Bitcoin (Blink) - Peer to Peer Global Finance System v1

Joby Reuben
joby@projectblink.org

Purva Chaudhari
purva@projectblink.org

Abstract : Bitcoin's PoW is replaced with a propagation competition on blocks sent across certain set of validators under a time interval stamped with cryptographic proofs to claim fees and solve forks as per proof weight. To achieve adaptable scalability, each block's size is determined in consensus among elected nodes to reduce transaction waiting time. Gossip systems are replaced with a privacy-centered direct messaging system by constructing encrypted paths to deliver unconfirmed transactions and confirmed blocks. Apart from bringing speed, we resolved the need for a single transaction fee token for a blockchain by bringing forth a novel non-custodial per-token staking system to allow users to pay in any token. The "bitcoin" as a currency will ensure network security and hold Layer-1 tokens with staking and yielding fees. Since Bitcoin script adapts a Turing-incomplete language, the fees imposed for renting UTXOs make the transactions cheaper and the chain's ledger size optimized. We propose solutions for regulation revolving around taxation within the self-custody wallet ecosystem without compromising users' privacy.

1 Introduction

The Bitcoin Network [1] and other altcoin blockchains with newer consensus and programmable money are unable to compete with centralized payment providers in speed and volume due to their sheer inability to scale with centralization issues. Many consensus models rely on external validation concepts such as finding a nonce and proving stake instead of rules tied to propagation itself. Imposing heavy reliance on users to acquire native chain tokens diminishes the adoption of blockchain, thus keeping them far from the wonders of this technology. Decentralized networks can effectively adapt to users' needs by 1. Increasing block size 2. Decreasing block time 3. Eliminating low-efficient nodes 4. Increasing the requirement for node joining 5. Choosing producers by performance. Retail Staking with non-custodial solutions encourages users to stake their bitcoin to become a world reserve currency for every financial instrument with an additional restrictive monetary policy that helps to reduce volatility in times of recession.

Instead of storing UTXOs for an indefinite period, which compromises storage, renting UTXOs and replacing them with a fingerprint after they have expired without altering the block's Merkle root provides cheaper fees. With Bitcoin's unlocking script and use of sCrypt [2], developers can create custom scripts with regulatory options involving various types of taxes within its UTXOs, performing identity verification off-chain, with signatures instructing nodes to validate regulated payments with self-custody of tokens. Decentralized Altcoins can be bridged one-way to facilitate payments. A floating rate stable coin [3] without pegging to fiat can be issued with bitcoin as collateral for staking and yielding fees in future. Basic banking solutions can be developed in Bitcoin Script, whereas common computable programs can be deployed to a Layer 2 State Machine [4], where nodes update the state by providing a Proof-of-Fee-Receipt paid in any token in the Bitcoin network.

2 Bitcoin (Blink)

2.1 Election

Block size denotes the amount of data that can be propagated across every node on the Bitcoin network hence, its success rate is directly dependent on the bandwidth each node allocates for confirmed block transmission. Block size is not limited and is fixed in every new block, which denotes that every validator of the block can send and receive the data size. Variable block size helps to scale the network by increasing transactions per block when the nodes upgrade and announce their bandwidth. A ZK-SNARK-based proof takes a node's bandwidth as a public input signal, which is compared to the threshold range of the desired bandwidth of the network. To prevent tampering, the node will commit a salt, which is a large random number that will be added to the bandwidth to generate the xyz hash which will be verified in the proof. Thus, the proof will attest its bandwidth output to the UTXO's script and get validated. The network in consensus can forbid

