Discrete-event Simulator For P2P CryptoCurrency Network

Fenil Mehta (203050054), Aditya Pradhan (203059006), Arnab Das (20305R005)

October 16, 2021

1 Introduction

We have implemented a discrete-event simulator for a peer-to-peer cryptocurrency network. A discrete-event simulator maintains an event queue and a global clock. From this queue, it executes the earliest event and also pushes some new events which were generated as a result of the execution of this event (if any). In this report, we describe some design choices and we analyse different simulations based on different system parameters specified in the configuration file.

2 Graph Generation For P2P CryptoCurrency Network

Analysing the referenced research paper, we found that Bitcoin network typically follows a social graph like structure where most nodes have very low degree, and few but non negligible number of nodes have their degree higher than some large number k. With increase in number of nodes such networks converge to scale free behaviour. A scale free network is a network whose nodes follows a power law distribution for degree distribution. [BFL14]

We used Barabási–Albert Model to implement a scale free graph using networkx library in python.

3 Transaction

Each node/peer generates transactions of the form A pays B C coins. Inter arrival time between Transactions follows an exponential distribution with a mean which is given as input.

3.1 Why queuing delay is inversely proportional to bandwidth?

The queuing delay depends on two things, the *Rate of Arrival* and the *Rate of Departure*. The Rate of Departure depends on transmission time as more the transmission time less the Departure Rate. We know that transmission time is inversely proportional to bandwidth, so queuing delay is also inversely proportional to bandwidth.

3.2 Why Exponential Distribution is used to determine Inter Arrival time of transactions?

In Bitcoin like scale-free Networks the process of transaction generation follows a Poisson process. This has been found by doing experiments as well as performing a queuing model analysis. Since it follows a Poisson distribution, the inter arrival time of two events (in this case Transaction) follows an exponential distribution. [GHJ20]

4 Experiments

We performed simulations with different values for the following system parameters:

1. Total number of nodes (n) = 10, all the nodes are slow
2. Total number of nodes (n) = 10, all the nodes are fast
3. Total number of nodes (n) = 30, 50% of the nodes are slow (z), transaction inter arrival mean (T_{tx}) = 5s, block average mining time (T_k) = 600s
4. Total number of nodes (n) = 30, 80% of the nodes are slow (z), transaction inter arrival mean (T_{tx}) = 10s, block average mining time (T_k) = 60s

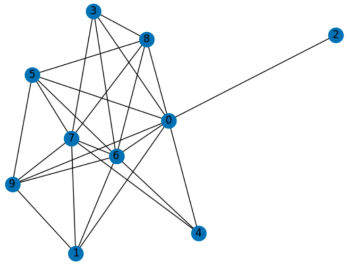

Figure 1: Network topology for experiment 1

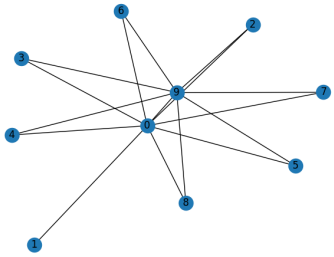

Figure 2: Network topology for experiment 2

4.1 Analysis

We divide our analysis into two parts: block generation analysis and blockchain tree analysis.

4.2 Reason For Choosing Specific T_k Value

T_k value influences hash power wastage. If the T_k value is very less then there is a high chance that more blocks will be generated in a very less time and the network won't be able to add those block in a single blockchain which will create forks in peers thus wasting CPU hash power for those peers whose blocks are not in the longest chain.
Can we increase T_k to a very high value? No, as there will be lot of pending transactions and confirmation time for the block transactions will increase drastically leading to various type of attacks. This value needs to set along with transaction inter arrival time because if there is very slow transaction generation then wastage will be more. This value needs to set in a way so that blocks are well filled with transactions. Satoshi, creator of Bitcoin network choose an average mining time of 10 minutes. We tried with this value and got fairly good results. we also tried with some other values in experiments section.

