(Pre) GENI Lab 3

Due: There is nothing to submit. However, make sure you watch my video on BlazeVIEW, read Section 0, and complete Section 1 before working on the actual Lab.

> Please find the actual GENI Lab 3 in a separate document.

0. Introduction

0.1. Overview

This experiment explores the problem of consensus in distributed systems in the context of **Bitcoin**, a distributed currency.

0.2. Background

This experiment explores the problem of reaching consensus in a distributed system. "Consensus" is the problem of getting members of a network to agree on something, e.g. a value. In some systems, there is a centralized control unit who can decide on the value and then broadcast it to the rest of a network. In a distributed consensus system, members of the group have to collectively reach consensus without the benefit of a centralized unit. Further complicating the problem, some members of the group may be lying or otherwise manipulating the group to try and reach a consensus that favors them over the "true" value.

There are various solutions to the problem of distributed consensus. In this experiment, we examine the approach used by Bitcoin, a distributed currency with no central bank or other authority. In order for a distributed currency to work, members of the network must agree on how many units of the currency each member holds at all times, in order to prevent members from **double spending**, i.e. re-using the same units of currency in multiple transactions. This is, of course, a distributed consensus problem.

Bitcoin uses a process called **mining** to reach a consensus. Members of the network who choose to take part in the process of reaching a distributed consensus are called **miners**. Mining involves forming a **block** containing a series of transaction records, then finding a valid **proof of work** for that block that satisfies certain rules. Specifically, miners increment a **nonce** until they find a value that gives the block's hash a certain

number of leading zeros, thereby "finding" the next block in the **blockchain**.

> Proof of Work – What it Is and How Does it Work?
>
> SHA256 online

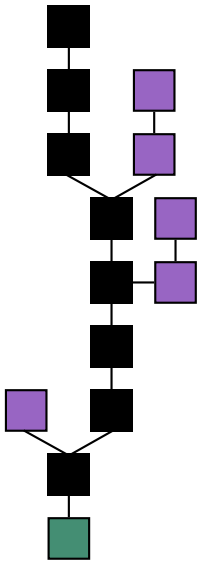
Once a block has been "found", it is broadcast to all other nodes in the network, who validate the transactions it holds and accept the block if it is valid.

Overall, the process is:

1. New transactions are broadcast to all nodes in the network.
2. Miner nodes collect the new transactions into a block.
3. Miner nodes try to find a proof of work for the block. **This is an extremely expensive procedure, computationally speaking.**
4. When a miner node finds a proof of work, it adds the new block to its blockchain and broadcasts the new block to all nodes in the network.
5. Receiving nodes validate the new block and then start work on the next block, containing transactions that have not yet been placed in a block.
6. The node that found the block is rewarded with some new Bitcoins, and also the value of the transactions fees for the transactions in the block.

Miners choose which transactions to include in the block they are mining based on priority algorithms that include metrics such as the age of the transaction and the value of its transaction fees. So it is possible for two nodes to arrive at different "next blocks," potentially at the same time, and broadcast different blocks to the rest of the network. If this occurs, the blockchain is said to be **forked**, and the network needs some way to reach a consensus on which block to accept as the next block.

Bitcoin clients operate on a simple rule: always trust the **longest** chain of blocks. If a node received two "next blocks", it will save both and start to work on the next block in one of the chains. If in the meantime it receives a new block for one of the chains, it will discard the shorter chain and start to work on a next block for the longer chain. (Miners have an incentive to work on the longest chain, rather than continuing to work on the chain following from the orphaned blocks, because they want to earn the Bitcoins associated with finding the next block in the longest chain.)



In the image above, the main chain (black) consists of the longest series of blocks from the genesis block (green) to the current block. Orphan blocks (purple) exist outside of the main chain.

A key property of these blocks is that in order to replace a block that is already in the blockchain with one of its own, a node would have to compute the proof of work for that block and all subsequent blocks that have been added to the blockchain after it, since in order to get Bitcoin clients to accept your version of the blockchain, you need to have the longest chain.

Finding proof of work is computationally difficult, so the more blocks have been added to the blockchain after the block containing a particular transaction, the more secure it is and the less likely it is for the block to be replaced (rolling back the transaction). Most Bitcoin clients consider a Bitcoin transaction to be **confirmed** after six more blocks have been added to the chain, and the Bitcoins awarded for finding a new block are considered to be confirmed after 100 blocks.

1. Lab Configurations

Create a new slice (please name the new slice "**lab3-your-initial**") and request resources by loading the RSpec from the following URL: https://git.io/vhcWT.

Be sure you have one ssh terminal to each of the five nodes. To make your life easier, please arrange your terminals such that they match the corresponding nodes in the network topology on the GENI portal.

> Since the RSpec file above contains instructions of installing the Bitcoin software on each node, before logging in, you might have to wait a few minutes after the nodes become green.

TO BE CONTINUED....