Simulation of a P2P Cryptocurrency Network
Report

Vignesh Badisa (22B0921)
Ravi Kishore (22B0959)
Samipa Samanta (24D0380)

Implementaion

Details of the implemented Discrete Event Simulator for a Peer-to-Peer Cryptocurrency network.

- Maintains an event-queue (sorted based on timestamp) and a clock that updates after every event
- The control flow of the program can bee seen in flowchart.pdf
- Respective nodes are called by simulator using Event attributes to execute the event, which adds new events to the queue if needed.

Reason to choose Exponential Distribution for Inter-arrival times

Each peer generates transactions randomly in time. The inter-arrival time between transactions generated by any peer is chosen from an exponential distribution whose mean time T_{tx} can be set as a parameter of the simulator. Now we show the theoretical reasons for why the inter-arrival times are chosen to be from an exponential distribution.

Consider the inter-arrival time between $(k+1)^{th}$ and k^{th} transaction as I(Random Variable which we shall show is from exponential distribution). Now for any small δ, where $\delta \to 0$, we can say that the probability of any peer generating a transaction in that time interval (δ) is proportional to δ:

$$P(\text{generating a new transaction in interval } \delta) \propto \delta$$

(More the sample time, more the probability.)

The proportionality constant has to be $1/T_{\text{tx}}$, since for $\delta = T_{\text{tx}}$, the total probability should be 1. Thus, we get:

$$P(\text{generating a new transaction in next } \delta \text{ interval}) = \frac{\delta}{T_{\text{tx}}}$$

Now, say I is atleast n folds of δ i.e $I > n\delta$. Then

$$p(I > n\delta) = (1 - \frac{\delta}{T_{\text{tx}}})^n$$

(i.e failing to generate txns in the first $n\delta$ intervals). Set $n\delta = x$. Now since x spans over the entire finite range ($0 \to \infty$, and δ being too small, we set n to be very large. i.e for $n \to \infty$, we can write

$$\text{As } n \to \infty, \quad P(I > x) = e^{-\beta x}, \quad \text{where } \beta = \frac{1}{T_{\text{tx}}}$$

which is exponential distribution. Hence Interarrival time follows exponential distribution. Now since we know the expected value i.e mean interarrival time is $1/\beta$ for exponential distribution. we get the mean interarrival time as T_{tx} which is hand with our analysis.

Mean of Queuing delay inversely related to link speed

The queuing delay d_{ij} models how long a message waits at node i before being forwarded to node j and c_{ij} is the link speed between node i and j.

If the link speed c_{ij} is slower, more and more messages pile up, resulting an increased queuing delay d_{ij}. If the link speed c_{ij} is faster then more messages can be pushed out , resulting a decreased queuing delay for other messages. So, the average queuing delay decreases if the link speed is faster. This is the reason $d_{ij} \propto \frac{1}{c_{ij}}$

Explanation for the choice of a particular mean for block inter-arrival time

Choice of Tk (Average Time required to mine a block) is very crucial as it has many trade-offs as follows,

- **Network delays :** If it is too small i.e, comparable to the network delays then there is a significant increase in forks.We can see that Fig 1 (Tk = 600) has no forks, Fig 2,3,4,5 (Tk = 6) have two forks from some blocks and Fig 6 (Tk = 1) has multiple forks.
- **With respect to N*=** $\frac{Tk*n}{Ttx}$**:** If it is large compared to Ttx ,it results in large number of pending transactions in the network, where as if it is comparable to Ttx then it results in wastage of hash-power as block have less no of transactions than the max value. This was observed in Experiments 2,3,4,5,6 which were creating blocks with less than 100 transactions.
- N* represents average no of new transactions generated during the average block mining time.

By considering the trade-offs one can choose Tk in the order of $1000 * \frac{Ttx}{n}$ such that 1000 new Transactions are created for every block mined, and Tk should be larger so that all peers receive the new mined block and update their chain before they mine. Following Experiments are conducted to verify these conclusions.

Experiments

We conducted the following experiments on 80 peers to verify the conclusions mentioned above and observe variations related to Node specifications.

