CS-765 Assignment-2 Report

Simulating a double selfish mining attack using the P2P Cryptocurrency Network

200050019 - Bagathi Shiv Kiran
200050020 - Bale Teja Rama Chandra Murthy
200050075 - Moganti Harshadeep

Selfish Mining :

The idea behind Selfish Mining Attack is for the selfish miner (Adversary) to keep his discovered blocks private, thereby intentionally forking the chain. The honest nodes continue to mine on the public chain, while the adversary mines on his own private branch. If the adversary mines more blocks, he develops a longer lead over the public chain and continues to keep these new blocks private. When the public branch approaches the adversary's private branch in length, he reveals blocks from its private chain to the public.

In the paper written by Eyal and Sirer, they discuss the protocol that protects against selfish mining pools that command less than 1 /4 of the resources. This threshold is lower than the wrongly assumed 1 /2 bound, but better than the current reality where a group of any size can compromise the system.

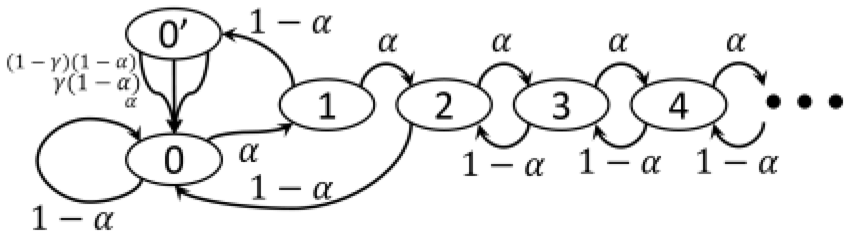


Fig. 1: State machine with transition frequencies.

Experiments and Insights:

Parameters we are varying in our experiments :

n : number of peers in the network
z0 : percentage of slow peers   = 50% in honest. Attackers are always fast.
h0: hash power fraction of attacker 0
h1: hash power fraction of attacker 1
ttx : mean inter-arrival time between transactions
I : average time taken to mine a block
save : save the events in the file

Definitions of Defined Ratios :

$$MPU_{node_{adv}} = \frac{\text{Number of block mined by an adversary in final public main chain}}{\text{Total number of blocks mined by this adversary overall}}$$

