Coded Transaction Broadcasting for High-throughput Blockchains

Authors: Lei Yang (MIT CSAIL), Yossi Gilad (Hebrew University of Jerusalem), Mohammad Alizadeh (MIT CSAIL)

Abstract

High-throughput blockchains require efficient transaction broadcast mechanisms capable of delivering transactions to most network nodes with minimal bandwidth overhead and latency. Existing schemes either incur high latency or are vulnerable to adversarial network nodes. Strokkur, a new transaction broadcasting mechanism, addresses these issues by avoiding explicit coordination through randomized transaction coding. Strokkur nodes send codewords—XOR sums of multiple transactions selected at random—ensuring that almost every codeword is useful for decoding new transactions. This eliminates the need for coordination to determine missing transactions. Strokkur's coding strategy builds on LT codes, extending them to support multiple uncoordinated senders with partially-overlapping transaction data streams. It includes mechanisms to handle adversarial senders and a rate control algorithm for optimal codeword sending. Implemented in Golang, Strokkur supports 647k transactions per second on a single CPU core, demonstrating 2–7.6× less bandwidth usage and 9× lower latency than existing schemes under certain adversarial conditions.

Introduction

The blockchain technology landscape has seen rapid evolution with the advent of decentralized applications such as cryptocurrencies, smart contracts, NFTs, and decentralized finance (DeFi). A blockchain network comprises thousands of geographically distributed nodes that collectively operate as a replicated state machine. Nodes establish a peer-to-peer (P2P) network, participating in a consensus algorithm to agree on the order of transactions, which are essentially state machine updates.

A critical aspect of blockchain performance is the transaction broadcasting mechanism. Clients initiate transactions by sending them to one or more nodes, which then broadcast these transactions throughout the network. Traditional broadcasting methods, such as flooding, where nodes relay transactions to all peers, are inefficient for high-throughput blockchains due to excessive bandwidth consumption and latency issues. Modern blockchains processing thousands of transactions per second require advanced broadcasting schemes that optimize both bandwidth usage and latency.

The performance and scalability of blockchain networks hinge on their ability to handle transaction broadcasts efficiently. Traditional methods like flooding, which involve sending transactions to all peers, lead to excessive bandwidth usage and high latency. These limitations are particularly problematic in high-throughput blockchains where the volume of transactions is substantial. Newer approaches aim to balance the trade-offs between low bandwidth overhead and low latency while ensuring robustness against adversarial nodes. This paper introduces Strokkur, a transaction broadcasting mechanism that leverages randomized transaction coding to address these challenges.

Background

Understanding the context and motivation for this research requires a look at the evolution of blockchain technology and the inherent challenges in transaction broadcasting. Traditional blockchains like Bitcoin use a simple broadcasting method where each node forwards transactions to all its peers. While straightforward, this approach is not scalable as it results in high bandwidth usage and increased latency, especially as the network grows.

To mitigate these issues, various optimized broadcasting schemes have been proposed. Some approaches focus on reducing redundant transmissions by ensuring that each transaction is sent only once. However, these schemes often require explicit coordination among nodes, which can introduce latency and complexity. Additionally, these methods can be vulnerable to adversarial nodes that can disrupt the coordination process, leading to inefficiencies and potential security risks.

High-throughput blockchains, such as those used for decentralized finance (DeFi) and other high-frequency applications, demand even more efficient and resilient broadcasting mechanisms. These networks must handle a high volume of transactions per second while maintaining low latency and being robust against malicious actors. The need for a scalable and secure solution is the driving force behind the development of Strokkur.

Methods

Strokkur introduces a novel approach to transaction broadcasting by leveraging randomized transaction coding, specifically LT codes—a popular class of rate less erasure codes. The core idea is to use codewords, which are XOR sums of randomly selected transactions, to transmit data. This approach significantly reduces the need for explicit coordination among nodes, as almost every codeword is useful for the receiver to decode new transactions.

