Simulation of a P2P Cryptocurrency Network

Aman Nehra (23B1051)

June 2025

1 Questions

1. What are the theoretical reasons of choosing the exponential distribution for interarrival time sampling?

Answer : The most important theoretical reason for using the exponential distribution for interarrival times is its **memoryless property**, which means that the probability of an event occurring in the future is independent of the past. Specifically:

$$P(X > s + t \mid X > s) = P(X > t)$$

This is particularly helpful as this makes the future transactions independent of the past which is true as all transactions are independently randomly generated.

It's correspondence to the Poisson distribution. The Poisson distribution is about the most basic count process, with constant event rate and no memory. This can help model the generation of transactions appropriately. Whenever a Poisson models the distribution of events then an Exponential models the distribution of inter-event times.

2. Why is the mean of dij (queuing delay) inversely related to cij (link speed)? Give justification for this choice.

Answer : The mean of $d_{i,j}$ (queuing delay) is inversely related to $c_{i,j}$ (link speed) because a higher link speed allows messages to be transmitted more quickly, reducing the time they spend in the queue. For example, if a link has higher capacity, it can process and forward packets at a faster rate, reducing the time packets spend in the queue. Conversely, a lower link speed results in congestion, as packets arrive faster than they are transmitted, increasing queuing delays. Therefore delay is inversely proportional to link speed.

3. The theoretical reason for choosing the exponential distribution for Tk and the choice of its values set by you during experiments.

Answer : The theoretical reason for choosing the exponential distribution for T_k (mean block arrival time) is the same as that for transaction interarrival time: the **memoryless property**. In the context of block mining, this means that the probability that a block will be mined in the next t seconds is independent of how much time has already elapsed without a block being found. Formally:

$$P(X > s + t \mid X > s) = P(X > t)$$

2 Analysis and Experimentation

$$\textit{longest chain contribution ratio} = \frac{\textit{total number of blocks mined by peer included in longest chain}}{\textit{total number of blocks mined by peer}}$$

Plots and their insights

While varying one parameter the fixed parameters were : n=40, z0=30, z1=30, t_tx=30, block_interval=100 , end_time = max(1000,10*block_interval)

The end_time is chosen in the above way so that a good amount of blocks are mined even if block_interval is as high as 800.

1. Varying values of z0(percentage of slow peers) versus Longest chain contribution ratio

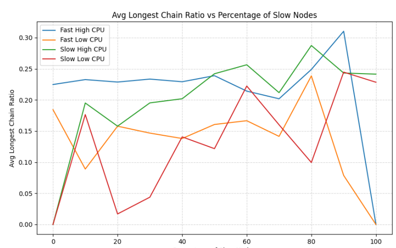

Figure 1: z0 v/s Average Longest chain ratio

The following observations and insights can be made:

- As the percentage of slow nodes (`z0`) increases(around 80%), the chain ratio for slow peers significantly drops.
- Fast peers, especially Fast High CPU nodes, consistently maintain a higher contribution to the main chain.
- Network speed has a dominant effect on longest chain participation.

2. Varying values of z1(percentage of low CPU peers) versus Longest chain contribution ratio

The following observations and insights can be made:

- High CPU nodes show stable chain ratios as the percentage of low CPU peers (`z1`) increases.

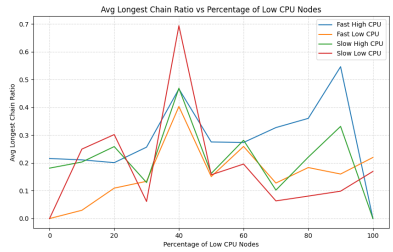

Figure 2: z1 v/s Average Longest chain ratio

- Low CPU peers contribute less as compared to High CPU peers to the main chain when the percentage of the Low CPU node increases(around 80-90%).
- CPU strength directly influences mining success.

3. Varying values of Transaction intervals(t_tx) versus Longest chain contribution ratio

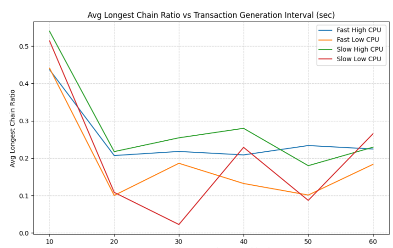

Figure 3: Transaction generation interval versus Average Longest chain ratio

The following observations and insights can be made:

- At higher transaction rate(i.e. low transaction generation intervals) we observe low ratios for all kind of peer which can be possible due to diverse blocks due to high transaction rates which implies more number of forks.
- When t_tx increases (fewer txns), blocks are more similar, leading to slightly higher ratios due to less number of forks.

