Question 3: P2P Questions and Solutions

Part 2: Open the block explorers of each of your nodes at the end of that script’s execution; do they appear synchronized? Now, repeat the experiment, but this time stop one of the nodes around halfway through the execution, and restart it at the end. What do you observe, and does this suggest any additions required to our gossip protocol (if so, what changes, otherwise, why not)?

After running nodes 2-6 I noticed that after execution of the script all nodes were synchronized. The blockchain looked the same which meets the property of a “public-ledger” system. When repeating the experiment and stopping node 2 halfway (Height 31) and restarting the node at the end the blockchain never caught up to other nodes 3-6 (Height of 68) (See Figure 1 that shows Node 2’s chain and Figure 2 that shows Node 5’s chain). This indicates that the gossip protocol is missing a way for disconnected nodes to view the current blockchain and be synchronized. One possible solution is if the blockchain was stored in a public forum and disconnected nodes can access the chain when they reconnect. The bitcoin protocol uses the inv message for a node to advertise its knowledge of one or more objects as a reply to getblocks. This can share the status of the chain if the disconnected node has its last known blockhash.

Graphical user interface, text, application

Description automatically generated

Figure 1. Node 2 After Stopping and Restarting Execution

Graphical user interface, text, application

Description automatically generated

Figure 2. Node 5 after completed script execution

Part 3: Notice that our node’s list of peers is hard coded in config.py. Does this suggest another missing component required to achieve a permissionless blockchain? Why not, or if so, what is the closest analogous message type in the Bitcoin p2p protocol documentation linked above?

Since the node’s list of peers is hard coded, we are limited when new peers join the p2p network. Currently the developers control the permissions on peers by manually adjusting the config.py. In the Bitcoin p2p protocol there is a message type ‘getaddr’ which request active peers to transmit addresses from a database of known active peers making it easier to find new active nodes and achieve a permissionless blockchain.