Report on: Simulation of a P2P Cryptocurrency Network along with demonstration of Selfish Mining Attack

Aniket Jadhav (22M0817)

1. Questions

1. What are the theoretical reasons for choosing the exponential distribution?

- The exponential distribution is ”memoryless,” which means that the probability of discovering a block at any given moment is unaffected by the amount of time since the discovery of the previous block. Thus it makes sense to simulate the process of block discovery.
- The average time between blocks is the only parameter needed to explain the exponential distribution, which has a straightforward shape. This makes it practical to use and offers a straightforward approach to mimic the block discovery procedure.

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

- The message queueing delay at node I before forwarding the message to node j is represented by the variables dij and cij, respectively Therefore, the speed of cij will determine how quickly a node can process its queue, giving rise to an inverse relationship between cij and dij.

3. Justification for the chosen mean value for Tk from an exponential distribution.

- Tk is the average time spent waiting for blocks to generate. Consequently, a larger value of Tk would result in fewer blocks being created, and vice versa. The number of transactions per block de- creases with an excess of blocks, whereas the node transaction queue fills up with an excess of blocks. Therefore, these two parameters must be balanced.

2. Observations

Let,

$$r = \frac{Number\ of\ blocks\ generated\ in\ the\ longest\ chain}{Number\ of\ blocks\ generated\ at\ the\ end\ of\ simulation}$$

N=No. of nodes
Ttx =Mean transaction inter-arrival time
Z0= number of slow nodes
Z1= number of low cpu nodes

CASE 1 :

N= 10
Ttx =60
Z0= 30
Z1= 20
r on slow node= 0.67
r on fast node= 0.58
r on low cpu= 0.00
r on high cpu=0.67

```
152
slow nodes list
[2, 0, 6]
slow cpu nodes list
[9, 5]
extra info for node  0
 number of block of its own in chain=  10  number of total created =  15  ratio is =  0.6666666666666666
extra info for node  1
 number of block of its own in chain=  3  number of total created =  21  ratio is =  0.14285714285714285
extra info for node  2
 number of block of its own in chain=  6  number of total created =  13  ratio is =  0.46153846153846156
extra info for node  3
 number of block of its own in chain=  7  number of total created =  12  ratio is =  0.5833333333333334
extra info for node  4
 number of block of its own in chain=  5  number of total created =  12  ratio is =  0.4166666666666667
extra info for node  5
 number of block of its own in chain=  0  number of total created =  11  ratio is =  0.0
extra info for node  6
 number of block of its own in chain=  2  number of total created =  26  ratio is =  0.07692307692307693
extra info for node  7
 number of block of its own in chain=  0  number of total created =  12  ratio is =  0.0
extra info for node  8
 number of block of its own in chain=  7  number of total created =  13  ratio is =  0.5384615384615384
extra info for node  9
 number of block of its own in chain=  0  number of total created =  17  ratio is =  0.0
```

CASE 2 :

N= 10

Ttx = 120

Z0= 30

Z1= 20

r on slow node= 0.57

r on fast node= 0.40

r on low cpu=0.0

r on high cpu=0.57

```
240
slow nodes list
[7, 3, 6]
slow cpu nodes list
[1, 7]
extra info for node  0
 number of block of its own in chain=  11  number of total created =  27  ratio is =  0.4074074074074074
extra info for node  1
 number of block of its own in chain=  0  number of total created =  16  ratio is =  0.0
extra info for node  2
 number of block of its own in chain=  3  number of total created =  16  ratio is =  0.1875
extra info for node  3
 number of block of its own in chain=  8  number of total created =  28  ratio is =  0.2857142857142857
extra info for node  4
 number of block of its own in chain=  1  number of total created =  34  ratio is =  0.029411764705882353
extra info for node  5
 number of block of its own in chain=  12  number of total created =  47  ratio is =  0.2553191489361702
extra info for node  6
 number of block of its own in chain=  12  number of total created =  21  ratio is =  0.5714285714285714
extra info for node  7
 number of block of its own in chain=  0  number of total created =  18  ratio is =  0.0
extra info for node  8
 number of block of its own in chain=  4  number of total created =  19  ratio is =  0.21052631578947367
extra info for node  9
 number of block of its own in chain=  1  number of total created =  14  ratio is =  0.07142857142857142
```

CASE 3 :

N= 10

Ttx =250

Z0= 20

Z1= 60

r on slow node=0.03

r on fast node=0.39

r on low cpu= 0.25

r on high cpu=0.38

```
290
slow nodes list
[6, 5]
slow cpu nodes list
[8, 4, 0, 6, 1, 5]
extra info for node  0
 number of block of its own in chain=  1  number of total created =  4  ratio is =  0.25
extra info for node  1
 number of block of its own in chain=  1  number of total created =  20  ratio is =  0.05
extra info for node  2
 number of block of its own in chain=  6  number of total created =  83  ratio is =  0.07228915662650602
extra info for node  3
 number of block of its own in chain=  15  number of total created =  57  ratio is =  0.2631578947368421
extra info for node  4
 number of block of its own in chain=  1  number of total created =  7  ratio is =  0.14285714285714285
extra info for node  5
 number of block of its own in chain=  1  number of total created =  27  ratio is =  0.037037037037037035
extra info for node  6
 number of block of its own in chain=  0  number of total created =  20  ratio is =  0.0
extra info for node  7
 number of block of its own in chain=  13  number of total created =  46  ratio is =  0.2826086956521739
extra info for node  8
 number of block of its own in chain=  1  number of total created =  8  ratio is =  0.125
extra info for node  9
 number of block of its own in chain=  7  number of total created =  18  ratio is =  0.3888888888888889
```

CASE 4 :

N= 10

Ttx = 300

Z0= 20

Z1= 80

r on slow node=0.53

r on fast node=0.17

r on low cpu=0.17

r on high cpu=0.53

```
246
slow nodes list
[7, 1]
slow cpu nodes list
[8, 7, 0, 5, 9, 2, 3, 6]
extra info for node  0
 number of block of its own in chain=  4  number of total created =  24  ratio is =  0.16666666666666666
extra info for node  1
 number of block of its own in chain=  17  number of total created =  32  ratio is =  0.53125
extra info for node  2
 number of block of its own in chain=  2  number of total created =  15  ratio is =  0.13333333333333333
extra info for node  3
 number of block of its own in chain=  3  number of total created =  17  ratio is =  0.17647058823529413
extra info for node  4
 number of block of its own in chain=  9  number of total created =  52  ratio is =  0.17307692307692307
extra info for node  5
 number of block of its own in chain=  3  number of total created =  17  ratio is =  0.17647058823529413
extra info for node  6
 number of block of its own in chain=  2  number of total created =  29  ratio is =  0.06896551724137931
extra info for node  7
 number of block of its own in chain=  0  number of total created =  23  ratio is =  0.0
extra info for node  8
 number of block of its own in chain=  0  number of total created =  17  ratio is =  0.0
extra info for node  9
 number of block of its own in chain=  0  number of total created =  20  ratio is =  0.0
```

CASE 5 :

N= 10

Ttx = 60

Z0= 50

Z1= 20

r on slow node=0.35

r on fast node=0.57

r on low cpu=0.57

r on high cpu=0.35

```
172
slow nodes list
[8, 3, 4, 6, 1]
slow cpu nodes list
[6, 0]
extra info for node  0
 number of block of its own in chain=  4  number of total created =  7  ratio is =  0.5714285714285714
extra info for node  1
 number of block of its own in chain=  1  number of total created =  20  ratio is =  0.05
extra info for node  2
 number of block of its own in chain=  3  number of total created =  20  ratio is =  0.15
extra info for node  3
 number of block of its own in chain=  0  number of total created =  19  ratio is =  0.0
extra info for node  4
 number of block of its own in chain=  9  number of total created =  33  ratio is =  0.2727272727272727
extra info for node  5
 number of block of its own in chain=  8  number of total created =  27  ratio is =  0.2962962962962963
extra info for node  6
 number of block of its own in chain=  0  number of total created =  12  ratio is =  0.0
extra info for node  7
 number of block of its own in chain=  4  number of total created =  13  ratio is =  0.3076923076923077
extra info for node  8
 number of block of its own in chain=  5  number of total created =  14  ratio is =  0.35714285714285715
extra info for node  9
 number of block of its own in chain=  0  number of total created =  7  ratio is =  0.0
```

3. Conclusion of Observations

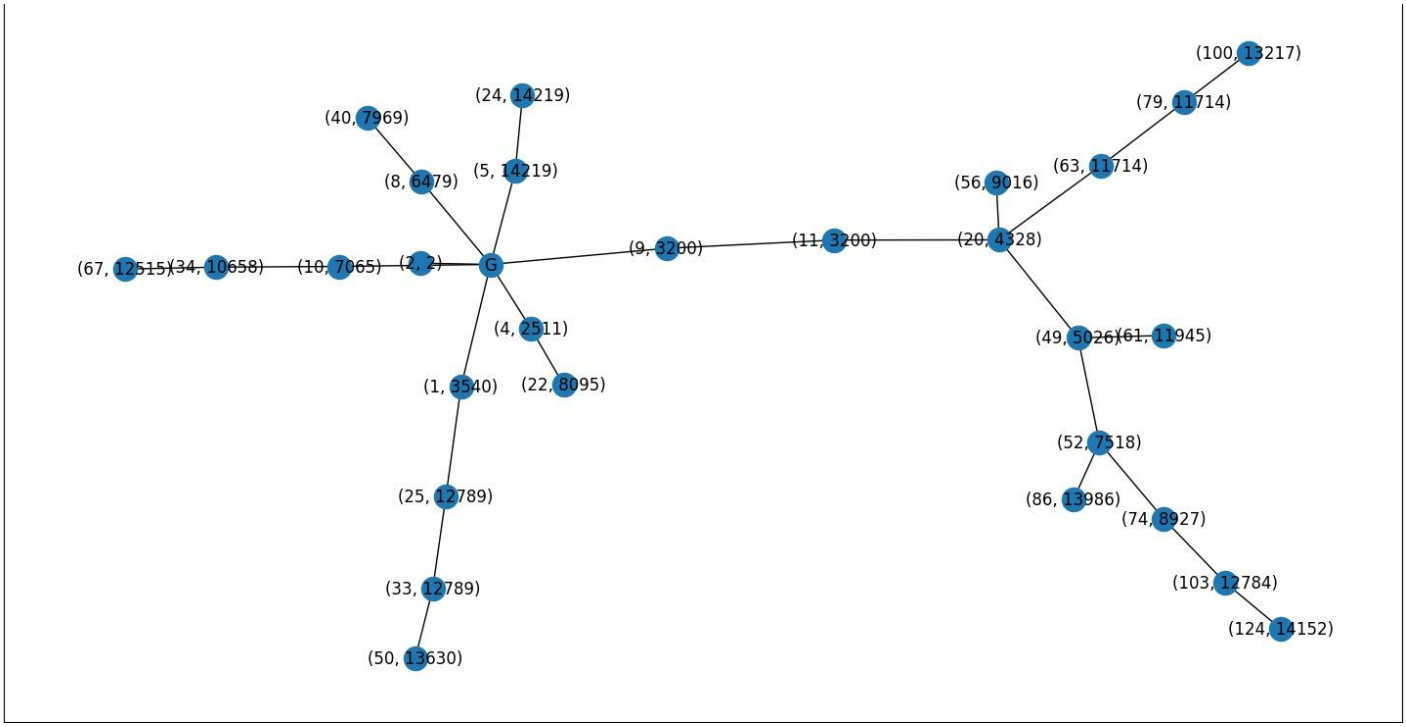
- r of high CPU nodes is always greater than that of low CPU nodes in most cases because the low CPU nodes has lesser mining power so due to this it is able to mine lesser blocks as compared to high CPU node so its blocks are lesser in the longest chain.
- r of slow nodes is higher than r of faster nodes in most cases because the slow node will broadcast the blocks at a slower rate to its neighbors than the fast nodes.
- r value decreases with an increase in the value of Ttx.

