NON-COMPETITIVE BLOCKCHAIN SIMULATION REPORT

Program Overview

The program consists of three parts:

1. A keyset and channel created on PubNub software
2. alice.py program file
3. bob.py program file

The program is a blockchain mining simulation, however this simulation is non-competitive, meaning that miner ‘Alice’ will mine blocks 1, 3, 5, 7, and 9, while miner ‘Bob’ will be mining blocks 2, 4, 6, 8, and 10.

Once a miner has mined a block, they will broadcast it to the established PubNub channel, where the other miner will receive the block, and verify that its ‘Hash’ value matches the hash of the previous block in their own records.

If the hash value of the broadcast block is valid, they will submit the block to their own record, and begin mining the next.

The miners will be mining blocks with a requirement of 20 ‘0 bits’ at the beginning of their SHA256 hashing algorithm. The hash will be represented in hexadecimal format, so each hash must be:

Hash < “00000fffffffffffffffffffffffffffffffffffffffffffffffffffffffffff”

After the programs reach the 10th block (ledger10.json), they will automatically stop.

Program Requirements

This simulation was written in python, it will require a minimum version of python 3.3.

This simulation was coded to run on a specific channel created by Jodie Soondra on software ‘PubNub’. To run, it will require these details, and the PubNub channel must be open:

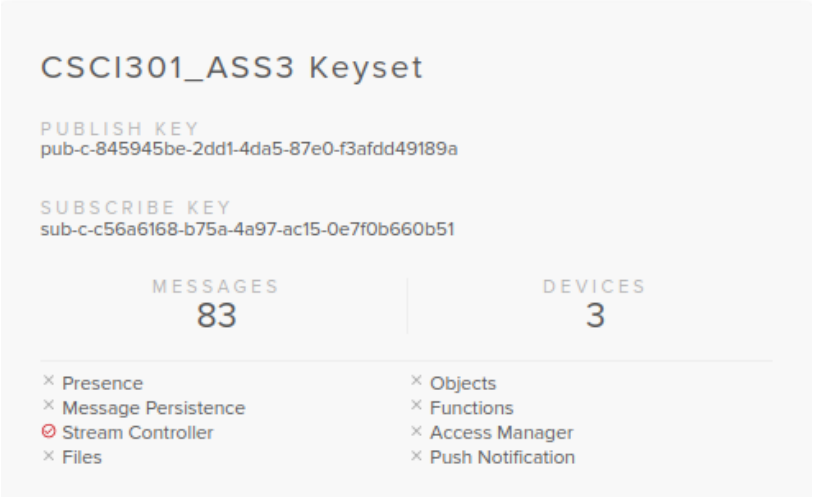
PubNub version: 6.3.1 (minimum)
Channel name: Channel-0rtj04rto
Subscriber key : 'sub-c-c56a6168-b75a-4a97-ac15-0e7f0b660b51'
Publish key : 'pub-c-845945be-2dd1-4da5-87e0-f3afdd49189a'

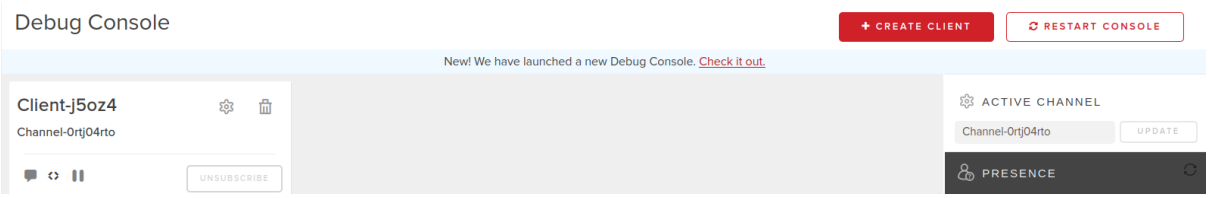
After opening the channel on PubNub, the program files ‘alice.py’ and ‘bob.py’ must be opened in separate terminals in separate folders.

It is better to run ‘bob.py’ before ‘alice.py’ so that miner ‘Bob’ can be listening on the channel, and receive the 1st block mined and then broadcast by Alice.

