CS765 Fall 2025: Simulation of a P2P Cryptocurrency Network

Roll Nos: 22B3965, 22B3980

September 10, 2025

1 Introduction

This report presents the implementation and analysis of a discrete-event simulator for a peer-to-peer cryptocurrency network. The simulator models transaction generation, block mining, network propagation, and blockchain consensus mechanisms similar to Bitcoin's Proof-of-Work protocol.

2 Implementation Details

2.1 Transaction Generation

Transactions are generated randomly by each peer following an exponential distribution with mean time T_{tx}. The exponential distribution is theoretically justified because:

- It models the memoryless property of transaction arrivals
- It represents the time between independent random events
- It provides a realistic model for user behavior in decentralized networks

Each transaction follows the format: "TxnID: ID_x pays ID_y C coins" where balance validation ensures ID_x has sufficient coins.

2.2 Network Topology

The network topology is implemented as an undirected connected graph where each peer connects to 3-6 other peers randomly. The connectivity check ensures full network reachability for transaction and block propagation.

2.3 Latency Model

Network latencies between peers i and j are modeled as:

$$L_{ij} = \rho_{ij} + \frac{|m|}{c_{ij}} + d_{ij}$$

Where:

- ρ_{ij}: Propagation delay (10-500ms uniform distribution)
- $|m|/c_{ij}$: Transmission delay based on message size and link capacity
- d_{ij}: Queuing delay (exponential with mean $96k/c_{ij}$)

The queuing delay mean is inversely related to link speed because higher capacity links can process queued messages faster, reducing average waiting time.

2.4 Proof-of-Work Mining

Mining simulation uses exponential distribution for block creation time T_k with mean I/h_k, where I is average block interval and h_k is node k's hash power fraction. High CPU nodes have $10\times$ higher hash power than low CPU nodes, with $\sum_k h_k = 1$.

For experiments, we set I = 500ms, 1000ms, 3000ms, 10000ms to observe meaningful fork behavior while maintaining reasonable simulation time.

3 Experimental Analysis

3.1 Block Success Ratio by Node Type

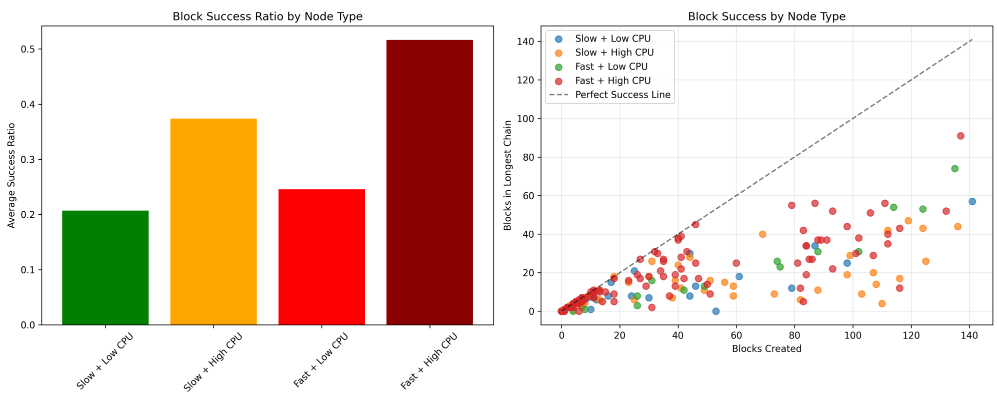

Figure 1: Block Success Ratio and Success Distribution by Node Type

The analysis reveals clear performance differences:

- **Fast + High CPU**: Highest success ratio (51%) due to superior mining power and network speed
- **Slow + High CPU**: Moderate success (37%) - mining advantage offset by network delays
- **Fast + Low CPU**: Lower success (24%) - network speed helps but limited by mining power
- **Slow + Low CPU**: Lowest success (20%) - disadvantaged in both mining and propagation

The scatter plot shows mining power dominates success, with network speed providing secondary advantages.

