Bitcoin (Blink) - Peer to Peer Global Cash System

[WORKING-DRAFT-v0.1]

Joby Reuben
joby@projectblink.org

Purva Chaudhari
purva@projectblink.org

Abstract : Bitcoin's PoW is replaced with a propagation competition on blocks sent across validators under a certain 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. Altcoins can be bridged one-way and collateralized for a floating rate stable coin directly used for staking and yielding fees along with bitcoin bringing utility. Basic banking solutions can be developed in Bitcoin Script, whereas common computable programs can be deployed to a Layer 2 State Machine, 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 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 a finite & unique set of nodes will validate and attach to the longest chain [1] . This set of nodes of following blocks n to m are called as a "ring". After each block's confirmation, the head of the ring eliminates itself and elects a tail 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. The election seed is based on Merkle Chain root and VDF Proof [3] of the newest confirmed block which is constructed by validators after resolving forks is taken as a random input value. 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. Each token per block's collateral requirement (y) is given in market price (USD) 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. Additionally an interest rate (r) is added along with the stake to restrict the free flow of bitcoins to increase demand in exchanges with a timelock of 21 days. This interest rate is determined by analyzing the market price of bitcoin from previous blocks. In future, a new collateralized floating-rate coin can be issued from lending L1 altcoins which can be used for staking to receive yield benefits.

2.3 Regulation

Regulation via centralized exchanges & custodians risks funds and doesn't encourage a self-custodial ecosystem. A regulator must have the authority to sign or approve transactions. Whitelisting specific hashed public addresses belonging to specific countries, verified and signed by government-assigned client wallets or regulators, by either doing full KYC or something minimal such as mobile number-based OTP verification, could work with maximum privacy.

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 in this case. Tax models such as capital gains slabs can be issued by governments trustlessly and are validated during the user's script execution. External taxes such as TDS and sales tax can be imposed off-chain with maximum flexibility. These three types of taxes provide almost all the tax models that a regulator can siphon into a centralized payment infrastructure.

2.4 Messaging

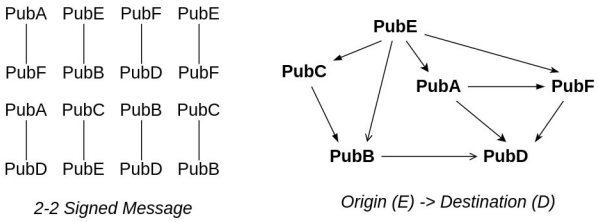

Figure 1: Topology

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 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 from a node's point of reference. Paths are encrypted end-to-end with routing [4] 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

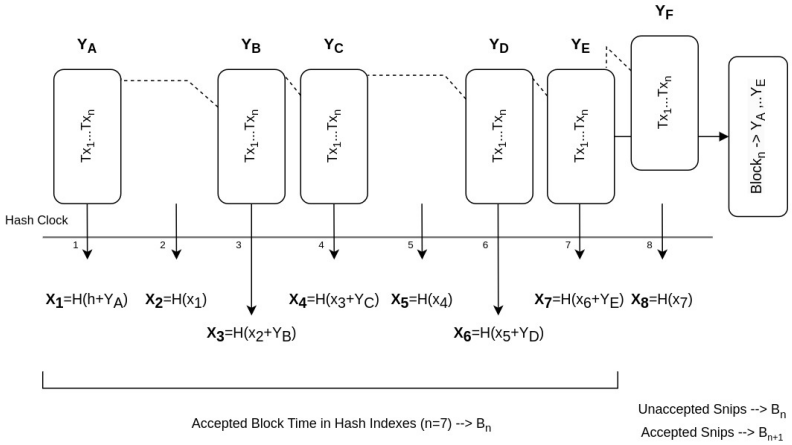

Figure 2: Snips

A Block is collectively validated but constructed as snips - divisible block chunks by the producer and directly messaged to the block's ring nodes 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 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 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 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 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 provides cryptographic timestamped proofs to validate each block's size and time, under which all snips have to arrive and win the time-based propagation competition. The competition is termed "Proof of Speed".

2.6 Rewards

Rewards are given for each snip hash concated with transactions validated by VDF proofs. Since newly minted bitcoins for a period of time are definite and each block's time is capped, new bitcoins can be supplied exactly proportional to the current Bitcoin issuance rate with halving. Each 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 next block producers of the halving period. This encourages nodes to attest to VDF proofs for receiving snips at the earliest. While Tax outputs are attached as zero input transactions within the snip it contains, the rewards and fee outputs when accounted as a separate snip mark 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

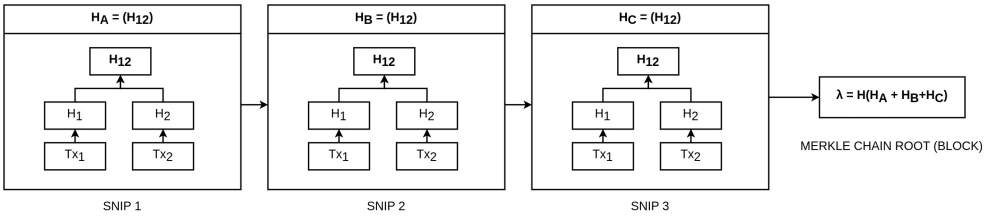

Figure 3: Merkle Chain

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. 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 giving it a purely randomized 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 will not be charged where users can combine UTXOs to a single balance holding UTXO with increased expiry value. This encourages users to store a single UTXO per wallet reducing transaction fees and also incentivizes nodes, clients, etc by saving disk space.