4. Pictures of Blockchains

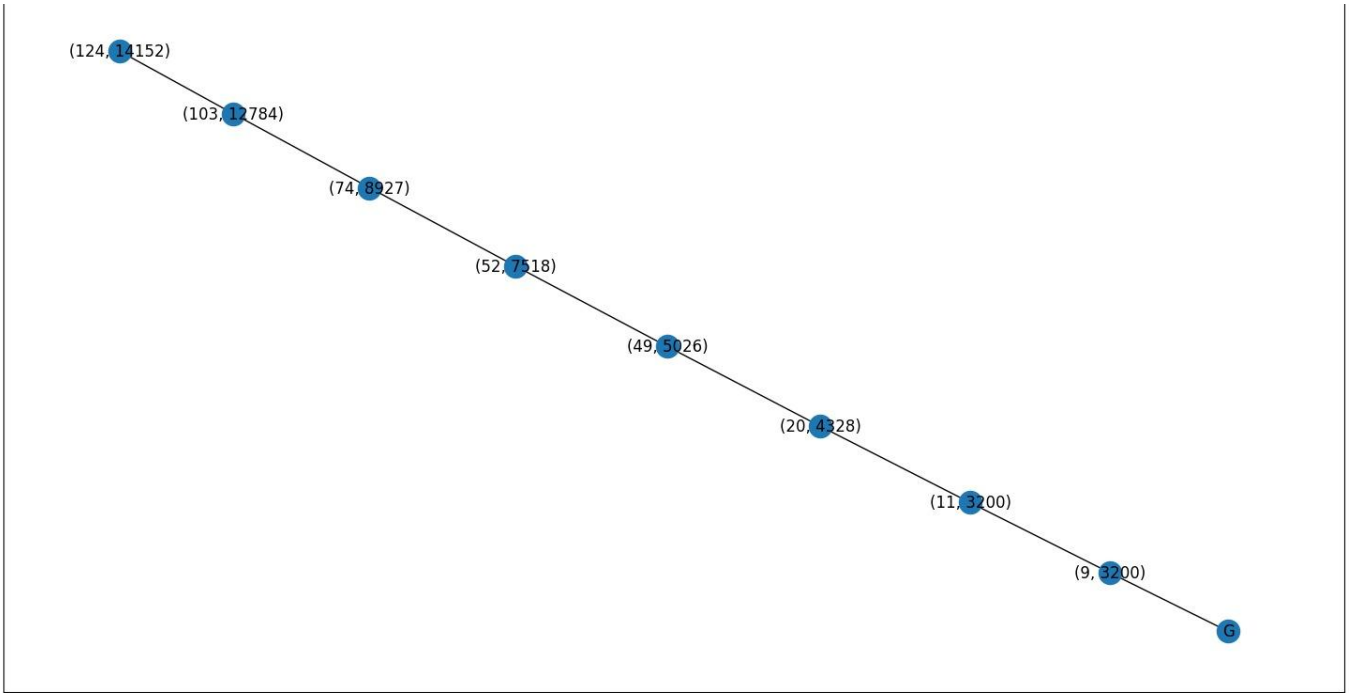
For,
number of nodes= 10
Ttx (TXN Interarrival mean time)= 60
percent of slow nodes = 30
percent of low CPU nodes=20
slow nodes={2,6,8}
fast node={0,1,3,4,5,7,9}
low cpu={5,6}
fast cpu={0,1,2,3,4,7,8,9}

