Blockchain Assignment 1

Yashvardhan, 200050162
Koustubh Rao, 200100176
Onkar Borade, 200050022
Neeshi Merchant, 200050087

February 2023

1 Report

1.1 What are the theoretical reasons of choosing the exponential distribution?

Since an exponential distribution has a memoryless attribute that closely resembles the behaviour of transaction arrival times in a decentralised network, it is frequently used to represent the interarrival time of transactions generated by peers in a blockchain. As a result, the likelihood that a transaction will come within a given time frame is independent of how long has passed since the last transaction arrived.

We present the following argument:
Let δ be very small tiny interval such that
P(txn is generated after n δ time) = P(txn not generated in first $n\delta$ time) = $(1 - T_{tx}\delta)^n$
Substituting t = $n\delta$ and denoting transaction generation time as Random Variable t_{tx}
$P(t_{tx} > t) = (1 - T_{tx}t/n)^n$
Since $\delta \to 0$, for t to be finite, we have $n \to \infty$ therefore,
$P(t_{tx} > t) \sim e^{-T_{tx}t}$
$P(t_{tx} < t) = 1 - e^{-T_{tx}t}$
$P(t_{tx} > t) = T_{tx}e^{-T_{tx}t}$

Therefore, we choose transaction interarrival time to be sampled from an exponential distribution.

1.2 Why is the mean of dij inversely related to cij ? Give justification for this choice.

The size of the packet being transmitted and the link's available bandwidth affect the queuing delay. Packets can be transmitted more quickly when the link speed is high since there is more bandwidth available. As a result of less queue congestion and faster packet transmission, the queuing delay at node I is decreased. On the other side, slower link speeds result in lower bandwidth availability, which lengthens the transmission time for packets. As a result, there is higher congestion in the queue at node I which lengthens the time packets must wait before being transferred. Therefore, the mean queuing delay at node i is inversely proportional to the link speed between node i and node j.

1.3 Justify the chosen mean value for Tk.

We have chosen the default value of Tk as 600 secs, which is 10 mins.
Justification: The latency for transmitting a fresh block A after mining it is roughly half a second for each link involving any two peers. Sending this block to every peer in the network will therefore take a few seconds. The number of blockchain forks should be kept to a minimum. A fork would happen if another node mined a new block before block A got there. As a result, we set Tk's default value to be substantially more than the transmission time. This lowers the likelihood of forking, and

we end up with a chain that almost entirely without forks. The number of blocks generated over a time period is likewise based on the value of Tk.

1.4 Find the ratio of the number of blocks generated by each node in the Longest Chain of the tree to the total number of blocks it generates at the end of the simulation.

