BitVote: A Decentralized Voting System for Improving the Bitcoin Development Process

Jennifer Fairhurst

1. Introduction

Bitcoin [1] is the most popular decentralized monetary network in the world [2]. However, its governance, specifically the process of upgrading the protocol, remains an informal social process. Changes are proposed via Bitcoin Improvement Proposals (BIPs), discussed on a Google Groups mailing list and online forums, and implemented only after rough consensus is achieved. While this conservatism protects the network, it creates a high barrier to entry and opacity regarding community sentiment, as well as taking up a lot of time from the developer and proposer. There is currently no verifiable way to measure the popularity of soft fork or protocol upgrade ideas.

We propose BitVote, a hybrid on-chain/off-chain voting system that aims to improve the Bitcoin development process in a way the broader Bitcoin community would accept. Unlike centralized forums or off-chain signaling tools, BitVote anchors its logic in smart contracts on the Stacks Layer 2 (L2) [3], ensuring that voting rules are immutable and execution is trustless. We introduce a level-up mechanism to filter ideas and filter spam, and a gas-efficient Optimistic Batch Cleanup protocol to allow for proposal status updates without centralizing authority. We also introduce a rewards system that allows users to get more on-chain BitVote voting tokens.

2. Background

2.1 Bitcoin

Bitcoin is a decentralized peer-to-peer electronic cash system. It solved the double-spend problem in systems without a central intermediary by utilizing a consensus mechanism known as Nakamoto Consensus, which combines public-key cryptography with Proof-of-Work (PoW). At its core, Bitcoin prioritizes self-sovereignty (decentralization), security, censorship resistance, verifiability, and immutability above all else, which is a direct reflection of the broader Bitcoin community values. The network is secured by miners who expend real-world energy (hashing power) to propose new blocks. This design ensures that rewriting history is economically infeasible. Additionally, the Bitcoin scripting language is intentionally non-Turing complete to reduce the attack surface. This means it cannot natively support complex state-heavy applications like voting systems.

2.2 Bitcoin Development Process

The standard for changing Bitcoin is the Bitcoin Improvement Proposal (BIP) process [4]. This requires a social consensus workflow. While intended to ensure high-quality standards, the process is tedious and can sometimes create friction for contributors.

The process begins when a contributor conceives of an improvement, soft fork, or new standard for the network. The author must first identify BIP 2, the meta-BIP that specifies the strict formatting and procedural requirements for all proposals. However, before writing, the author must perform an exhaustive search to ensure the idea has not been proposed previously and must find the existing developer context for their idea. This is a significant hurdle due to the fragmentation of Bitcoin discourse. Relevant context is often scattered

across decades of archives in the bitcoin-dev mailing list, the Bitcoin Talk forum, various Subreddits, Discord servers, X debates, Telegram group chats, and open and closed Pull Requests in the BIPs GitHub repository. If the idea appears novel, the author must fully draft the proposal, adhering strictly to the technical format and structure mandated by BIP 2, and then submit it to the bitcoin-dev mailing list for peer review. It is notable that this list is hosted by Google Groups. This reliance on a centralized infrastructure provider has been a source of significant controversy within the community, raising concerns about potential censorship, data harvesting, and the irony of hosting decentralized protocol discussions on Google servers.

The author incorporates feedback from the mailing list, revising the draft repeatedly until they believe it meets the technical standards required for formal submission. They then format the proposal and open a pull request to the official bitcoin/bips GitHub repository. More reviewing occurs in the GitHub comments section, focusing on technical correctness and formatting. The assigned BIP editors review the pull request, and if it meets the necessary criteria and demonstrates a degree of rough consensus, the editors assign it a BIP number. Only at this stage is the proposal officially under consideration. If these criteria are not met, the proposal may be deferred or rejected. The BIP is further refined until it reaches a terminal state: final, rejected, withdrawn, obsolete, or replaced.

This workflow is notoriously time-consuming, often taking years to move from ideation to finalization. Additionally, with discussions fragmented across varying platforms, it is difficult for developers to gauge the true sentiment of the broader economic majority. This ambiguity can fracture the community, leading to internal disputes, such as the discussions around SegWit2x (BIP 102 [5]). In extreme cases, the inability to reach consensus effectively can lead to hard forks, such as what happened in the case of Bitcoin and Bitcoin Cash [6].