Experiment 1: Block chain : z0 = 30, z1 = 30, ttx = 6, tk = 600, t = 60000
Experiment 2: Block chain : z0 = 30, z1 = 30, ttx = 6, tk = 6, t = 600
Experiment 3: Block chain : z0 = 30, z1 = 70, ttx = 6, tk = 6, t = 600
Experiment 4: Block chain : z0 = 70, z1 = 30, ttx = 6, tk = 6, t = 600
Experiment 5: Block chain : z0 = 70, z1 = 70, ttx = 6, tk = 6, t = 600
Experiment 6: Block chain : z0 = 30, z1 = 30, ttx = 1, tk = 1, t = 100

The following are the block chain drawn using visualization tool for each of the above experiment

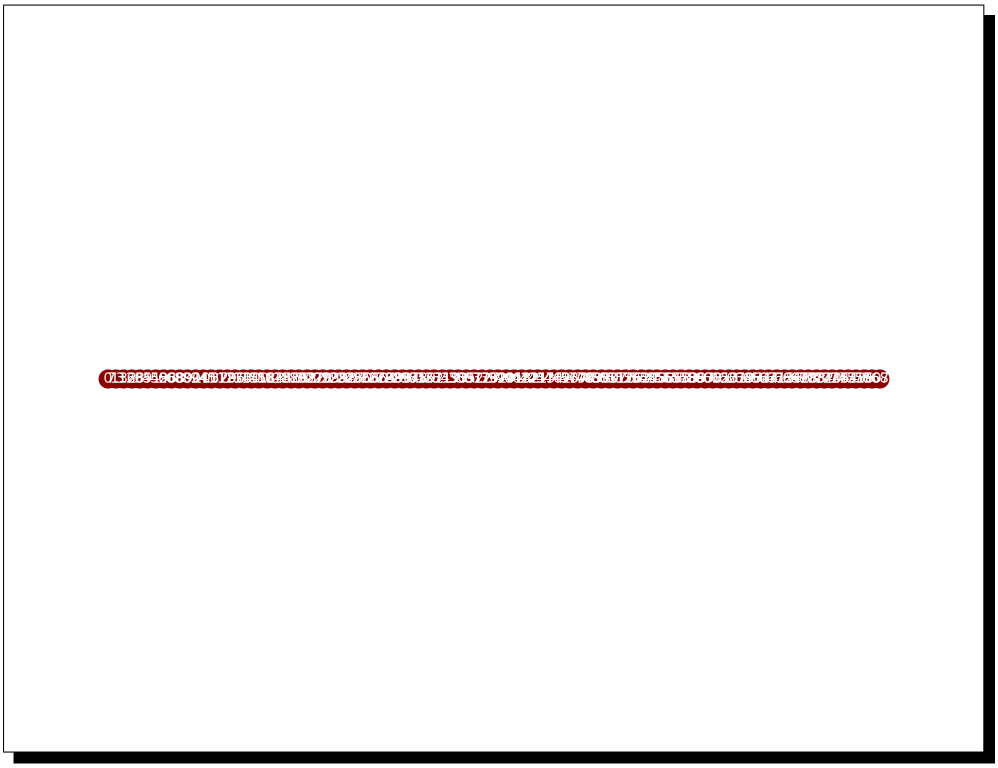

Figure 1: Block chain : z0 = 30, z1 = 30, ttx = 6, tk = 600

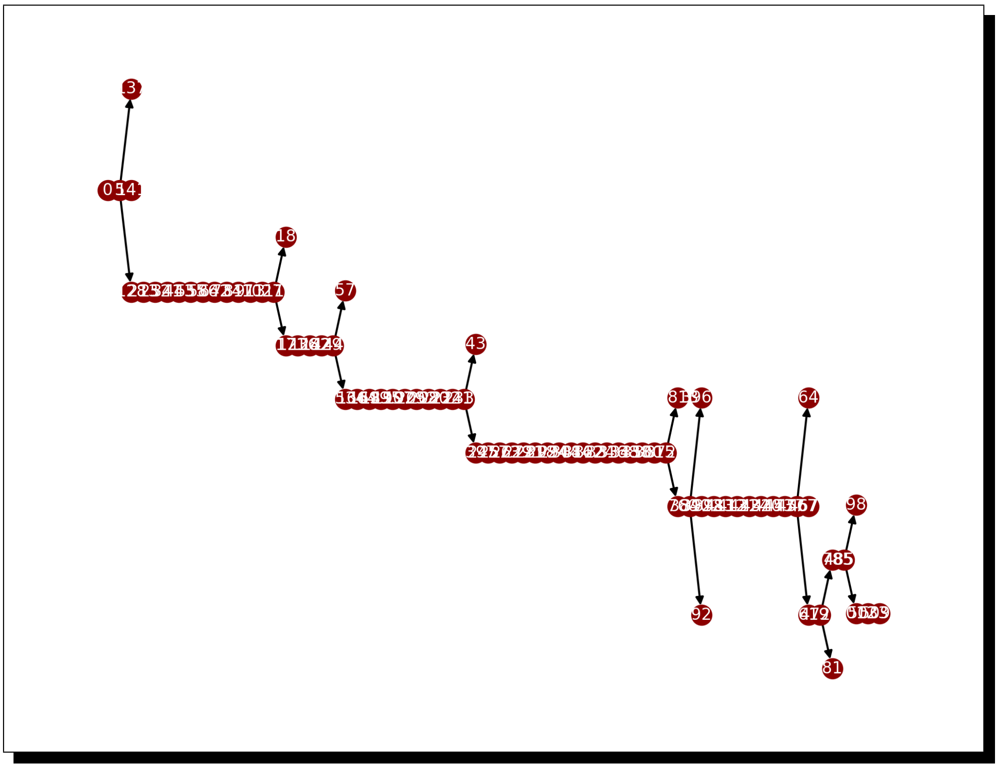