Slow nodes + Fast CPU

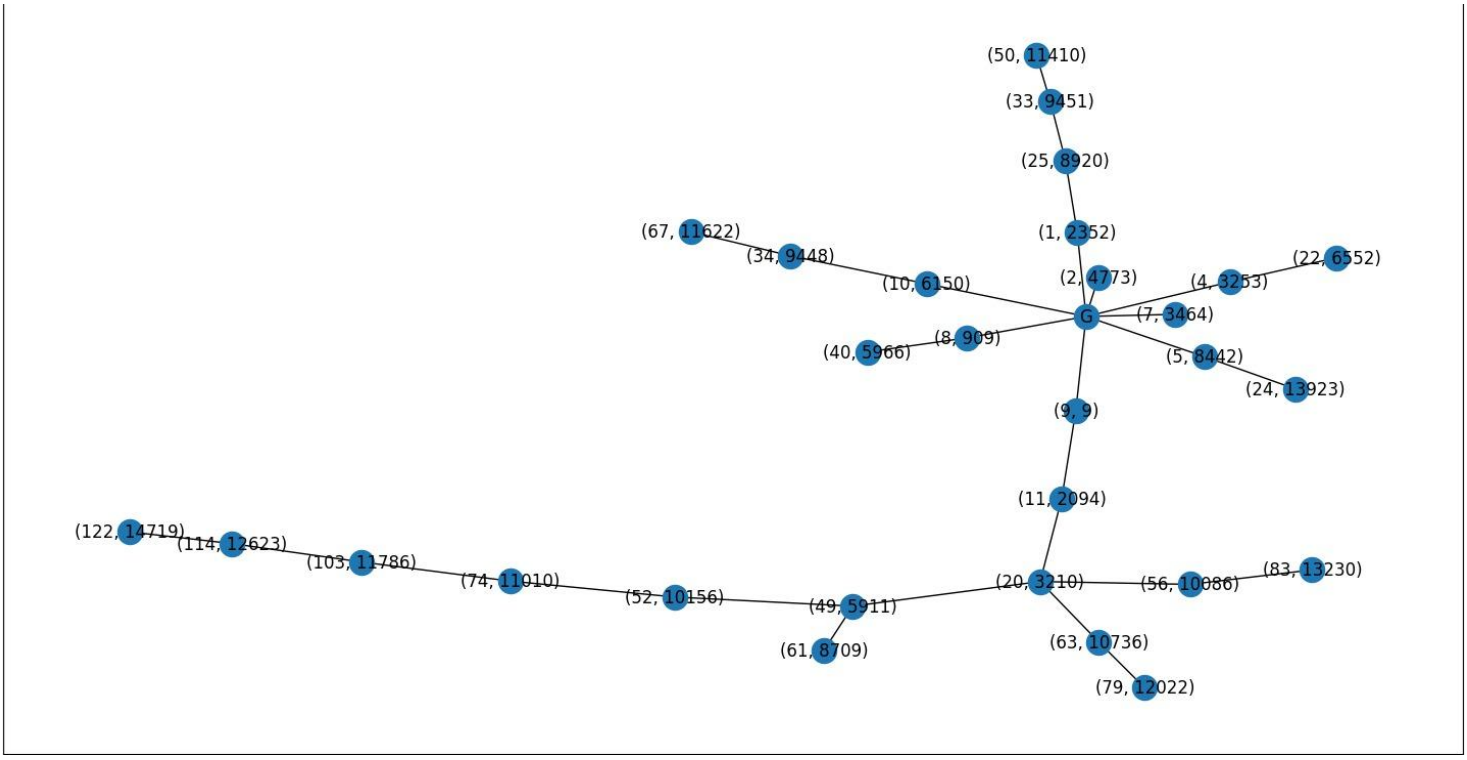
1.1 Tree of node 2



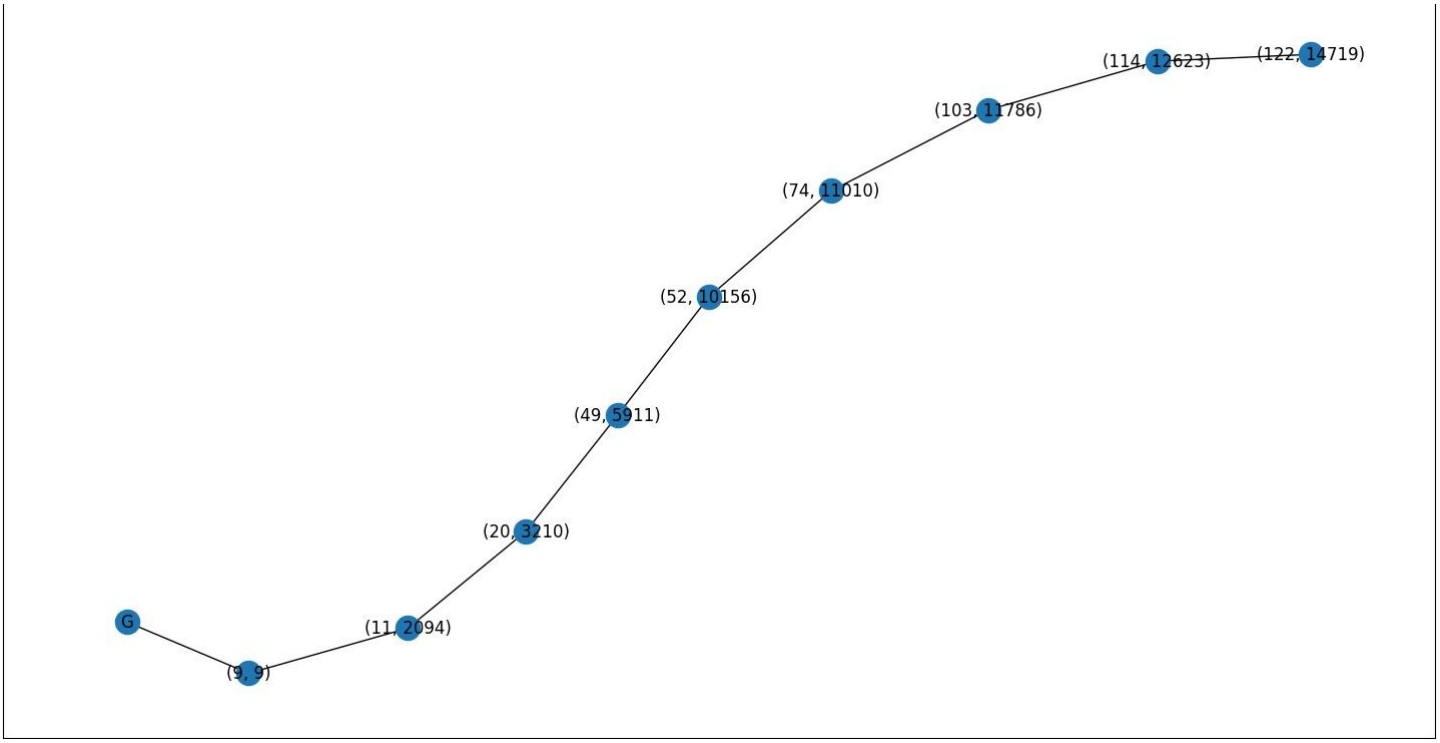
1.2 Longest chain of node 2



2.1 Tree of node 8

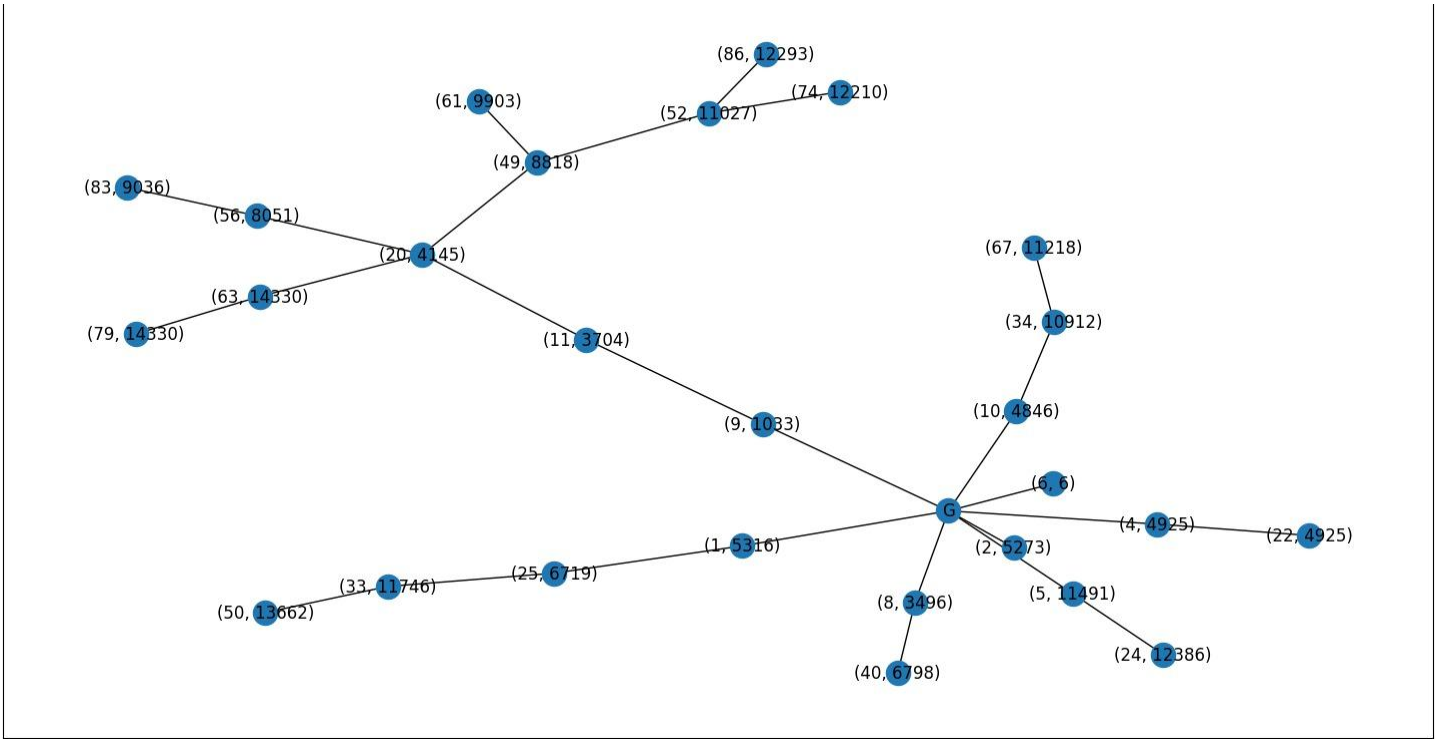


2.2 Longest chain of node 8

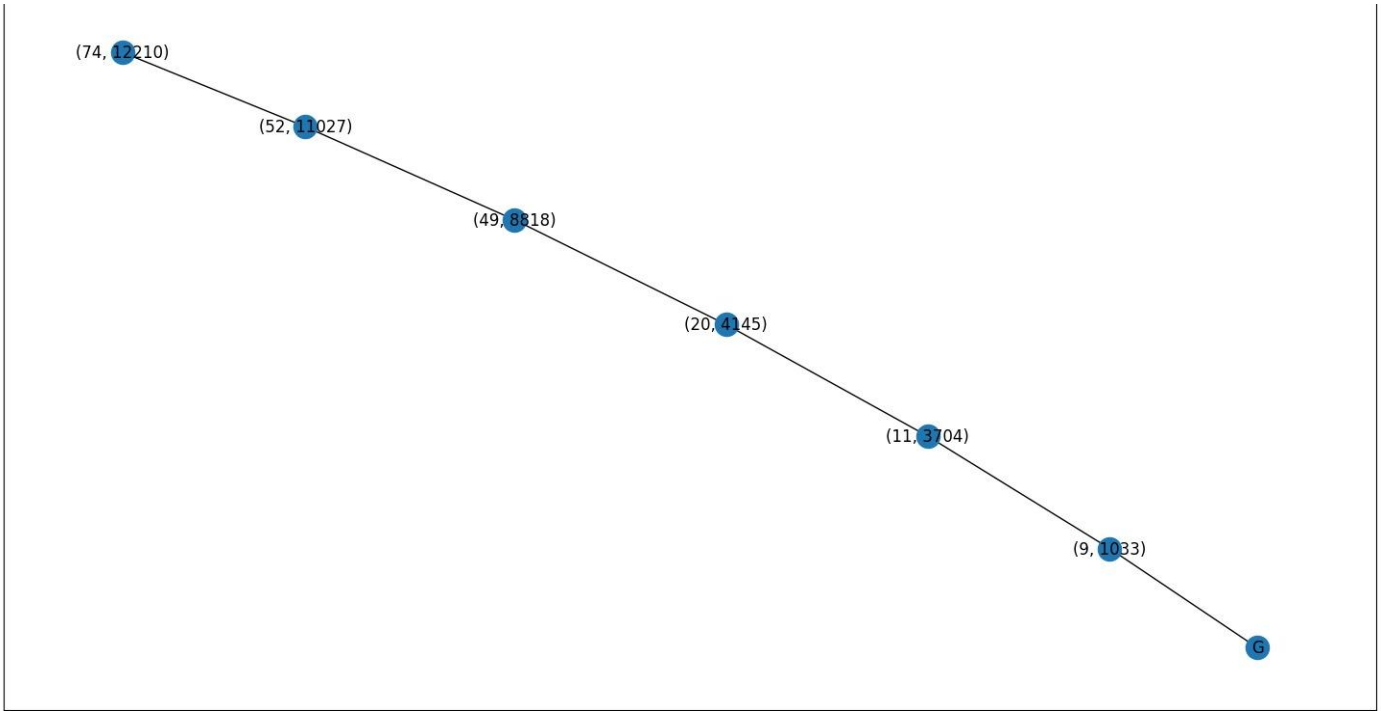


Fast Node + low CPU

3.1 Tree of node 5

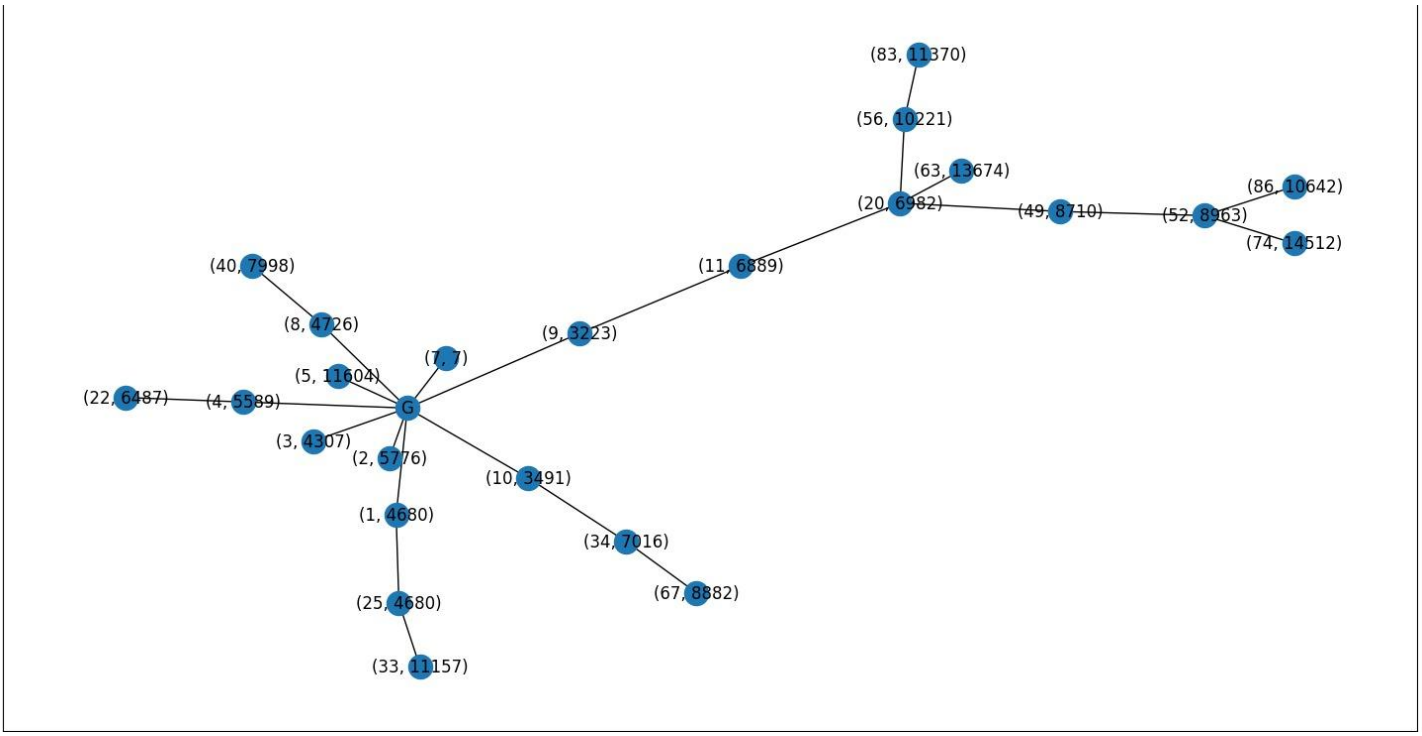


3.2 Longest chain of node 5

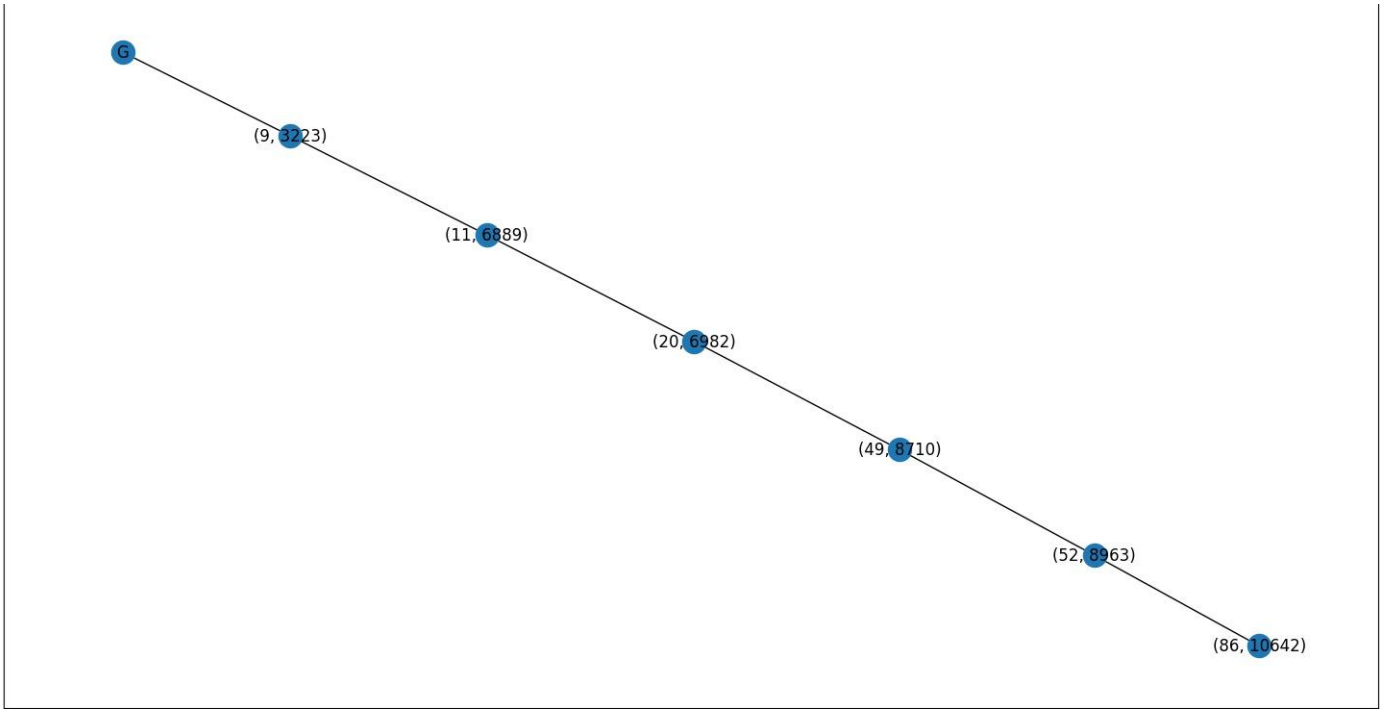


Slow Node + low CPU