Mined Blocks per Unit Node_adversasry

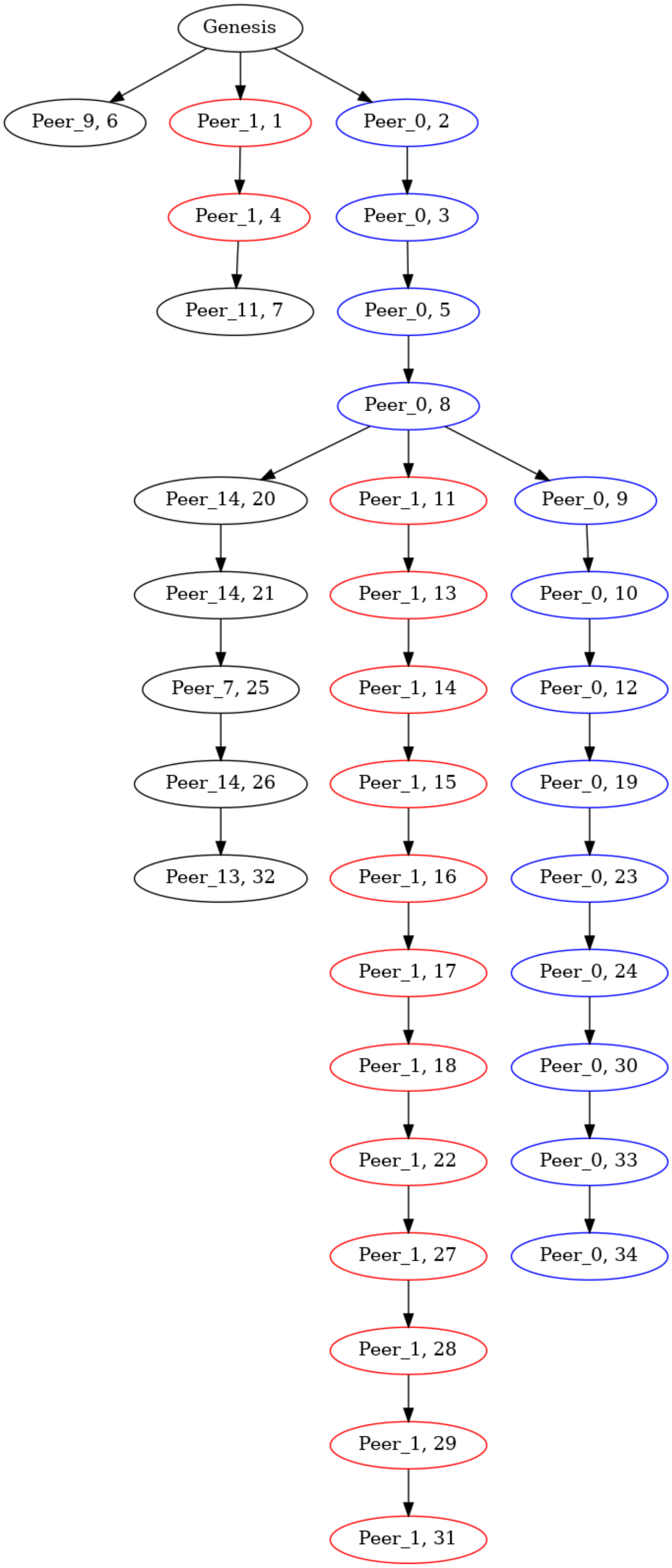
$$MPU_{node_{overall}} = \frac{\text{Number of block in the final public main chain}}{\text{Total number of blocks generated across all the nodes}}$$

Case 1 : I = 1000.0, h0 = 0.3, h1 = 0.4

```
Simulating the cryptocurrency network with 15 peers
z0 = 50, ttx = 10, I = 1000.0, h0 = 0.3, h1 = 0.4, stop = 1.0
MPU of selfish miner 0:  0.46153846153846156
MPU of selfish miner 1:  0.6875
MPU overall:  0.4864864864864865
Blocks created by selfish miner 0:  13  Blocks in longest chain:  6
Blocks created by selfish miner 1:  16  Blocks in longest chain:  11
Length of longest Chain :  18  Total Blocks :  37
```

```
z0 = 50, ttx = 10, I = 1000.0, h0 = 0.3, h1 = 0.4, stop = 1.0
MPU of selfish miner 0:  0.3076923076923077
MPU of selfish miner 1:  0.8571428571428571
MPU overall:  0.4857142857142857
Blocks created by selfish miner 0:  13  Blocks in longest chain:  4
Blocks created by selfish miner 1:  14  Blocks in longest chain:  12
Length of longest Chain :  17  Total Blocks :  35
```

Blockchain :


Genesis
Peer_9, 6
Peer_1, 1
Peer_0, 2
Peer_1, 4
Peer_0, 3
Peer_11, 7
Peer_0, 5
Peer_0, 8
Peer_14, 20
Peer_1, 11
Peer_0, 9
Peer_14, 21
Peer_1, 13
Peer_0, 10
Peer_7, 25
Peer_1, 14
Peer_0, 12
Peer_14, 26
Peer_1, 15
Peer_0, 19
Peer_13, 32
Peer_1, 16
Peer_0, 23
Peer_1, 17
Peer_0, 24
Peer_1, 18
Peer_0, 30
Peer_1, 22
Peer_0, 33
Peer_1, 27
Peer_0, 34
Peer_1, 28
Peer_1, 29
Peer_1, 31

Observations from the Blockchain :

We observe that our adversaries release their blocks only in the cases that we have forks i.e., unless an honest releases their block, an adversary never reveals his block(unless we broadcast at the end).AlsoRed Chain (Peer1) wins because it has higher hashing power than Blue Chain (Peer 0). It generates more blocks and has a longer chain.