Key Components of Strokkur:

1. **Randomized Transaction Coding**: Nodes send out codewords, which are combinations of multiple transactions. This randomness ensures that nodes do not need to explicitly coordinate to determine which transactions are missing at the receiver end.
2. **LT Codes Extension**: Strokkur extends LT codes to support multiple, uncoordinated senders and partially overlapping streams of transaction data. This extension is crucial for maintaining high throughput and robustness in a decentralized environment.
3. **Adversarial Resilience**: The design includes mechanisms to detect and cope with adversarial senders that might inject corrupt codewords into the network. This is achieved through verification processes and selective retransmissions.
4. **Rate Control Algorithm**: Each node independently determines the appropriate sending rate of codewords for each peer, ensuring efficient use of bandwidth and maintaining network stability.

The methodology behind Strokkur involves several critical innovations. First, the use of randomized transaction coding means that nodes broadcast codewords, which are XOR combinations of multiple transactions. This randomness ensures that nearly every codeword is useful for the receiving node, thereby reducing the need for explicit coordination to determine which transactions a node is missing.

The extension of LT codes to support multiple uncoordinated senders is another significant advancement. Traditional LT codes are designed for single-sender scenarios, but Strokkur adapts these codes to function

effectively in a decentralized network with many nodes broadcasting simultaneously. This adaptation is crucial for maintaining high throughput and ensuring that the network can handle a large volume of transactions efficiently.

To ensure robustness against adversarial nodes, Strokkur incorporates mechanisms to detect and mitigate the effects of corrupt codewords. Adversarial nodes may attempt to disrupt the network by sending invalid or malicious codewords. Strokkur's design includes verification processes that help nodes identify and discard corrupt codewords, maintaining the integrity of the transaction broadcasting process.

The rate control algorithm is another essential component of Strokkur. This algorithm allows each node to independently determine the optimal rate at which to send codewords to its peers. By adjusting the sending rate based on network conditions, nodes can optimize bandwidth usage and ensure stable and efficient broadcasting across the network.

Implementation

Strokkur was implemented in Golang, chosen for its efficiency and support for concurrency. The implementation involved several steps to ensure the system could handle high transaction throughput while maintaining low latency and robustness against adversarial nodes.

The first step in the implementation was developing the core coding mechanism based on LT codes. This involved creating algorithms for generating and decoding codewords, ensuring that they could handle multiple uncoordinated senders and overlapping transaction streams. The implementation also included mechanisms for detecting and mitigating corrupt codewords, crucial for maintaining the integrity of the broadcast process.

The rate control algorithm was another critical part of the implementation. This algorithm was designed to allow each node to dynamically adjust its sending rate based on current network conditions. The goal was to optimize bandwidth usage and maintain stable and efficient broadcasting across the network.

The final implementation step involved extensive testing and evaluation. The system was tested in various network conditions, including scenarios with adversarial nodes. The tests measured key performance metrics such as transaction throughput, bandwidth usage, and latency to ensure that Strokkur met its design goals.

Results

The implementation of Strokkur in Golang demonstrated the system's capacity to handle 647,000 transactions per second using a single CPU core. The evaluation involved a 19-node Internet deployment and large-scale simulations to compare Strokkur's performance against existing schemes like Bitcoin's and Shrec's.

Key Findings:

1. **Bandwidth Efficiency**: Strokkur was found to consume 2 to 7.6 times less bandwidth than Bitcoin's existing transaction broadcasting scheme. This reduction is attributed to the efficient coding strategy that minimizes redundant data transmission.

2. **Latency Reduction**: Compared to Shrec, Strokkur achieved 9 times lower latency when only 4% of the nodes were adversarial. This significant reduction in latency ensures quicker transaction confirmations, which is critical for high-throughput blockchain applications.
3. **Adversarial Robustness**: The mechanisms to handle corrupt codewords and the rate control algorithm proved effective in maintaining performance and security even in the presence of malicious nodes. The system's ability to quickly identify and mitigate the impact of adversarial behavior is a crucial aspect of its design.

The performance evaluation of Strokkur involved several key metrics, including transaction throughput, bandwidth usage, and latency. The system was tested in a real-world 19-node Internet deployment and through large-scale simulations to assess its effectiveness under various conditions.

