Summer 2020

ECE8930 Introduction to Blockchain Technology
Programming Project 2

Lu Yu
luy@g.clemson.edu
July 15th, 2020

You can use C/C++, Python or Java for this project. You must include a detailed instruction on how to compile/execute your code in the submission.

For this project, you will implement a "modified" PoS-based voting algorithm. The original Casper FFG consists of two layers:

1. The bottom layer is the tree of transactions.
2. The top layer is the sub-tree of checkpoints extracted from the tree of transactions.

For the sake of simplicity, the following assumptions are made:

1. There are **10 validators** in the system. The set of the validators are fixed. The deposits of the validators are given in the table below:

Validator ID	0	1	2	3	4	5	6	7	8	9
Deposit Amount	500	100	300	250	150	500	600	350	200	150

2. The **checkpoint tree** is already given.
3. The checkpoint tree is a **full binary tree**.
4. Each node/checkpoint in the checkpoint tree has a number associated with it. Given a node of number n, its left child is $2n + 1$, and its right child is $2n + 2$. The **root node** is labeled **0**.
5. The length of the **link** contained in each **vote** is always 1. This means a validator **cannot** skip any checkpoint in its vote, i.e., the target of the link is either the left child or the right child of the source. (Note: in the real Casper FFG, a vote can skip some checkpoints. An example is given on page 27 of the slides, where link $b_2 \rightarrow b_3$ skips two checkpoints.)
6. For each vote, the probability of selecting the left child as the target is **the same** as the probability of selecting the right child.
7. If the sum of the deposits of the nodes voting for a link *L* exceeds **1/2** of the total deposit, *L* is the **supermajority link**. (Note: this is different from the 2/3 rule used in real Casper FFG.)

In your program, starting from the genesis node, you will run the Proof-of-Stake (PoS)-based voting **10 rounds**, which yields 10 supermajority links.

Output:

- The blockchain formed by the 10 supermajority links, which is a list of the node/checkpoint numbers; and
- The IDs of the validators who **finalized** each checkpoint.

Notes: Your submission is a zip file, which contains your code and the report. Things that MUST be contained in your report include three example outputs, and the detailed instruction on how to run your code.