4.2.1 Block generation analysis

For this analysis, we define a term **R value** which is the ratio of number of blocks mined by a node which are present in the longest chain to the total number of blocks mined by that node which can be present in any branch of the blockchain. We consider experiments 1 and 2 for this analysis. Table 1 and Table 2 shows the *R value* of each node in the network along with their CPU Power. The CPU Powers greater than the median of CPU Power of all nodes is considered as High and others as low.

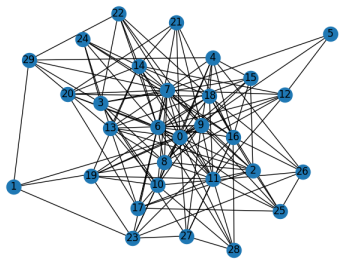

Figure 3: Network topology for experiment 3

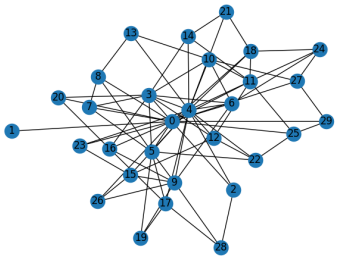

Figure 4: Network topology for experiment 4

	n_0	n_1	n_2	n_3	n_4	n_5	n_6	n_7	n_8	n_9
CPU Power	Low	Low	Low	High	Low	High	High	High	High	Low
R value	0.44	0.05	0.17	0.00	0.00	0.67	0.50	0.64	0.86	0.00

Table 1: *R value* of nodes in experiment 1

	n_0	n_1	n_2	n_3	n_4	n_5	n_6	n_7	n_8	n_9
CPU Power	High	High	High	High	Low	Low	Low	High	Low	Low
R value	0.67	0.82	0.09	0.45	0.00	0.00	0.00	0.58	0.22	0.12

Table 2: *R value* of nodes in experiment 2

From the tables, we can observe that the ratio depends on Hash Power and Network Power. Higher the hash power of a node the faster it can generate the blocks. We observed that the nodes with high CPU power always have more advantage over the nodes with low CPU power. So, more hash power increases the denominator of the *R value*.
Does it influence the numerator? Yes, there is a correlation as faster block generation may lead to addition of that block in the longest chain of peers but numerator mainly depends on the network connectivity because faster the block gets delivered to peer nodes, higher the chance of it being added as a tail of the longest chain, thus increasing the numerator of the *R value*. The blocks of a node in the longest chain (i.e. the numerator of *R value*) also depend on the topology of the peer-to-peer network.

Node Type	**Rank in influence of R Value**
High CPU Power with Fast Network	1
High CPU Power with Slow Network	2
Low CPU Power with Fast Network	3
Low CPU Power with Slow Network	4

Table 3: Type Of Node vs R value

4.2.2 Blockchain tree analysis

In the generated graph of our simulator each leaf maintains its index in the path from genesis block to itself. So, $index + 1$ of any leaf is the branch length having that block as the tail.

We consider experiments 3 and 4 for this analysis. In experiment 3, the transaction inter arrival mean is only 5 seconds and block mining takes 600 seconds on average, which leads to zero forks and there is only one branch (the longest branch) which includes 11 blocks. Figures 5 and 6 show the starting blocks and ending blocks of this branch respectively.

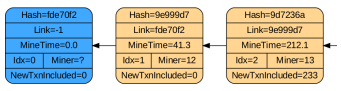

Figure 5: Starting blocks of the resulting blockchain tree (experiment 3)

In experiment 4, we modified our parameters such that 80% of the nodes are slow and the transaction inter arrival mean is 10 seconds which is high, the block average mining time is 10 times less as compared to that of experiment 3. This means that nodes will generate more blocks.

