Avrio: A Infinitely scalable Cryptocurrency

Ariel Hurdle
areilhurdle@gmail.com
www.avrio.network

Abstract. A peer-to-peer currency using multiple blockchains rather than one, validated by a network of economically incentivised "nodes".

1. Introduction

In the year 2009 bitcoin was released. It provided the first ever decentralised method of monetary exchange. At first it seemed like a solution to a major problem in modern finance - centralised control of money, however as it gained more users a major flaw emerged. Bitcoin could handle only ~7 transactions per second while centralised payment vendors such as visa could handle up to 24,000 per second. This is largely down to bitcoins core design in which all transactions are stored on a public blockchain. each transaction has to wait to be included in a block and blocks can only contain a few transactions meaning the tps is hugely capped. We now propose a new protocol, avrio, which replaces the single public blockchain and PoW blocks with wallet chains.

2. Nodes

Like bitcoin anyone can run a node, however in avrio you get rewarded for doing so. To be rewarded you must fit certain conditions:

1) Have a *full* and correct copy of every wallet chain

2) Maintain *at least* 20 connections

3) Have a read/write delay of under x seconds (*x* is dictated by the network based on current node average to prevent low power nodes from getting rewards)

4) Have a average network ping of under *x* seconds (the average ping of the node is calculated by creating an average of every connections reported time, *x* is dictated by the network based on current node average to prevent low speed nodes from getting rewards)

The reward each node receives also changes based on a number of factors to do with the node. This is done by letting every node that is connected to that node vote on it based on the following factors:

1) Read reaction time. The time it takes to read a random block from memory

2) Network reaction time. The time it takes for a node to respond to a ping

3) Total uptime. A node only receives rewards while online however nodes that are offline for less periods receive a larger reward each time

4) The amount of valid new blocks, peers and uptime proofs they have received from that peer

The nodes then vote on how well the node is performing and whenever the node wants to collect a reward, he gets the average vote then times that by the number of connections. he then calculates the maximum reward he could have got by multiplying number of connections by 100 (the maximum vote) he then calculates the reward by working out what each vote is worth (the maximum reward is 10 AIO, if you have 100 connections the maximum votes is 100000 therefore each vote would be worth 10 / 100000 = 0.0001 AIO. he then times the number of votes by each vote worth to get his reward). Nodes who lock up funds also get a larger reward (percentage based). He publishes a block to the network sending himself the reward with the votes he received from each reward (each one is signed by the full node for verification). anyone can easily validate the amounts by checking the maths.

To ensure a full node has a full up to date copy of the blockchain it hashes a Merkle tree of the entire blockchain (this is done during syncing and updated constantly) and then encrypts that final hash using AES-NI with their node public key; Every node stores a Merkle tree of the blockchain so can easily verify that the fullnode has the blockchain by simply decrypting it.

2. Wallet chains

In Avrio every wallet has its own chain. This chain is stored by the nodes and only has a block added to it when a transaction related to them occurs (e.g. a receive transaction or they send a transaction). This removes the bottleneck introduced by a central blockchain and allows the blockchain to scale linearly to the number of fullnodes.

A transaction can add a byte of extra data, this can be used to set up user-based deposits. This allows services to integrate avrio as a payment method then provide a unique address to each customer (which will send to their wallet chain but with the data automatically added) and wait for the transaction to their wallet with the correct extra data. This is similar to cryptonote's payment id but much more compact as it only takes up one-byte extra space as opposed t 64.

3. Transactions

There are two main types of transactions: send txns and reward txns. A send transaction is created by a sender and includes recipient, amount gas details (price, max etc.) and a digital ECDSA signature to check integrity. A reward transaction is created by a node - it contains everything in a send tx but not the sender, it must also be broadcast with a list of votes signed by each voter (node) to ensure reward amount is correct. There are a few other less common txs:

1) Username registration tx - this transaction is used to register a new username for a wallet

2) Message Tx - using the extra data field of the transaction a sender can send a message to any wallet by simply paying the gas fee.

3) Fund Lock Tx - used by fullnodes to optionally lock some funds in order to receive a larger reward. the lock time is set to 30 days later at which point the fullnode automatically renews its certificate and relocks the funds.

4) Burn tx - a user can choose to burn coins and allow them to be re-rewarded to fullnodes (this will possibly be used to prevent nodes from misbehaving by burning some of their funds if they misbehave)

A block can contain upto 100 transactions (though the receiving chain only stores the txns from that block that are relevant to them in a block) however they cannot be conflicting. a block is considered a state of the blockchain - each new block alters the old state of the blockchain (eg registering a new node or changing the balance of wallet x) and thus cannot contradict each other (eg send x funds to allice then send these funds to bob as well). A node first checks the estimated work to complete the transaction(/s) will be less than the max gas then the transaction is processed by the node and it updates the state of thier blockchains, node list, balance list or anything else that is changed by the tx. He then calculates the work done and works out the fee based on the gas price and work done.

4. Usernames

Most cryptocurrencies utilise long and un-recitable wallet "addresses". This makes end user adoption very difficult as communicating where to send money becomes extensively difficult. To solve this Avrio will allow users to register a username tied to their public key, meaning you don't have to use a long and hard to remember address. The usernames are case sensitive and must be 10 or less characters allow all letters, numbers and – _ . and ! allowing a total posable combinations of unimaginable magnitude. You can also create "subusernames" for a lower fee. this allows services to register a username (eg printersco) and register a subusername for every user for deposits (eg maxcarter.printersco). Avrio Core wallet also allows tracking of each subusername balance for easy deposit tracking.

5. Blocks

Each Block has the current block version, the creators chain key, the previous block hash, the timestamp the block was signed by the creator, the height of the block, the txns in the block, any extra data added to the block, a signature from the creator and a signature from the first node to validate it (to work out the number of blocks validated by each node). The previous block's hash is included to link each block in the wallet's chain together.

Because a block cannot be altered or removed once it is shared across the network forks will only happen if a node is disconnected from the network, if a conflicting block is presented to a node it polls its connections for their top block, if over 50% of the network have that block then it is accepted otherwise it is rejected.

6. Speed

Due to avrio's parallel blockchain technology and lack of mining avrio's only main bottle neck is the computational power of the nodes running avrio. The number of users is presumed to increase every year. The nodes will most likely follow that path at a smaller rate and slow down once the user count gets to a mass adoption level. Using current affordable technology, a node can validate roughly 5000-15000 transactions per second. bitcoin has a average of 400,000 transactions confirmed per 24 hours, therefore to run a bitcoin sized network avrio would need (((400,000 / 24) / 60) /60) / 10,000 = only 0.4 nodes. moores law states that computational power doubles every two years, therefore if we can gain new (nodes per year * 2) * 10,000 new users per year.

12. Conclusion

In this paper we have presented a p2p decentralised payment system allowing a max transaction per second way over any other payment vendor on earth, not only that but one that scales with new users (new users = new nodes).