CS 4121, Fall 2019 Dr. Zhiguang Xu

GENI Lab 3

Due: 3:30pm Sept 25, 2019

0. Introduction

0.1. Overview

See Pre-lab3 for details.

0.2. Background

See Pre-lab3 for details.

1. Lab Configurations

See Pre-lab3 for details.

2. Lab Details, Part I: Set up Bitcoin network

Run the following on **each** node:

```
mkdir ~/.bitcoin
echo "rpcuser=test\nrpcpassword=test\n" > ~/.bitcoin/bitcoin.conf
```

Then run

```
bitcoind -regtest -daemon -deprecatedrpc=generate
```

> **IMPORTANT:** Notice that the `-deprecatedrpc=generate` option in the command above MUST be there or the rest of the lab won't work. However, it is not in the video "Lab3_section_2" that I recorded last year. If you accidentally typed the command as in the video, you will have to stop the `bitcoind` daemon by running the command `killall -9 bitcoind` and restart it by running the command with the `-deprecatedrpc=generate` option.

to run the bitcoin daemon in "*regtest*" mode on **each** node. This mode lets us create a private blockchain to experiment with, and create new blocks on demand (otherwise it would be impossibly difficult for us to mine blocks on our ordinary CPUs. We don't want spend tons of money purchasing a bunch of GPUs, do we?)

On **each** node, add all the other nodes as peers:

```
bitcoin-cli -regtest addnode node-0 onetry
bitcoin-cli -regtest addnode node-1 onetry
bitcoin-cli -regtest addnode node-2 onetry
bitcoin-cli -regtest addnode node-3 onetry
bitcoin-cli -regtest addnode node-4 onetry
```

Verify that **each** node sees all the others with

```
bitcoin-cli -regtest getpeerinfo
```

3. Lab Details, Part II: Generate blocks

WARNING: Be extremely careful with the commands below and the nodes on which they are executed. If you mess them up, the block chain generated will be diffeerent from the correct one and you will have to redo the commands in Section 2 above all over again.

Now that all our peers are connected, we are going to generate blocks, and watch them get propagated through the network.

Generate a block on **node-0** with

```
bitcoin-cli -regtest generate 1
```

and check that **all the other nodes** become aware of it with

```
bitcoin-cli -regtest getblockcount
```

all nodes should see a blockcount of 1. That means, the length of the blockchain on each node is 1 now.

Check whether **node-0** that generated the block has earned a reward by running

```
bitcoin-cli -regtest getbalance
```

It should still have a balance of 0, because the block it generated has not yet been confirmed by 100 additional blocks.

Let's now add more blocks to the blockchain; run

```
bitcoin-cli -regtest generate 20
```

on **each of the five node**. Verify that the blockchain length seen on **all** nodes is 101 with

```
bitcoin-cli -regtest getblockcount
```

Also, if you watch the end of the bitcoin log file on **any** of the nodes,

```
tail --lines=100 ~/.bitcoin/regtest/debug.log
```

you should see "update tip" messages indicating that new blocks have been added to the end of the blockchain.

Now that there are 101 blocks in the blockchain, if you run

```
bitcoin-cli -regtest getbalance
```

on **node-0** -- the node that generated the first block - you should see that it has earned 50 bitcoins for its troubles. The other nodes still have a balance of zero, until the blocks they contributed are, in turn, confirmed.

If you run

```
bitcoin-cli -regtest listunspent
```

on **node-0**, you should see further details about its unspent bitcoins:

```
[
  {
    "txid": "fa6d540d4b18655f19e71ae3c3579161c135a24c3888ffd6ae4a3d63ab4fcc56",
    "vout": 0,
    "address": "n2TasUFFLx872onVxRiaWXWenAF3TrJjof",
    "scriptPubKey": "210307b00d28af8f94d1e719341013e36145dccd1caea622ac0d5952ae90fd15171cac",
    "amount": 50.00000000,
    "confirmations": 101,
    "spendable": true,
    "solvable": true,
    "safe": true
  }
]
```

(ACTION) Take a screenshot of the node-0 ssh terminal and save it in a single image file: "node_0_yourinitial.jpeg".

> Make sure your screenshot includes all the commands (and their outputs) that you ran on node-0 above, **or no point will be assigned**.
>
> If you can't capture all commands in one screenshot, take multiple screenshots of node-0 and name them "node_0_yourinitial_part_1.jpeg", "node_0_yourinitial_part_2.jpeg", ..., etc.
>
> ```
>   bitcoin-cli -regtest generate 1
>   bitcoin-cli -regtest getblockcount
>   bitcoin-cli -regtest getbalance
>   bitcoin-cli -regtest generate 20
>   bitcoin-cli -regtest getblockcount
>   bitcoin-cli -regtest getbalance
>   bitcoin-cli -regtest listunspent
> ```

4. Lab Details, Part III: Fork the network

WARNING: Be extremely careful with the commands below.