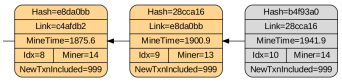

Figure 6: Ending blocks of the resulting blockchain tree (experiment 3)

Since there are a high number of slow links, the chances of fork are also high. The resulting blockchain tree contains 9 branches with lengths 49, 57, 93, 111, 120, 157, 168, 202 and 235. As compared to only 11 blocks in experiment 3, the longest chain includes 235 blocks in this case. Figures 7, 8 and 9 show some branch leaves of the blockchain tree.

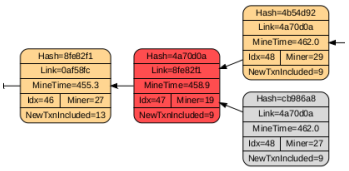

Figure 7: Leaf of the first branch in the resulting blockchain tree (experiment 4)

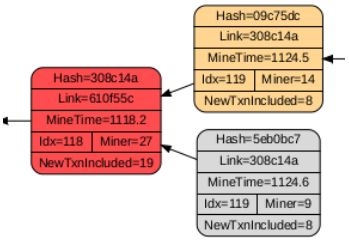

Figure 8: Leaf of the fifth branch in the resulting blockchain tree (experiment 4)

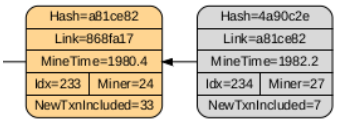

Figure 9: Leaf of the longest branch in the resulting blockchain tree (experiment 4)

5 Simulating selfish mining and stubborn mining attacks

In addition to the honest nodes and infected nodes (which might generate invalid transactions), we have introduced another class of nodes, called "Attacker" nodes. The aim of attacker node(s) is to perform either the selfish mining attack [ES13] or the stubborn mining attack [NKMS16] in the P2P cryptocurrency network. The type of the attacker node (selfish or stubborn) and other attack related parameters (number of individual attacker nodes, hash power of an attacker node, percent of honest nodes to which the attacker node is connected (ζ)) can be specified in the configuration file.

5.1 Experiments and Analysis

We performed several experiments using the following configurations:
Total number of nodes (n) = 100,
Number of honest slow nodes (in percentage) = 50%,
Number of infected nodes (in percentage) = 0,
Transaction inter arrival mean (T_{tx}) = 1000s,

Block average mining time (T_k) = 60s,
Number of individual attacker nodes = 1,
Attacker node type = "selfish", "stubborn",
Attacker node mining power (in percentage) = 10%, 20%, 34%,
Fraction of honest nodes an attacker is connected to (ζ) = 25%, 50%, 75%,
Total execution time = 10000s

We kept the transaction inter arrival mean high because it allows our simulator to execute fast and to see the bigger picture in less execution time. Figures 10, 11, 12 and 13 show the results of these experiments. We now present the analysis of these experiments.