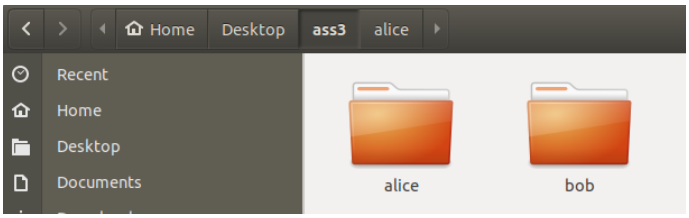
Running the Program

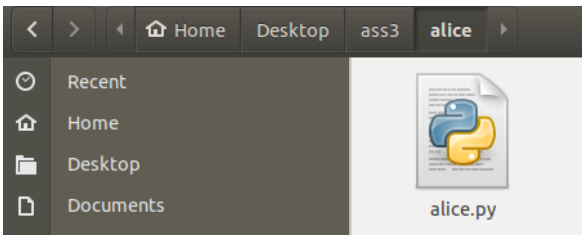
1. *First, open the PubNub channel with the specified keysets and channel name.*

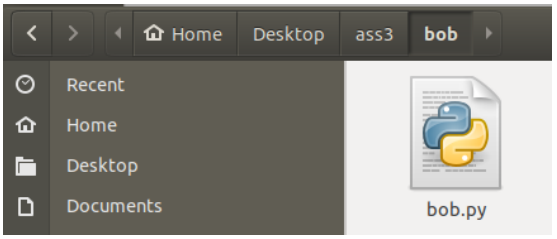




2. *Next, ensure you have alice.py and bob.py in separate folders:*





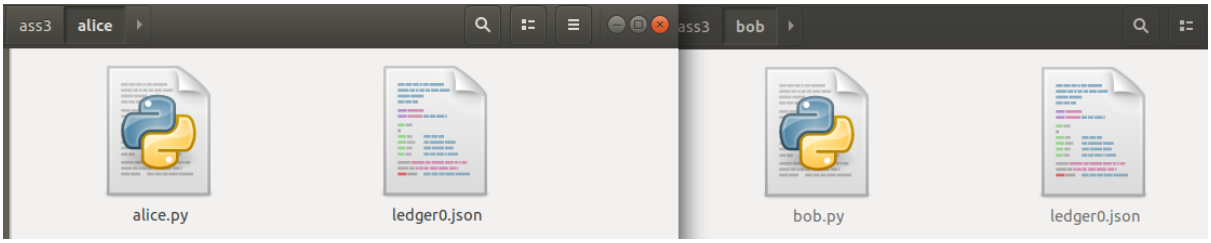


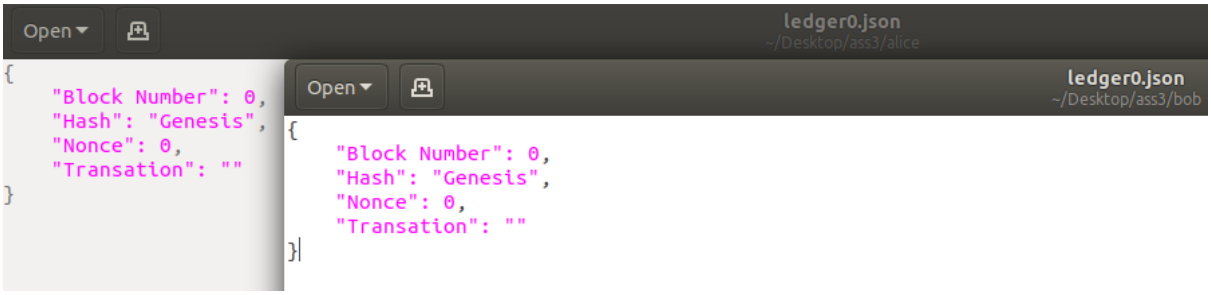
3. *Run the files in separate terminals, run bob.py first:*

```
jodie@jodie-VirtualBox:~/Desktop/ass3/bob$ python3 bob.py
Bob Connected
```

