Discrete Event Blockchain Simulator

Vivek (210050077)
Chanikya (210050053)
Hemanth (210050117)

February 16, 2024

Contents

0.1 Question 2

The reason for choosing exponential distribution is that it is memoryless. This means that the time between two transactions is independent of the time that has passed since the last transaction. This is a good model for the real world where transactions are generated randomly in time.

0.2 Question 5

d_{ij} is randomly chosen from an exponential distribution with some mean 96kbits/ci,j. The mean of d_{ij} is inversely related to c_{ij} because as the link speed increases, the queuing delay decreases. This is because the queuing delay is the time taken to forward the message to the next node. As the link speed increases, the time taken to forward the message decreases. This is why the mean of d_{ij} is inversely related to c_{ij}.

0.3 Question 7

I: Desired average inter-arrival time between any two blocks from any two nodes. h_i: Hashing power of the i^{th} node. T_k: Meantime for each peer.

The probability that a peer with mean time $T_k = \frac{I}{h_i}$ generates a block in time x is given by:

$$P(I > x) = e^{-\frac{x}{T_k}} = e^{-\frac{hx}{I}},$$

where h is the hashing power of the peer.

The probability that none of the peers generate a block in time x is the product of the probabilities of each peer not generating a block in time x, Using independence of random variables, we get:

$$P(I_1 > x, I_2 > x, \ldots, I_n > x) = \prod_{i=1}^{n} P(I_i > x)$$

further simplifying, we get:

$$P(I_1 > x, I_2 > x, \ldots, I_n > x) = \prod_{i=1}^{n} e^{-\frac{h_i x}{I}} = e^{-\frac{x}{I}\sum_{i=1}^{n} h_i}.$$

using the fact that the sum of the hashing powers of all peers is 1, we get:

$$P(I_1 > x, I_2 > x, \ldots, I_n > x) = e^{-\frac{x}{I}}.$$

The average time interval between two blocks is given by:

$$\int_0^\infty P(I_1 > x, I_2 > x, \ldots, I_n > x)dx = I.$$

The variance of the time interval between two blocks is given by:

$$\int_0^\infty x^2 P(I_1 > x, I_2 > x, \ldots, I_n > x)dx - I^2 = I^2.$$

Setting the meantime of $T_k = \frac{I}{h_i}$ achieves the desired goals: an average inter-arrival time of I between any two blocks from any two peers and ensuring that nodes with higher hashing power generate blocks in lesser time.

0.4 Insight and critique

We created 30 blocks in each of the case, and observed the fraction of longest chain. We observed the following insights:

0.4.1 Number of peers

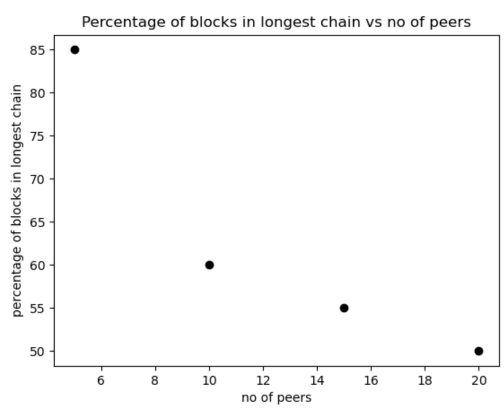

Figure 1: Number of peers vs fraction of longest chain.

On increasing the number of peers, the branching in the blockchain increases. This is because the number of peers that generate blocks in the same time interval increases. This leads to a higher number of branches in the blockchain. So the fraction of longest chain decreases. Other parameters being 0.1 for slow peers, 0.1 for low cpu peers, 100 for transaction interval time, 200 for block interval time. There should be a reasonable number of high cpu peers because if the high cpu peers are very low then the number of blocks generated will be very low and the blockchain will not be able to grow.1

0.4.2 transaction interval time

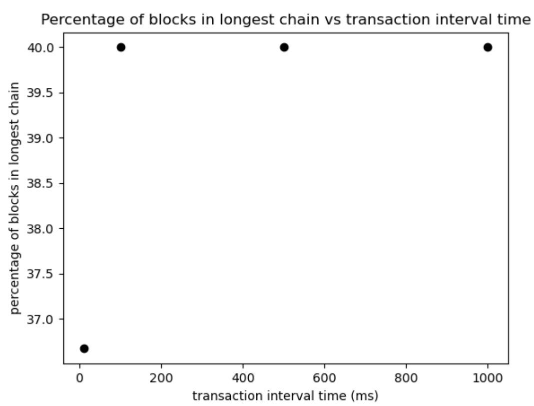

Figure 2: Transaction interval time vs fraction of longest chain.

