CS765 Project Part-1
Simulation of a P2P Cryptocurrency Network

Sanchit Jindal (200020120), Sarthak Mittal (200050129), Virendra Kabra (200050157)

Spring 2023

Contents

1 Questions

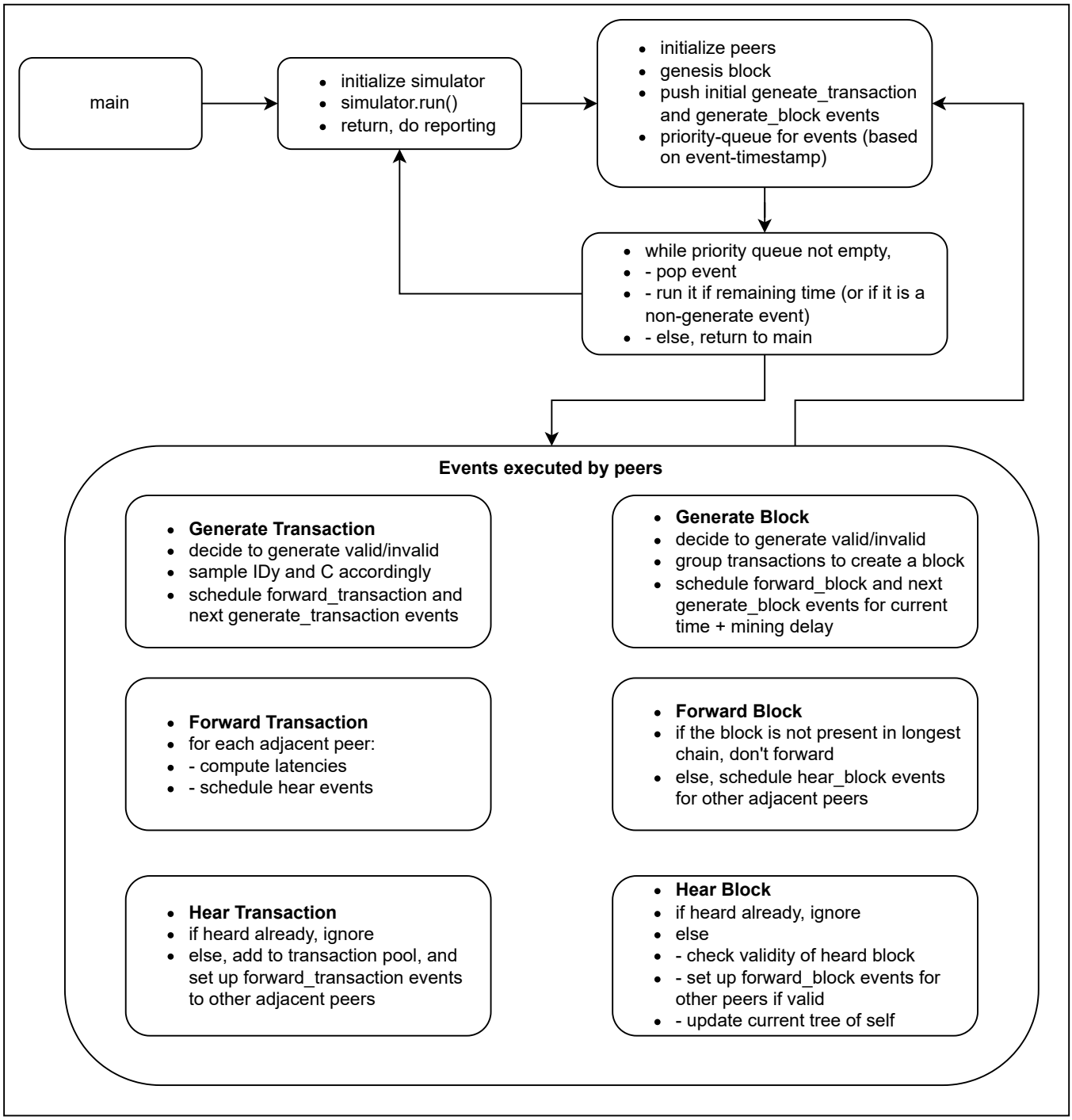
2. What are the theoretical reasons of choosing an exponential distribution for generating transactions?

- The Poisson distribution models events occurring with a constant mean rate (λ) and independent of the time since the last event. Transactions follow this memorylessness property. So, the inter-arrival time of generating transactions follows an exponential distribution (with mean inter-arrival time $\frac{1}{\lambda}$).

4. Why is the mean of d_{ij} inversely related to c_{ij}? Give justification for this choice.

- Link speed (c_{ij}) measures the number of bits that any of the two nodes can push through their connection. Queuing delay (d_{ij}) is the time for which a message waits in a queue at the respective node. If the link speed is less, then a message takes longer to transmit, and longer the waiting time of other messages at that node. Hence the inverse proportionality.