4.1 Tree of node 6

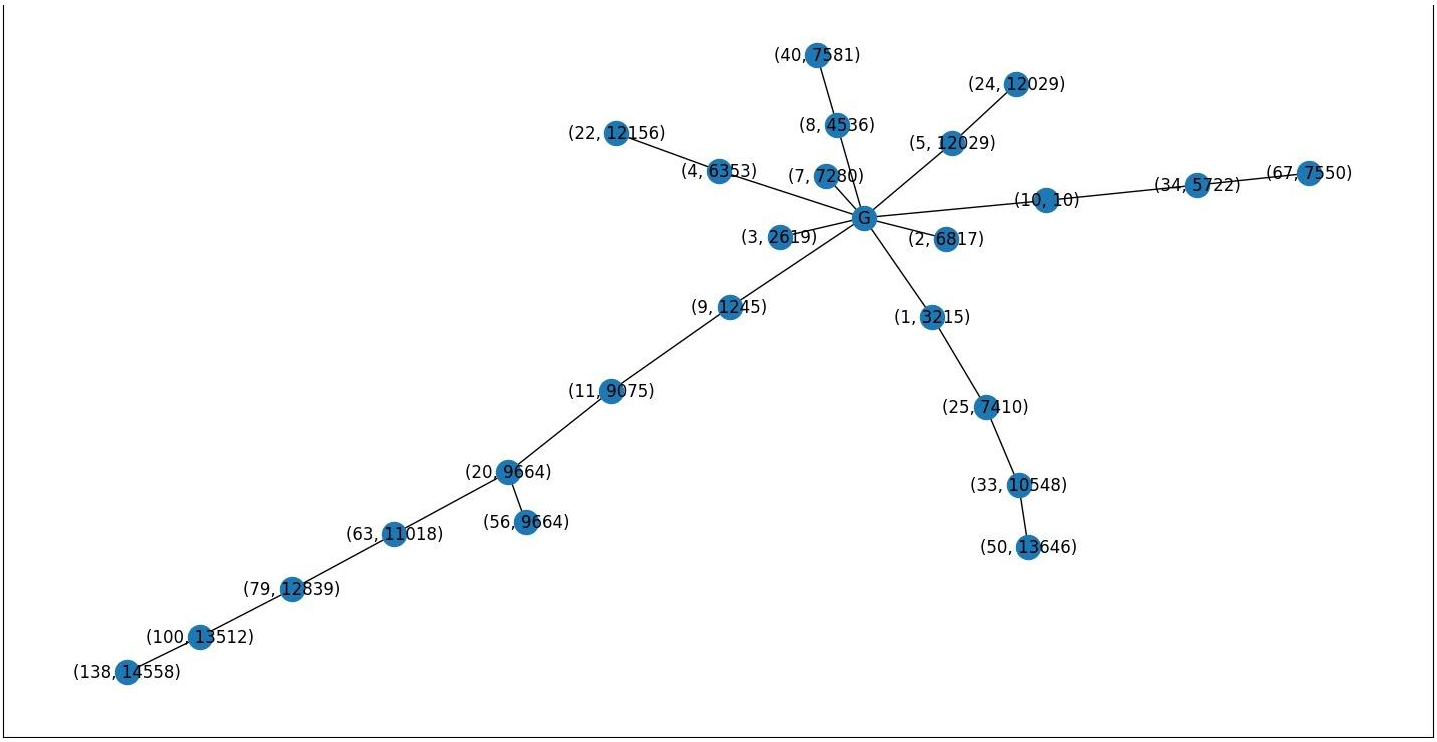


4.2 Longest chain of node 6

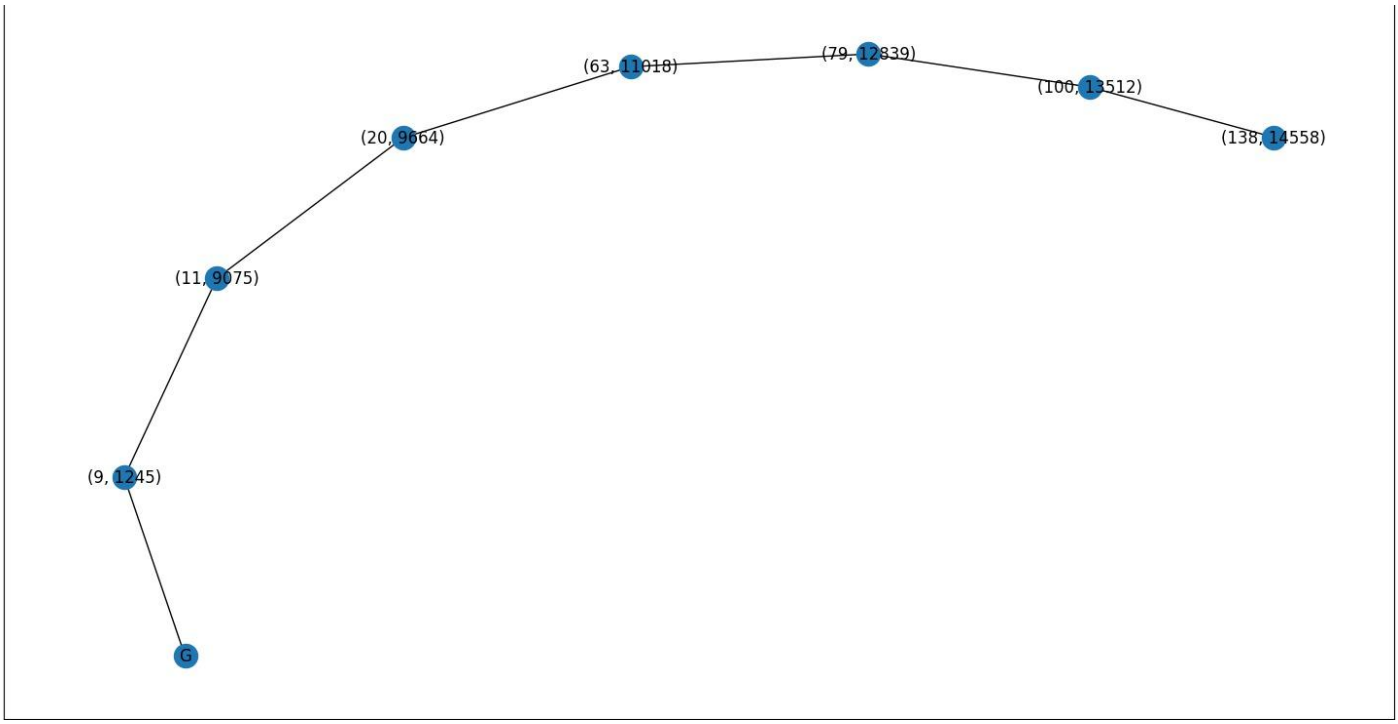


Fast Node + High CPU

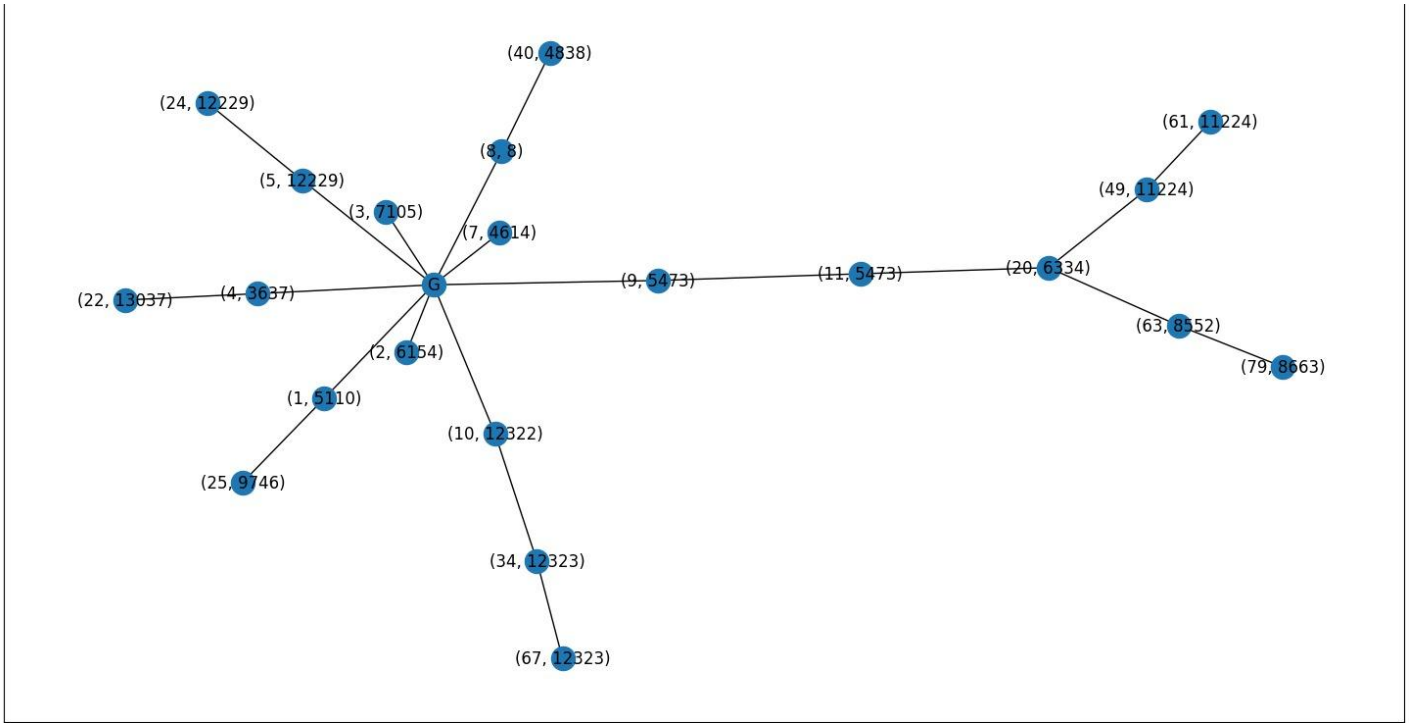
5.1 Tree of node 9



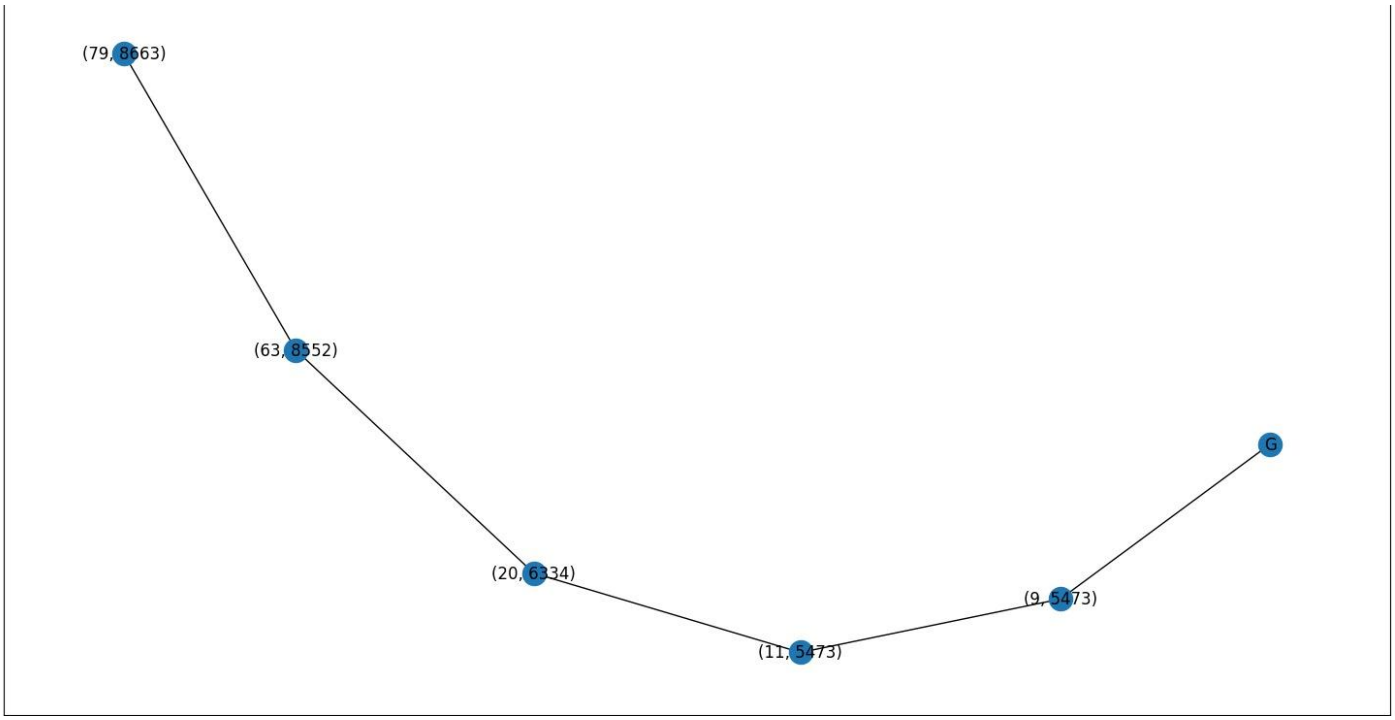
5.2 Longest chain of node 9



6.1 Tree of node 7



6.2 Longest chain of node 7



- **Simulating Selfish Attack**

Parameters

- N=number of nodes
- Ttx= Transaction Interarrival time mean
- z0 = Percent of slow nodes