On increasing the transaction interval time, the number of transactions in the blockchain decreases. But the block inter arrival time remains the same. This is because the number of transactions generated in a time interval decreases, but the time interval between two blocks remains the same, as the time interval between two blocks is independent of the number of transactions generated. Fraction of longest chain remains almost the same.2 Also if the transaction interval time is very low means it takes larger time to create same number of blocks with large transaction interval time. Other parameters being 0.3 for slow peers, 0.5 for low cpu peers, 10 for no of peers, 200 for block interval time.

0.4.3 block interval time

On increasing block interval time, each peer takes more time to create a block, so branching decreases. fraction of longest chain increases. Other

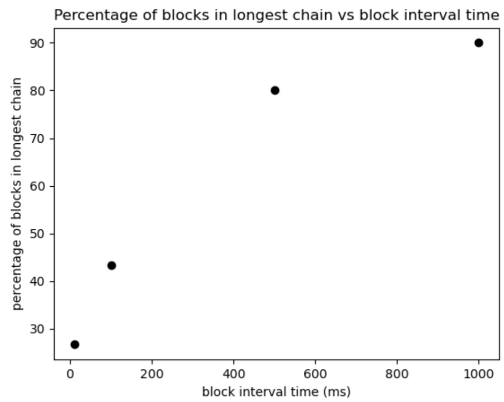

Figure 3: Block interval time vs fraction of longest chain.

parameters being 0.3 for slow peers, 0.5 for low cpu peers, 10 for no of peers, 100 for transaction interval time.3

0.4.4 num of slow peers

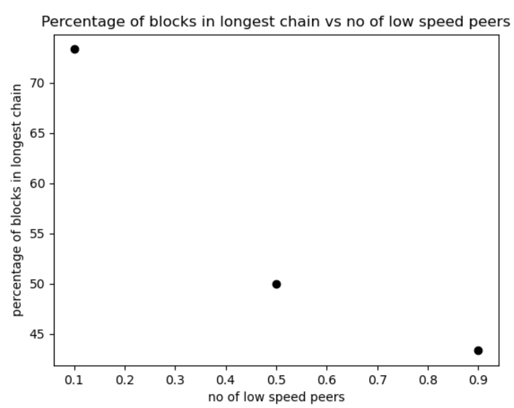

Figure 4: Number of slow peers vs fraction of longest chain.

In this case on increasing the number of slow peers, the branching in the blockchain increases. This is because the number of peers that generate blocks in the same time interval increases. This leads to a higher number of branches in the blockchain, because the propagation time of the block increases. So the fraction of longest chain decreases. Other parameters being 0.9 for low cpu peers, 10 for no of peers, 100 for transaction interval time, 200 for block interval time.4

0.4.5 num of low cpu peers

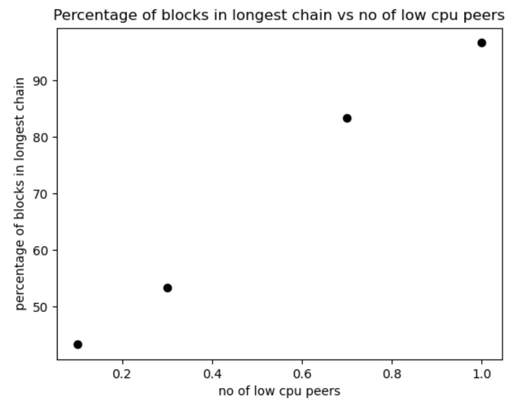

Figure 5: Number of low cpu peers vs fraction of longest chain.