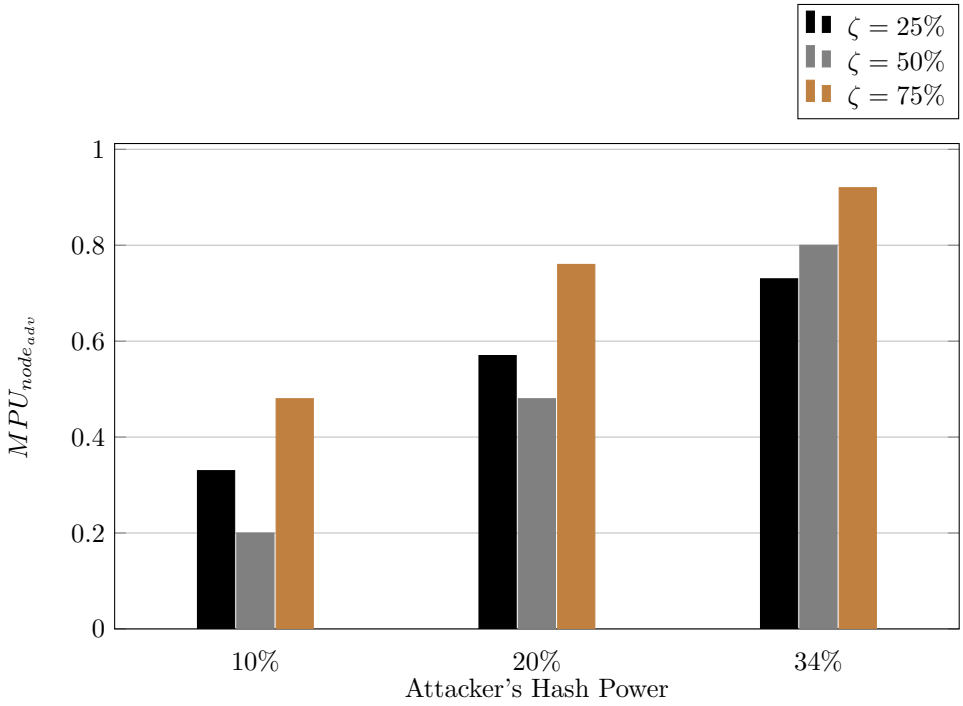

Figure 10: $MPU_{node_{adv}}$ v/s Attacker's Hash Power (for selfish mining attack)

The $MPU_{node_{adv}}$ and $MPU_{node_{overall}}$ are defined as follows:

$$MPU_{node_{adv}} = \frac{Number\ of\ blocks\ mined\ by\ the\ attacker\ in\ longest\ chain}{Total\ number\ of\ blocks\ mined\ by\ the\ attacker}$$

$$MPU_{node_{overall}} = \frac{Number\ of\ blocks\ in\ the\ longest\ chain}{Total\ number\ of\ blocks\ generated\ across\ all\ the\ nodes}$$

We observe that as the mining power of attacker increases, the $MPU_{node_{adv}}$ also increases. This is reasonable because with high mining power, the attacker will be able to mine more blocks privately as compared to low mining power. $MPU_{node_{adv}}$ also depends on how quickly the attacker is able to broadcast it's private chain block(s) to the network, this becomes important in the situations when the length of public branch becomes equal to that of attacker's private branch. With increased value of ζ (percentage of honest nodes to which the attacker is connected), we observe that in case of tie situations, the number of honest nodes that mine on the attacker's block is more as compared to when the value of ζ is low.

The value of $MPU_{node_{overall}}$ decreases with increase in the mining power of the attacker because the attacker is able to generate more blocks privately and hence there will be more forks in the blockchain, with some of the honest nodes mining on the attacker's block.

The theoretical values obtained in [ES13] show that in the worst case when $\gamma = 0$ (the fraction of honest nodes that mine on the attacker's node in case of a tie), the mining power threshold required is about $\frac{1}{3}$ of the total mining power. Our experiments also reveal that the attacker is able to push over 90% of it's blocks into the main chain with a mining power of 34% of the total mining power.

In case of the stubborn mining attack, we observed that the value of $MPU_{node_{adv}}$ increases (on averaging the values obtained over every ζ) with increase in mining power, however, this value is