- z1 = Percent of low CPU nodes
- Hashing Power(alpha)
- Fraction of peers that are connected to adversary(zeta)

Observations for selfish Mining

- Hashing power(alpha)=0.15

Zeta	0.25	0.50	0.75
MPU Adversary	0.666666	0.66	0.64
MPU Overall	0.95	0.875	0.8
R_pool	0.105263	0.093	0.125

- Hashing power(alpha)=0.33

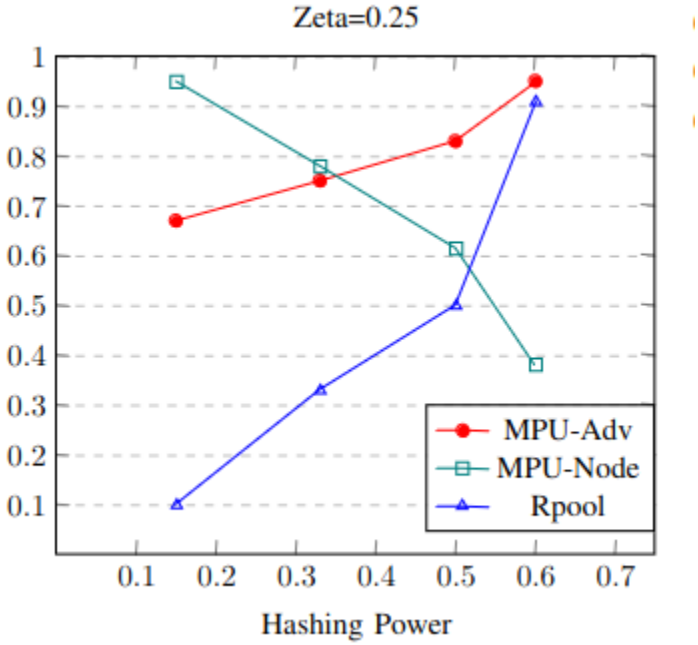
Zeta	**0.25**	**0.50**	**0.75**
MPU Adversary	0.75	0.75	0.66
MPU Overall	0.78	0.80	0.70
R_pool	0.33333333	0.34	0.35

