received packages and correspondence of the received hash to the hash of TN's last block. This is the proof that the node that sent correctly validated signature has up-to-date blockchain and is capable of validating transactions in the next round. After the Trusted Node received first N-amount of correct hashes of the last block, it forms ordered list of nodes' public keys which will be the candidates for Trusted Nodes in the next round and will be included in the package of the first stage of consensus.

Validation of transactions and formation of characteristic function

Trusted Node validates transactions in the list of transactions of the round formed on the node. It should be noted that the process of transactions validation is performed on each node for all transactions independently from other nodes and only upon this process conclusion the nodes can exchange information about the results.

During validation, the following is checked:

1. If all transaction fields are filled correctly; (is valid)
2. If a transaction is unique (exclusion of "double spending")
3. If the transaction signature is correct;
4. If after the transaction is made the balance is more than 0.

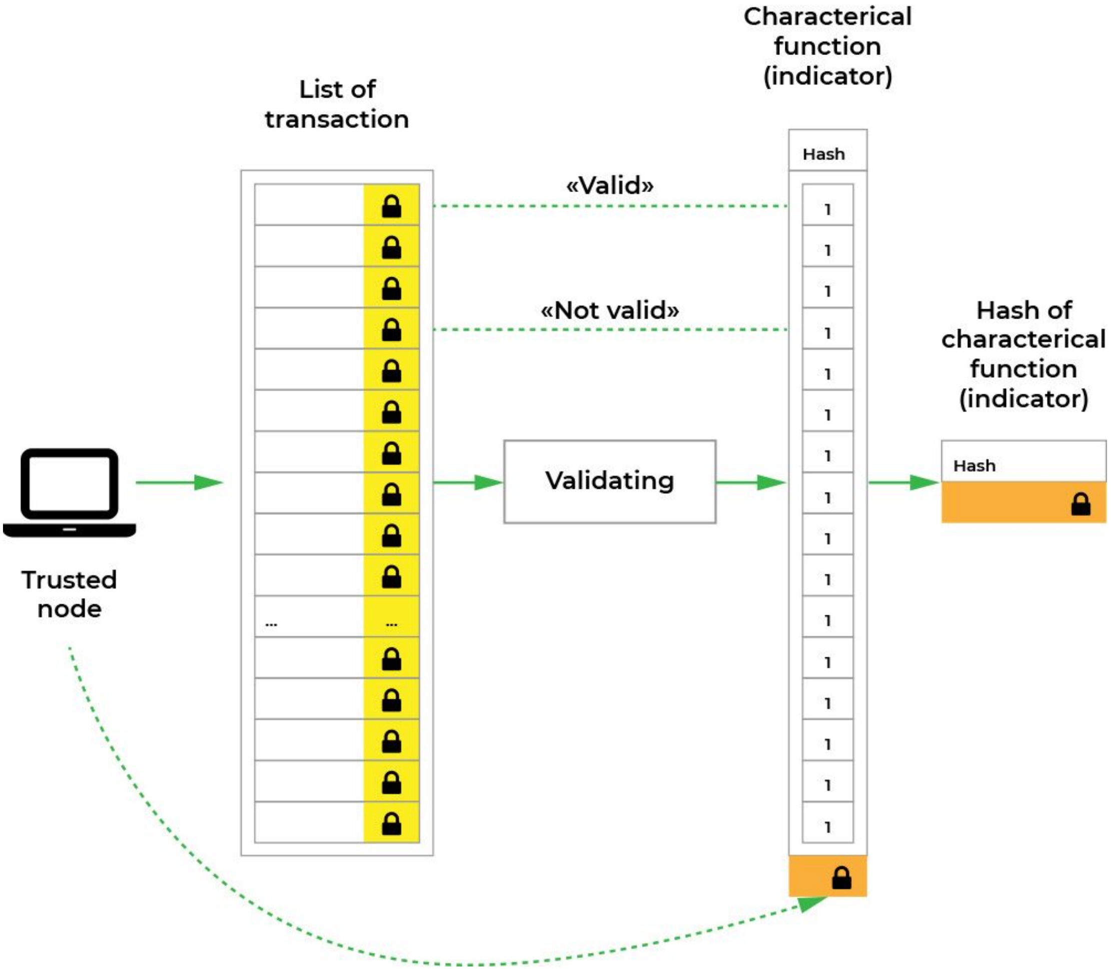

Characterical function (indicator)
List of transaction
Hash
«Valid»
«Not valid»
Hash of characterical function (indicator)
Validating
Hash
Trusted node
1
...