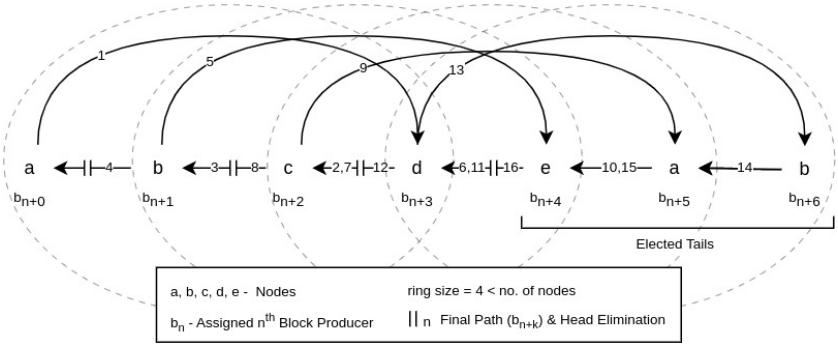

Figure 1: Ring Validation

low bandwidth nodes from participating in the election to produce blocks. Bandwidth requirements are increased for nodes to get elected for next block production can assure the capacity to hold more transactions to scale further. Elections will be conducted based on nodes bandwidth and each node's honesty weight. Nodes identities are masked by their public keys encouraging privacy. For each block, the elected node will produce, and set of nodes will validate and attach to the longest chain [1] . This set of nodes are finite and unique for every block and are called as a "ring validators" (see Fig.1). After each block's confirmation, the head of the ring is eliminated and a tail is elected which can be validated by nodes during propagation. The ring head after receiving the confirmed block will acquire bandwidth to gossip it's block over to other full or SPV nodes to secure it's rewards. The tail election's random seed is taken from concatenating Merkle Chain root and VDF Proof [5] of the recent head's confirmed block. Thus, the election is conducted for every block where a tail node is assigned.

2.2 Staking

The chain native token - bitcoin can be staked for node's public keys with specified token IDs, where the collateral can be used only once for a block. This results in a stake per token per block bringing the throughput (tps), per token basis. The proofs are based on Zk-Snark based Semaphore protocol [6] which will allow users to prove their membership in a group and send signals without revealing the original identity. Each token per block's collateral requirement is given in market price by taking the mean volume of previous blocks. Staked bitcoins can be withdrawn at any time, with no vesting period, except when the block is being created. This brings in a retail and non-custodial solution as opposed to security

deposit-type PoS chains. Delegators won't lose their stakes as slashing is done directly on the fees during forks. For specific nodes, bitcoins can be staked to collateralize or lock in its allocated block. Clients can provide their general users with a savings wallet in which they can benefit from an annual percentage yield by providing liquidity for staking. In this way, for a specific token's transaction to be included in a block, the first transaction should prove the collateral.

2.3 Regulation

Regulation via centralized exchanges & custodians risks funds and doesn't encourage a self-custodial ecosystem. Regulations are carried out by whitelisting verified hashed public addresses. Regulatory agencies can verify a public address by either doing minimal KYC or something such as mobile number-based OTP verification to encourage privacy through client wallet applications.

UTXOs are stamped with region proof on their unlocking script based on specific spending conditions that will only allow a transaction on-chain if taxes are deducted properly. Bitcoin scripts can work efficiently and securely, as opposed to turing-complete smart contracts. Tax models such as capital gains slabs can be issued by regulators trustlessly and are validated during the user's script execution. External taxes such as TDS and sales tax can be imposed off-chain by wallet-providers with flexibility. These three taxes in a trustless-decentralized way provides almost all the tax models that a regulator can siphon into a centralized payment infrastructure.