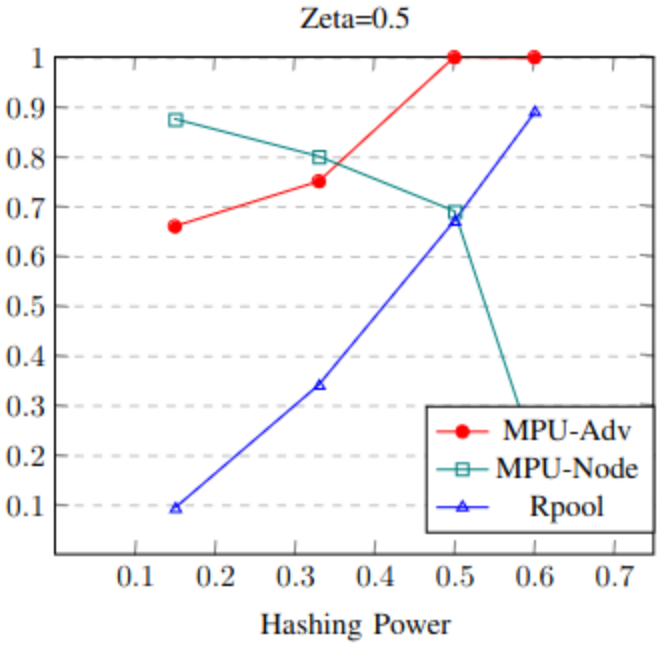
- Hashing power(alpha)=0.50

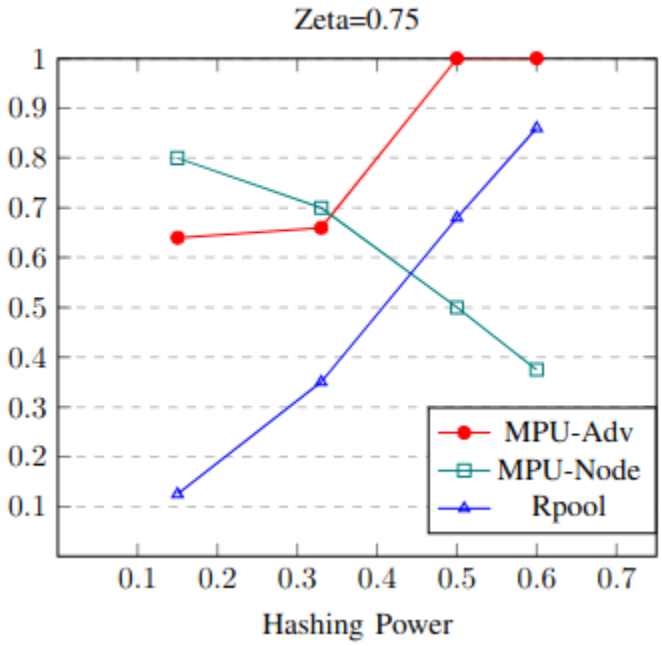
Zeta	**0.25**	**0.50**	**0.75**
MPU Adversary	0.8333	1.0	1.0
MPU Overall	0.615	0.6923076	0.5
R_pool	0.5	0.6666666	0.684210

- Hashing power(alpha)=0.60

Zeta	0.25	0.50	0.75
MPU Adversary	0.95	1.0	1.0
MPU Overall	0.382	0.212	0.375
R_pool	0.91	0.89	0.86







Observations for Stubborn Mining

- Hashing power(alpha)=0.15

Zeta	0.25	0.50	0.75
MPU Adversary	0.25	.33	0.52
MPU Overall	0.89	.90	0.905
R_pool	0.052	0.058	0.052

- Hashing power(alpha)=0.33

Zeta	0.25	0.50	0.75
MPU Adversary	0.75	0.75	0.80
MPU Overall	0.87	0.78	0.61
R_pool	0.23	0.27	0.44

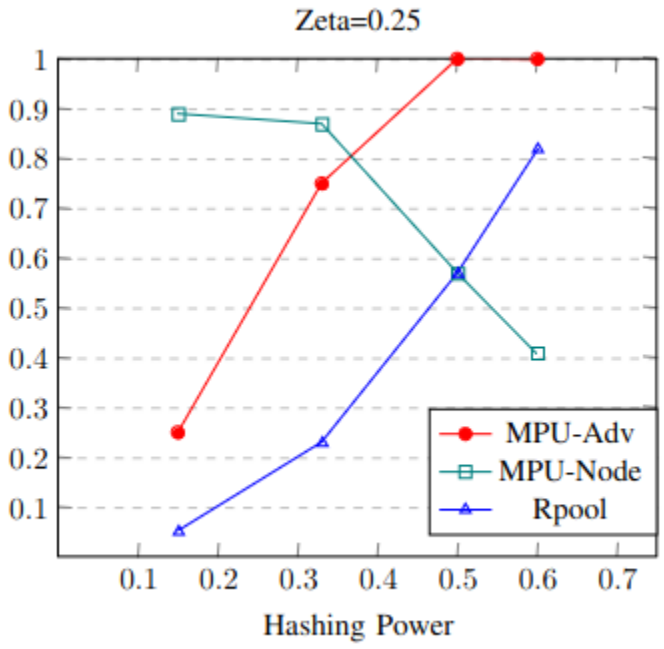
- Hashing power(alpha)=0.50

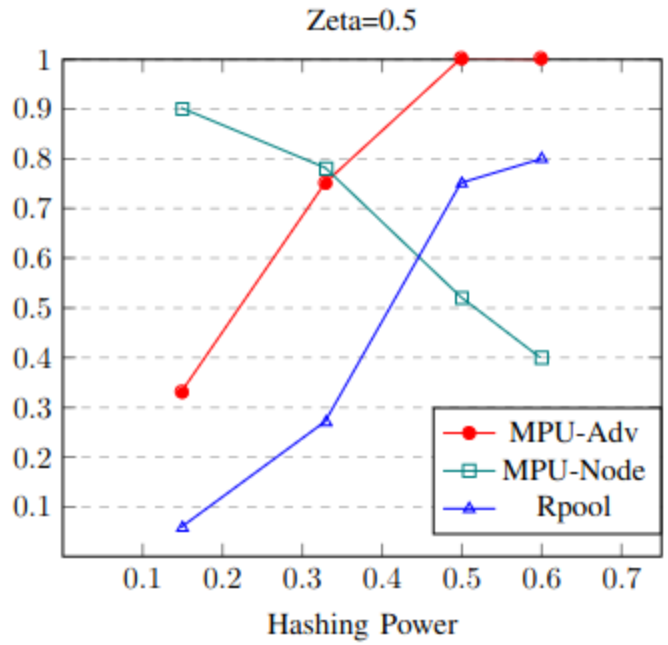
Zeta	0.25	0.50	0.75
MPU Adversary	1	1	1
MPU	0.57	0.52	0.53

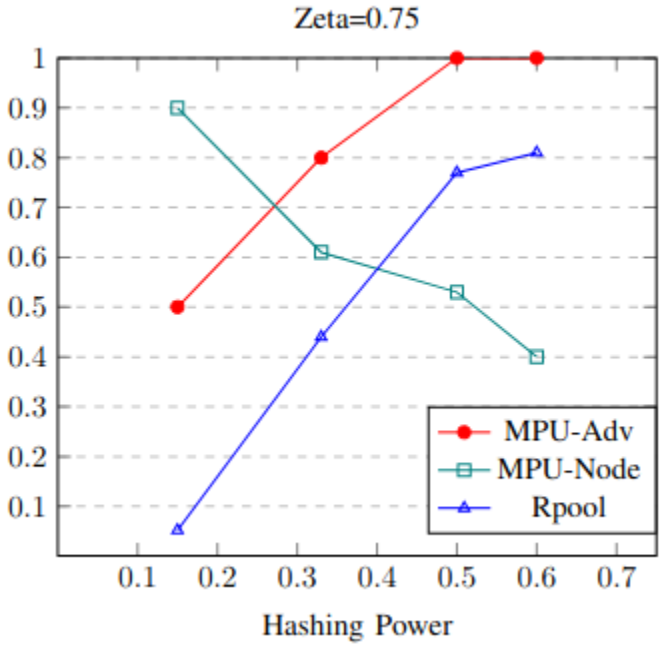
Overall			
R_pool	0.57	0.75	0.77

- Hashing power(alpha)=0.60

Zeta	**0.25**	**0.50**	**0.75**
MPU Adversary	1	1	1
MPU Overall	0.4	0.42	0.41
R_pool	0.82	0.8	0.81





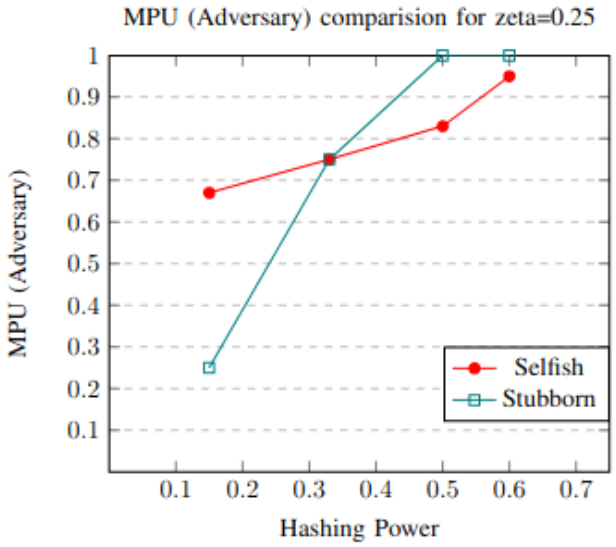


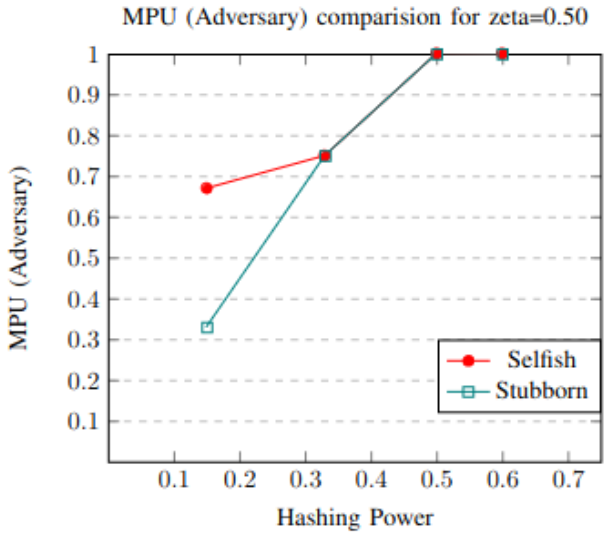
Observation regarding MPU_adv, MPU_overall and R_Pool:

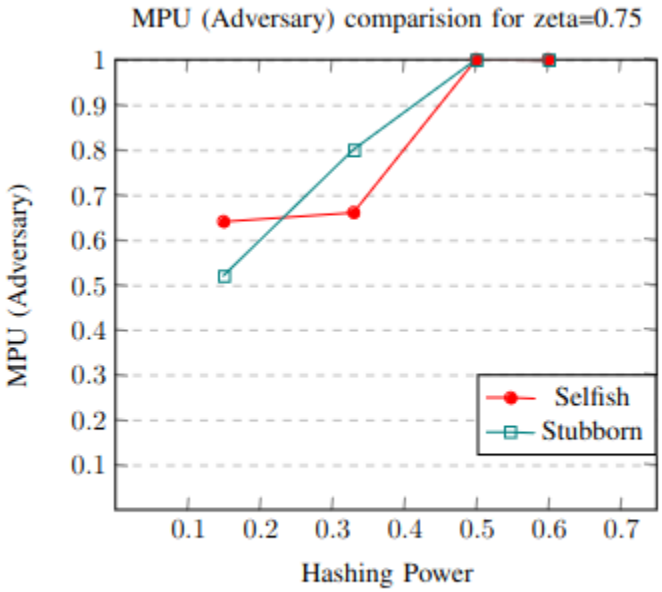
- From the above graphs, we can see that with increased hashing power, the ratio of adversary block in the main chain increases and finally becomes 1. Since the attacker has more power, it has more chances to mine a new block than its peers.
- We see that with increased hashing power, the value of MPU_Node increases as more of the attacker's mined blocks are now becoming part of the public chain. Hence, a lesser number of attackers block are getting into forks, or when in forks, there is a high chance that it gets adopted as public chain in later stages.
- We also observe that MPU_overall decreases with the adversary's hashing power. It is because now more and more of honest miners block's are getting abandoned as the adversary releases a private block to complete with honest miners' block. Due to lesser power than the adversary, the next block on the honest block gets mined later, and hence by the time it gets mined attacker has already mined a block and taken some lead, and thus this honest block gets abandoned.

Variation of MPU Ratio with hashing power for both attacks

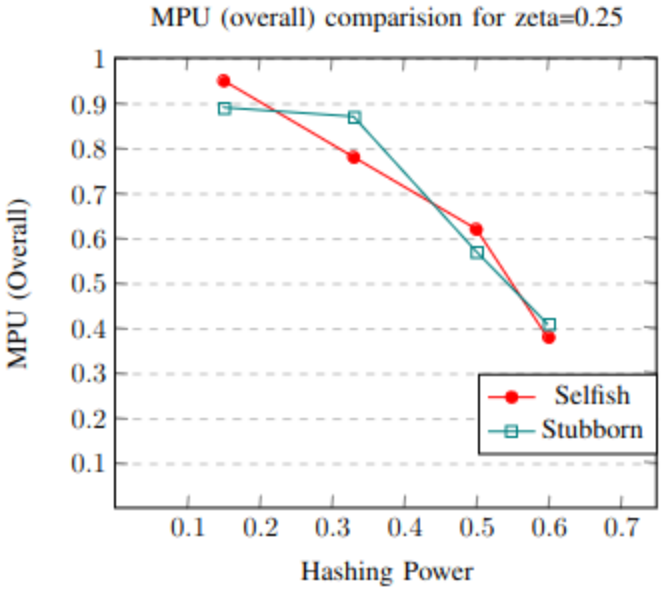
1. MPU (Adversary) for varying hashing power.

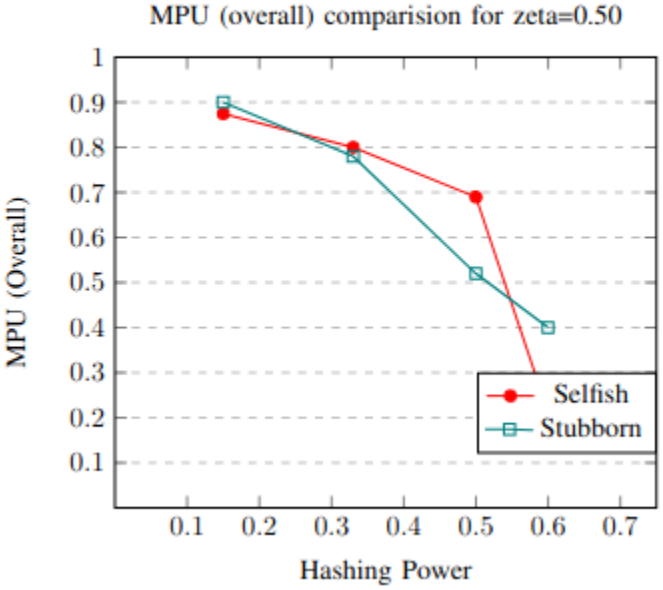


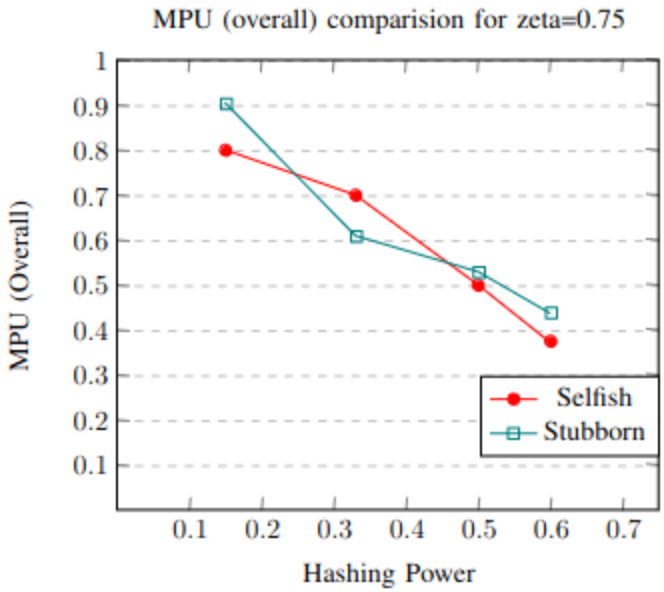




2. MPU (Overall) for varying hashing power.


MPU (overall) comparision for zeta=0.25
MPU (Overall)
Hashing Power
Selfish
Stubborn
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.1
0.2
0.3
0.4
0.5
0.6
0.7


MPU (overall) comparision for zeta=0.50
MPU (Overall)
Hashing Power
Selfish
Stubborn
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.1
0.2
0.3
0.4
0.5
0.6
0.7



We see the following details regarding the Selfish mining attack: MPU overall trend and MPU adv have contrasting patterns. MPU adv grows as the adversary's hashing power increases. This is expected because as the attacker's mining capacity increases, more and more of its blocks are added to the main chain. After all, it can quickly wipe out the honest blocks. If more blocks produced by honest miners are discarded, the number of honest blocks in the main chain will drop, which will have a negative impact on the MPU overall.

In the case of stubborn mining, when the adversary has lesser hashing power and hence will mine blocks at a slower rate. Since the attacker does not release multiple blocks to win the race, then there is a high chance that due to the equal length of the forked public chain, it is possible that the honest miners keep on mining the public block seen earlier. Hence, there

are higher chances that attackers mined blocks are not getting added to the public chain, and hence the private chain of the attacker gets orphaned at a higher rate. Therefore MPU adversary is increasing at lower rate then then that of the selfish miner.

Conclusions

On changing hashing power, we observe the following:

- As hashing power increases, the value of MPU node_adv increases on both selfish and stubborn mining.
- As hashing power increases, the value of MPU node_overall decreases on both selfish and stubborn mining.
- As hashing power increases, the ratio of adversary's blocks in the main chain increases.

On changing the fraction of honest nodes ζ, an adversary is connected to:

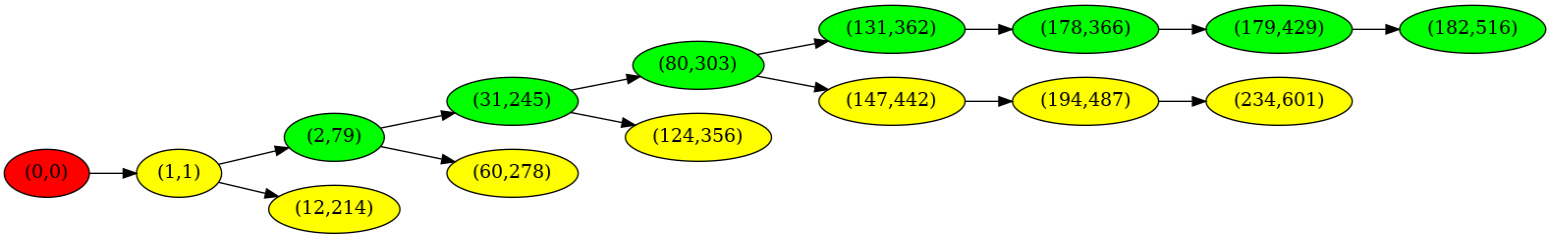
- When Hashing power is low, then there is no significant pattern observed on both MPU node_adv and MPU node_overall
- When Hashing power is more then the MPU node_adv reaches 1.0 at exponentially, and MPU node_overall is low in almost every ζ