2.3 Layer 2 (L2) and Stacks

To add programmability to Bitcoin without altering its base layer, known as layer 1 (L1), Stacks, a Bitcoin L2, is utilized. Stacks uses Proof-of-Transfer (PoX), a consensus mechanism that anchors Stacks blocks to Bitcoin blocks. This means that after approximately 100 blocks, a Stacks transaction is as immutable as a Bitcoin transaction. Stacks uses Clarity, a decidable smart contract language that prevents common reentrancy attacks found in Solidity, making it ideal for high-stakes governance.

3. Related Work

The problem of decentralized governance has been explored extensively across the blockchain ecosystem, with many different system designs being proposed. Snapshot [7] is an example of an Ethereum based off-chain voting system. Users sign messages with their private keys, which are then stored on a decentralized storage network known as IPFS [8]. A centralized server aggregates these signatures and calculates the results based on a specific block height. While gasless and free, it relies on a system external to the blockchain for the most important parts of the voting process: the actual voting. This adds another layer of trust required from users.

Beyond general-purpose governance tools, specific academic models have been proposed to secure electronic voting using blockchain technology. One such example is the Auditable Blockchain Voting System (ABVS) [9], which employs a hierarchical network structure consisting of polling stations, super-nodes, and trusted nodes. In this model, polling stations allow for capturing votes, while super-nodes are the primary validators responsible for counting votes and committing them to the main blockchain. To ensure data availability and fault tolerance, this chain is replicated across a network of trusted nodes. ABVS faces a significant limitation in accessibility in that it is designed for physical attendance. Voters are required to visit

designated polling stations to cast their ballots, making the system unsuitable for the decentralized, globally distributed nature of the Bitcoin community.

4. BitVote Design

It is within the culture of extreme verification and resistance to centralization described in Section 2 that BitVote is situated. A voting system for Bitcoin cannot simply be a majority rules mechanism, as that violates the core principle of minority protection. Instead, BitVote serves as a verifiable signaling tool to cryptographically prove community sentiment regarding specific proposals without enforcing the result via code. BitVote follows the following design goals:

1. **Align with the community values, needs, and beliefs**: in other words, it should fully align with the Bitcoin design principles and respect the social layer of Bitcoin's consensus while upgrading the tooling used to measure it.
2. **Mandate the proposal structure set in a simple way**: to reduce time and search cost for formatting.
3. **Lower the effort barrier for participation**: the steps to vote on and propose ideas should be intuitive and straightforward.
4. **Ensure only serious stakeholders put forth proposals**: this allows spam reduction and avoids wasting the community members' time.
5. **Allow for structured evolution of a proposal**: to ensure fairness in the decision process, and it allows for faster and more structured evolution of ideas.
6. **Enable differing Voting Powers**: allows rules and rewards to be put in place in order to encourage community participation.