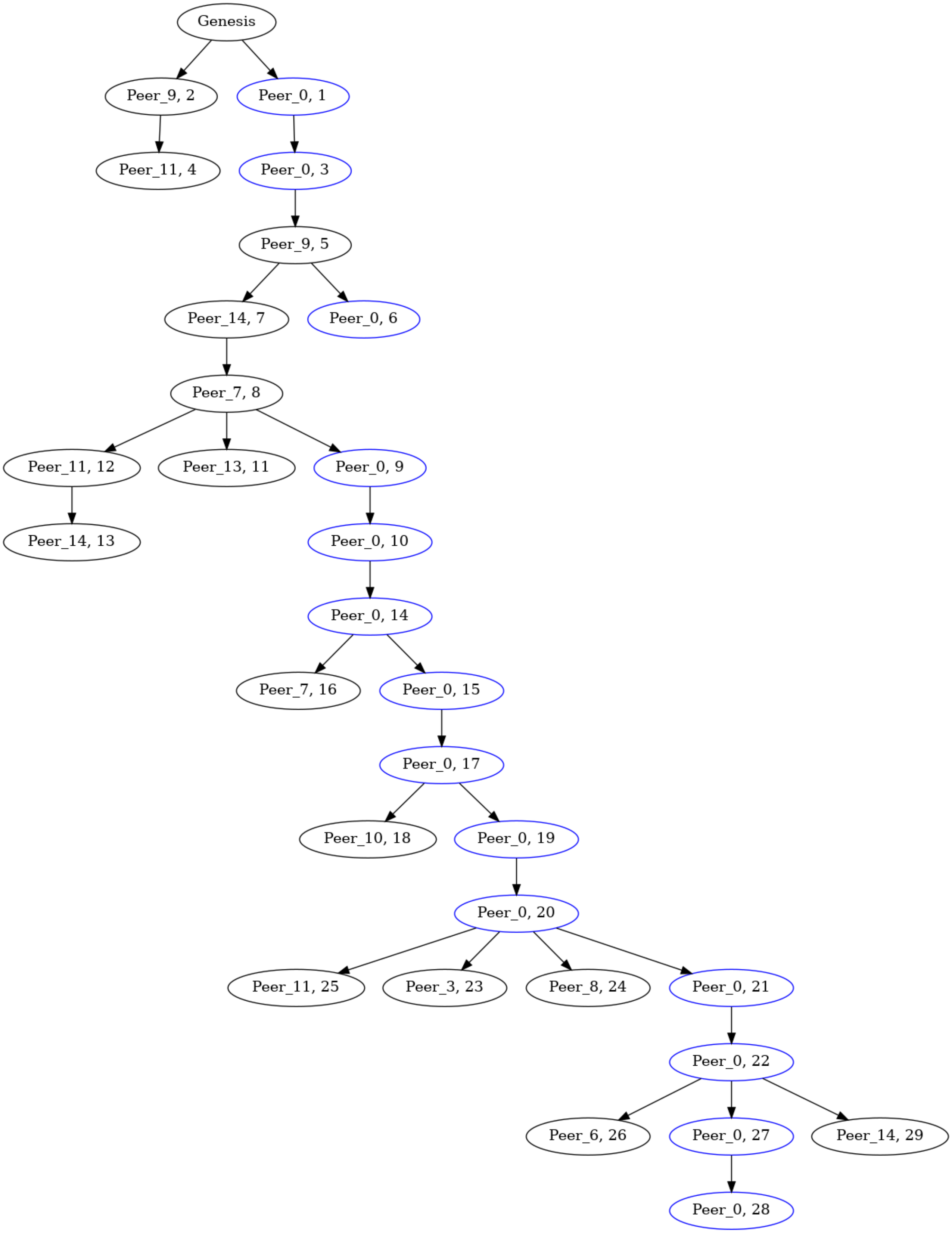
Case 2: :

```
Simulating the cryptocurrency network with 15 peers
z0 = 50, ttx = 10, I = 1000.0, h0 = 0.4, h1 = 0.0001, stop = 1.0
MPU of selfish miner 0:  0.8666666666666667
MPU of selfish miner 1: 0
MPU overall:  0.5483870967741935
Blocks created by selfish miner 0:  15  Blocks in longest chain:  13
Blocks created by selfish miner 1:  0  Blocks in longest chain:  0
```