Figure 3. Formation of an indicator (characteristic function)

The result of validation is a characteristic function (Indicator) - a sequence of bits, where 1 means valid transaction and 0 - invalid. The characteristic function is given a header which contains a hash calculated with Blake2s hashing algorithm and also signed by a Trusted Node. For consensus execution, a short package is formed consisting of characteristic function hash (Indicator's hash), "Timestamp" field and a digital signature. "TimeStamp" field each Trusted Node fills with its local time. As a result of information exchange, each Trusted Node will have information about what "TimeStamp" each TN made and after writing node will be known, its TimeStamp will also be known and it'll be possible to assemble block from meta block.

Trusted nodes exchange – Stage I

Each TN forms the following structure:

- Node's ordinal number in the list of current round Trusted Nodes
- Indicator's hash
- Timestamp field
- An ordered list of the hashes of the next round transactions packages
- List of candidates for the Trusted Nodes of the next round

Then, a hash of the received package is calculated using Blake2s algorithm, the number of the current round is added as well and all of this is signed by the node's private key using ED25519.

The signature is then added to the original structure and stage 1 package is sent to all Trusted Nodes of the current round.

It should be noted that the hash itself and round's number are not shared that's why to validate digital signature the nodes that received this package have to preliminary perform the same operations independently

Public Key for validation is extracted from the rounds table available on Trusted Node by index contained in the first field of stage 1 package.

During validation of the received Stage 1 package the following verifications are performed:

1) The sender is in the list of Trusted nodes of the current round. In case of incorrect validation, the package is rejected.
2) Package's restoration (adding hash and round's number) and digital signature validation. In case of incorrect validation, the package is rejected.
3) If all verifications are passed, the hash of the stage 1 package and digital signature are put in the Vector of signatures with the number of the node that formed and sent the package. It should be noted that Stage 1 packages are sent not through intermediate nodes, but delivered directly.