On increasing the number of low cpu peers, the branching in the blockchain decreases. This is because the peer with low cpu generates blocks in more time, so the branching decreases. So the fraction of longest chain increases, with less branching. Other parameters being 10 for no of peers, 100 for transaction interval time, 200 for block interval time , 0.1 for slow peers.5

0.4.6 Some pictures of the simulation while generating 30 blocks

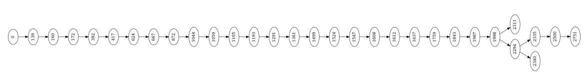

Figure 6: Simulation with 5 peers, 100 transaction interval time, 200 block interval time, 0.1 slow peers, 0.9 low cpu peers.

we increased the number of slow peers and low cpu peers, and we can see that the branching in the blockchain increases.6 7

we decreased the number of low cpu peers, and we can see that the branching in the blockchain decreases.6 8

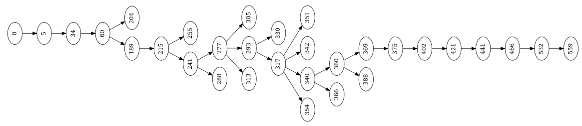

Figure 7: Simulation with 5 peers, 100 transaction interval time, 200 block interval time, 0.9 slow peers, 0.9 low cpu peers.

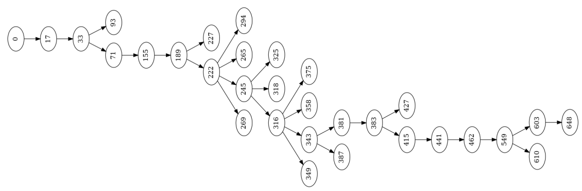

Figure 8: Simulation with 5 peers, 100 transaction interval time, 200 block interval time, 0.1 slow peers, 0.5 low cpu peers.

we increased the number of peers with reasonable number of high cpu peers, and we can see that the branching in the blockchain increases.6 9

we increased the transaction interval time, and we can see that branching in the blockchain remains almost the same.6 10

we decreased the block interval time, and we can see that the branching in the blockchain increases.6 11

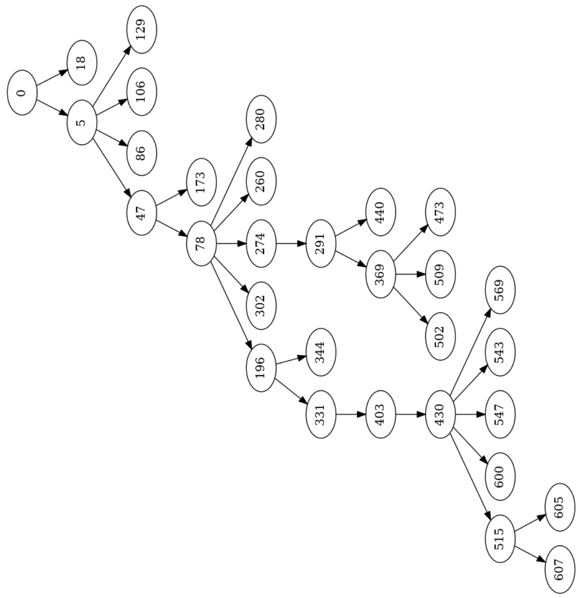

Figure 9: Simulation with 20 peers, 100 transaction interval time, 200 block interval time, 0.1 slow peers, 0.1 low cpu peers.

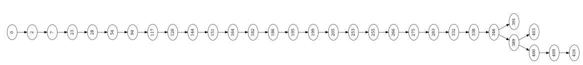

Figure 10: Simulation with 5 peers, 500 transaction interval time, 200 block interval time, 0.1 slow peers, 0.9 low cpu peers.

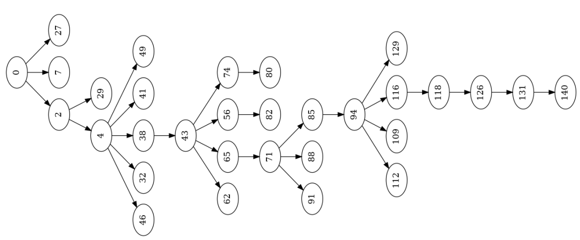

Figure 11: Simulation with 5 peers, 100 transaction interval time, 10 block interval time, 0.1 slow peers, 0.9 low cpu peers.