Observations from the Blockchain :

We observe that our adversaries release their blocks only in the cases that we have forks i.e., unless an honest releases their block, an adversary never reveals his block(unless we broadcast at the end). Peer1 has no blocks as its hashing power fraction is 0.

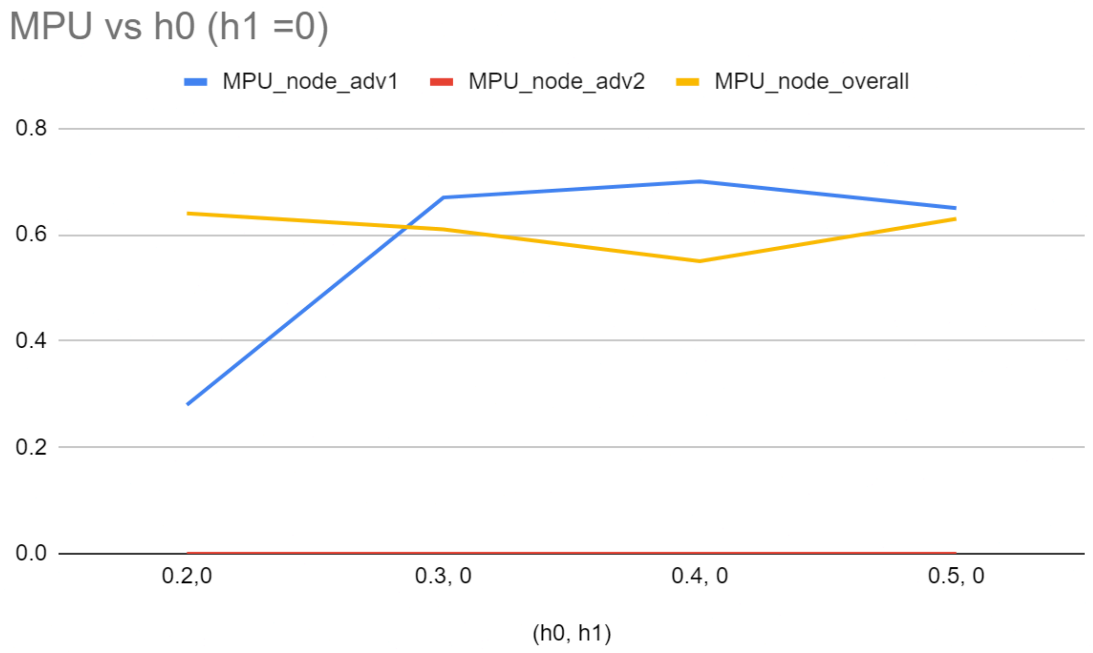
Blockchain:


Genesis
Peer_9, 2
Peer_0, 1
Peer_11, 4
Peer_0, 3
Peer_9, 5
Peer_14, 7
Peer_0, 6
Peer_7, 8
Peer_11, 12
Peer_13, 11
Peer_0, 9
Peer_14, 13
Peer_0, 10
Peer_0, 14
Peer_7, 16
Peer_0, 15
Peer_0, 17
Peer_10, 18
Peer_0, 19
Peer_0, 20
Peer_11, 25
Peer_3, 23
Peer_8, 24
Peer_0, 21
Peer_0, 22
Peer_6, 26
Peer_0, 27
Peer_14, 29
Peer_0, 28

