CS 765 Assignment 2

Simulation of a Selfish Mining Attack

Parth Dwivedi - 200050100

March 2023

1 Selfish Mining

All simulations in this section involve 100 nodes, 50 percent slow nodes, 50 percent low CPU nodes, and mean block time 1000 ms, with the attacker node as a fast node.

Attacker Power	Attacker Connections	MPU(adversary)	MPU(tot)
25	25	0.402	0.598
25	50	0.525	0.619
25	75	0.600	0.720
50	25	0.985	0.498
50	50	0.991	0.420
50	75	0.992	0.473
75	25	0.992	0.500
75	50	0.993	0.503
75	75	0.992	0.502

When the hashing power is 25 percent, according to the results of the Eyer Paper, we should get a percentage of nodes around 30 percent of the attacker, which is lower than the estimates obtained here.

I think that this is because of the fact that the propagation delay has not been taken into account in the paper. If the propagation delay increases, then the attacker (by basis of it having a much higher connectivity than other nodes) is not affected as much as other nodes, and hence releases blocks earlier. Due to this, it is difficult to break its chain.

This is also supported by the fact that on increasing the connectivity, the ratios change by a large margin. This implies that propagation delay of blocks has a significant role to play here.

When the hashing power reaches around 50 percent, we get a ration close to 1. (Indeed in almost all the testcases with 50 and 75 percent hashing power, all the blocks in the chain belonged to the attacker except the genesis block, so

the longer we let the simulation run the large the ratio will be).

As far as the MPU(tot) ratio goes, we see that with lower hashing power, a comparatively higher percentage of nodes belong to the main chain. And the ratio seems to have a maximum cap of 50 percent.

Assume that we have an attacker with a very high hashing power. Now, every time the honest nodes generate a block, the attacker will release another block, effectively always having 2 chains with the attacker chain winning always in the end. This makes the ratio tend to 50 percent, one chain of honest nodes and one chain of dishonest nodes.

With a lower hashing power, at times the attacker will lose out in some of the chains, which will lead to a single chain at some points. This implies a lower value of the ratio MPU(tot).

2 Stubborn Mining

All simulations in this section involve 100 nodes, 50 percent slow nodes, 50 percent low CPU nodes, and mean block time 1000 ms, with the attacker node as a fast node.

Attacker Power	Attacker Connections	MPU(adversary)	MPU(tot)
25	25	0.624	0.316
25	50	0.628	0.329
25	75	0.811	0.279
50	25	0.992	0.494
50	50	0.984	0.473
50	75	0.992	0.446
75	25	0.992	0.492
75	50	0.978	0.503
75	75	0.985	0.501

Here, we see a similar trend as explained above. At times, we see apparent contradictions, such as the MPU(adversary) ratio decreasing as connections increase, while we expect the opposite trend. But this is because we ratios are so high that a single node, if it comes by chance, can cause the ratio to change at the last decimal places. This uncertainty causes the slight variation in the ratios.

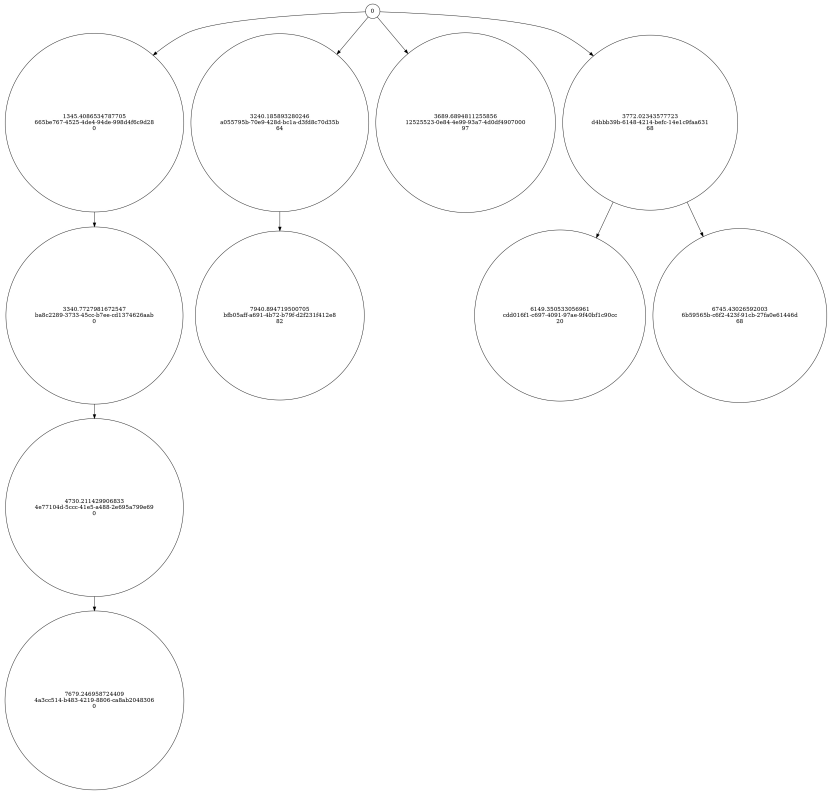
We also see that the Stubborn Mining seems to have higher ratios as compared to Selfish Mining. We chalk this up to the fact that since the forks are more likely to be won by the attacker due to its high connectivity and due to consideration of propagation delay, trying to increase the length of the chain by only releasing one block at a time is more advantageous for the attacker in the

long run.