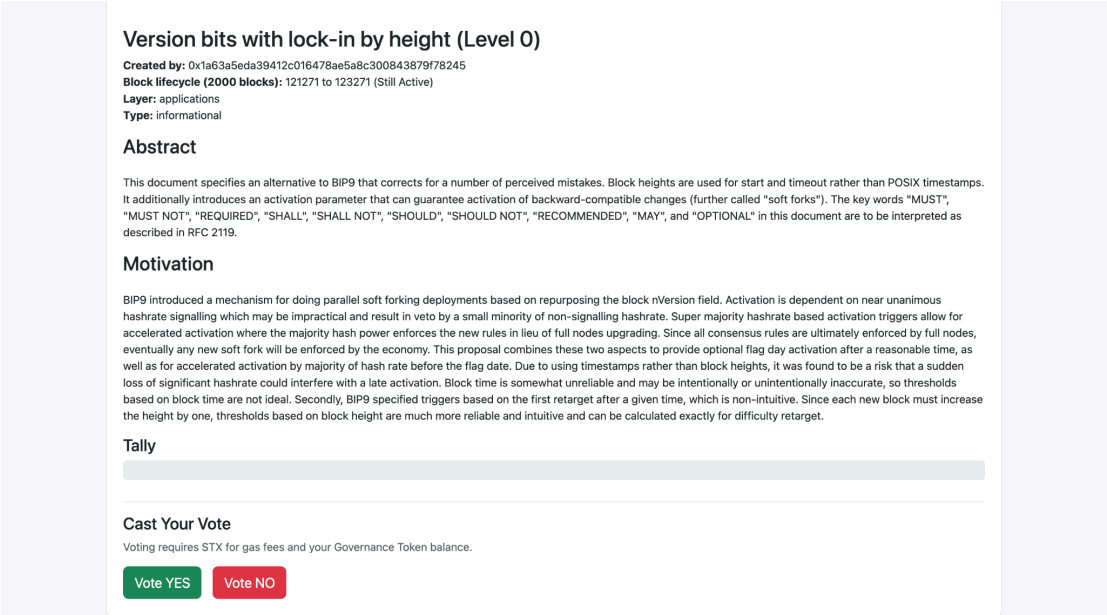

Figure 1: Mock active level 0 proposal voting page.

4.1 The L2 Contracts

To implement the immutable logic required for trustless signaling, BitVote utilizes Clarity smart contracts deployed on Stacks, a Bitcoin Layer 2 network. This allows the system to run on the Stacks testnet during development and inherit Bitcoin's finality on mainnet after deployment. The system relies on two primary contracts: governance-token and proposal-contract.

4.1.1 governance-token

This contract defines the bitvote-token, a SIP-010 fungible token [10] that represents a holder's voting power. Unlike systems that rely on wealth-weighted voting (1 token = 1 vote), this contract includes a mechanism that allows official users, those who connected their wallets, to claim a one-time welcome balance of 1,000,000 tokens. This balance represents a single vote unit, standardizing general participation. It also has methods that reward Bitcoin improvement actions, such as incentivizing high-quality contributions by rewarding proposers with 1,000,000 tokens upon successfully advancing a proposal from Level 0 to Level 1, and 2,000,000 tokens on leveling up beyond that. The token balance is committed on-chain at the L2 layer, ensuring the integrity of voting power cannot be manipulated by the application interface, as it is verifiable.

4.1.2 proposal-contract

This contract serves as the central registry and state machine. It stores the metadata and information for every proposal, enforcing strict compliance with the BIP 2 standard (requiring fields such as Title, Abstract, Motivation, and Copyright). Crucially, it manages the voting lifecycle, ensures the system rules are being followed (no double-voting and such), tracks the live vote tally, the current status (passed, failed, or undecided), and the level of the proposal.

The level attribute tracks a proposal's progression through the governance tiers, reflecting its growing consensus over time. The higher the level, the higher the quorum needed and the higher the passing threshold. An example of this is shown in Figure 2. This allows only the best proposals with the highest consensus to progress to later levels. Additionally, as the level of a proposal increases, the proposer is required to add more information about the proposal. For level 0, the proposal only has a title, layer, type, abstract, and motivation. If it passes the level's voting requirements, the proposer has the option to level it up to the next level by editing previous information and providing new information, such as the copyright, specification, and backward compatibility. For the final level, level 2, the proposal is required to have a reference implementation.

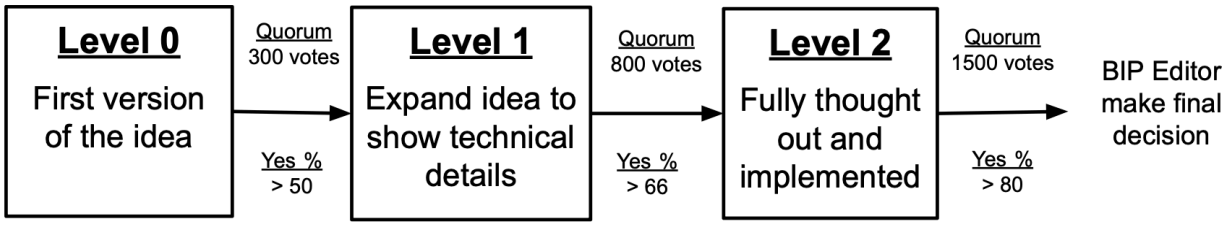

Figure 2: Example of a proposal going through a version of BitVote with 3 levels that have 3 differing quorum and approval percentages