The results showed that Strokkur could handle up to 647,000 transactions per second using only one CPU core. This high throughput demonstrates the efficiency of the coding mechanism and the scalability of the system. In terms of bandwidth usage, Strokkur consumed significantly less bandwidth compared to Bitcoin's existing broadcasting scheme, with reductions ranging from 2 to 7.6 times. This efficiency is attributed to the randomized transaction coding, which minimizes redundant data transmissions.

Latency is another critical metric for high-throughput blockchains. The results showed that Strokkur achieved 9 times lower latency compared to Shrec when only 4% of the nodes were adversarial. This low latency is crucial for applications that require fast transaction confirmations, such as financial services and decentralized exchanges.

The evaluation also demonstrated Strokkur's robustness against adversarial nodes. The mechanisms for detecting and mitigating corrupt codewords proved effective, ensuring that the system could maintain performance and security even in the presence of malicious actors. These findings highlight Strokkur's ability to provide a secure and efficient transaction broadcasting solution for high-throughput blockchains.

Discussion

The discussion section delves into how Strokkur addresses the inefficiencies and vulnerabilities of existing high-throughput blockchain transaction broadcasting methods.

Innovations and Contributions:

1. **Elimination of Explicit Coordination**: By utilizing randomized transaction coding, Strokkur eliminates the need for nodes to coordinate explicitly. This innovation reduces the complexity and overhead associated with traditional coordination mechanisms.
2. **Scalability**: Strokkur's ability to support multiple, uncoordinated senders with overlapping transaction streams ensures scalability in large, decentralized networks. The system can handle a high volume of transactions without significant degradation in performance.
3. **Robustness Against Adversarial Nodes**: The incorporation of adversarial resilience mechanisms enhances the network's security and reliability. The ability to detect and address corrupt codewords ensures the integrity of the transaction broadcasting process.
4. **Efficient Use of Resources**: The rate control algorithm optimizes the use of network bandwidth and computational resources, allowing each node to independently manage its broadcasting rate based on current network conditions.

The discussion highlights several key innovations and contributions of Strokkur. One of the most significant is the elimination of explicit coordination among nodes. Traditional broadcasting methods often require nodes to coordinate explicitly to determine which transactions are missing, leading to increased complexity and overhead. Strokkur's use of randomized transaction coding eliminates the need for such coordination, simplifying the broadcasting process and reducing overhead.

Scalability is another critical aspect of Strokkur. The system's ability to support multiple uncoordinated senders with overlapping transaction streams ensures that it can handle a high volume of transactions efficiently. This scalability is crucial for large decentralized networks that require high throughput and low latency.

The robustness of Strokkur against adversarial nodes is another important contribution. The mechanisms for detecting and mitigating corrupt codewords enhance the security and reliability of the network. These mechanisms ensure that the transaction broadcasting process remains intact even in the presence of malicious actors, maintaining the integrity of the blockchain.

The rate control algorithm is also a significant innovation. By allowing each node to independently manage its broadcasting rate based on current network conditions, the algorithm optimizes the use of network bandwidth and computational resources. This efficient use of resources ensures stable and efficient broadcasting across the network.

Implications for Blockchain Networks

Strokkur's design principles and implementation have significant implications for the future of high-throughput blockchain networks. By improving bandwidth efficiency and reducing latency, Strokkur enables faster and more reliable transaction processing. This is particularly beneficial for applications requiring high transaction throughput, such as financial services and decentralized exchanges.

The findings from this research suggest that Strokkur could play a critical role in enhancing the performance and scalability of blockchain networks. By providing a more efficient and robust transaction broadcasting mechanism, Strokkur addresses some of the key challenges faced by modern blockchains. This could lead to broader adoption of blockchain technology in various high-frequency applications, further driving innovation and development in the field.

Strokkur's ability to reduce bandwidth usage and latency has several practical implications. For financial services and decentralized exchanges, faster transaction confirmations can enhance user experience and enable more efficient trading. For other high-frequency applications, the improved performance and reliability of the network can support more complex and demanding use cases.