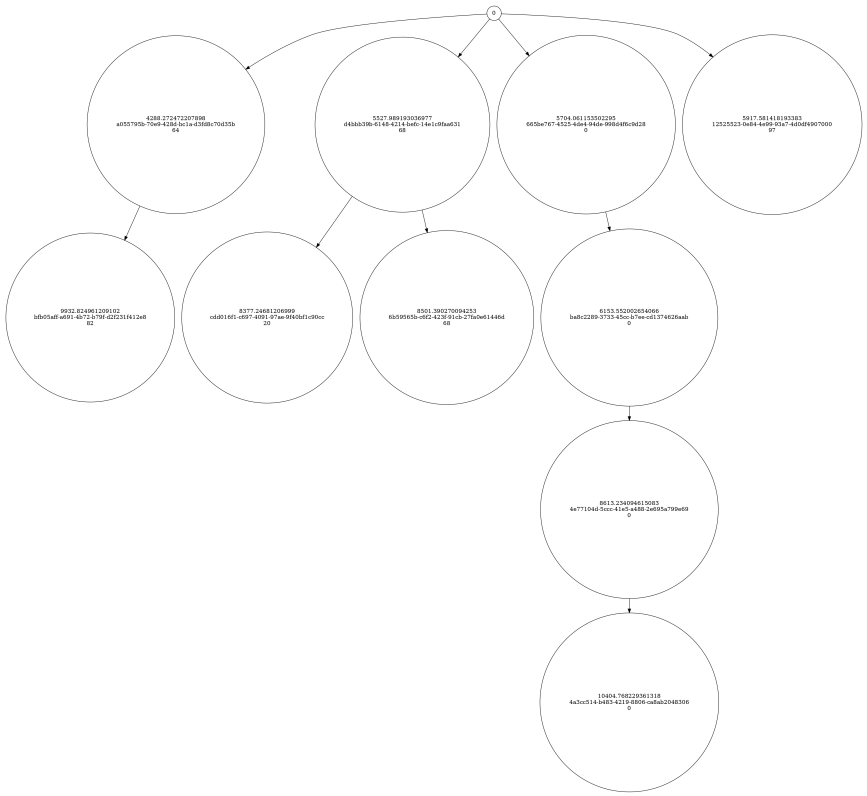
This is also what is reflected in the results we see.

3 Graphs

The following is for Selfish Mining with power 50 percent and connections 50 percent, MPU(adv) ratio around 0.8.