2.8 Forks

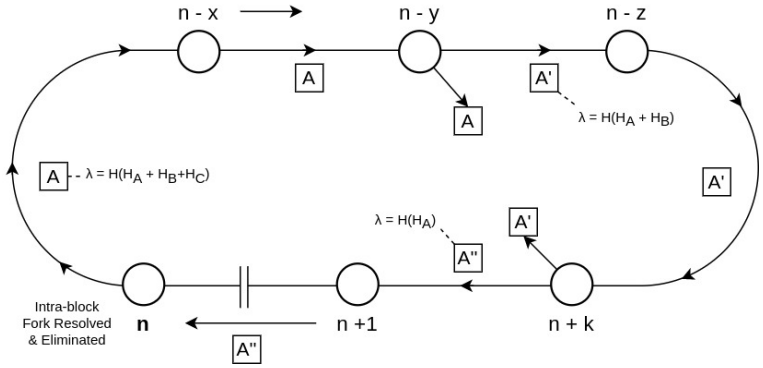

Figure 4: Resolving Forks

Confirmed blocks as snips are streamed in a backward ring manner. 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, this 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 not electing them for the upcoming blocks. Each new block will include the previous block's VDF proof which resolves the intra-block forks by selecting the snips and updating the chain using Merkl Chin Roots. During offline activity of ring validators, Inter-block forks arises. This can be resolved by attesting VDF Proofs of offline activity which will add negative weights to the node.

2.9 Oracles

Oracle data provide real-time market rates to unlocking scripts for every x interval of blocks. Various oracle protocols [5] fall to freeloading and duplication issues due to its trustless nature. A per-feed subscription-based price-oracle protocol is introduced with participants - exchanges, nodes, and users. Exchanges provide real-time market rates along with traded volume for weighted aggregation of data. Nodes subscribe per request to exchange price feeds for each token id. Each exchange's feeds will have a reputation score similar to prediction markets maintained by users to avoid faulty data. Exchanges are incentivized by subscription fees, nodes provided with positive weights for the next elections, and users profit from removing faulty data oracles. Freeloading is avoided by exchange participation and equally distributing positive weights per feed to producer nodes.

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). Each has its privileges provided to developers, investors, and community members.

Votes can decide to add or remove contracts stating x amount per y term for z period withdrawn during payouts. Votes can also decide to kick a temporary member due to any circumstantial issues or failure in active contribution. Permanent members cannot be kicked out due to their external contribution to building the project. Permanent and Temporary members cannot be added after the base script deployment, but can appoint heirs to their membership. Contracts are paid out initially and the rest is provided to other memberships as dividends according to their weight. Core-Developers are only rewarded for their active contribution, not indefinitely like holding a fungible token. A decentralized open-sourced organization structure is maintained with decisions involving votes bringing forth 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, and mirrored wallets with current banking. 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.

Since Layer 1 is dynamically scalable, the need for an offline digital cash-payment system will be developed with suitable hardware wallets. To provide a decentralized programmable environment, a State Machine [6] featured with multiple high-level languages using LLVM [7] 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. Moreover, research will be conducted to merge various Bitcoin and altcoin chains 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] A. Yakovenko, "Solana: A new architecture for a high performance blockchain v0. 8.13," *Whitepaper*, 2018.

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

[5] S. Ellis, A. Juels, and S. Nazarov, "Chainlink : A decentralized oracle network v1," *Whitepaper*, 2017.

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

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

Appendix

A Onchain Protocol

A.1 VDF & Forks

@Joby

A.2 Ring Validation

@Joby

A.3 Election

@Joby

A.4 Block

@Joby

A.5 Fork

@Joby

A.6 Merkle-Chain

@Ajay

A.7 Hash-Reward

@Joby

A.8 Network Graph

@Purva

A.9 Routing Path

@Purva

A.10 Renting

@Joby

A.11 Stake Requirement

@Joby

B Zk-Circuits

B.1 Bandwidth Proof

@Purva

B.2 Individual Hash Rate

@Purva

B.3 Ownership Proof

@Purva

C Bitcoin Scripts

C.1 Tax

@Ajay

C.2 Staking

@Purva

C.3 Oracle

@Joby

C.4 DAO

@Purva