Monadring: A lightweight consensus algorithm for organizing subnets upon a blockchain system

Zhang Yu

June 15, 2024

Abstract

Existing blockchain networks are often large-scale, requiring transactions to be synchronized across the entire network to reach consensus, and on-chain computations can be prohibitively expensive, making many CPU-sensitive computations infeasible. Inspired by the structure of IBM's token ring networks, we propose a lightweight consensus algorithm called Monadring to address these issues. Monadring allows nodes that are already part of a large blockchain network to form a smaller subnetwork to perform computations more quickly and cheaply, while still maintaining the same security guarantees as the main blockchain network. The paper details the design and implementation of the Monadring algorithm, and evaluates its performance and feasibility through simulation experiments. This research contributes to enhancing the practical utility of blockchain technology in large-scale application scenarios.

1 Introduction

Recent blockchain systems have often adopted faster and more energy-efficient consensus protocols like Ouroboros, BABE, and Tendermint. Blockchain networks are typically open and permissionless, and we can define the degree of decentralization of such networks by the number of participating nodes. Leveraging the consensus algorithms mentioned earlier, it is possible to rapidly construct new blockchain networks. However, decentralized applications often cannot flexibly customize their degree of decentralization based on the importance of their underlying data.

For example, decentralized social media applications may require faster response times and lower storage costs compared to decentralized finance (DeFi) applications, and hence may need a lower degree of decentralization.

This flexibility to adjust the level of decentralization based on the needs of the application is an important consideration that has not been fully addressed in existing blockchain architectures.

Building a subnetwork over an existing blockchain system is a possible way to address this issue. By drawing inspiration from the token ring architecture, we propose a novel consensus algorithm called Monadring that aims to enable nodes within an existing blockchain network to form smaller, lightweight subnetworks capable of performing computations more efficiently and cost-effectively, while still maintaining the same security guarantees as the main blockchain.

Token ring network operates at the data link layer (Layer 2) of the OSI/RM model. It was designed to solve the problem of physical link contention just like Ethernet. In a token ring network, a token is passed sequentially from one node to the next, granting the holder the right to transmit data. This token-based system for managing access to the shared medium bears some similarity to the consensus mechanisms used in blockchain networks for selecting block producer.

Despite the fact that token ring networks have largely fallen out of favor in modern networking due to their limited scalability and other drawbacks, the underlying principles of their decentralized, token-based structure could provide valuable insights for designing a lightweight consensus algorithm for small blockchain systems.

2 Model and Definitions

Hostnet and subnet. Consider a network composed of a set of participants V in which the majority obey a protocol to reach Byzantine Agreement and finality[**?**] over ledger L. S_i is a subset of V, $V = S_0 \cup S_1 \cup .. \cup S_{n-1}$.

A participant $v_j \in V$ could be a member of any S_i at meantime. We call V is a *hostnet* and S_i is a *subnet* of V.

Subnet ledger, topology and token. A subnet S_i maintains an independent ledger L_i, but it is not a sub-set of the ledger L of the hostnet. Instead, only the state root of the subnet ledger would go into the hostnet ledger, $L_i \nsubseteq L, f(L_i) \in L$. The state root could be simply calculated by using a sparse merkle tree generated from the subnet ledger L_i.

Assume the hostnet ledger L maintains all the subnets information in form of a (*subnet id*, *node list*) map which is under Byzantine Agreement.

The participants of the subnet strictly follows the sequence of the *node list* to form a ring topology.

A *token* T of a subnet with n nodes is a special signal circulates around the ring. The token carries a group of transactions from its sender and the sender's forehead recursively. Each transaction contains a modification event over the subnet ledger and a number q.

Whenever a node receives the token, it ought to apply all the transactions to its local ledger and decrease the associated number q by 1. If the number q of a transaction decreased to zero, it should be removed from the token. After handling all transactions in the token, the node should handle transactions from its local queue and put them all into the token with initial number $q = n$ and deliver the token to its successor.

Blind challenge.