Bitcoin NG: A Scalable Blockchain Protocol

Bitcoin NG is a blockchain protocol designed to improve scalability and transaction throughput by decoupling leader election from transaction serialization. Unlike Bitcoin, where block creation is limited by the leader election process, Bitcoin NG allows continuous transaction processing.

Protocol Overview

Bitcoin NG introduces two types of blocks:

- **Key Blocks**: Used for leader election, containing proof-of-work and a public key.
- **Microblocks**: Created by the elected leader to serialize transactions at a higher rate without additional proof-of-work.

Key Features & Improvements

- **Continuous Transaction Processing**: Unlike Bitcoin, where transactions must wait for the next block, Bitcoin NG allows transactions to be included in microblocks instantly.
- **Reduced Forking & Mining Waste**: Since microblocks do not affect chain weight, forks occur less frequently, minimizing wasted mining power.
- **Fair Compensation Model**: Mining rewards are split between the current and next leader (40% vs. 60%) to incentivize honest behavior and discourage selfish mining.
- **Security & Double-Spending Prevention**: Microblock forking is mitigated through a "poison transaction" mechanism that penalizes dishonest leaders.

Performance & Experimental Results

Bitcoin NG improves transaction throughput by increasing microblock frequency while maintaining network security. Experiments show that Bitcoin NG achieves lower consensus delay, better mining power utilization, and improved fairness compared to Bitcoin.

Challenges & Considerations

- **Microblock Forks**: Frequent but quickly resolved.
- **Censorship Resistance**: A malicious leader can withhold transactions, but only for their short epoch.
- **Mining Power Variation Resilience**: Key block difficulty adjustment ensures stability despite mining power fluctuations.

Overall, Bitcoin NG enhances Bitcoin’s scalability and efficiency while preserving its core security principles, making it a promising solution for high-throughput blockchain applications.

Snow Protocol Family and Avalanche Payment System

The Snow Protocol family operates by randomly sampling the network, ensuring rapid convergence to an irreversible state. Like Nakamoto consensus, it provides a probabilistic safety guarantee but is more energy-efficient as it does not rely on Proof-of-Work (PoW). Instead, it eliminates bottlenecks by requiring only O(1) communication per node per round, unlike classical consensus protocols that require O(n).

Model and Goals

- **Safety:** The protocol provides a probabilistic safety guarantee comparable to traditional consensus models.

- **Liveness:** Ensures the system can finalize decisions efficiently unless an adversary controls a significant portion of the network.

Protocol Design

1. **Slush:** A gossip-based protocol where nodes adopt a transaction's color based on responses from randomly sampled peers. It ensures eventual agreement but lacks Byzantine Fault Tolerance (BFT).

2. **Snowflake:** Enhances Slush by introducing a counter to track a node's conviction in a color. The counter resets on color change and increments when queries yield consistent responses.

3. **Snowball:** Further strengthens Snowflake by tracking accumulated confidence over multiple rounds. Nodes switch colors only when the new color has higher confidence.

Analysis

- **Safety:** Slush ensures rapid convergence, Snowflake introduces probabilistic irreversibility detection, and Snowball reinforces decisions, making reversals exponentially harder.

- **Liveness:** The system guarantees termination if the buffer zone remains positive. Adversarial control over nodes slows but does not prevent finalization.

Avalanche: A Peer-to-Peer Payment System

Avalanche builds on Snowball and organizes transactions in a **Directed Acyclic Graph (DAG)** to enhance efficiency. Each transaction is a vertex, inheriting votes from its ancestors to reduce redundant validations.

- **Transaction Structure:** Transactions reference parents, forming an ancestor-progeny relationship leading back to the genesis vertex.

- **Conflict Resolution:** Avalanche runs a Snowball instance for each conflict set, ensuring transaction validity and preventing double-spending.

- **Multi-UTXO Model:** Inherits Bitcoin's UTXO model, using transaction input pairs as DAG vertices. A transaction is only accepted if all inputs are validated.

Optimizations

- **Efficient Confidence Updates:** Only updated when a descendant gains a chit, reducing costly DAG traversals.

- **Optimized Query Processing:** Nodes stop querying once a threshold is met, improving speed and efficiency.

By leveraging these protocols, Avalanche provides a scalable, energy-efficient alternative to PoW-based systems while ensuring fast and secure transaction finalization.