Analysis of MPU_adv, MPU_Overall :

h1 = 0

(h0, h1)	MPU_node_adv1	MPU_node_adv2	MPU_node_overall
0.2,0	0.28	0	0.64
0.3, 0	0.67	0	0.61
0.4, 0	0.7	0	0.55
0.5, 0	0.65	0	0.63

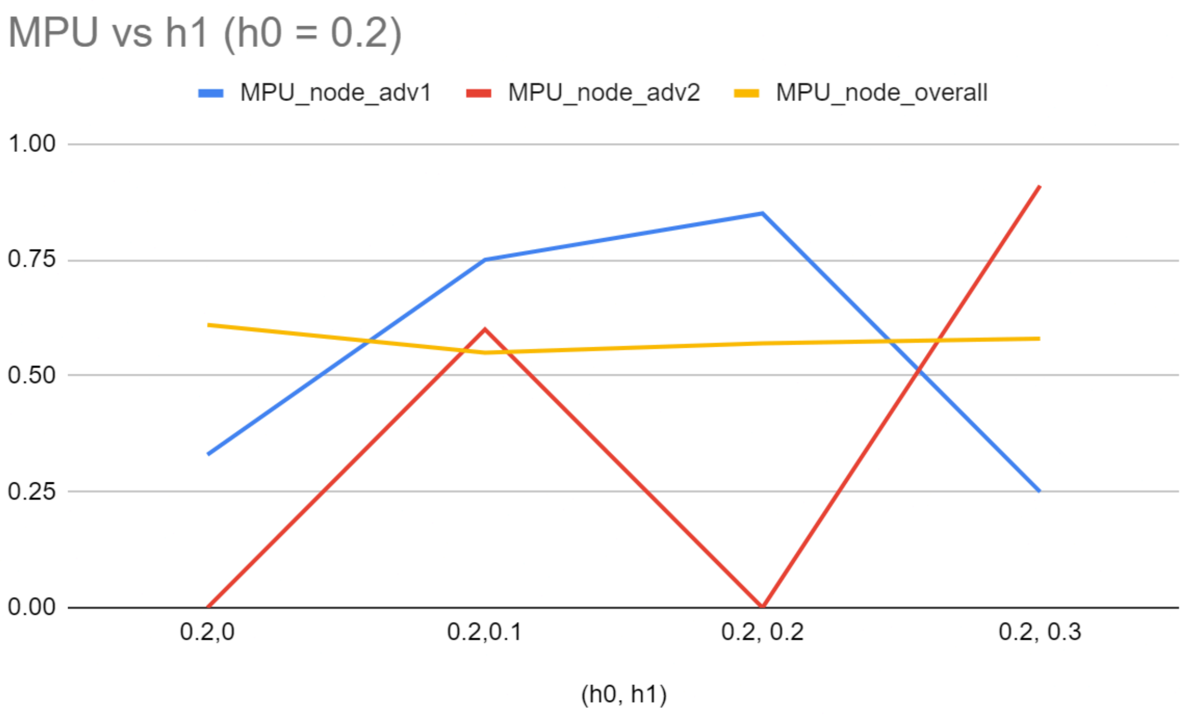


MPU adversary is able to get increased reward when he has around 0.3 to 0.4, after that we observe that the reward is not increasing as steeply as before

MPU Overall when in between 0.3-0.4 it is decreasing because attacker has sufficient hashing so that he is always ahead of the honest chain and even the honest Peers have enough considerable hashing so that too can generate blocks, causing more forks and subsequently less MPU_overall but when hashing of adversary is 0.5 the honest Peers don’t have sufficient hashing power so number of blocks created by them decreases. Hence decreasing the number of forks and increasing the MPU overall

