Homework assignment # 4: The Bitcoin Network

Deadline: February 2, 2023

1. Administration

Each homework in this course will contribute 20 % to the final grade. There will be 4 homework assignments and one final project. If needed, we might have a bonus assignment, which can replace your worst homework, but not the final project.

Your solution should be sent to the following email as a text file:

- domagojvrgoc@gmail.com

There is no restriction or penalty for using materials you found online. It is a good practice to mention those if you use them. This will not result in any penalty to your homework.

2. The homework

In this homework assignment, we will simulate what a full Bitcoin node does upon entering the network: downloading full blocks (and not just the headers).

Using the code explained in the lectures (`block.py` and `network.py`), you should add the support for the messages that will allow us to download the first 20 blocks from the Bitcoin testnet (the genesis block is already defined in our code). Here you need to download the entire blocks, and not just the headers (that is, you you need to download all the transactions as well).

For this, you will need to do two things. First, you need to define the message that allows to obtain an entire block from a full Bitcoin node. As a reply to this message, the node will send you a serialization of the block. To manipulate this object, you need to implement the class `FullBlock` in `block.py`. The code you received with this homework already contains the skeleton of this class, together with all the methods necessary for its correct operation. To define the method `parse` in the class `FullBlock`, you need to parse *all* the transactions in the block. To parse a (single) transaction, you can use the code available in `txP2PKH.py`.

All the details of the serialization and the logic of the messages that are sent on the Bitcoin network can be found at `https://en.bitcoin.it/wiki/Protocol_documentation`. If you do not understand how to define a message, or how to send to a node, you should review the code we provided when explaining network communication (Week 11/12).

Once yo implement these methods, you should do the following:

1. [**15 points**] Using your implementation, you should download the first 20 blocks from the Bitcoin testnet, and print their hashes, the MerkleRoot, and (a hash of) all of their transactions.

2. [**5 points**] Once you have the blocks, try to validate the transactions in each block using the class `Tx` defined in `txP2PKH.py`. The execution will result in an error. Explain why this error occurs. It is not necessary to correct the error, you just need to explain why the execution fails.

Submissions: For this assignment, you need to hand in the code you used to solve the assignment, together with: (i) an explanation of how to use the code to obtain the blocks; and (ii) an explanation of why an error occurs when you try to validate the transactions in each block.

Bonus tasks In this bonus you will do the same as in the homework itself, but now connecting to several different nodes at the same time. There are two parts of the bonus:

1. [**10 points**] Using the implementation available in `network.py`, yo ushould connect to **three** different nodes, and download the first 20 blocks from the **mainnet** of Bitcoin. *Again, not from testnet, but from the mainnet!* Following this, verify that the blocks received from the three nodes coincide.

2. [**5 points**] Obtain an address of at least one node using the `getaddr` message. The reply you will receive is a message of the type `addr`, defined in `https://en.bitcoin.it/wiki/Protocol_documentation`, so you will need to deserialize (parse) this message.

For the bonus you should hand in the code, and explain how to use it in order to obtain the desired information.