COMS 6998-006 (Foundations of Blockchains): Homework #3

Due by 11:59 PM on Thursday, October 14, 2021

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.

(9) Some of these problems are difficult, so your group may not solve them all to completion. In this case, you can write up what you've got (subject to (4), above): partial proofs, lemmas, high-level ideas, counterexamples, and so on.

Problem 1

(25 points) Recall the Tendermint protocol for state machine replication from Lectures 6 and 7.

(a) (10 points) Prove that if $f \geq n/3$, the protocol no longer satisfies consistency. That is, give a specific strategy for the adversary (both the behavior of the Byzantine nodes and the message delivery choices) that leads to two honest nodes writing different blocks (at the same height) to their local histories.

[If it's helpful for this or the next part, you can assume that the adversary also gets to pick the fixed round-robin order in which the protocol rotates leaders.]

(b) (15 points) In our liveness proof for the Tendermint protocol, we showed that, provided $f < n/3$, a transaction known to all honest nodes will eventually appear in all honest nodes' local histories. Show that the conclusion no longer holds if we assume only that a transaction is known to a single honest node. That is, give a specific strategy for the adversary (with $f < n/3$) that censors the transactions known to a given honest node indefinitely.

Problem 2

(15 points) Recall the permissioned version of longest-chain consensus from Lecture 8. Recall that a sequence of leaders is w-balanced if every window of at least w consecutive leaders contains a strict majority of honest leaders.

(a) (5 points) Suppose leaders are chosen round-robin, using an arbitrary order through the n nodes. Assume that at most f nodes are Byzantine. What is the smallest value of w that guarantees that this round-robin leader sequence is w-balanced? (In this and the next two parts, prove both an upper bound and a matching lower bound.)

(b) (5 points) The previous part effectively studied worst-case robin-robin orderings. This part and the next consider best-case orderings. Suppose $f < n/2$. What's the smallest value of w for which there exists a round-robin order that generates a w-balanced leader sequence?

(c) (5 points) Repeat part (b) under the stronger assumption that $f < n/3$.

Problem 3

(25 points) This problem considers the balancedness of random sequences of leaders. Consider such a sequence in which each leader is chosen independently, and is Byzantine with probability α and honest with probability $1 - \alpha$. (This is equivalent to the case in which $f = \alpha n$ and each leader is chosen uniformly at random.) Assume that $\alpha < \frac{1}{2}$.

(a) (12 points) Let $X_1, X_2, \ldots, X_n$ denote n i.i.d. Bernouilli (i.e., 0-1) random variables, with $\mathbf{E}[X_i] = \mathbf{Pr}[X_i = 1] = p$. Let μ denote np. By the linearity of expectation, $\mathbf{E}[\sum_{i=1}^n X_i] = \sum_{i=1}^n \mathbf{E}[X_i] = \mu$. The Chernoff bounds state that the random variable $\sum_{i=1}^n X_i$ is almost always close to its expectation. One of these bounds (which you should assume without proof) asserts that, for $\gamma \in (0, 1)$:

$$\mathbf{Pr}\left[\sum_{i=1}^{n} X_i < (1-\gamma)\mu\right] \leq e^{-\frac{\gamma^2\mu}{2}}. \tag{1}$$

Fix a failure probability δ. Prove that there exists a constant c_1 such that, with probability at least $1 - \delta$, a random leader sequence of length at least $c_1 \ln \frac{1}{\delta}$ has a strict majority of honest leaders. (The constant c_1 can depend on the fraction α of Byzantine nodes.)

(b) (8 points) Prove that there exists a constant c_2 such that, with probability at least $1 - \delta$, a length-T leader sequence is $c_2(\ln \frac{1}{\delta} + \ln T)$-balanced. (Again, c_2 can depend on α.)

[Hint: Union bound.]