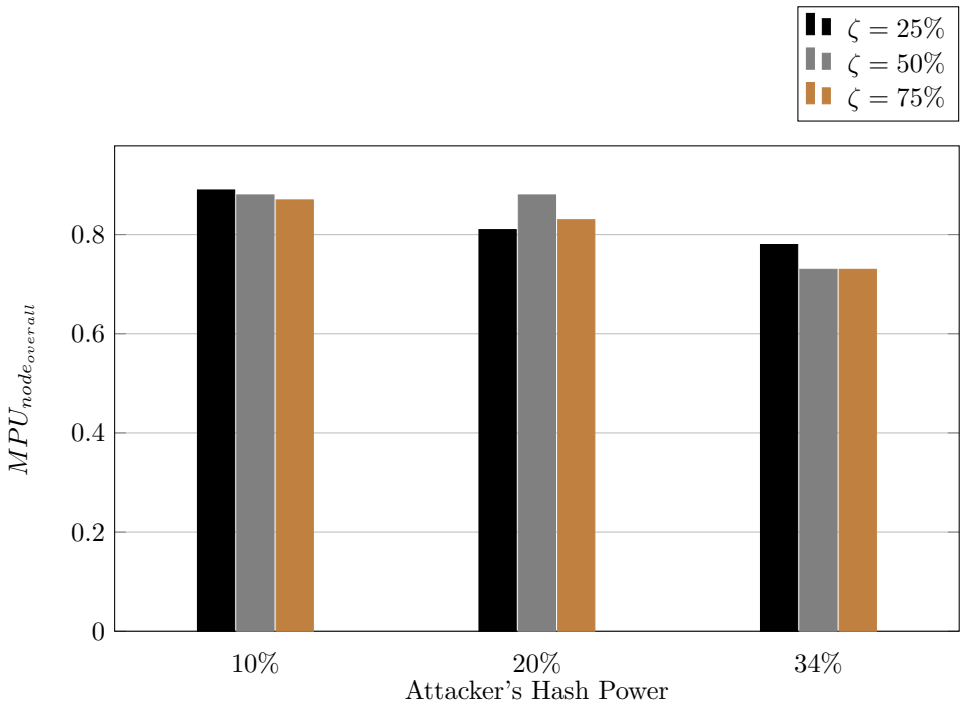

Figure 11: $MPU_{node_{overall}}$ v/s Attacker's Hash Power (for selfish mining attack)

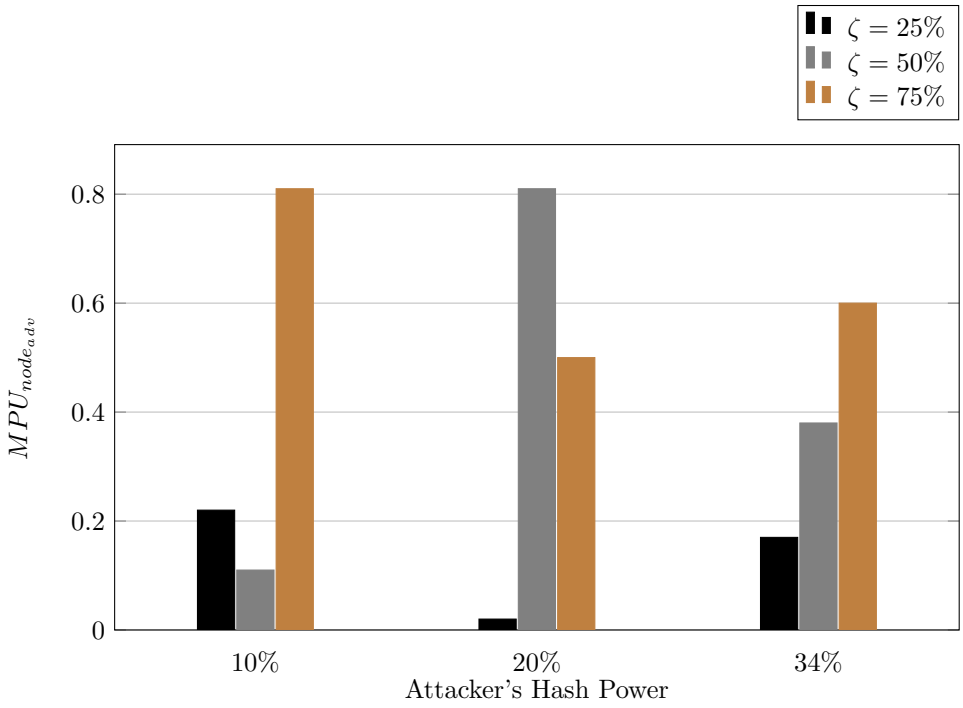

Figure 12: $MPU_{node_{adv}}$ v/s Attacker's Hash Power (for stubborn mining attack)

less than the one obtained in case of selfish mining attack because stubborn mining attack tries to take more risk and leads to more scenarios where the length of public chain become equal to that of private chain. As a result, the number of forks in the blockchain are also large and the value of $MPU_{node_{overall}}$ decreases with increase in mining power.