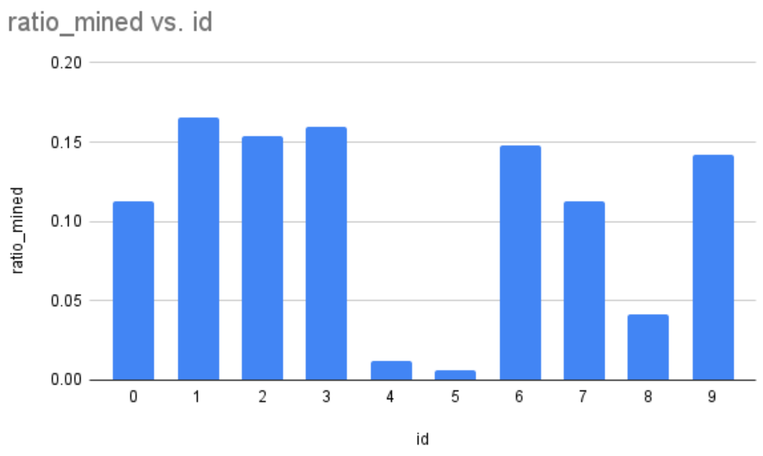


Figure 1: Caption

1.5 How does this ratio vary depending on whether the node is fast, slow, low CPU, high CPU power etc.?

1.5.1 With respect to Slow

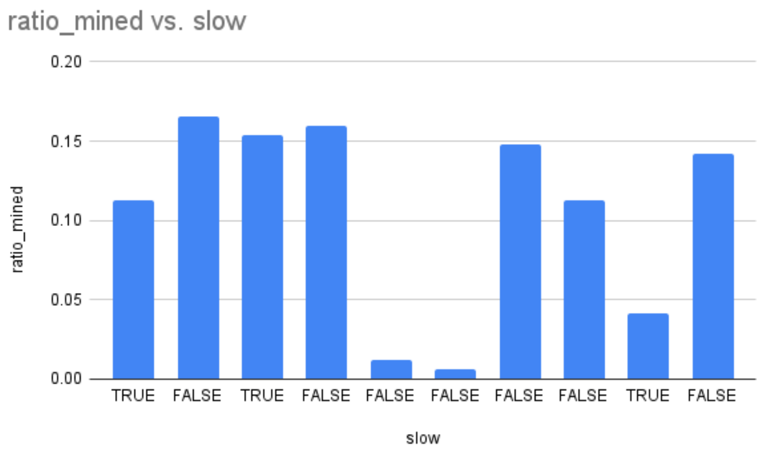


Figure 2: WRT SLOW

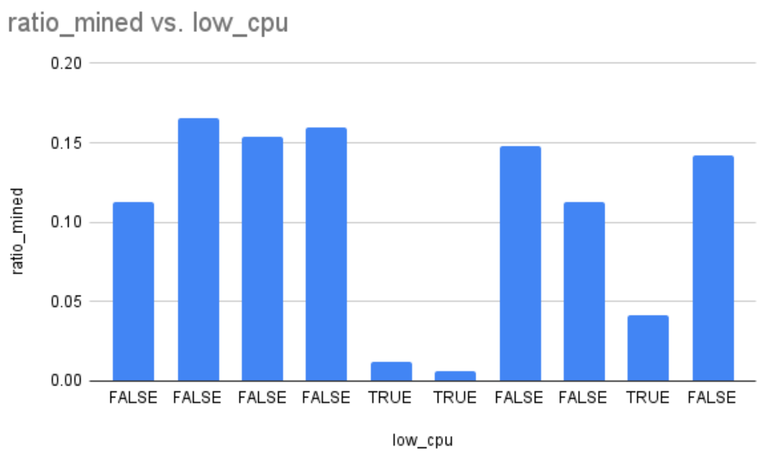
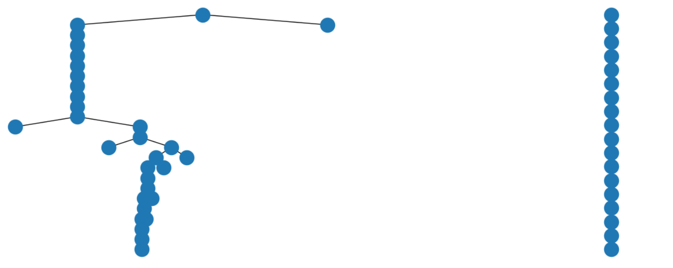


Figure 3: WRT LOW CPU

1.5.2 With respect to low cpu

- In our analysis we found that ratio of blocks generated by peer in the final blockchain mostly depends only on the whether the cpu is fast or slow.
- A peer which is fast but has low CPU, will not generate as much blocks as a peer with high CPU but is slow.
- In the best results we for the peers who had high CPU and were fast (for eg. peer 1, 3).
- This suggest that the ratio of block in final blockchain depends significantly on CPU and little on transmission speed.

1.6 How long are branches of the tree measured in number of blocks?



(a) Tree

(b) Chain

Figure 4: 2 Figures side by side

- Number of Forks generated in blockchain (tree) is directly proportional to inter-arrival time and inversely proportional to hashing power of peer
- With inter-arrival time = 600 secs and defualt hashing power we did not find any forks in our blockchain.
- But on reducing the inter-arrival time to 600 ms we found the the above blockchain(tree)