(c) (5 points) Let's plug in some concrete numerical parameters. Suppose $T = 1000$ (roughly one week of blocks in Bitcoin, for example). According to your analysis, in longest-chain consensus, how big should you take k (the number of unconfirmed blocks at the end of the longest chain) to ensure consistency (i.e., no blocks get rolled back) over the next T blocks with probability at least 90%?[1]

Problem 4

(15 points) Prove Theorem 3 from Lecture 8. That is, prove that with an infinitely long $2k$-balanced leader sequence, if a transaction is at some point known to every honest node, then it will eventually be included in a (confirmed) block of $B_k(G)$.

[1] BTW, here's the proof of Theorem 2 (self-consistency/finality of longest-chain consensus) that we didn't have time for in Lecture 8. Toward a contradiction, suppose that a block b belongs to $B_k(G_i)$ but not $B_k(G_j)$ for some $j > i$. In particular, b is in every longest chain of G_i but is excluded from some longest chain of G_j. Consider the first $h \in \{i+1, i+2, \ldots, j\}$ for which there is a longest chain excluding b. In G_h, there is a tie between two longest chains. These chains differ in their last k blocks (because b is k blocks deep on one of the chains and excluded from the other one). But Theorem 1 (the well-definedness of $B_k(G)$ proved in lecture) asserts that this can never happen (under the $2k$-balancedness assumption), a contradiction.

Problem 5

(30 points) This problem extends the "super-synchronous" model from Lectures 8 and 9 to include bounded message delays. We assume proof-of-work leader selection (and thus can proceed in the permissionless setting), and that an $\alpha < \frac{1}{2}$ fraction of the hashrate is Byzantine. There is no limit on the number of (honest or Byzantine) nodes in the system.

The new model is:

- There are time steps $t = 1, 2, \ldots, T$. Messages sent at a time t are received by the beginning of time step $t + 1$.[2] (Honest nodes only send messages via broadcast. Byzantine nodes can send different messages to different honest nodes.)
- During each time step, each node successfully produces 0 or 1 proof-of-work solutions (encoding a block and its predecessor). A node that controls an h fraction of the total hashrate produces a block in a given time step with probability $h \cdot \nu$, where ν is a small constant. (Thus, ν is the expected number of blocks produced per time step.)
- An honest node always seeks to extend the longest chain (breaking ties between longest chains arbitrarily), and in every time step broadcasts all of the blocks that it knows about.

The key point is that, in this model, it is possible for more than one node to produce a block in the same time step. Even honest nodes may therefore inadvertently create a fork. Because ties are broken arbitrarily, honest nodes may then try to extend different branches of the fork.

State and prove analogs of Theorems 1–3 from Lecture 8 for this model.

[Hints: Feel free to assume that ν is at most a sufficiently small constant of your choosing (perhaps depending on α); your theorem statements can depend on ν. Call a time step *relevant* if at least one block is produced. Call a relevant time step *good* if exactly one node produces a block, and that node is an honest node. Prove that, with high probability, for every sufficiently long sequence of relevant time steps, an almost $1-\alpha$ fraction of those time steps are good (see also Problem 3). If you need to bound from above the total number of blocks found by the attacker or more generally in non-good time steps, use the fact that there are Chernoff-style concentration inequalities for Poisson random variables.[3] Modify the proofs of Theorems 1–3 from lecture, with good time steps playing the previous role of honest blocks.]

[2] The analysis extends to the general synchronous model with an arbitrary a priori known bound Δ on the maximum message delay, as you're invited to check. (This problem corresponds to $\Delta = 1$.) The key assumption needed is that the expected number of blocks produced in an interval of Δ time steps is small (analogous to the parameter ν in this problem being small). For example, if in Bitcoin we (generously) set Δ to 10 seconds (e.g., $\Delta = 10000$ with one-millisecond time steps), then, because puzzle difficulty is tuned so that one block is produced every ten minutes on average, the probability of anyone producing a block in a length-Δ time window is less than 2%.

[3] See, for example, https://math.stackexchange.com/questions/2434883/chernoff-style-bounds-for-poisson-distribution. You may use these inequalities without proof. But be sure to explain why your random variable of interest has a Poisson distribution.