2 Design


main
initialize simulator
simulator.run()
return, do reporting
initialize peers
genesis block
push initial geneate_transaction and generate_block events
priority-queue for events (based on event-timestamp)
while priority queue not empty,
- pop event
- run it if remaining time (or if it is a non-generate event)
- else, return to main
Events executed by peers
Generate Transaction
decide to generate valid/invalid
sample IDy and C accordingly
schedule forward_transaction and next generate_transaction events
Generate Block
decide to generate valid/invalid
group transactions to create a block
schedule forward_block and next generate_block events for current time + mining delay
Forward Transaction
for each adjacent peer:
- compute latencies
- schedule hear events
Forward Block
if the block is not present in longest chain, don't forward
else, schedule hear_block events for other adjacent peers
Hear Transaction
if heard already, ignore
else, add to transaction pool, and set up forward_transaction events to other adjacent peers
Hear Block
if heard already, ignore
else
- check validity of heard block
- set up forward_block events for other peers if valid
- update current tree of self

3 Observations

The images are available within the code folder. Various experiment results are present in the outputs folder. The tree has been generated using the trees text file. Peer trees can be formed using peer text files. Ratios and lengths can also be plotted. Examples are shown below.

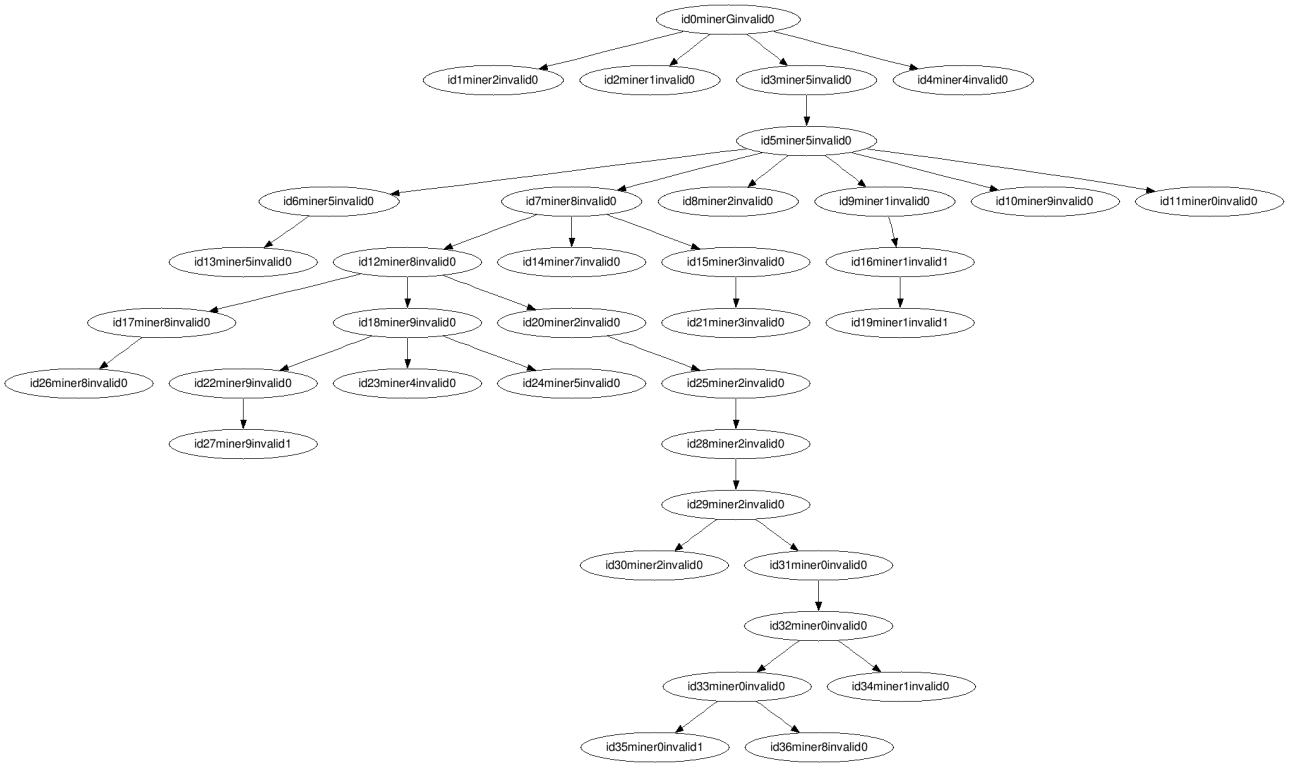


Figure 1: A sample Blockchain

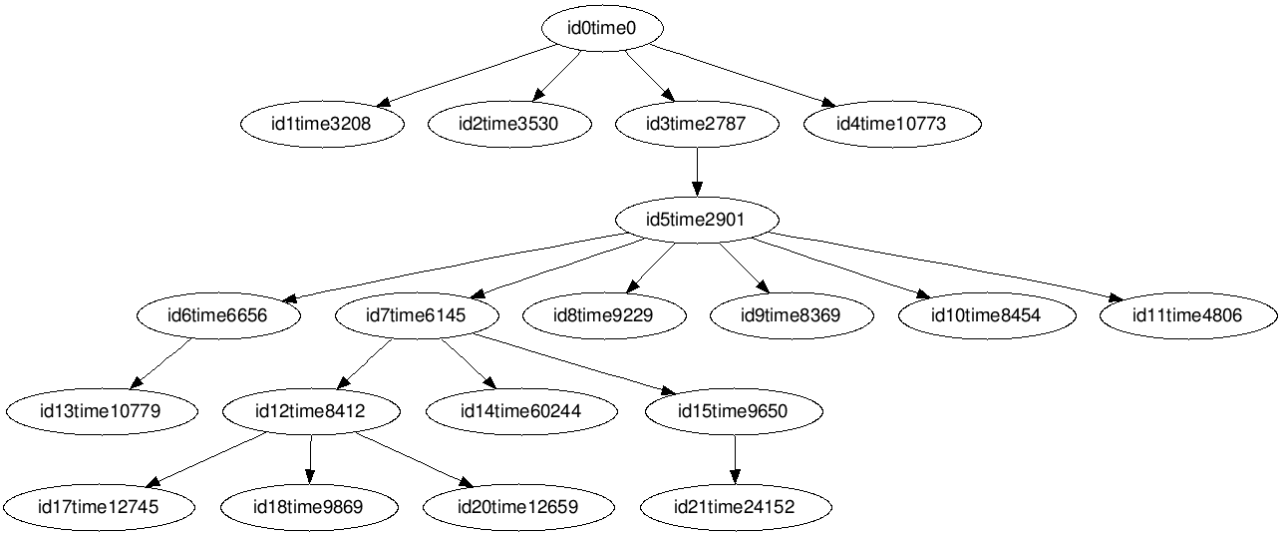


Figure 2: A sample tree of peer 1

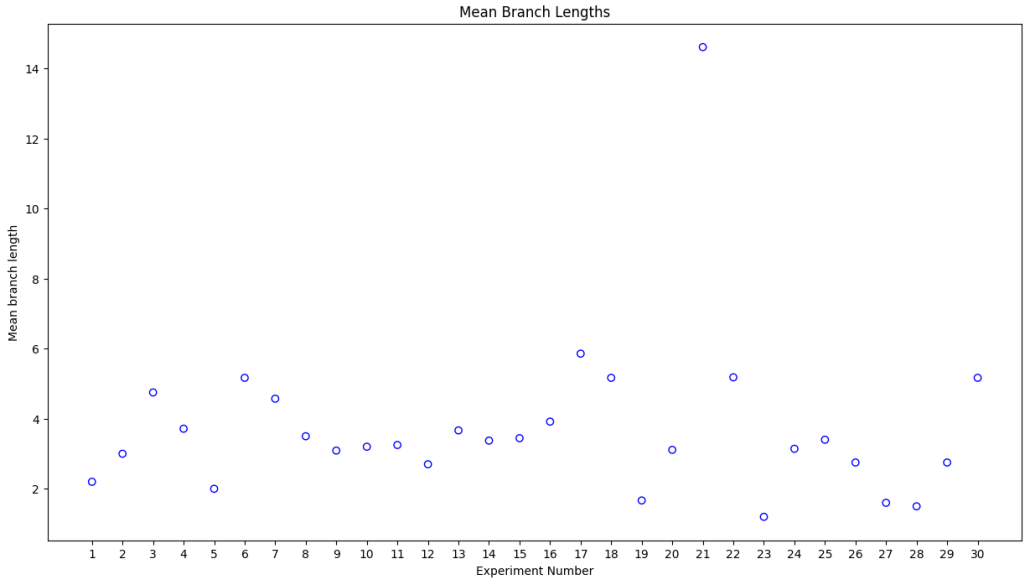


Figure 3: Average branch length

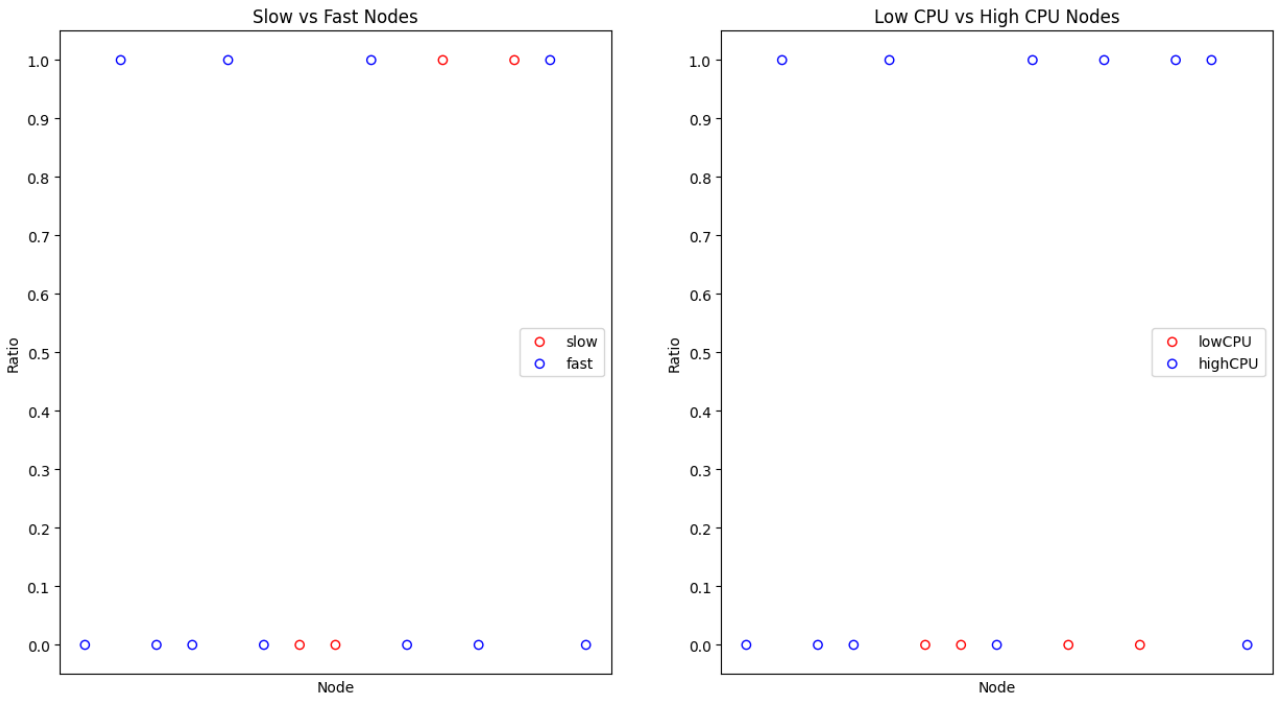


Figure 4: Ratio of generated nodes

3.1 Number of nodes

- In our system the number of nodes in the system are the number of active miners that mine block after certain time interval, and create transactions between each other
- Some of the nodes can be malicious that is they may allow invalid blocks to be added to their chain and forward such blocks to the neighbours

3.2 Slow Nodes

- Slow Nodes are nodes which have a lower Network Bandwidth and and any communication with them take more time
- Due to this there is a larger latency for any transaction or block traversal to and from these nodes
- As a result there blocks are usually heard less than the blocks of the other nodes and they have a smaller ratio of blocks in main chain to the blocks generated
- In our implementation we can provide the percentage of nodes that are slow

3.3 LowCPU Nodes

- Nodes with low CPU power require more time to perform operations and thus take more time to generate Blocks than the other nodes
- Due to this They have a smaller fraction of hashing power in the system and it is less likely that they will generate a node that is added to the longest blockchain
- In our implementation we can provide the percentage of nodes that have low CPU capabilities

3.4 Transaction Inter Arrival Time (T_{tx})

- Transaction Inter arrival time is the mean time that it takes for a node to generate transactions
- In our System Transactions have a 10% chance to be malicious that is they try to spend amount that they don't have
- Increasing Inter arrival time between leads to less number of transactions through all the nodes
- Increasing the transactions inter arrival time leads to the situation where all transactions are usually known to everyone (that is they are broadcast through the network faster than the generation rate)

3.5 Block Inter Arrival Time(T_{blk})

- Block Inter Arrival Time is the mean time between block generation by all the nodes
- In our implementation blocks consist of the parent, peer who mined them, the coinbase transaction providing 50 coins to that peer and a list of transactions that the generating node have already seen
- Increasing this value mean that it takes longer to generate blocks
- Hence, with high probability the blocks are broadcast before competing blocks can be made by other miners
- So, as the Inter Arrival Time increases there are fewer and fewer branches in the block chain and the the height of the tree also increases
- On the other hand if the Inter Arrival Time is decreased then most of the nodes create blocks around the same time without knowing about the other blocks, Hence multiple branches are created and the height of the tree also reduces