COMS 4995-001 (Science of Blockchains): Homework #2

Due by 11:59 PM on Wednesday, February 12th, 2025

Instructions:

(1) Solutions are to be completed and submitted in pairs.

(2) We are using Gradescope for homework submissions. See the course home page for instructions, the late day policy, and the School of Engineering honor code.

(3) Please type your solutions if possible and we encourage you to use the LaTeX template provided on the Courseworks page.

(4) Write convincingly but not excessively. (We reserve the right to deduct points for egregiously bad or excessive writing.)

(5) Except where otherwise noted, you may refer to your lecture notes and the specific supplementary readings listed on the course Web page *only*.

(6) You are not permitted to look up solutions to these problems on the Web. You should cite any outside sources that you used. All words should be your own. Submissions that violate these guidelines will (at best) be given zero credit, and may be treated as honor code violations.

(7) You can discuss the problems verbally at a high level with other pairs. And of course, you are encouraged to contact the course staff (via the discussion forum or office hours) for additional help.

(8) If you discuss solution approaches with anyone outside of your pair, you must list their names on the front page of your write-up.

Throughout this homework assignment, unless otherwise specified, we consider a network of n validators, denoted by $\{v_1, \ldots, v_n\}$.

Problem 1

(10 points) This question concerns Protocol C from lecture 4. In that lecture, we saw that protocol C satisfies consistency and liveness in the presence of f crash faults for any $f < \frac{n}{2}$, in a partially synchronous network. In this question, we consider the case in which $\geq \frac{n}{2}$ crash faults may occur. Specifically, does protocol C remain consistent under the threat of $\geq \frac{n}{2}$ crash faults? Provide either a proof of consistency or a valid execution exhibiting a consistency violation as your result. For the purposes of this question, you may use properties of protocol C stated in the lecture slides without reproving them (but make sure to refer to the slide with the employed statement).

Problem 2

(10 points) This question concerns protocol C from lecture 4. Suppose we changed a single line of the code, as follows: At timestep $3\Delta \cdot v + \Delta$, instead of "let A = most recently proposed of these (i.e., with max view number)," we use "let A = longest of these (i.e., with the largest number of blocks, breaking ties arbitrarily)." Provide an explicit example of an execution in which consistency is violated for this modified version of protocol C.

Problem 3

(20 points) This problem is a reading response problem. Specifically, you should read up on practical implementations of Raft. Pointers to many such implementations can be found in the "production use of Raft" section of the Raft Wikipedia page. Then, answer the following questions. [Target length: 2-3 paragraphs for each part. The more specific you can be, the better.]

1. Some implementations of Raft adopt the *stable leader* approach, in which the leader stays the same between views until it misbehaves in some way. Describe what are some of the potential protocol changes required to adopt such an approach. What are the potential advantages/disadvantages of the stable leader approach (compared to the round-robin approach we saw in lecture)?

2. An additional variant employed by some practical implementations of Raft is that of *read only validators*. Keeping protocol C in mind, we say that a validator is *read only* in the case that it contributes to read quorums but not to write quorums. Concretely, in terms of protocol C, the *only* change in the code for a read only validator i is that it never executes the line "i sends 'ack A^*' message to all other validators." In particular, a read only validator does update its local chain (A_i) whenever it promptly receives a proposal A^* from the leader.

   Suppose now that k of the n validators are read-only (where $0 \leq k \leq n$). How does the value of k affect the consistency and liveness properties of protocol C? What are some of the potential advantages/disadvantages of employing one or more read only validators in the protocol?