2.4 Messaging

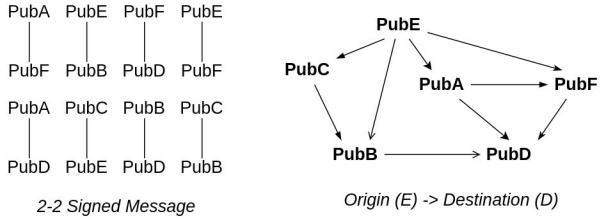

Figure 2: Topology

Delivery of unconfirmed transactions to nodes plays an important role in finality. Shared mempools defile the network with duplicated data, resulting in a poor choice of transactions by accepting higher fees to include in a block. A direct-messaging system should be deployed with messaging instructions specific to each party as opposed to gossip protocols. Paths are attached with unconfirmed transactions directly from the constructed network graph, which is available to all nodes with public keys as pseudonymous identities protecting privacy. Two peering parties mutually sign a 2–2 random message for every x blocks (ring size) and are gossiped across the network to identify the connection as online. All the signed random messages prove that each pubkey signature can display a network topology map (see Fig.2) from a node's point of reference. Paths are encrypted end-to-end with routing [7] instructions so that the origin cannot be traced. Nodes route transactions to the destinations where producers can attach the transaction to their allocated block. Since the stake information is available in the public ledger, client wallets can construct transactions along with path information that routes to the nearest blocks for instant finality. Transactions are always atomic, providing a solution to queuing issues. Responsibilities are provided to all participants, where nodes only receive the transactions that they need to include, and client wallets should construct shorter paths to provide the best user experience.

2.5 Propagation

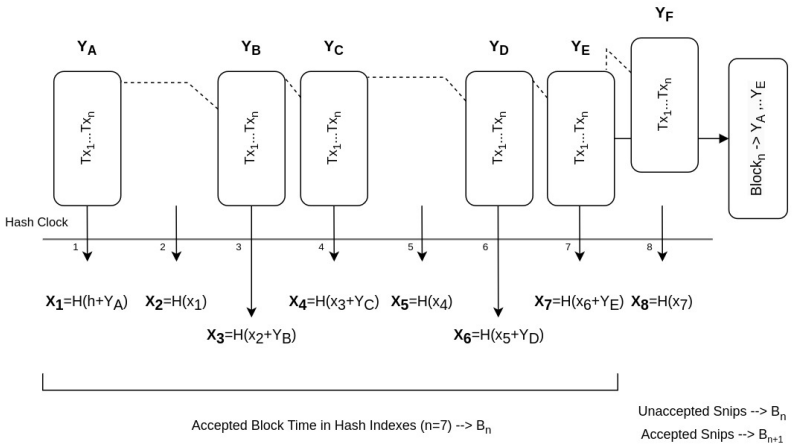

Figure 3: Snips

A Block is collectively validated but constructed as snips - divisible block chunks by the producer and directly messaged to the block's ring validators with routing instructions to propagate and get confirmed. Each snip references the previous snip's hash similar to the chain of blocks for proper identification of each block's snips. For a block, a competition to deliver all snips under x time interval in hashes is required to win rewards and avoid slashing of fees. When a block fails to win, it will do a intra-block fork and not mint its last of snips which will contain the fees and rewards. VDF proofs are attached for every snip (see Fig.3) during propagation among its assigned ring validators to declare the state of each block's competition and resolve forks. Failed block fees and rewards are slashed by sending them to a burn address by next blocks and results in addition of negative weights that indirectly slashes the node's bandwidth costing capital. Null-Blocks are self-minted by its ring validators with proofs.

To synchronize time, each node's hash rate per second of a specific hash function is proved cryptographically onchain and taken in multiples of a common hardware's hash rate. An Individual hash-rate proof is provided along with bandwidth proof for every x blocks (ring size) before it expires which trustlessly synchronizes all nodes as a single hardware producing continuous hashes concated with all snips. The ZK IHR proofs are algorithmically similar to the bandwidth proofs, where the hash rate provided by the node is compared to the threshold range of the desired hash rate and salt is used to prevent tampering attacks. This time-based competition can be termed as "Proof of Speed".