Figure 2: Block chain : z0 = 30, z1 = 30, ttx = 6, tk = 6

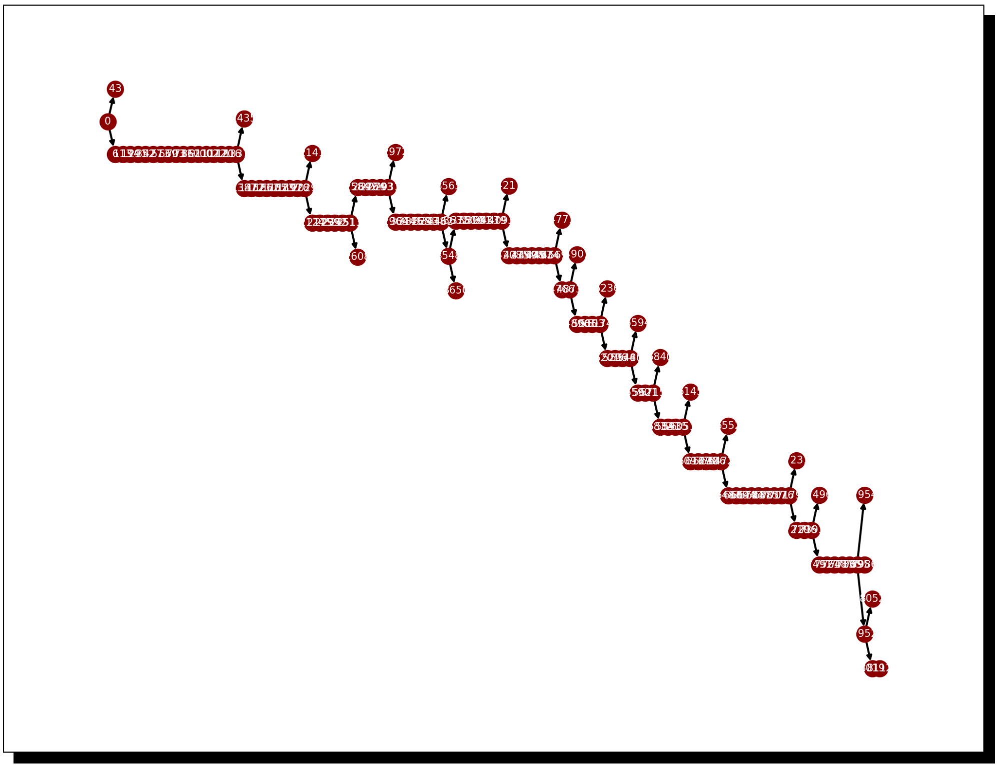

Figure 3: Block chain : z0 = 30, z1 = 70, ttx = 6, tk = 6

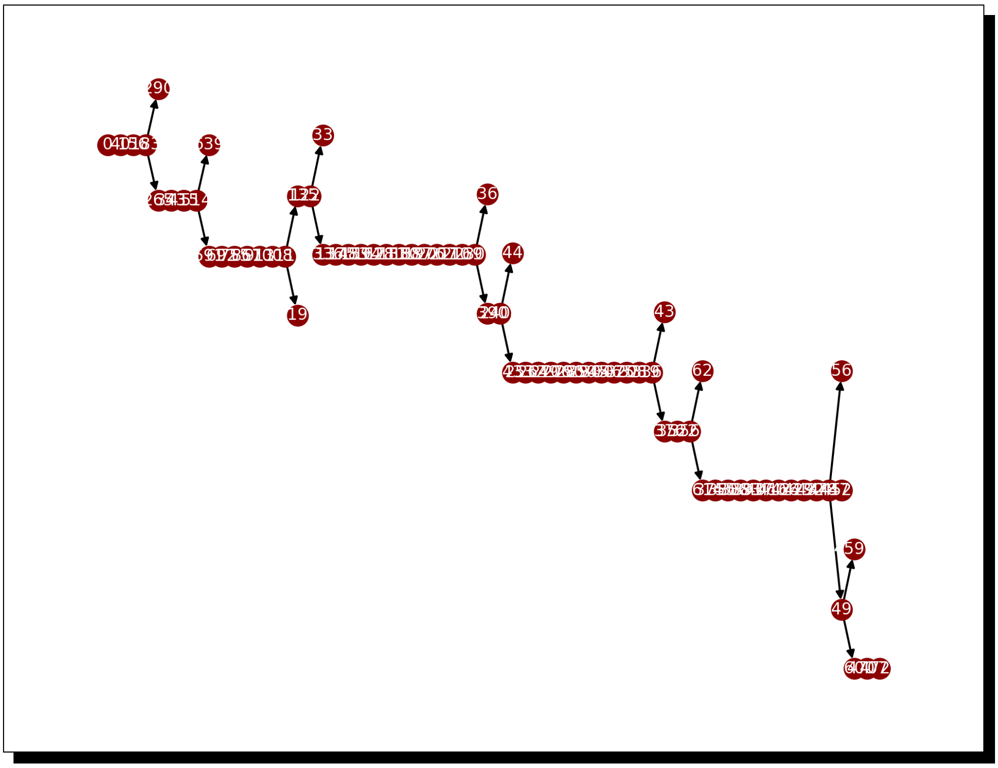

Figure 4: Block chain : z0 = 70, z1 = 30, ttx = 6, tk = 6

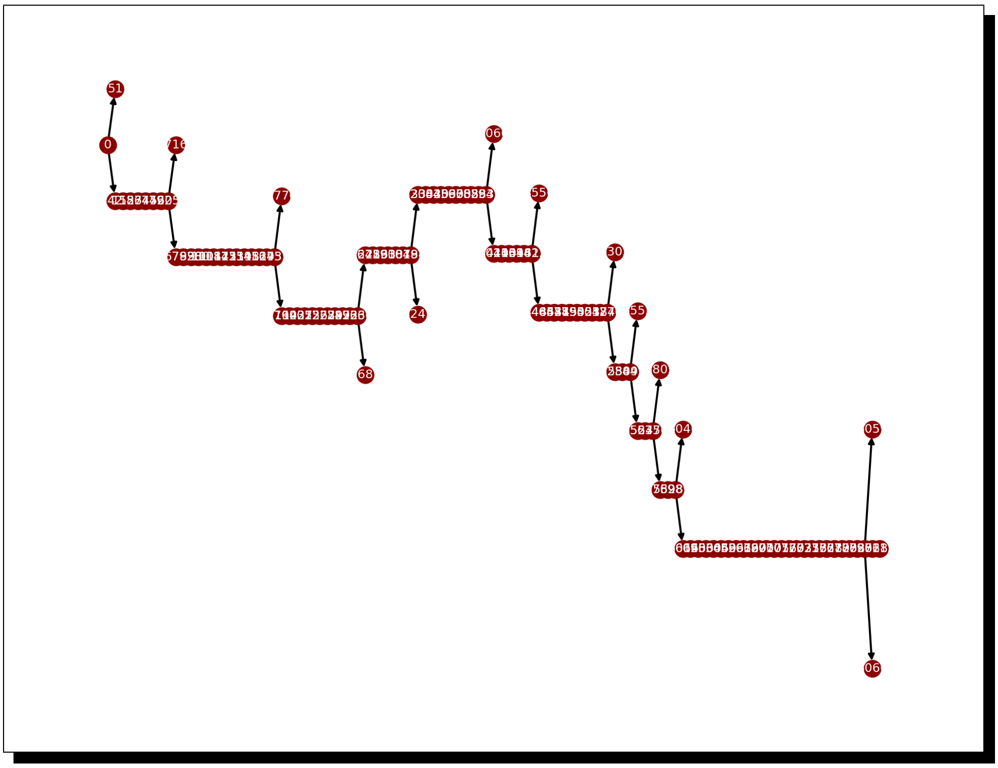

Figure 5: Block chain : z0 = 70, z1 = 70, ttx = 6, tk = 6

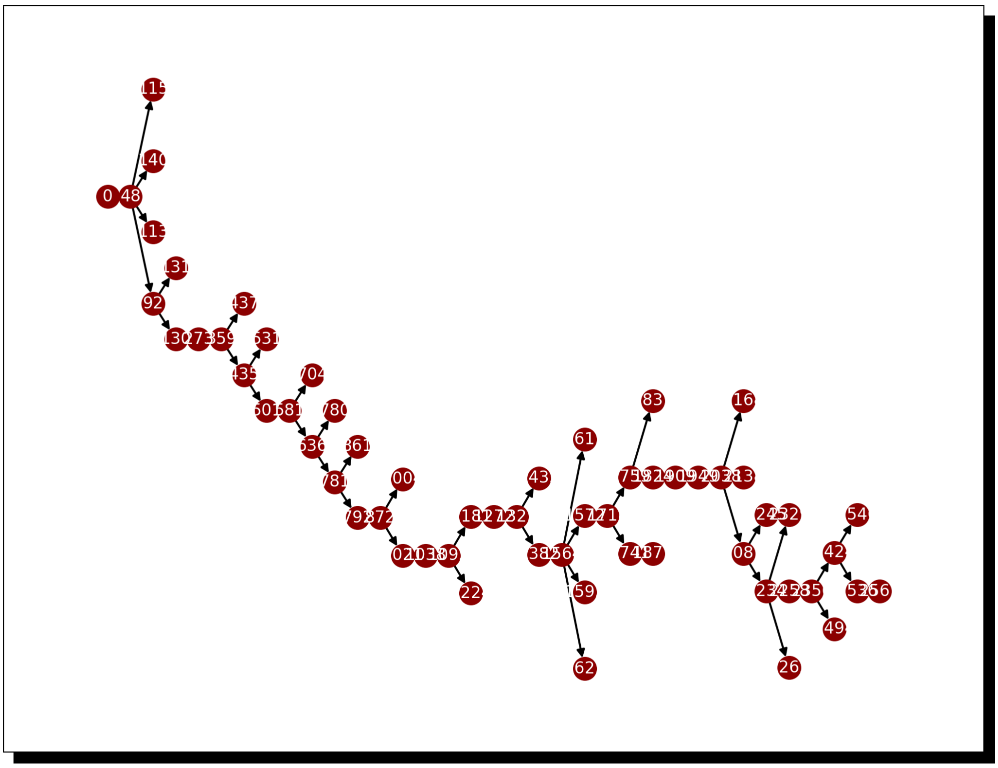

Figure 6: Block chain : z0 = 30, z1 = 30, ttx = 1, tk = 1

Further Observations

Apart from the observation on how blockchain changes with respect to different values of Tk and Ttx as discussed in previous section, we now analyse the pattern of rvalue and its dependency on hashpower and type of Newtork.