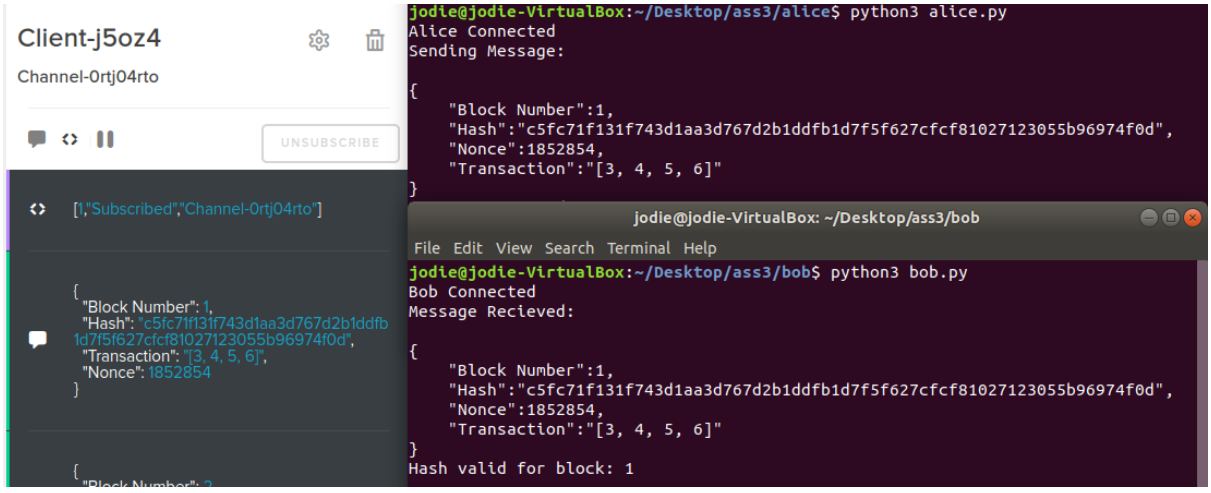
```
jodie@jodie-VirtualBox:~/Desktop/ass3/alice$ python3 alice.py
Alice Connected
```

4. *Both files will now generate their ‘ledger0.json’ block under the same parameters.*