2.6 Rewards

Rewards are given for each snip hash concated with transactions validated by VDF proofs. Since newly mined bitcoins (5BTC/10mins) for a period of time are definite and each block's time is capped in blink implementation, new bitcoins can be supplied exactly proportional to the current Bitcoin issuance rate with halving. Halving of Bitcoin issuance is done every $R \cdot (1.26 \cdot 10^8)$ hashes, where R is the common hardware's single thread hashrate. Each VDF hash represents work done by nodes on validating and propagating transactions. When a fork arises due to rejected snips, the rest of the block time and its allocated rewards are slashed. Ring validators also propagate proof for empty hashes without snips within themselves for accurate measures. Meanwhile, when a block is fully minted, the rest of its allocated rewards are distributed to the current halving period. This incentivizes nodes to attest and receive snips at the earliest to get increased block reward

in the next blocks. While Tax outputs are attached as zero input transactions (coinbase tx) within the snip it contains, the rewards and fee outputs created as a separate snip denotes the end of a block.

During staking, producers announce their accepted tokens for which they will directly withdraw the commission. For other tokens, delegators can stake with a condition that their stake in bitcoins will be traded for the collected fees. During the commission withdrawal of non-accepted tokens, the producer will deposit collected fees to delegators and inflate the stake 1:1 ratio to withdraw the collateral. Users can pay in any token, delegators incur the risk, and producers get paid in tokens of their choice to validate transactions.

2.7 Renting

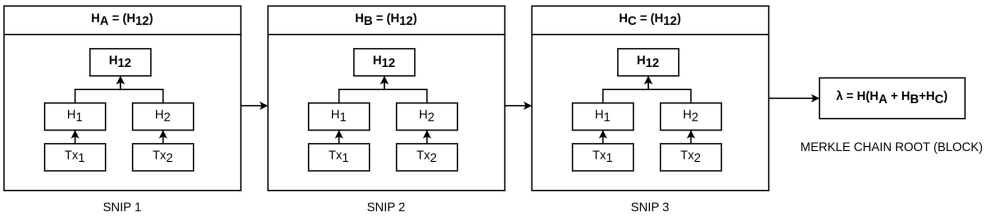

Figure 4: Merkle Chain

Instead of taking the Merkle roots of all the transactions inside a block, a snip's Merkle root is taken and linearly hashed to find the Merkle Chain root (see Fig.4). Since snips can be rejected by validators and cannot be tampered once it is streamed to ring validators, it is unsure to predict a Merkle chain root making it a pseudo-random value. Inside a snip contains parsed transactions whose hashes are taken to find the snip's Merkle root can be pruned if the UTXOs are spent, burnt, or expired. Each UTXO expiry block height is embedded in its script, and can be scanned by nodes, and pruned to optimize their data storage. Client Wallets can store each of their users' transaction history and can be audited onchain using Merkle chain roots. Renting rates can be given in market price independently voted by producer nodes per byte per block. Users cannot directly pay for rent, but rather each new UTXO created is charged a transfer fee in the range of 0.05% - 0.005% decided based on the total volume of all transactions settled on previous blocks.

Transfer fee charges more fees for higher value utxos and less for lesser value utxos bringing ease to transact for retailers. Each UTXO's transfer fee will set an expiry date to itself. UTXOs doing state updates [8] will not be charged a fee, where users are incentivized to combine UTXOs to a single balance holding UTXO with an increased expiry value saving validators disk space.

2.8 Forks

Confirmed blocks as snips are streamed in a backward ring manner (block_n to block_{n+k}...block_{n+1}). For each block to prove block time by providing VDF proofs, a set of validators is initialized by ring's size denoting number of blocks and their producers. Intra-block forks arise when ring validators reject snips of a block (see Fig.5) which can be resolved by the very next block providing instant finality to users. Intra-block forks can be minimized by keeping the ring size variable and always lesser than the total number of nodes, a better alternative to sharding. Block size and time are decided on consensus by the block's ring validators bringing a healthy propagation. Bandwidth upgrades are needed for the nodes which initiate the intra-block fork and are punished by the network by imposing negative weights. Each new block will include the previous block's VDF proof which resolves the intra-block forks by verifying the accepted snips and updating the chain using Merkle Chain Roots.