The following are tables (All slow and All fast) for simulation of 10 nodes to find the ratio of number of blocks in longest chain to that of those mined by it.

	Node 1	Node 2	Node 3	Node 4	Node 5	Node 6	Node 7	Node 8	Node 9	Node 10
is_highhash	1	1	1	0	1	0	1	1	1	0
rvalue (x/y)	0.91	0.90	1.00	0.00	0.86	0.67	1.00	1.00	1.00	1.00

Table 1: Node characteristics and computed r-values.

	Node 1	Node 2	Node 3	Node 4	Node 5	Node 6	Node 7	Node 8	Node 9	Node 10
is_highhash	0	1	1	1	1	0	1	0	1	1
rvalue (x/y)	1.00	1.00	0.88	0.93	1.00	1.00	1.00	1.00	1.00	1.00

Table 2: Node characteristics and computed r-values.

From the table we can see that the numerator of the rvalue no of mined blocks in the blockchain depends mainly on network connectivity (i.e, Fast or slow), but the denominator number of mined blocks is mainly depend on the hash power of the node. It can be explained as follows -

- High network connectivity implies, the mined block is propagated through most of the nodes as soon as it is mined increasing its chances of getting added to the blockchain, where as in case of low connectivity there is a higher chance that a fast peer with low hashpower may get his block added.
- High hash power implies less amount of time for mining compared to others which leads to higher number of mined blocks.