4. Varying values of Mean block arrival interval(I) versus Longest chain contribution ratio

The following observations and insights can be made:

- An optimal block interval around **200–300s** maximizes consensus, with chain ratios close to 1.0.

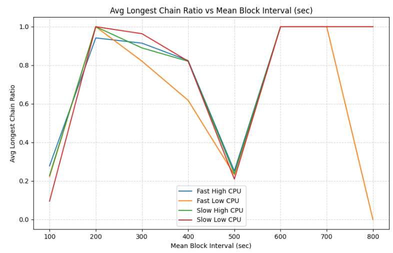

Figure 4: Mean block arrival interval versus Longest chain contribution ratio

- Too small intervals cause frequent forks, while too large intervals reduce mining activity, both leading to lower chain ratios.
- Thus, **extreme block intervals** can hurt network efficiency, highlighting a trade-off in block time selection.

5. Varying values of Number of Peers(n) versus Longest chain contribution ratio

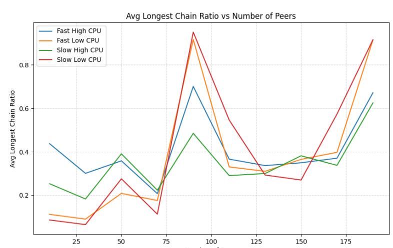

Figure 5: Number of Peers versus Longest chain contribution ratio

The following observations and insights can be made:

- The chain ratio varies non-monotonically with peer count; moderate network sizes (80-100) often show higher consensus.
- Increasing the number of peers can increase propagation delays and fork frequency, reducing the chain quality.
- An optimal number of peers balances decentralization and stability, avoiding both underutilization and network congestion.

Analysis of Branches

The run on which analysis was done was:
`python3 Simulator.py 30 40 --n 50 --Ttx 10 --I 30 --end 300`

1. Number of Branches per Peer

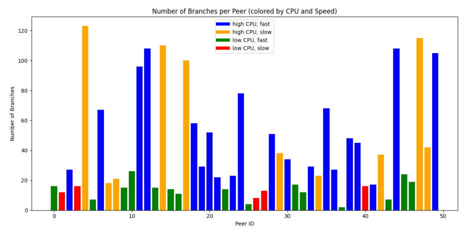

Figure 6: Number of Branches per Peer

The following observations and insights can be made:

- Peers with high computational power (especially `High CPU, Fast`) tend to create more branches due to frequent block generation.
- Many branches may indicate frequent forks, suggesting a trade-off between mining power and chain consistency.
- Reducing network latency or improving block propagation can mitigate excessive branching.

2. Branch length versus Peer Type

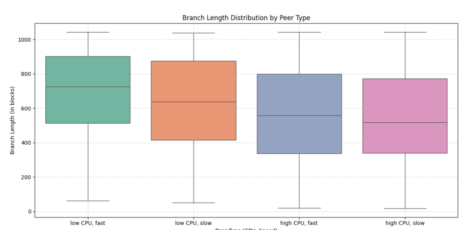

Figure 7: Branch length versus Peer Type

The following observations and insights can be made:

- `Low CPU` peers, especially when fast, tend to produce longer branches, possibly due to poor block propagation.
- These longer branches may consist of stale or orphaned blocks that never reach the main chain.

- `High CPU` peers generate shorter, higher-quality branches, suggesting better integration with the main chain.

3. Longest chain contribution ratio versus Peer ID and Type

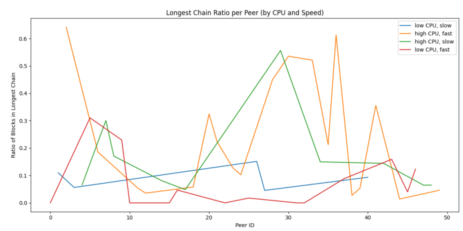

Figure 8: Longest Chain contribution ratio versus Peer ID and type

The following observations and insights can be made:

- `High CPU, Fast` peers dominate the longest chain, showing the highest contribution rates.
- `Low CPU` peers, especially slow ones, contribute minimally, indicating heterogeneous influence.
- This suggests a natural centralization towards more capable peers, relevant for incentive mechanisms.

Pictures of Block Trees

The following are the pictures of block trees of a few peers formed during the following run of the simulator:

`python3 Simulator.py 30 40 --n 10 --Ttx 40 --I 50 --end 300`

Figure 9: Block Tree of peer_0

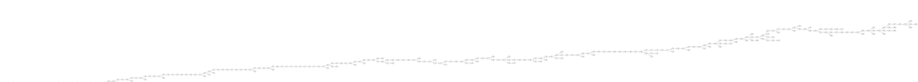

Figure 10: Block Tree of peer_1