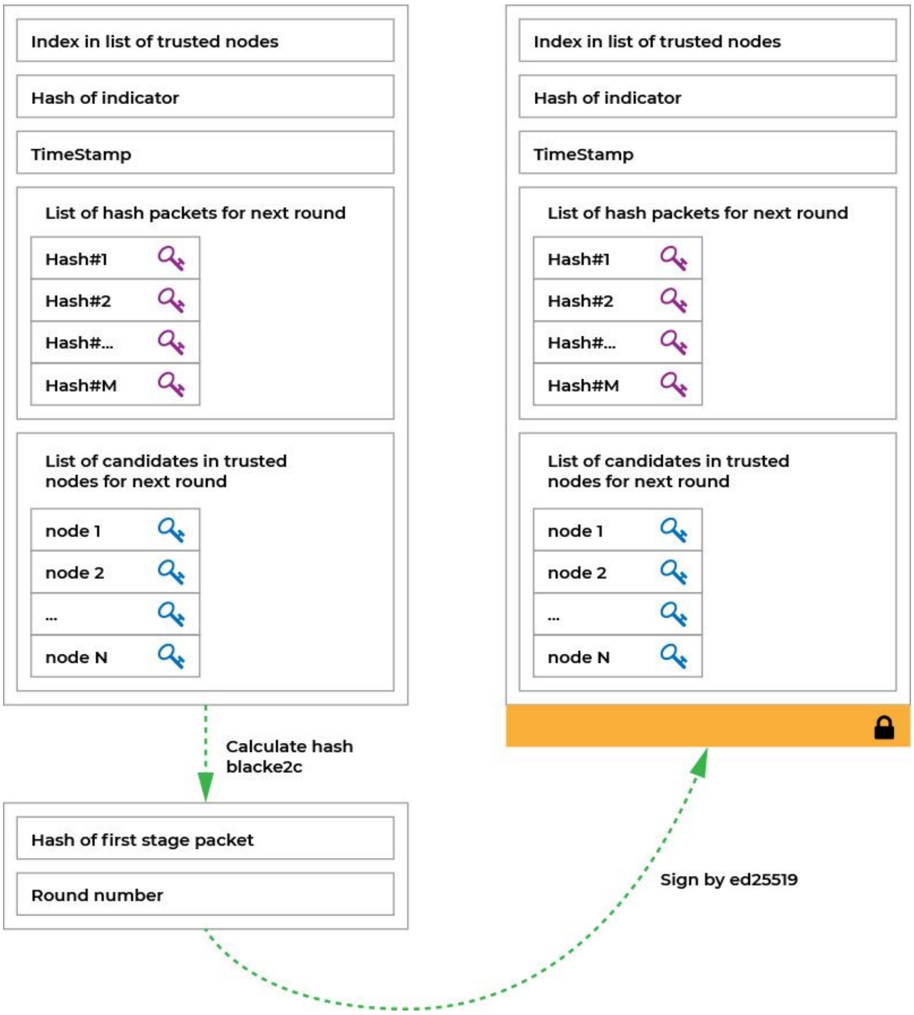

Index in list of trusted nodes
Hash of indicator
TimeStamp
List of hash packets for next round
Hash#1
Hash#2
Hash#...
Hash#M
List of candidates in trusted nodes for next round
node 1
node 2
...
node N
Index in list of trusted nodes
Hash of indicator
TimeStamp
List of hash packets for next round
Hash#1
Hash#2
Hash#...
Hash#M
List of candidates in trusted nodes for next round
node 1
node 2
...
node N
Calculate hash blacke2c
Hash of first stage packet
Round number
Sign by ed25519

Figure 4. Formation of Stage I package

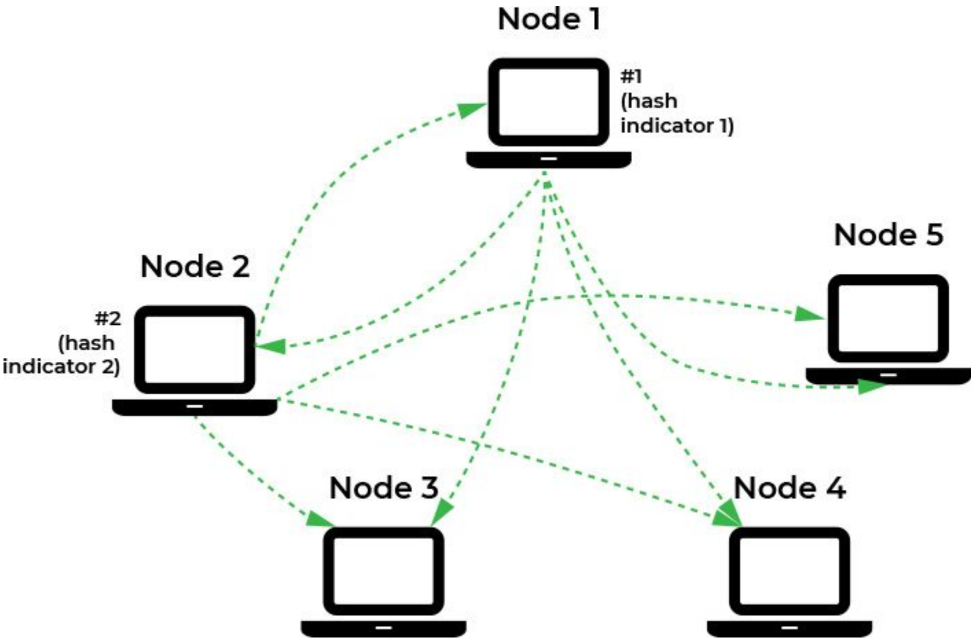
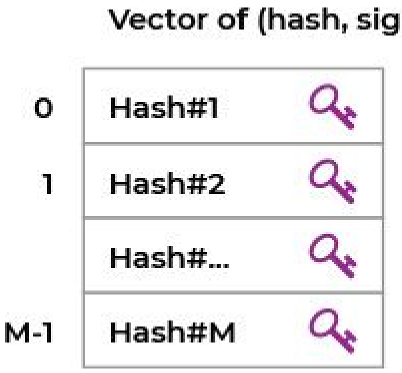