Now we are going to split our Bitcoin network, allow unconnected parts of the network to generate blocks, and see what happens when they reconnect.

On **node-3**, use

```
ifconfig
```

and look for the interface that has IP address "10.10.3.2 ". This is the interface between node-3 and node-2, i.e. the interface that connects nodes 3 and 4 to nodes 0, 1, and 2. Find the name of the interface (e.g. eth1, eth2) and bring it down with

```
sudo ifconfig eth1 down     <== Substitute eth1 with your actual interface name !!!
```

> Mine interface is eth1. Yours might be different. Make sure you use yours.

Generate some blocks on **node-4**:

```
bitcoin-cli -regtest generate 20
```

Make a note of the output, e.g.

```
[
  "6d38fd107717b7d831cbb1abec80318aa99876fb5db1eaba9605e427abd99d0a",
  "5a3517a77aba647716f09ae8cb55c375ae23b5ea2b26592fd79e18da45ba4109",
  "5621aaacec39cf12753add7344ec089c9ece585866d803795cbd946a117ba243",
  "678621830f7e801d86af35282eb1fa15f33b2ed3511a49574e4c4971ac454094",
  "70bbaf43709a31a7ed094f60c65c767f84a485be86dd81d44b6c8b98ba088c80",
  "054347b36d3c3f122a4b7093a3541846cfc9817451a3c046aefb816bacd8323b",
  "4aede78ca5cf3ab28eab1efb14b774ba30f3e3a860f411a0a221fd49c3c241ec",
  "6ad4a3c1f182c0b1ed68ede62f48c5afeab056e01b6f4abac3821b01636d1e95",
  "7c6e34cf879fa8bdc509a7526a93655e9547abb3532d54f31b11c6e3920e6000",
  "64561623be647e239979516cb68029f7e940b6dcd872ad6fdf65586386da6ca6",
  "1648df6822896b18d571887f0af367340123908f1858c85ddcf22914cd6da461",
  "4366acd56aaf981f1df545905c8def7664784759066c250aa4878eb43b02f6f0",
  "54bc4b71f3fc37ff57823bfd8a3d1d8068a2e484405eb4b59e64263d467db80c",
  "3e746dfa9e7f5c700f209445214bb6df689372c8646282813c4b47e86a03ced3",
  "43559c2e01ee082a0c209e283e7b4e26fcb0fb4a06c8452d1204828c54d459e5",
  "6c2d2ffcf1b1fba4f401997ad2c4c20c0a83380e53c2f5bed0d000cf45c4643c",
  "5d062946b0c5134ab98eaa2a6d92636612aa3b03d87a09192d0a9283449357d6",
  "2a7c819abe0cbdd1b52a9a5700fd2b66bb15f01d4d62613ccbb37d9744d1a2a0",
  "53ccb611d75f4839766dbe68bb29aa11e8f0a4c51ea02e6e420a7240f5569aa3",
  "13b2772b008e5d7b5970409c6622722dadb88e4d71e8c05f1b1f1f4cabda83c8"
]
```

Use

```
bitcoin-cli -regtest getblockcount
```

on **both** sides of the network. On **node-3**, you should see a blockchain length of 121, and if you

```
tail -f .bitcoin/regtest/debug.log
```

you can see that it got exactly the blocks generated by node-4. On the other hand, **node-0**, node-1, and node-2 will still have a blockchain length of 101.

Now generate 20 blocks on the other side of the network. On **node-0**, run

```
bitcoin-cli -regtest generate 20
```