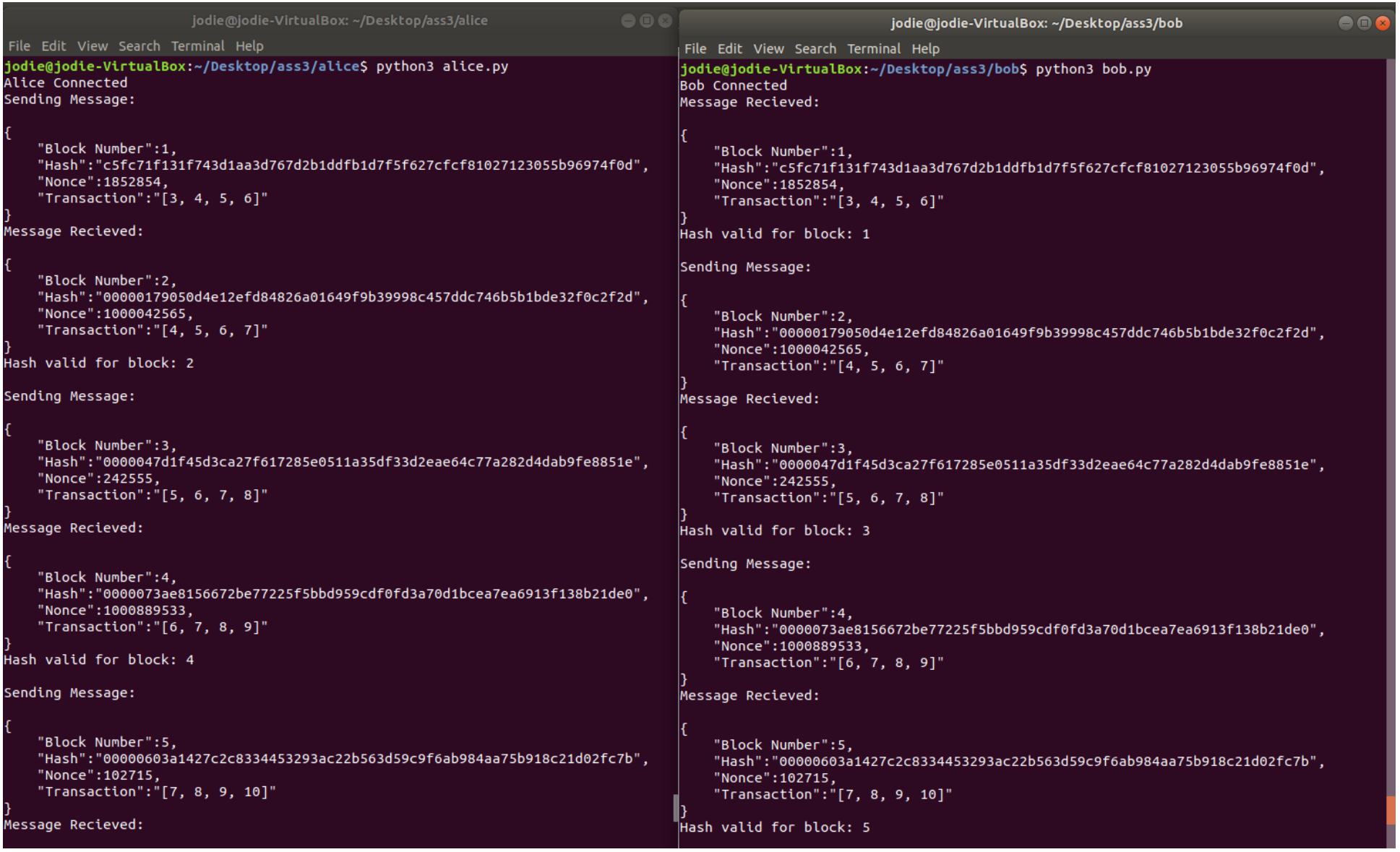


5. *Once Alice has created their genesis block, they will begin mining the first block in the chain ‘ledger1.json’. Once they have mined it, they will broadcast it:*

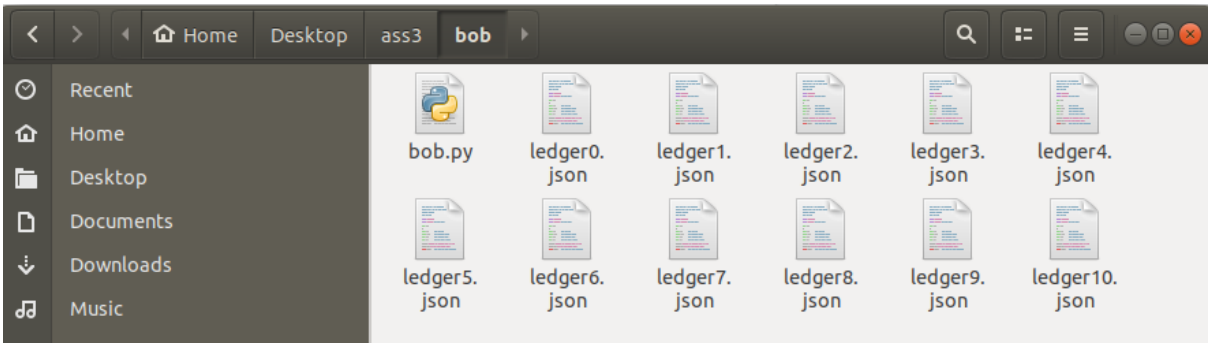


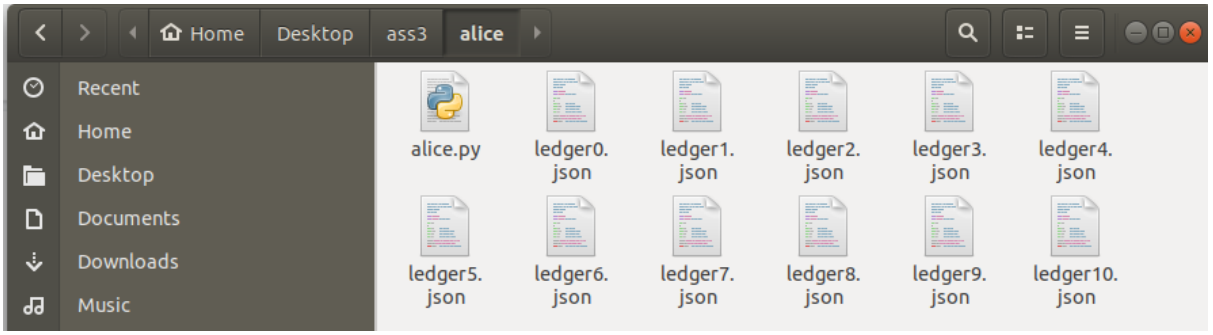
Once Bob receives the block, they will check its hash value. If it is valid, they will mine for the next block, and then broadcast it.

