Bachelor Thesis Topic Proposal

Title: Distributed Ledger Technology Mechanism in a Dynamic P2P Network

The goal is to implement a DLT system, within a P2P network of edge servers to record and manage network state information. This system must adapt to the dynamic nature of the network, where bootstrap nodes can change location and characteristics, impacting factors like latency. Incorporating a Distributed Ledger Technology (DLT) system into a dynamic, decentralized peer-to-peer network of edge servers offers a solution for managing and recording network state information. This implementation becomes particularly crucial in networks with dynamic bootstrap nodes.

1.1. Definition and Collection of Network Metrics: The first step involves defining the critical network metrics to be collected, including the number of peers, latency, bandwidth, throughput, and node capacity. Each edge server in the network is equipped with automated tools to periodically collect these metrics. This constant data collection ensures the provision of up-to-date information regarding the network's state.

1.2. Blockchain Integration and Node Setup: The integration of blockchain technology within the network is pivotal. It involves selecting a blockchain platform that aligns with the network's requirements. This platform could be an existing blockchain framework or a custom-built solution. Each edge server in the network operates as a blockchain node, maintaining the blockchain ledger and ensuring network decentralization.

1.3. Data Formatting and Writing to Blockchain: The collected network state data is formatted into transactions or records suitable for blockchain entry. Each edge server then adds its data to the blockchain by creating new blocks and broadcasting them to the network. Cryptographic methods like digital signatures secure and authenticate the data, maintaining the integrity of the blockchain.

1.4. Consensus Mechanism for Data Validation: A suitable consensus mechanism is chosen to align with the network's operational and performance requirements. This mechanism validates new blocks broadcasted by edge servers, ensuring that only accurate and verified data is appended to the blockchain.

1.5. Access, Synchronization: New edge servers access the blockchain through a current bootstrap node to analyse the network's state and performance history. All servers in the network continuously synchronize their local blockchain ledger with the network.

1.6. Possibility of Network Switching: If the increased latency significantly impacts performance, the edge server might evaluate the possibility of switching to a more optimal network (e.g., the other nearby P2P network with lower latency). This decision can be automated based on predefined thresholds for acceptable performance metrics, including latency.

2. Decision Process of an Edge Server in Overlapping P2P Networks:

2.1. Initial Data Gathering: An edge server, located in a region with overlapping P2P networks (for example Network A and Network B), starts by gathering data about each network. This involves connecting to the blockchain ledger of each network, through the currently designated bootstrap nodes.

2.2. Analysing Network Metrics: The edge server analyses the blockchain data from both networks. This data includes metrics such as overall network latency, throughput, number of active peers, resource utilization, and historical performance metrics.

2.3. Evaluating Network Health and Suitability: The edge server assesses each network's health and how well they align with its requirements. For instance, if the edge server is more latency-sensitive, it might prefer a network with lower average latency. In addition to current metrics, the server considers trends and historical data, which can provide insights into network stability and performance fluctuations.

2.4. Considering Dynamic Bootstrap Node Locations: Since bootstrap nodes can dynamically change locations, the edge server also evaluates the network's resilience to these changes. A network that maintains stable performance despite changes in bootstrap node locations might be more desirable. Based on this comprehensive analysis, the edge server decides which network to join. The decision is primarily driven by which network best meets its operational requirements and the needs of its clients.

2.5. Joining the Chosen Network: After deciding, the edge server initiates the joining process. This might involve interacting with the network's blockchain to register itself as a new node and start contributing to the network's operations. Post-joining, the edge server continuously monitors network conditions. If network circumstances change significantly (e.g., a drastic increase in latency), the edge server might reassess its network affiliation.

2.6. Example Scenario: Consider an edge server that serves latency-sensitive applications. It is located where Network A and Network B overlap. Network A has lower latency but higher variability. Network B has slightly higher latency but more stability. The edge server, after analysing blockchain data from both networks, might choose to join Network B, valuing stability for its latency-sensitive applications.

2.7. Process:

- Metric Collection: Edge Servers collect network metrics.
- Data Format and Write: Format data and write to the blockchain.
- Consensus Validation: Other nodes validate the new data via consensus.
- Bootstrap Update: Update and synchronize bootstrap node records.
- New Server Analysis: New edge server analyses blockchain data.
- Join Network: Server joins the network based on analysis.