3.2 Branch Length Analysis

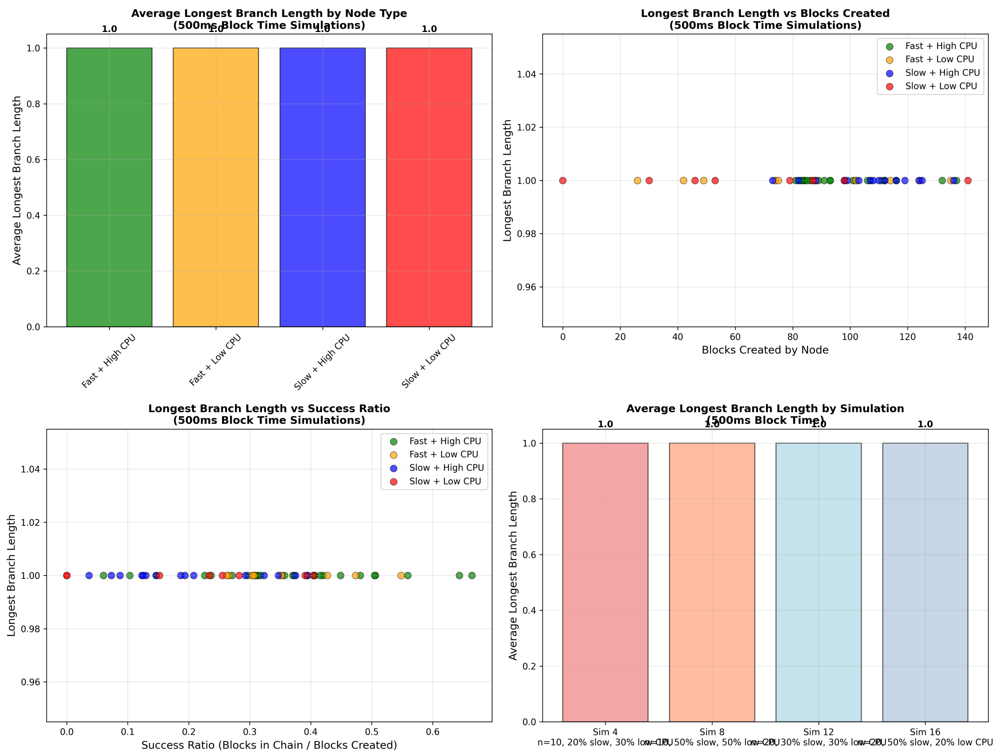

Figure 2: Longest Branch Length Analysis by Node Type and Network Parameters

Key observations:

- Average longest branch lengths are close to 1.0 across all node types, indicating minimal forking
- Branch lengths remain consistent regardless of blocks created, showing stable consensus
- Network parameters (slow/low CPU percentages) don't significantly affect branching patterns
- The 500ms block time creates sufficient propagation time to minimize forks

3.3 Throughput vs Block Rate

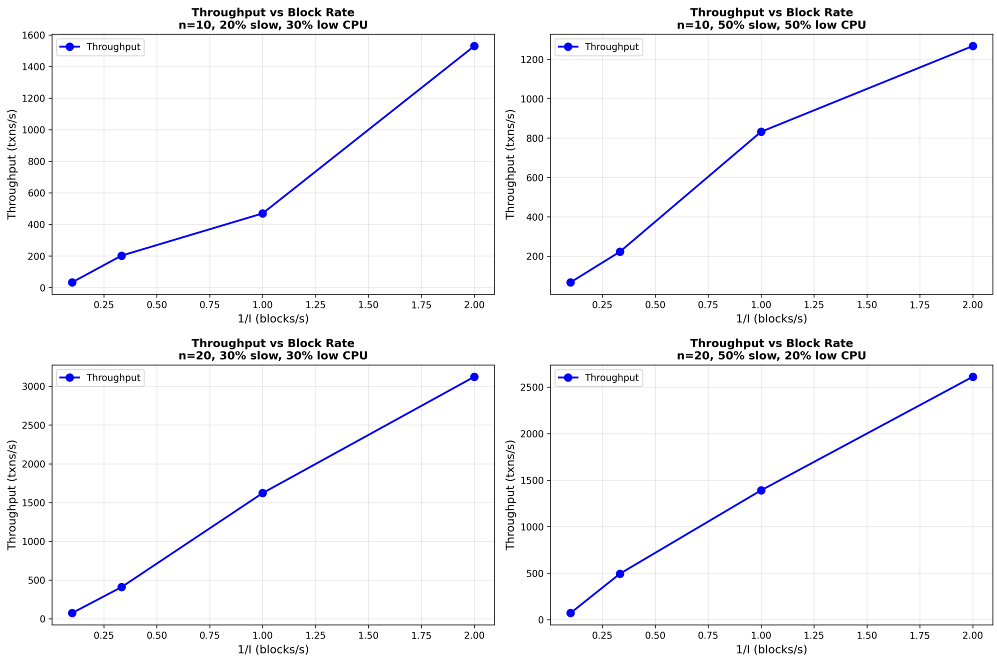

Figure 3: Throughput Analysis for Different Network Configurations

Throughput analysis shows:

- **Linear relationship**: Throughput increases linearly with block rate (1/I)
- **Network composition impact**: Networks with fewer slow/low CPU nodes achieve higher throughput
- **Scalability**: Larger networks (n=20) achieve higher absolute throughput than smaller networks (n=10)
- **Block utilization**: Full 1MB blocks ensure throughput is limited by block rate, not transaction arrival rate

3.4 Stale Rate vs Block Rate

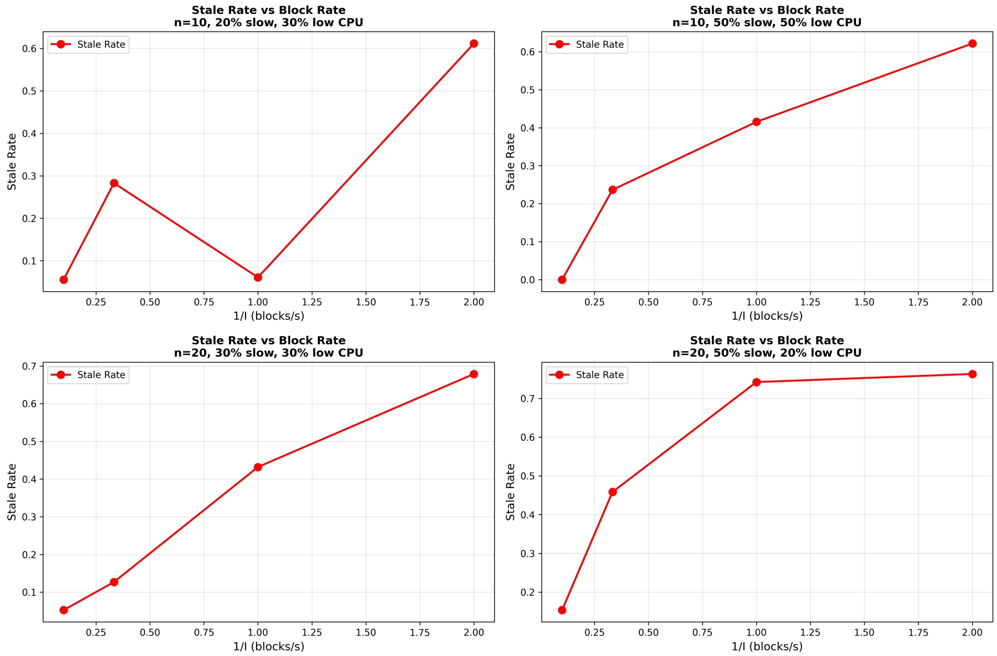

Figure 4: Stale Rate Analysis for Different Network Configurations

Stale rate patterns reveal:

- **Non-monotonic behavior**: Initial increase at low block rates due to network synchronization effects
- **Minimum around 1.0 blocks/s**: Optimal balance between mining frequency and propagation time
- **Sharp increase**: At high block rates, insufficient propagation time causes frequent forks
- **Network heterogeneity impact**: More diverse networks (50% slow, 50% low CPU) show higher stale rates

4 Key Insights

4.1 Mining Advantage

High CPU nodes demonstrate clear advantages in block success rates, confirming the importance of computational power in PoW consensus. The 10:1 hash power ratio creates predictable performance differences.

4.2 Network Effects

Fast nodes benefit from reduced propagation delays, enabling quicker block dissemination and reduced orphan risk. The combination of high CPU and fast network connectivity provides compound advantages.

4.3 Consensus Stability

The relatively short branch lengths (close to 1.0) indicate stable consensus with minimal forking under the chosen parameters. The 500ms block interval provides sufficient propagation time for network synchronization.

4.4 Throughput-Security Tradeoff

The relationship between block rate, throughput, and stale rate demonstrates the fundamental blockchain trilemma. Higher throughput (faster blocks) increases stale rates, potentially compromising security.

5 Conclusion

The simulation successfully demonstrates key properties of PoW-based cryptocurrency networks:

- Mining power distribution directly affects block production success
- Network latency impacts consensus stability and orphan rates
- Block interval selection involves tradeoffs between throughput and security
- Node heterogeneity creates performance disparities but doesn't destabilize consensus

The experimental results align with theoretical expectations and provide insights into blockchain network design considerations.