REPORT-PROJECT PART 2

Maloth Maheer 180050054
Bhukya Varun Tej 180050025
Bandameedi Harshavardhan 180050020

Connections:

For the honest nodes, we used the power law distribution to form connections. The degree of the nodes, P(k) follows this distribution.

$$P(k) \sim c.k^{(-y)}$$

Here c is the normalisation constant, k is the degree of the node and y is known as the scaling parameter.

For the honest nodes, the algorithm is as follows:

- We assumed each peer to have at least “m” edges.
- For the first m+1 peers, we ensured there was full connectivity.
- For the next nodes, we are calculating probability of connecting to the existing nodes by $\Pi i = \eta i.ki / \Sigma(\eta j.kj)$
- Where ηi is the strength of a node (assume high value for fast node and low value for a slow node. Both greater than 0).
- Then we choose m nodes based on the probability we calculated.

Note: as mentioned in the question, 50% of honest nodes are slow and 50% are fast.

As far as the connections of the adversary node are concentrated , we have randomly sampled the indices of the honest nodes by choosing zeta percentage of them and made connections with the adversary node.

Observations:

We have used only one adversary node for all our analysis.

MPU_node_adv is directly linked to the strength of the attack by the adversary. High value of MPU_node_adv indicates strong attack

For the same configuration the values of MPU_node_adv for selfish attack will be greater than that of stubborn attack (honest power is > 50%). This is because in a selfish attack in the lead of 2 or 1 we completely release the private chain and we don't do that in stubborn mining. So chances of replacing the honest blocks is more in selfish mining.

Effect of n:
As n increases the total number of blocks generated will increase and so will the number of blocks in the main chain, so MPU_node_overall would be the same.
But the adversary hashing power will decrease as a fraction if we keep one adversary itself. So MPU_node_adv will decrease

Slow vs fast nodes:
Fast nodes will have slightly higher values of MPU_node_overall because that is nothing but the ratio of number of blocks in the main chain to the total number of blocks generated. Fast blocks would receive blocks quicker with the connection explained above so MPU_node_overall should be slightly more. However in practice the trend has been slightly irregular.

Effect of Tk:
If we reduce the value of Tk for the adversary this means we are increasing the hashing power of the adversary. This will increase the value of MPU_node_adv and will lead to stronger attack

Effect of zeta:
On increasing zeta MPU_node_adv increases. This is because improved connectivity helps the adversary to release his blocks in less time thus creating more blocks in the honest chain. However there have been some irregularities with the trend.