Figure 5. Exchange of Indicators' hashes

Figure 5 illustrates the example where 5 TNs participate in the round and 2 of them (Node1 and Node2) sent their formed packages of Stage 1 to all their neighbors.



Vector of (hash, signing)

0	Hash#1	🔑
1	Hash#2	🔑
	Hash#...	🔑
M-1	Hash#M	🔑

Figure 6

Indicator's hash is signed by ED25519 and validated by other nodes upon its receipt.

Stage II

Trusted Nodes exchange of Signatures vectors - Stage II

After Trusted Node receipt of the packages from Stage 2, it checks signature with Public Key of the TN that sent the package and if the validation is positive it adds "Hash-Signature" pair to Signatures vectors.

After all packages from Stage 1, formed Signatures vector (Fig. 6) is signed by ED25519 and the received Stage 2 package is exchanged again with all other TN.

Validation steps of the Stage 2 packages (signed signatures vectors):

1) Validation of package signature. Public key for the validation is extracted from “sender” field of transport level.
2) Then, strings that differ from the majority are detected in the vector. In case hash of the string matches digital signature, Trusted Node with the index of that string is marked as “incorrect”.
3) In case digital signature differs from the rest of Trusted Nodes and the hash doesn’t, it means that the sender substituted the signature. In this case the “sender” is marked as “incorrect”.
4) The same hashes must be found in most Trusted Nodes (>51% from total number of Trusted Nodes), otherwise round is finished ahead of time.

Upon receipt of the package of stage 2 the signature is validated. Then, by using of BFT algorithm “incorrect” Trusted Node (“liar”) is determined.

Time of exchange for Stage 1 and 2 is fixed and, thus, the nodes that won’t manage to exchange the packages in time are automatically marked as “incorrect” and loose the right to generate block.

Consensus procedure

Upon receipt of the Signatures vectors from all TNs-colleagues a TN assesses received information via the following algorithm:

- Step 1. Indicator’s hash is checked. All indicators’ hashes received from TN are checked. The most frequent hash (>51%) is considered as the consensus’ solution. All nodes that sent different hash are marked as “incorrect” Trusted Nodes. In case there is no hash with the frequency of >51%, the consensus is canceled and the round is finished.
- Step 2. A list of hashes of transactions packages to be processed in the next round is formed. Only the hashes of the transactions packages stored in the TN of the current round, but not included in the current processing are considered. Check for uniqueness (hash of the transactions package of the next round within one TN should appear no more than once) which allow excluding doubling the same transactions packages. If the TN has not passed the check, it’s marked as “incorrect”.

Formation of the list of next round Trusted Nodes

Next TNs are exchanging the lists of candidates for becoming a TN of the next round and the occurrence frequency of a candidate in the created list is analyzed. In case the candidate has a frequency of 50% (i.e. it’s included in the list of candidates in >50% of all TNs of the current round) it automatically becomes a TN of the next round and is included in the vector “next_round_trusted”.

All other nodes that have a frequency of less than 50% are selected by a pseudo-random algorithm, by the result of which on all TNs operating under the protocol the same list of next round Trusted Nodes “next_round_trusted” is formed.