h0 = 0.2

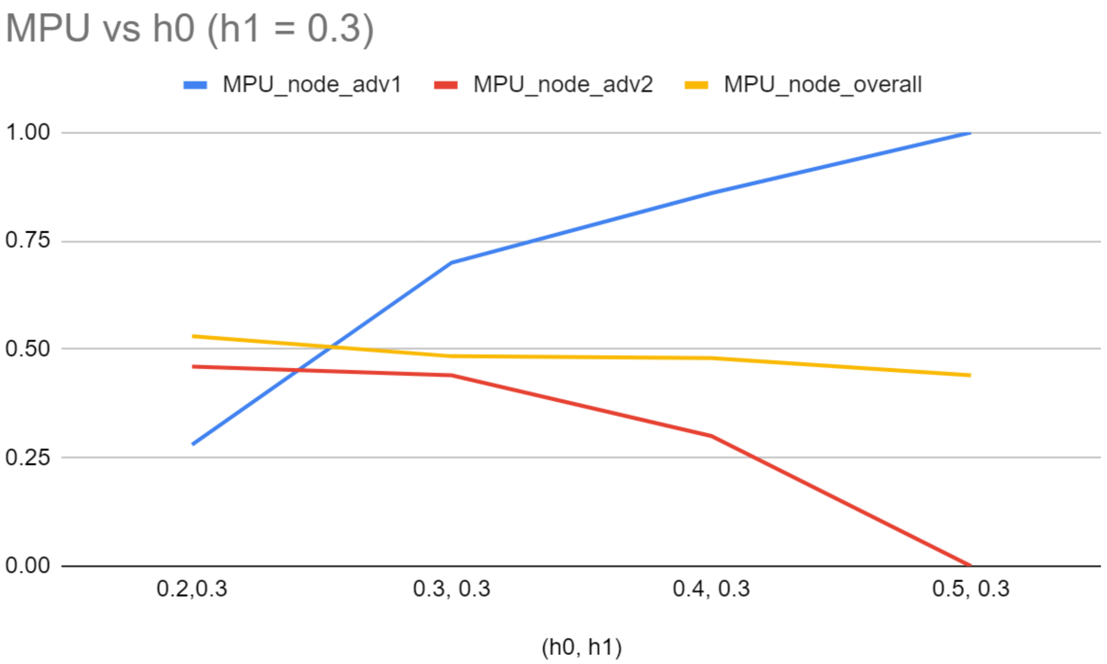
(h0, h1)	MPU_node_adv1	MPU_node_adv2	MPU_node_overall
0.2,0	0.33	0	0.61
0.2,0.1	0.75	0.6	0.55
0.2, 0.2	0.85	0	0.57
0.2, 0.3	0.25	0.91	0.58



h1 = 0.3

(h0, h1)	MPU_node_adv1	MPU_node_adv2	MPU_node_overall
0.2,0.3	0.28	0.46	0.53
0.3, 0.3	0.7	0.44	0.484
0.4, 0.3	0.86	0.3	0.48