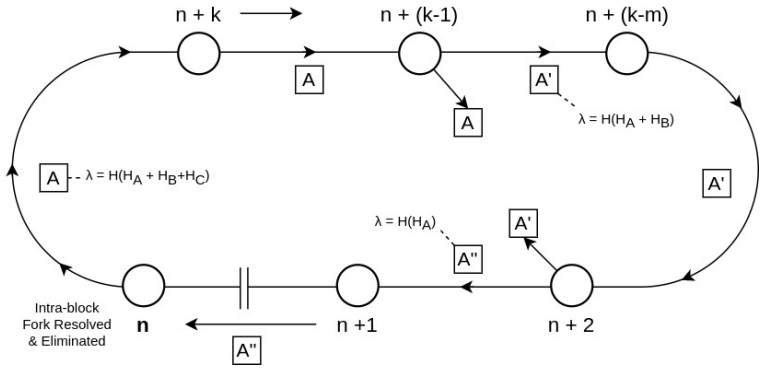

Figure 5: Resolving Forks

During offline activity of ring validators, Inter-block forks arises which can be resolved by attesting VDF Proofs of offline activity and thereby adding negative weights to it. Hence, each block will have its ring validators who are assigned by onchain accounted bandwidth and honesty weights, where the heaviest VDF proofed version of the block$_n$ is added to the longest chain.

2.9 Exchange

3 Treasury

For the active development and sustainability of the project, a DAO Treasury is set up to fund developers and the community. A minimal commission is imposed on producer fees and deposited to a treasury script. Memberships are non-fungible and non-transferable 1. Temporary (Core-Developers), 2.Permanent (Investors, Community), 3. Contracts (Employees, Operations) given in x amount per y term for z period. Except Permanent members, votes are taken to decide membership and treasury decisions. Temporary members can be kicked for failure in active contribution whereas Permanent members cannot be kicked out due to their external contribution (funds) towards building the project. Permanent and Temporary members cannot be added after the first mainnet, but can appoint heirs to their registered membership. Contracts are paid out initially and the rest is provided to other memberships as dividends according to their weight. Participants are only rewarded for their active contribution, not indefinitely like holding a fungible token or speculative participation. A decentralized open-sourced organization structure is maintained with decisions involving votes brings forth a sustainable growth for the future of project blink.

4 Future

The future of the Bitcoin network includes building finance-specific L1 applications such as exchanges, bridges, lending & borrowing, insurance, token minting and bank mirrored wallets. Since these applications are developed inside an unlocking script, it requires preimage construction off-chain and settles onchain - inheriting the security of Bitcoin that centralized applications don't offer. Privacy can be improved by obscuring amounts similar to Monero with tax validation assisting regulators and masking financial information of a specific country. Emphasis on Zk delivery and validation of snips can reduce influence attacks in the future.

To introduce limitless scalability, an offline digital cash-payment system will be developed with suitable hardware wallets. To provide a general-programmable environment, a State Machine featured with multiple high-level languages using LLVM [9] IR code will be deployed as a layer that can update its global & contract state by providing a gas limit through a receipt-proof paid in Bitcoin Network. The smart-contracts will not contain balances and EOAs, rather purely executed for business-logic to build DApps without a financial scope. To store client-side assets & data files to build fully decentralized applications, a CDN system will be developed in Bitcoin Network without duplicating data among nodes and provide faster content delivery to end-user. Moreover, research will be conducted to merge various Bitcoin and altcoin chains to a single decentralized global scalable infrastructure for a clean experience on the whole of finance and computing.

Implementations can be contributed openly to `https://github.com/projectblink`

References

[1] S. Nakamoto, "Bitcoin : A peer-to-peer electronic cash system," *Whitepaper*, p. 9, 2008.

[2] "scrypt — a high level smart contract language for bitcoin sv." `https://scrypt.io/`.

[3] "Dai stablecoin purple paper." `https://web.archive.org/web/20210127042114/https://makerdao.com/purple/`.

[4] G. Wood *et al.*, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, no. 2014, pp. 1–32, 2014.