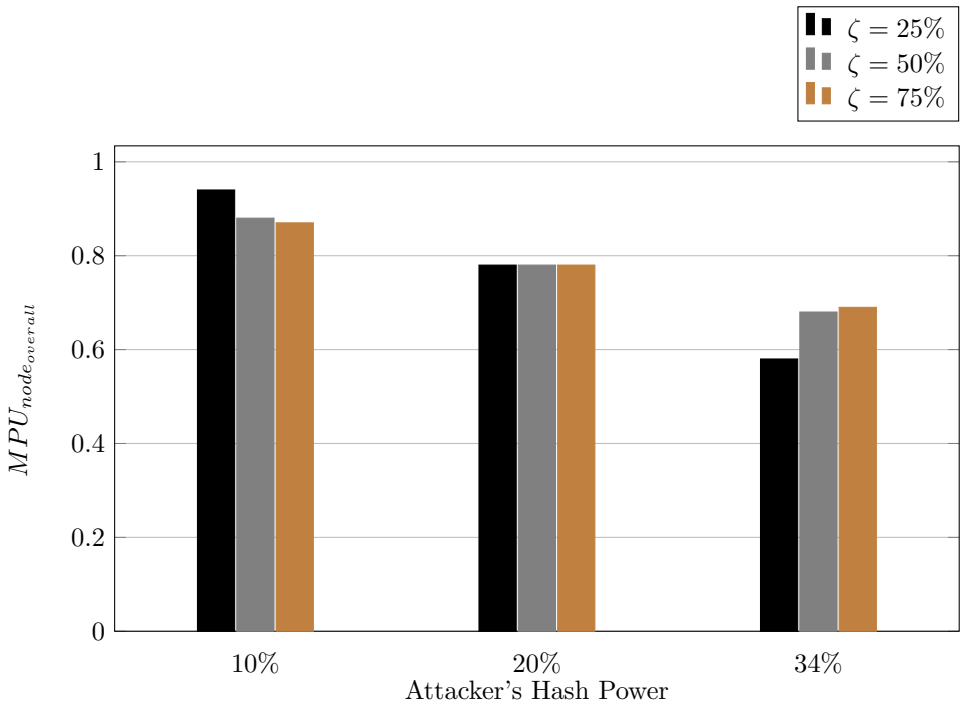

Figure 13: $MPU_{node_{overall}}$ v/s Attacker's Hash Power (for stubborn mining attack)

References

[BFL14] Annika Baumann., Benjamin Fabian., and Matthias Lischke. Exploring the bitcoin network. In *Proceedings of the 10th International Conference on Web Information Systems and Technologies - Volume 2: WEBIST,*, pages 369–374. INSTICC, SciTePress, 2014.

[ES13] Ittay Eyal and Emin Gün Sirer. Majority is not enough: Bitcoin mining is vulnerable. *CoRR*, abs/1311.0243, 2013.

[GHJ20] Befekadu G. Gebraselase, Bjarne E. Helvik, and Yuming Jiang. Transaction characteristics of bitcoin. *CoRR*, abs/2010.10858, 2020.

[NKMS16] Kartik Nayak, Srijan Kumar, Andrew Miller, and Elaine Shi. Stubborn mining: Generalizing selfish mining and combining with an eclipse attack. In *2016 IEEE European Symposium on Security and Privacy (EuroS P)*, pages 305–320, 2016.