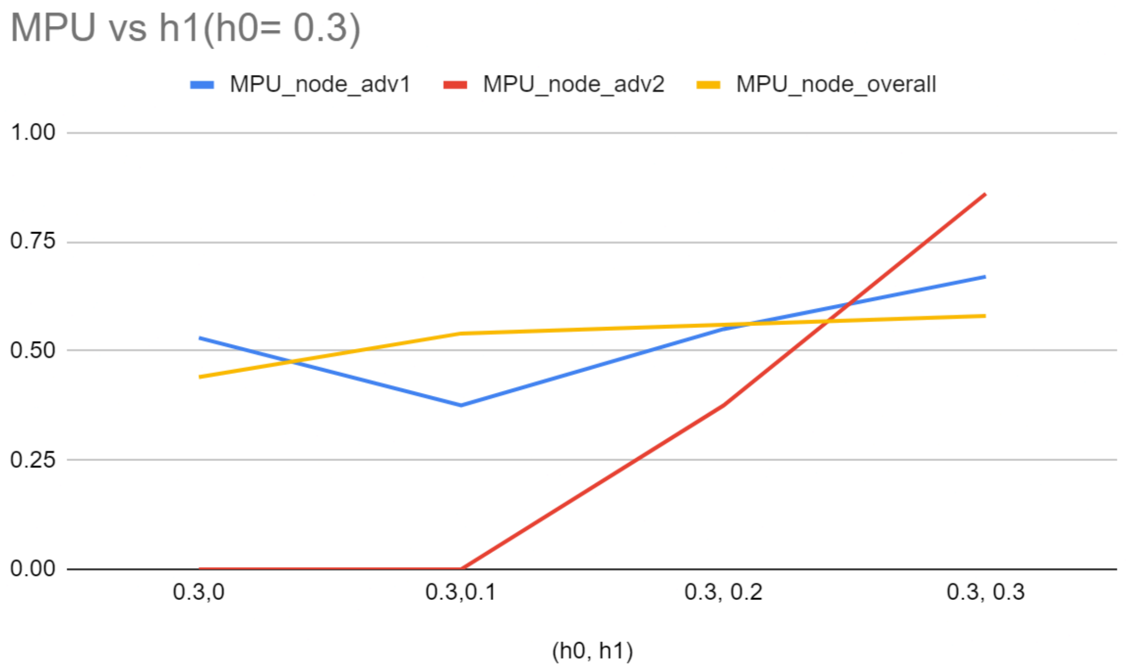
0.5, 0.3	1	0	0.44



MPU_node_adv1 as expected is increasing as we are increasing the hashing fraction of the adversary 1.
As the total Hashing Fraction of adversaries increasing beyond 0.5, we observe that MPU overall is around 0.5 (i.e fork due to competition between two adversaries only)

h0 = 0.3

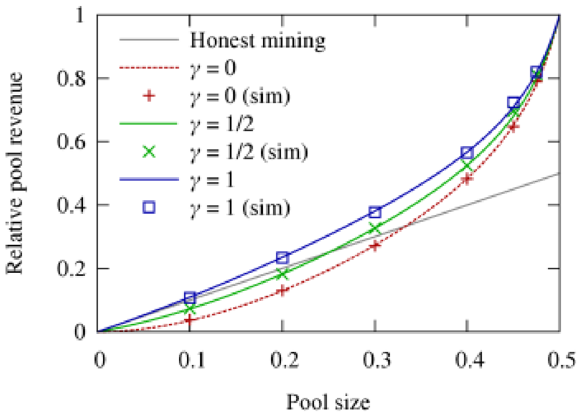
(h0, h1)	MPU_node_adv1	MPU_node_adv2	MPU_node_overall
0.3,0	0.53	0	0.44
0.3,0.1	0.375	0	0.54
0.3, 0.2	0.55	0.375	0.56
0.3, 0.3	0.67	0.86	0.58



Until h1 reaches h0 value as the trend doesn’t matter to h0, the hash fraction be it of honest or selfish miner.

Observations & Overall Insights :

1. The reason for dip in (MPU_overall) at 0.4 might be because when compared to 0.3 case, hashing power of the adversary is somewhat high, so the attacker always takes the lead, and the honest nodes also have considerable hashing power and they create considerable blocks. All these blocks will be in a fork which will ultimately be won by the adversary , thus decreasing the MPU_overall. In the 0.5 case, the hashing power of honest nodes decreased and they could not create more blocks, thus decreasing the number of forks.So MPU_overall increases.
2. There is no effect of Ttx (Transaction Interarrival Time) on selfish Mining
3. When I is low, there are latency effects and has more forks from honest Peers but this doesn't effect the Private Blockchain of the adversary facilitating in creating longer chains irrespective of the forks present in the main blockchain.
4. MPU_adversary0 is showing an upward trend as h0 increasing as expected.
5. Adversary 0 is getting increased rewards when h0 is in 0.3-0.4



6. Even when the number of Peers is large, we get a similar steep increase in rewards for selfish miners between 0.3-0.