[5] A. Yakovenko, "Solana: A new architecture for a high performance blockchain v0. 8.13," *Whitepaper*, 2018.

[6] "semaphore-protocol/semaphore: A zero-knowledge protocol for anonymous signalling on ethereum." `https://github.com/semaphore-protocol/semaphore`.

[7] J. Poon and T. Dryja, "The bitcoin lightning network: Scalable off-chain instant payments," 2016.

[8] "Stateful contract — scrypt 0.0.1 documentation." `https://scryptdoc.readthedocs.io/en/latest/state.html`.

[9] "The llvm compiler infrastructure project." `https://llvm.org/`.

A Appendix

A.1 Individual Hash Rate

```
// Public inputs
signal input : node_ihr
signal input : ihr_hash
// Private signals
signal input : salt
Signal input : required_ihr
// Output signal
signal output : if_pass
// Range proof check
signal buffer
signal range_check
if (node_ihr > required_ihr - buffer&& node_ihr < required_ihr + buffer) :
    range_check = true
// Verify hash
signal hash
signal hash_check
// RIPEMD160 to calculate the hash
hash = RIPEMD160 (salt, required_ihr)
if (hash == ihr_hash)
  hash_check = true
if (range_check && hash_check)
  if_pass = true
else
if_pass = false
```

Listing 1: IHR Circuit

A.2 VDF Proofs

A common hardware single threaded hash-rate in seconds () is taken where the IHR proofs synchronize all the nodes in multiples of () to identify time in means of () hash-rate per second. All nodes IHR proof provided along with its PubKey for identification.

VDF Proofs provide timestamps of external data by concatenating it to a continuous hashing function restricted in a single threaded operation. During Multi-core validation, it can prove that a delay in real-time happened and the exact delay value can be calculated in hashes. Since our nodes synchronize as a constant hash-rate network, we can convert eitherwise from hash-rates to seconds.

Each node will have a hash-clock that produces continuous hashes and oncatenates the external data received. Clocks are initiated with the arrival of first external data (transactions)

Transactions are divided as snips and a bunch of snips form a block. Each snip shall be the external data concated to the clock's hashes.

Proofs are attached along with the external data when sent to the next node. Proofs include pre images and its signature which includes the public key. A Block time in hash-count will be provided for each block. This hash-count is given in () where the nodes shall take it in multiples and integer.

From snip 1 to last snip it should be inside the hash-count or else reject the upcoming snips. If the block size is more also reject the upcoming snips. This creates a intra-block fork and resolved in the next block.

A.3 Merkle-Chain

Algorithms

A.4 Ring Validation

For each block a set of nodes should validate and attest the VDF proofs to confirm the block, these set of nodes are predictable, assigned and verified. These set of validators of a block is called as ring validators. The name came because a linear structure of elected nodes are treated as a ring to validate a block.

The ring validators should always be unique, finite and its count should be always lesser than the total number of nodes. Each node is treated as equal and can participate in producing blocks in the Bitcoin Network. This eliminates centralization and network halting that's done in popular scalable blockchains.

For VDF Proof attaching, the producer of the block is known as head shall propagate his block in a backward ring manner from n to n+k to n+1, it forms a ring like structure where it ensures that the block with or without intra-block fork shall have the heviest VDF proof block with n+1 block producer.

Since the n+1 producer is the next block producer he will resolve the intra-block forks by updating the heaviest proof of previous block to the following nodes.

When the n+1 producer receives the block, it will propagate to the producer and eliminates him from the ring validation. Such wise, the n+1 producer becomes the head of the ring and the network shall elect a tail of the ring to continue to the propagation of next block's snips. Since tails are newly joined ring validator, upon receiving a snip from the head, it starts its hash-clock, upon triggered. Rest of nodes of ring validators shall continue it hashing upon elected as a tail till it gets eliminated as a head.

When the ring head is eliminated from the ring validator set, it can gossip its confirmed block with heaviest VDF proofs to other nodes connected to the network. Since bandwidth is not consumed for confirmed block propagation, it can be well used to propagate its block for further validation and attaching to all nodes longest chain.