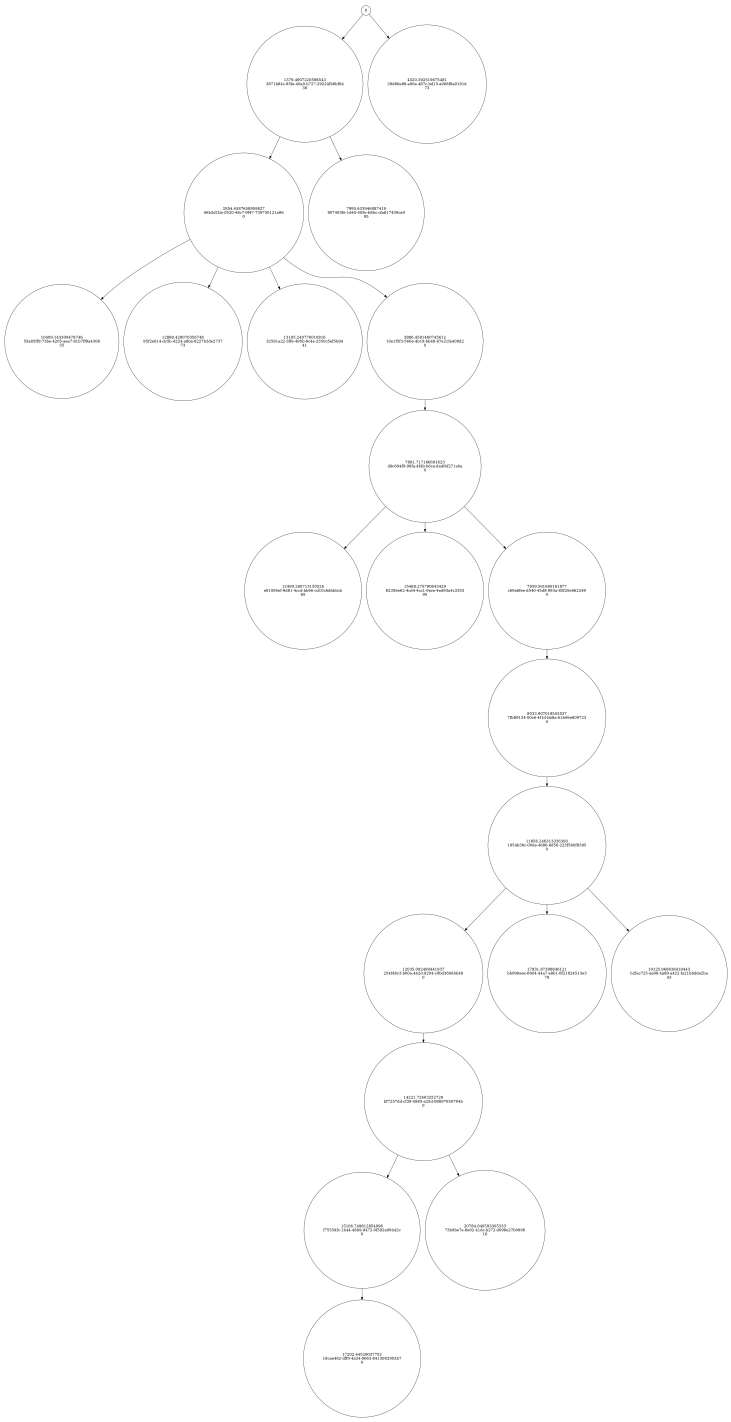
For the attacker node:-



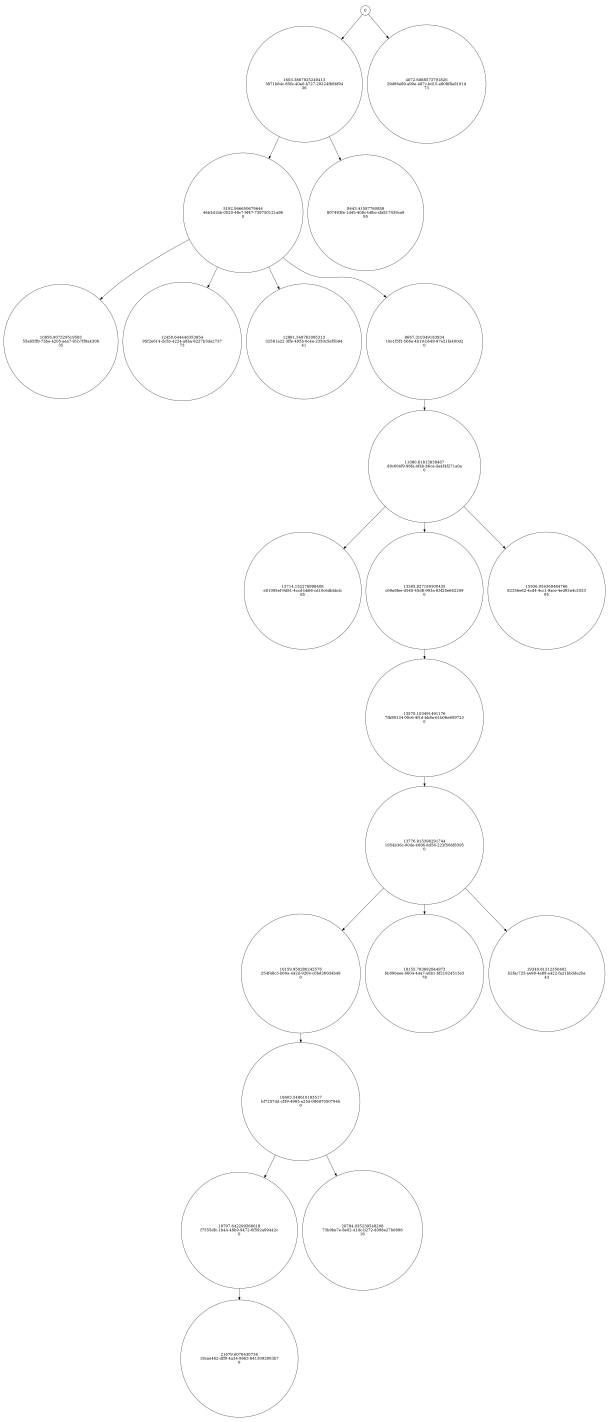
For an honest node:-

Huge graphs have an issue with displaying here, so have to make do with slightly smaller trees.

The following are trees for stubborn mining, with the same parameters as above. The obtained MPU(Adv) ratio is similar (0.818).



For the attacker node:-



For an honest node:-