4.2 The Website Engine and Frontend

While the ultimate truth is stored on-chain, the user experience is powered by a hybrid off-chain architecture that splits responsibilities between a Python script and a client-side JavaScript (JS) script. The Python script uses Flask for backend routing purposes and the request library to make calls to the Stacks API to query the read-only raw contract state from the L2 blockchain and data maps. It then structures that information for user-friendliness on the web. This component handles the computational heavy lifting that is too expensive to perform on-chain, such as filtering many inactive proposals to identify those that are stale (expired but not yet finalized status-wise). On the other hand, the JS helps with frontend matters, such as connecting to a wallet using the Stacks Connect library and getting the token reward the first time only, proposing a change, and casting a vote. All these actions cost a small fee, with the initial proposal being the most expensive (but not

prohibitively high) to reduce spam. BitVote also implements an Optimistic Batch Cleanup protocol. When a user casts a vote or proposes a new improvement, the client uses the stale data identified by the backend code and bundles a cleanup instruction into the same transaction. This distributes the maintenance cost of the system across active users, preventing state bloat without requiring a centralized administrator to pay the cost of updating the status of proposals on-chain. The process is optimistic in that another user's cleanup instruction may have already resolved a subset of those stale proposals.

4.3 The Discussion Section

A voting signal is meaningless without the context of debate. BitVote aims to maintain the discussion and feedback culture of mailing lists and online forums, but changes the platform to align with decentralized values. To achieve this, BitVote integrates the Nostr protocol [11] for its discussion layer. Rather than storing comments in a central database (which could be censored), comments are cryptographically signed notes broadcast to decentralized relays. The BitVote interface aggregates these notes, effectively centralizing only their viewing to allow for easy searching, filtering, and context gathering, while keeping the data itself decentralized and sovereign. This ensures that the history of the debate remains immutable and accessible, even if the BitVote website itself were to go offline.

5. Conclusion

BitVote demonstrates that decentralized governance for Bitcoin is possible without altering the base layer. By utilizing Stacks, we achieve a system that aligns with Bitcoin's philosophy and introduces the programmability required to have a voting system built on Bitcoin. The implementation of the level-up mechanism and the proposal fee provide a solution to governance spam, while the Optimistic Batch Cleanup Protocol ensures the system remains efficient and sustainable to run.

6. References

[1] Nakamoto, S. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*.
[2] Jacoby, L. (2025, August 29). 12 most popular types of cryptocurrency. Bankrate. https://www.bankrate.com/investing/types-of-cryptocurrency/
[3] Ali, M., et al. (2021). *Stacks 2.0: Apps and Smart Contracts for Bitcoin*. Whitepaper.
[4] Dashjr, L. (2016) BIP 2: BIP process, revised, *Bitcoin Improvement Proposals*. https://github.com/bitcoin/bips/blob/master/bip-0002.mediawiki
[5] Garzik, J. (2015) "BIP 102: Block size increase to 2MB," *Bitcoin Improvement Proposals*. https://github.com/bitcoin/bips/blob/master/bip-0102.mediawiki
[6] Token Metrics. (2023, August). *Bitcoin vs Bitcoin Cash - Key Differences and Similarities*. https://www.tokenmetrics.com/blog/bitcoin-vs-bitcoin-cash
[7] Snapshot Labs. (2020). *Snapshot: Gasless Voting*. https://docs.snapshot.box/
[8] Benet, J. (2014). *IPFS - content addressed, versioned, P2P file system*. arXiv. https://doi.org/10.48550/arXiv.1407.3561
[9] Pawlak, M., Poniszewska-Maranda, A., & Kryvinska, N. (2018). Voting process with blockchain technology: Auditable blockchain voting system (ABVS). In *10th International Conference on Intelligent Networking and Collaborative Systems (INCoS-2018)* (pp. 233–244). Springer. https://doi.org/10.1007/978-3-030-00350-4_23
[10] F. Müffke, "SIP-010: Standard Trait Definition for Fungible Tokens," *Stacks Improvement Proposals*, 2020. [Online]. Available: https://github.com/stacksgov/sips
[11] fiatjaf. (2020). *Nostr: Notes and other stuff transmitted by relays* [Source code]. GitHub. https://github.com/nostr-protocol/nostr