Pictures Of Blockchains for some particular cases

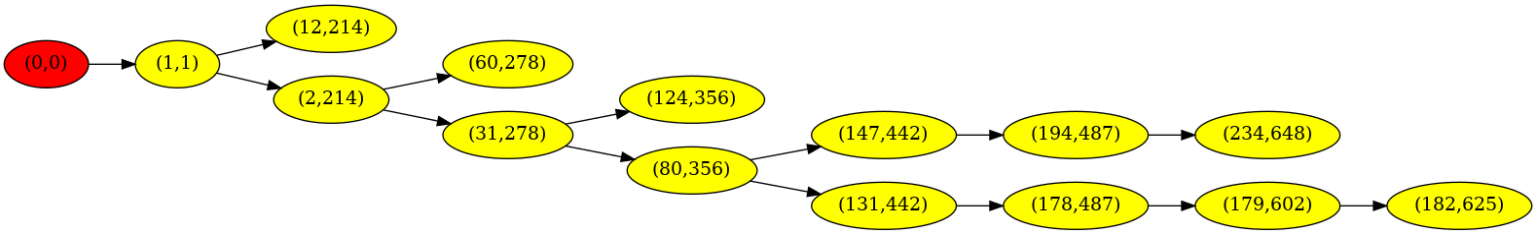
1. Stubborn Mining

a. n=25, Ttx=150, z0=50, z1=10, <u>alpha=50, zeta=75</u>

i. Adversary node

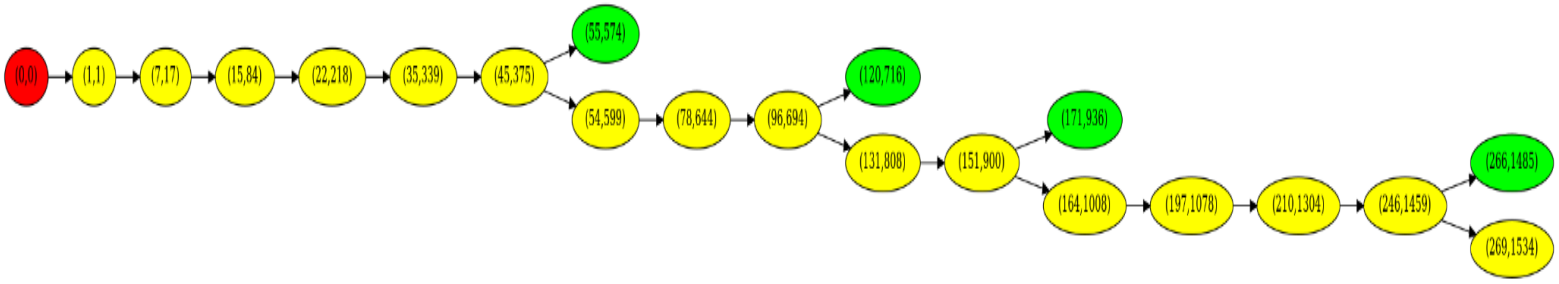


ii. Honest Node

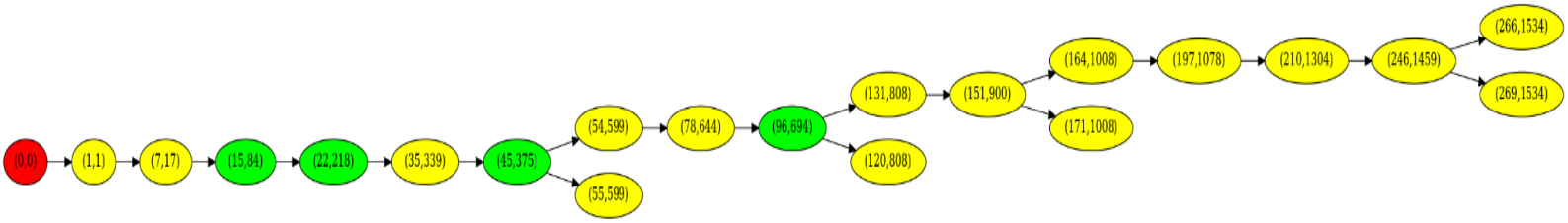


b. n=25, Ttx=150, z0=50, z1=10, <u>alpha=15, zeta=50</u>

i. Adversary Node

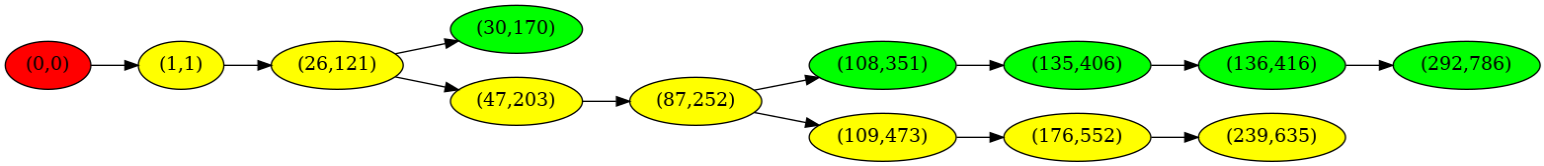


ii. Honest Node

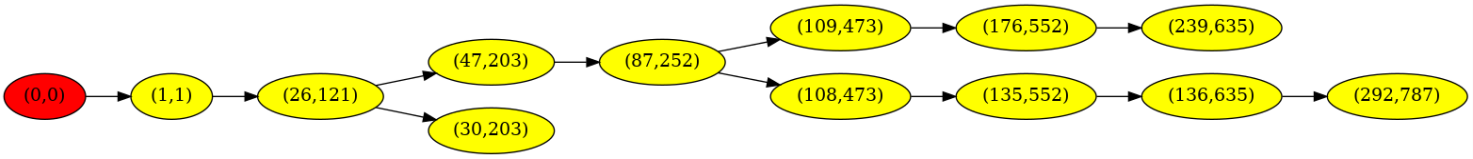


c. n=25, Ttx=150, z0=50, z1=10, alpha=33, zeta=25

i. Adversary Node



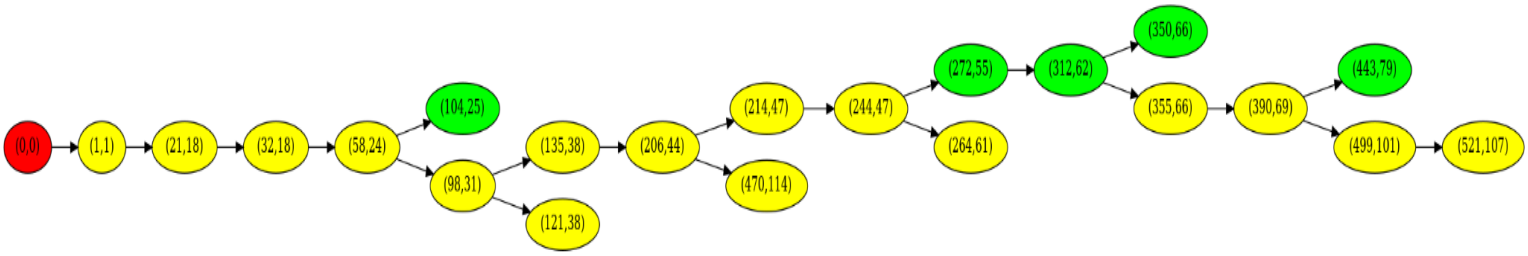
ii. Honest Node



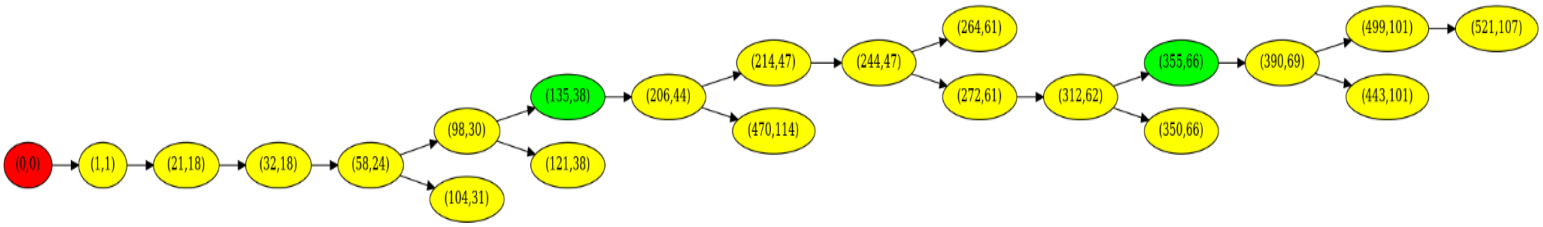
2. Selfish Mining

a. n=25, Ttx=150, z0=50, z1=10, alpha=15, zeta=50

i. Adversary Node

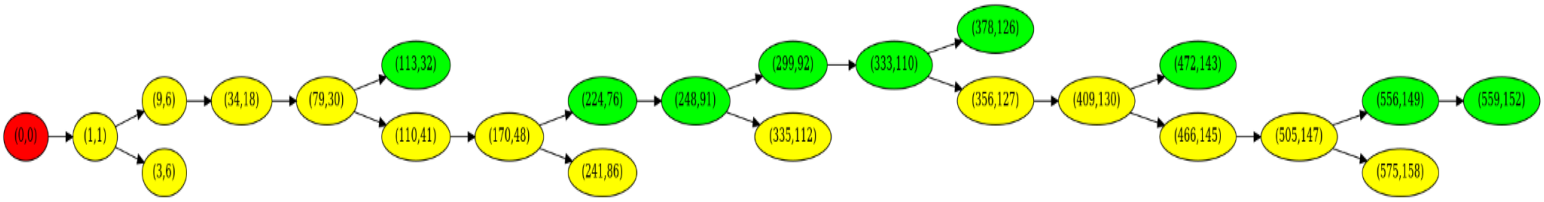


ii. Honest Node



b. n=25, Ttx=150, z0=50, z1=10, <u>alpha=33, zeta=50</u>

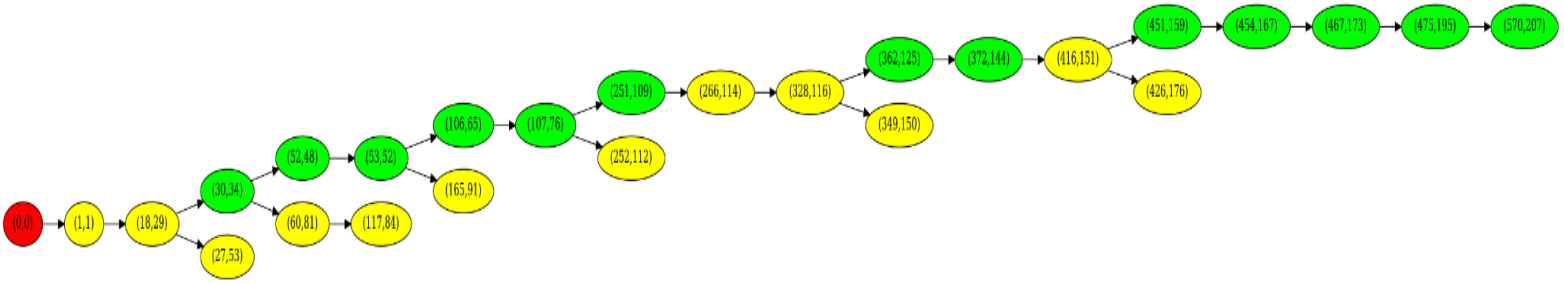
i. Adversary Node



ii. Honest Node

c. n=25, Ttx=150, z0=50, z1=10, <u>alpha=50, zeta=75</u>

i. Adversary Node



ii. Honest Node