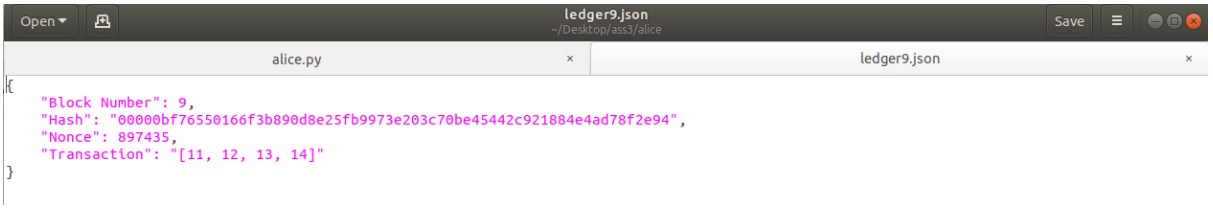
6. *This will then continue until blocks 0-10 are mined.*


jodie@jodie-VirtualBox: ~/Desktop/ass3/alice
File Edit View Search Terminal Help
jodie@jodie-VirtualBox:~/Desktop/ass3/alice$ python3 alice.py
Alice Connected
Sending Message:
{
"Block Number":1,
"Hash":"c5fc71f131f743d1aa3d767d2b1ddfb1d7f5f627cfcf81027123055b96974f0d",
"Nonce":1852854,
"Transaction":"[3, 4, 5, 6]"
}
Message Recieved:
{
"Block Number":2,
"Hash":"00000179050d4e12efd84826a01649f9b39998c457ddc746b5b1bde32f0c2f2d",
"Nonce":1000042565,
"Transaction":"[4, 5, 6, 7]"
}
Hash valid for block: 2
Sending Message:
{
"Block Number":3,
"Hash":"0000047d1f45d3ca27f617285e0511a35df33d2eae64c77a282d4dab9fe8851e",
"Nonce":242555,
"Transaction":"[5, 6, 7, 8]"
}
Message Recieved:
{
"Block Number":4,
"Hash":"0000073ae8156672be77225f5bbd959cdf0fd3a70d1bcea7ea6913f138b21de0",
"Nonce":1000889533,
"Transaction":"[6, 7, 8, 9]"
}
Hash valid for block: 4
Sending Message:
{
"Block Number":5,
"Hash":"00000603a1427c2c8334453293ac22b563d59c9f6ab984aa75b918c21d02fc7b",
"Nonce":102715,
"Transaction":"[7, 8, 9, 10]"
}
Message Recieved:
jodie@jodie-VirtualBox: ~/Desktop/ass3/bob
File Edit View Search Terminal Help
jodie@jodie-VirtualBox:~/Desktop/ass3/bob$ python3 bob.py
Bob Connected
Message Recieved:
{
"Block Number":1,
"Hash":"c5fc71f131f743d1aa3d767d2b1ddfb1d7f5f627cfcf81027123055b96974f0d",
"Nonce":1852854,
"Transaction":"[3, 4, 5, 6]"
}
Hash valid for block: 1
Sending Message:
{
"Block Number":2,
"Hash":"00000179050d4e12efd84826a01649f9b39998c457ddc746b5b1bde32f0c2f2d",
"Nonce":1000042565,
"Transaction":"[4, 5, 6, 7]"
}
Message Recieved:
{
"Block Number":3,
"Hash":"0000047d1f45d3ca27f617285e0511a35df33d2eae64c77a282d4dab9fe8851e",
"Nonce":242555,
"Transaction":"[5, 6, 7, 8]"
}
Hash valid for block: 3
Sending Message:
{
"Block Number":4,
"Hash":"0000073ae8156672be77225f5bbd959cdf0fd3a70d1bcea7ea6913f138b21de0",
"Nonce":1000889533,
"Transaction":"[6, 7, 8, 9]"
}
Message Recieved:
{
"Block Number":5,
"Hash":"00000603a1427c2c8334453293ac22b563d59c9f6ab984aa75b918c21d02fc7b",
"Nonce":102715,
"Transaction":"[7, 8, 9, 10]"
}
Hash valid for block: 5

The folders for alice and bob should now be populated with blocks:







NOTE:
I did not calculate a ‘Nonce’ value for the *ledger0.json* block, as I am unsure if a genesis block needs to be “mined”, so to speak.