Formation of the list of next round transactions packages

After “incorrect” nodes are detected a summary table of hashes of next round transaction packages with specifying the frequency of each transactions package hash in the general list. Hashes of the packages that are present on more than 50% of all TNs are included in the list to be processed in the next round and are written in the vector “next_round_hashes”. Usage of the single algorithm to form the list of the hashes of the next round transaction packages on all TNs operating under the same protocol should result in the same set of hashes. All the rest TNs are “incorrect”.

The result of this stage execution is a formed list of transactions to be processed in the next round which is saved on each TN in the vector “next_round_hashes”

Stage III

Writing transactions in a block

Let’s designate general list of transactions of the round as a, and the indicator (characteristic function) of a writing node as $I_{\text{ПУ}R(i)}$. A univocal correspondence of the indicator and its hash can be presented as $f_{hash}\left(I_{R(i)}\right) \rightarrow \#_{R(i)}$. Thus, if $\#_{\text{ПУ}R(i)} = \#_{validR(i)}$, we think that $I_{\text{ПУ}R(i)} = I_{validR(i)}$. By specifying the list of valid transactions as $T_{validR(i)}$ we get the following formula to determine the list of valid transactions by an indicator: $T_{validR(i)} = T_{R(i)} \times I_{validR(i)}$

According to this formula from the list of transactions “correct” nodes (from the list “real_trusted”) form the list of valid transactions which are included in the block.

Each Trusted Node that formed a block includes the following information in the meta block:

1. Indicator (Characteristic function)
2. Vector «realTrusted»
3. Vector «next_round_trusted»
4. Vector «next_round_hashes»
5. Separate signature of the block’s Writing Node (without the block itself)

Each Trusted Node must generate a block, correctly generate a meta block, sign it and send to other TNs in the form of signed receipt confirmation of the correct consensus completion. Each TN that’s received more than 50% receipts-confirmation from “correct” TNs of the round (from the vector “real_trusted) can become a round’s Writing Node (WN), in case WN selected by the general consensus of “correct” TNs couldn’t generate the meta block for any reason. A TimeOut mechanism is used for this purpose as well as an order set in the vector “real_trusted”. In other words, initially WN is a node “real_trusted[0], if it doesn’t generate signed meta block within a certain time frame, the node “real_trusted[1]” becomes WN and so forth.

Meta block Formation

Each node from the vector “real_trusted” generates a block based on characteristic function that’s passed consensus.

Transactions that are not marked in the indicator with non-zero sign, information about block’s (round’s) number, timestamp of the writing node, hash of the previous block and vector “real_trusted” are written in the block. The block is added to preliminary storage and its hash is calculated

Writing Node Selection

The list of “correct” and “incorrect” nodes is stored in the vector “real_trusted”. This vector is formed at the Stage 1 and is filled with values “0” (correct node) and “1” (incorrect node). Index, or ordinal number, correspond to the index (ordinal number) of a Trusted Node in the list of current round’s TNs.

At the moment of the selection of WN all TN operating under the protocol contain the same vectors “real_trusted”. Protocol defines pseudo-random function that allows all TNs reaching general consensus and selecting from all “correct” nodes (with “0” in the vector “real_trusted”) the node that will become WN.

Number of the WN is determined in the following way: the last 4 bytes of the last block’s hash are interpreted as unsigned integer and divided modulo the number of “correct” nodes (number of “0” in the vector “real_trusted”). The result of the division is the ordinal number of the “correct” node that by general consensus is accepted as WN.

Then, all “incorrect” nodes are removed from the vector and a list of “correct” nodes of the current round is formed, based on which a fee will be calculated. At the same time WN is the first one [index 0] in the list and the following it “correct” node is the second one [index 1] and so forth.

The result of this stage is a vector “real_trusted” which is included in the data package of the 3rd stage of consensus and also the adoption of the time in a “timestamp” of the WN by all nodes.

Adding new block to the blockchain by the nodes

Once the nodes receive meta block generated by WN, they form the list of valid transactions, generate the block, calculate the hash and verify signatures. In case all verifications are passed, the block generated by the node is added to the blockchain and the node provided that it has up-to-date blockchain sends the hash of the block to all TNs from the list “next_round_trusted”...