A.5 Bandwidth & Weights

Bandwidth proof here

Weights shall increment or decrement the posted bandwidth costing capital to nodes due to dishonest, lower performance or bandwidth.

$$\begin{aligned}
\text{Null Block} &= -0.015 \cdot (\widetilde{x}|\mathbb{R}|) \\
\text{Time-outs} &= -0.01 \cdot (\widetilde{x}|\mathbb{R}|) \\
\text{Forks} &= -0.01 \cdot (\widetilde{x}|\mathbb{R}|) \\
\text{Fork Proof} &= +0.01 \cdot (\widetilde{x}|\mathbb{R}|) \\
\text{Age} &= +0.05 \cdot (\widetilde{x}|\mathbb{R}|) \\
\text{Extras} &= +x \cdot (\widetilde{x}|\mathbb{R}|)
\end{aligned}$$

Null Block weight is added if a producer fails to add any snip. Time-outs refers to intra-block forked block with no fees or rewards collected. Futher forks refer to the fork created by the other ring validators. Fork proofs can be provided by any ring validator before the fork to get incentivized, whereas the nodes in the path shall be equally punished. Age is a positive weight provided to producers with full block production without forks for the epoch. Extras is given for further native protocols such as oracles, etc.

A.6 Election

Tail election is conducted for every block to assign nodes to mint blocks.

- Tails will be assigned from available nodes bandwidth proofs - Weights are ordered in heavy - From the current ring validators the mean value is taken - The mean value is the joining requirement - From the confirmed block, the block's merkle chain root and VDF proof is concated and produce a MD160hash - The Nodes with weights more than the mean value is taken - The node's pubkey which is lesser than the hash is assigned as node - If there are no lesser value, then the nearby greater pubkeys are taken. - If there are no pubkeys greater than the mean value, then the heaviest weight shall be taken, ignoring the hash - All validators can verify the assigned node, as weights are available in the ledger and when the intra-block fork is resolved each validator will have the produced hash MD160 and it verifies - If the assigned node is some other, the ring validators will not accept the block from it and it is treated as a null block.

A.7 Block Requirement

From each ring validator set, the block size is assigned

The minima of all the nodes bandwidth is taken and it is assigned as block size per second

The per block time is set by taking the mean value of previous two blocks.

The per block size is set by taking the per block time

The per block time is given in common hardware's hash counts, and it is validated by each IHRs.

A.8 Ring Size

The ring size is variable and it changes accordingly if a block is being subjected to intra block fork - If fork ring size –, no tail - If not fork ring size ++, 2 tails added

A.9 Fork Proof

Each Snip that is to be included with its VDF Proofs will have a node which will reject further snips. The origin of the fork can be taken by the previous node by providing the VDF Proof of unknown which can be validated. This can impose negative weights on the producers

A.10 Hash-Reward

Take ihr hashes of producer, convert to global, check hash reward.

Halving of Bitcoin issuance is done every $R \cdot (1.26 \cdot 10^8)$ hashes equals 210,000 10-minute bitcoin blocks.

Every epoch (ϵ_n) will have a g amount of bitcoins to be issued. Hence for every epoch a target $g(\epsilon_n)$ new bitcoins is set to mint. For forked blocks, the supply is slashed. For fully minted blocks with remaining hashes, the rewards are distributed to the next hashes equally incentivizing producers.

$$g[R(\epsilon_n)] = \frac{\Delta g(\epsilon_n)}{\epsilon_n(\sum_1^n R)}$$

Rewards per block,

$$g[R(\epsilon_n(b))] = \epsilon_n(b(\sum_1^n R)) \cdot g[R(\epsilon_n)]$$

$$\Delta g(\epsilon_n) = g(\epsilon_n) - [\sum_1^n g[R(\epsilon_n)] \in b(\epsilon_n)]$$