and make a note of the output, e.g.

```
[
  "7c94263c113d56bfc0e389a88cc7cbb1be5e29cdd702e7a8dce0f7b7853c72bc",
  "484330624031a90584ebd9ca809ba4c52fa03a82ffba31952a871187c668fc74",
  "242ece46c5eebde71d27b02d858093a7faf91f4c7c7f9f582665097e282da3a0",
  "54af4d371562fa61a18d3e684bd73f7790b8a78df0696eb09343b206b3db104c",
  "4cbf65ba85ff3c4626f525fba87c3590f5b94db8c07ccb19446813640528587a",
  "4e4b8c6694a400c76244864e03df6e80dc82888e1c53cd8f754f18bdc2e66c73",
  "3e762a6c3ba071e1fda895c66888a5962bad50b551bce388b54bfeef8f2d8c2b",
  "052a211945bdb199fbf7909d5e29cf9765d7d32a7d12e2ed3df8b3edc8fbc49c",
  "7c99066a3e6de06017babe2edbfb84a0e86dc1ab4d3e65102bc859570a1e5fd8",
  "0a0f1a1b4b1578e4d79ed2873657769f50ab9a4fc7e8cdc5f23df5549e71da58",
  "0b145998e3772f427cfd98467e7c4057fc6f45e747e9bfc99259ef87d7d72de1",
  "74770c5f70a17427e6f57f4bf32d5a97b5667a3d23d714d98d6ef3405ff0c35b",
  "639e6caf5bc5900edf5f7723e06fd4bbb5b0cf5f294c12679e288cbc6563966e",
  "426665108f89b362b84d0f2be80898455f3637af6120ad5ff03f7528b7d5d24f",
  "438a437e192d3eaed019a8aaadb51e0d61cb4c79c21370c31a458ac20e41b5b7",
  "687c1241638e0dcb09a5741619930bd98c92d1a3da89b46b4d8b89c080a04bfc",
  "742f0252fc0f8ffc9d18321460b42eb940c6bc45d44e7e70cd6d74ab6cf3abc5",
  "2b19bc407300fb2711131cb4e2dda520340c19bca42c682be503b9b6ff6dc3f9",
  "27bb7a14b4c599ca3ebc23036a737cfdc325505006b0109d6246833963caf7de",
  "391c411ca00b5ca341bbe28aa9930fcac272f66235e8b2b8ee59f4c08f6a5170"
]
```

We now have a blockchain with 101 blocks in common between both sides of the network, followed by 20 different blocks. Let's reconnect the network.

On **node-3**, run

```
sudo ifconfig eth1 up     <== Substitute eth1 with your actual interface name !!!
```

> Again, my interface is eth1. Yours might be different. Make sure you use yours.

All nodes in the network should still respond with "121" to

```
bitcoin-cli -regtest getblockcount
```

however, the specific blocks in their blockchains will vary. Compare the 121st block in the blockchain on both sides by running the following command on **node-0** and **node-3**:

```
bitcoin-cli -regtest getblockhash 121
```

See the difference?

Let's reconcile the conflict. On **node-0**, run

```
bitcoin-cli -regtest generate 1
```

Now repeat

```
bitcoin-cli -regtest getblockcount
bitcoin-cli -regtest getblockhash 121
bitcoin-cli -regtest getblockhash 122
```

on **all** the nodes to see how the network has "agreed" on a specific blockchain, with identical blocks, and abandoned what has become the shorter blockchain.

(ACTION) Take a screenshot of the node-0 ssh terminal and save it in a single image file: "node_0_fork_yourinitial.jpeg".

> Make sure your screenshot includes all the commands that you ran on node-0 above, **or no point will be assigned**.
>
> If you can't capture all commands in one screenshot, take multiple screenshots of node-0 and name them "node_0_fork_yourinitial_part_1.jpeg", "node_0_fork_yourinitial_part_2.jpeg", ..., etc.
>
> ```
>   bitcoin-cli -regtest getblockcount
>   bitcoin-cli -regtest generate 20
>   bitcoin-cli -regtest getblockhash 121
>   bitcoin-cli -regtest generate 1
>   bitcoin-cli -regtest getblockcount
>   bitcoin-cli -regtest getblockhash 121
>   bitcoin-cli -regtest getblockhash 122
> ```

(ACTION) Take a screenshot of the node-3 ssh terminal and save it in a single image file: "node_3_yourinitial.jpeg".

> Make sure your screenshot includes all the commands that you ran on node-3 above, **or no point will be assigned**.
>
> If you can't capture all commands in one screenshot, take multiple screenshots of node-3 and name them "node_3_yourinitial_part_1.jpeg", "node_3_yourinitial_part_2.jpeg", ..., etc.

```
ifconfig
sudo ifconfig eth1 down
bitcoin-cli -regtest getblockcount
tail -f .bitcoin/regtest/debug.log
sudo ifconfig eth1 up
bitcoin-cli -regtest getblockcount
bitcoin-cli -regtest getblockhash 121
bitcoin-cli -regtest getblockcount
bitcoin-cli -regtest getblockhash 121
bitcoin-cli -regtest getblockhash 122
```

5. What to Turn in?

Submit the following files:

- node_0_yourinitial.jpeg
- node_0_fork_yourinitial.jpeg
- node_3_yourinitial.jpeg

For each file above, in case you need to take multiple screenshots and submit multiple jpeg files, you must follow the naming conventions that I specified above or **no point will be assigned**.