The robustness of Strokkur against adversarial nodes also has significant implications for the security of blockchain networks. By ensuring that the transaction broadcasting process remains secure even in the presence of malicious actors, Strokkur enhances the overall security and trustworthiness of the network. This is crucial for applications that require high levels of security and reliability, such as financial services and supply chain management.

Conclusion

Strokkur presents a robust, efficient, and scalable solution for transaction broadcasting in high-throughput blockchains. Its use of randomized transaction coding, based on LT codes, and innovative mechanisms for adversarial resilience and rate control, sets it apart from existing schemes. The empirical results demonstrate substantial improvements in bandwidth usage and latency, highlighting Strokkur's potential to enhance the performance and reliability of modern blockchain networks.

By addressing the core challenges of transaction broadcasting, Strokkur contributes to the broader field of blockchain technology, offering a viable path forward for the development of high-performance decentralized applications. Future work could explore further optimizations and adaptations of Strokkur's principles to other aspects of blockchain network operations.

In summary, Strokkur's innovative approach to transaction broadcasting addresses some of the key challenges faced by high-throughput blockchains. The use of randomized transaction coding, combined with mechanisms for adversarial resilience and rate control, provides a scalable, efficient, and robust solution. The empirical results demonstrate substantial improvements in bandwidth usage and latency, highlighting Strokkur's potential to enhance the performance and reliability of modern blockchain networks.

The research presented in this paper has significant implications for the future of blockchain technology. By providing a more efficient and robust transaction broadcasting mechanism, Strokkur can support the development of high-performance decentralized applications. This could lead to broader adoption of blockchain technology in various high-frequency applications, driving innovation and development in the field.

Ideas Related to Coded Transaction Broadcasting for High-throughput Blockchains

-Adaptive Coding Schemes for Dynamic Network Conditions Building on the principles of Strokkur's randomized transaction coding, an adaptive coding scheme could be developed to dynamically adjust coding parameters based on real-time network conditions. This approach would involve nodes monitoring network metrics such as latency, bandwidth availability, and node density, and then dynamically tuning the coding strategy to optimize performance. For instance, in periods of high network congestion, nodes could increase redundancy to ensure reliable delivery, whereas in low-traffic periods, they could reduce redundancy to conserve bandwidth. This idea connects with Strokkur's focus on efficiency and robustness, extending its adaptability to varying network states, enhancing its practical deployment in diverse environments. This concept can be explored further in works like "Network coding for distributed storage systems" by Dimakis et al., which discuss dynamic coding strategies.

-Integration of Machine Learning for Predictive Transaction Broadcasting Another idea is to integrate machine learning algorithms with the coded transaction broadcasting mechanism to predict transaction patterns and optimize the broadcasting process. Machine learning models could analyze historical transaction data to predict future transactions, allowing nodes to preemptively encode and broadcast transactions more efficiently. For example, nodes could predict peak transaction times and adjust their coding and broadcasting strategies accordingly. This predictive capability could further reduce latency and improve bandwidth usage by anticipating and mitigating network congestion before it

occurs. This idea is inspired by the intersection of blockchain technology and machine learning, as discussed in works like "Machine Learning for Predictive Blockchain Analytics" by Chiang et al., which highlights the potential of predictive models in optimizing blockchain operations.

-Enhanced Security Protocols Using Homomorphic Encryption While Strokkur includes mechanisms to handle adversarial nodes, integrating homomorphic encryption could further enhance the security of transaction broadcasting. Homomorphic encryption allows computations to be carried out on ciphertexts, generating encrypted results that, when decrypted, match the results of operations performed on the plaintext. Applying this to transaction broadcasting means nodes could encode transactions and broadcast them without exposing sensitive information, even if adversaries intercept the coded data. This would provide an additional layer of security, ensuring the confidentiality and integrity of transactions in hostile environments. The concept relates closely to the need for robust adversarial resilience in Strokkur and aligns with advancements in cryptographic methods discussed in "Fully Homomorphic Encryption Using Ideal Lattices" by Gentry, which explores practical implementations of homomorphic encryption for secure data processing.