Slashing forks $\Delta\lambda$,

$$\forall \epsilon_n(b_n), \Delta g(\epsilon_n) = \begin{cases} g(\epsilon_n) - g(\epsilon_n(b_n)), & \text{if } \Omega(b_n) = 0 \\ \Delta g(\epsilon_n) + b, & \text{if } \Omega(b_n) = \mathbb{N} \wedge \lambda \\ \Delta g(\epsilon_n) - b, & \text{if } \Omega(b_n) = \mathbb{N} \wedge \Delta\lambda \end{cases}$$

$$\Omega(b_n) = g(\epsilon_n(b)) - [\sum_{1}^{n} g[R(\epsilon_n)] \in b_n(\epsilon_n)]$$

A.11 Network Graph

A.12 Routing Path

A.13 Renting

$$5 \cdot 10^{-4} \leq \Delta f \leq 5 \cdot 10^{-5} == \sigma[\Psi(\epsilon_{n-3}, \epsilon_{n-2})] > 0.75$$

$$f' = f + x \begin{cases} x = +5 \cdot 10^{-6}, & \text{if } \Psi(\epsilon_{n-3} > \epsilon_{n-2}) \\ x = -5 \cdot 10^{-6}, & \text{otherwise} \end{cases}$$

$F = \text{UTXO value} \times f'$

A.14 Staking

Delegators provide bitcoins or the staking token for accepted token id to nodes. For non-accepted tokens, delegators ($h1 \in D_T$) can create their script along with specifying node (M).

$$\mathbb{SS} = lim, id, R, D_T(am_n, Pub_k, e_i), P(am_n, Pub(M), er_i), F_t(am, er_i). \sum D_T(am_n)$$

When deposited, each stake of delegators (D_T) is identified along with its (k) Pubkey and (er_i) initial exchange rate to deduct gains tax during withdrawal (W). Each script ($\mathbb{SS}$) will have a limit (lim) in bitcoins or staking token along with id for what token (t) it is being staked.

$$W(\mathbb{SS}) \leq \begin{cases} [D_T(am_n) \cdot R] + [F_t(am) \cdot \frac{D_T(Pub_k(am_n))}{\sum D_T(am_n)}], & \text{if } D_T(Pub_k) \\ o = F_t(am) \cdot C & \text{if } P(Pub(M)) \wedge Pub(M) = h1 \in D_T \\ \frac{(\sum D_T(am_n) \cdot er_c(T)) - (o \cdot er_c(t))}{er_c(T)} & \text{if } P(Pub(M)) \wedge Pub(M) \neq h1 \in D_T \end{cases}$$

During the production of blocks, producers $P(M)$ can reject their stake script transactions to freeze the collateral as proof for validation. Each script can only be proved once where the sum of all deposits will be made constant. Each validator shall have a commission percentage given in $C = n\% \cdot 10^{-2}$. For accepted tokens, validators create outputs for themselves and deposit fees. For non-accepted tokens producers inflate the ratio (R), to withdraw staked tokens (T) for commissions.

$$y = (F_t(am) \cdot er_c(t))(C)$$

$$R = \frac{[\sum D_T(am_n) \cdot er_c(T)] - y}{\sum D_T(am_n) \cdot er_c(T)}$$

A.15 Tax

A.16 Exchange

A.17 DAO

Seperate scripts are provided for every x blocks from which 66% blocks shall be payout, rest 44% will be dividend yields.

Memberships Temporary - To remove a temporary member - Vote percentage - (Temporary members 100% (except the member to be removed), Permanent Members 0%, Employees - 75%) - Period is first 66% block period. To assign Heir of a temporary member (Temporary - 51%),

Permanent members can not be removed, can assign heir anytime

Contract - To include a contract (Temporary members - 51%, Permanent members - 0% (can vote), Employees - 30%) - Period is first 66% block period. To remove a contract -(Temporary members - 51%, Permanent members - 0% (can vote), Employees - 30%) - Period is last 44% blocks