Take-home Assessment*

Due: Tues, 2-March-2021, 1pm – Gradescope entry code: `N8XV23`

1. The point distribution (in % of the total) is provided in brackets. The total points is 105%.
2. **This take-home assessment is open-book. However, you are not allowed to consult or solicit help from other people, in particular, but not limited to, classmates or Internet forums or chats. Please abide by the Stanford Honor Code.**
3. Please upload your answers to Gradescope. Start a new page for every problem. LaTeXing your answers is not necessary; handwritten is fine. For the programming/simulation questions you can use any reasonable programming language (please no assembly, brainfuck, etc. ☺). **Comment your source code and include the code and a brief overall explanation with your answers.**

*Version: 1 – Last update: 1-Mar-2021

1. (35%) In Lecture 10, we analyzed the latency of Prism in terms of the effective vote $v(t)$ on a voter chain, where

$$v(t) := \Pr[H(\tau) > A(\tau), \forall \tau \geq t] \tag{1}$$

where $H(\tau), A(\tau)$ are the number of honest and adversary voter blocks mined in $[0, \tau]$ on voter chain 1. (We drop the indexing on H and A to simplify the notation.) Recall that the analysis is in the special case when the block inter-arrival times are much larger than the network delay.

   a) (2%) If λ_h and λ_a are the honest and adversary mining rates (in voter blocks per second) respectively, state the distributions of the random variables $H(\tau)$ and $A(\tau)$.

   b) (4%) What is $v(0)$? What happens to $v(t)$ when $t \to \infty$ and why? Answer this question for all possible values of λ_h and λ_a.

   c) (4%) We now want to obtain an expression for $v(t)$ when $\lambda_h > \lambda_a$. First, derive an expression for $\Pr[H(t) - A(t) = \ell]$ for any integer ℓ. (The answer can be left in a summation.)

   d) (4%) Using the previous result or otherwise, derive an expression for $v(t)$ for any t. (Hint: you may find the Gambler's Ruin result Nakamoto used in the bottom of page 6 of the Bitcoin whitepaper useful.)

   e) (6%) Let $\beta := \lambda_a/(\lambda_h + \lambda_a)$ be the adversarial fraction of the hashing power. Suppose $\lambda_h = 1/15$ blocks/s. Plot $v(t)$ as a function of t for $\beta = 0.3$ and $\beta = 0.4$. (You can evaluate $v(t)$ numerically if you have an analytic expression of $v(t)$ in part (d). In case you cannot get one, you can try to generate this plot via simulations, but make sure your simulated results are accurate.)

   f) (4%) If a single proposer block is mined at time 0 and no other proposer block is mined at the same level, state a confirmation rule for Prism, assuming a large number of voter chains. Explain why this confirmation rule is not good when there are only a small number of voter chains.

   g) (4%) What are the confirmation latencies when $\beta = 0.3$ and $\beta = 0.4$ for this confirmation rule?

   h) (3%) What is the impact of β on the latencies in part (g)? Explain this intuitively.

   i) (4%) The confirmation latency of Ethereum 1.0, which has roughly the same block time, is much longer, of the order of 30 minutes or more. Give a qualitative explanation of how Prism speeds up the confirmation.

2. (30%) Figure 1 shows the throughputs and latencies of implementations of some of the protocols we studied in this course. Longest Chain refers to the basic Bitcoin protocol but with tuning of the block size and the mining difficulty. The target tolerable adversarial fraction of hashing power, β, is a tunable parameter in Longest Chain, Bitcoin-NG and Prism.

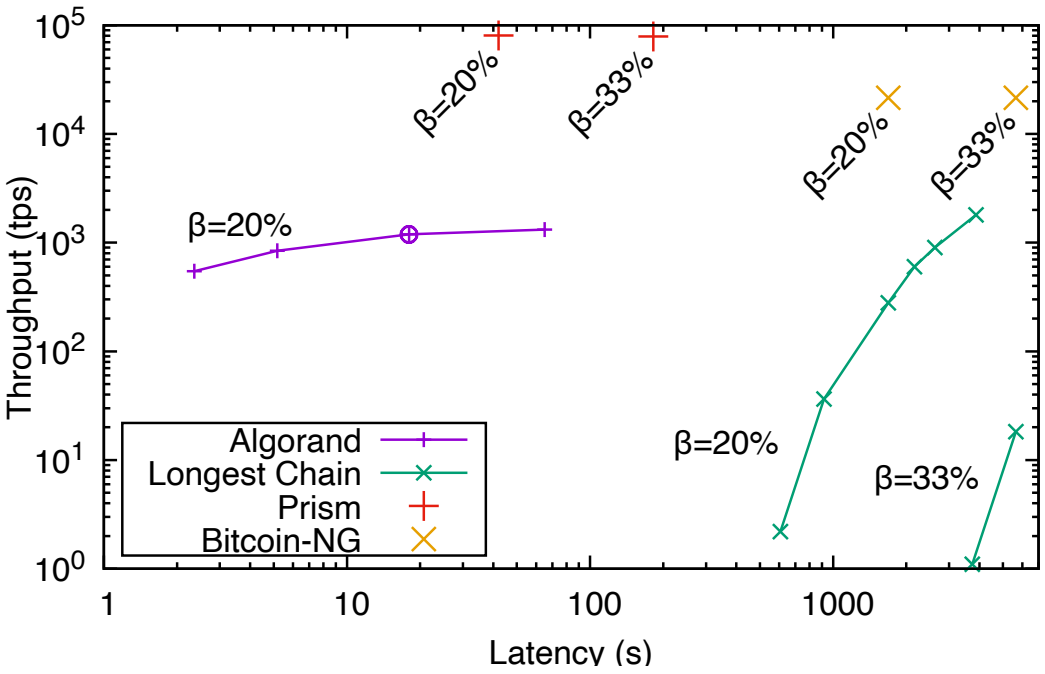

Figure 1: Plot of throughput versus latency for several protocols.

a) (3%) Explain why the throughput of Bitcoin-NG is insensitive to the value of β.

b) (3%) Explain why the throughput of Prism is insensitive to the value of β.

c) (8%) Give reasons why the latencies of (i) Longest Chain, (ii) Bitcoin-NG and (iii) Prism increase with increasing β. There may be more than one reason for each protocol, and the reasons may be different for different protocols. (Note that you should not assume that the network delay is zero here, since we are dealing with real systems!)

d) (6%) We have not studied Algorand yet, but for the purpose of this question, we can think of it as a BFT protocol like Streamlet. Give reasons why this protocol achieves better latency than the other protocols in Figure 1.

e) (10%) Suppose in the Longest Chain protocol, the network delay Δ depends on the block size B (in bytes) as:

$$\Delta = B/C + D, \tag{2}$$

where D is the speed-of-light propagation delay, in seconds, and C is the processing rate by the peer-to-peer relaying nodes, in bytes per second.

i. Using this relationship, show how the block size B and the mining difficulty can be tuned to achieve a throughput-latency tradeoff curve for the Longest Chain protocol at a fixed β, like the one shown in Figure 1 for $\beta = 20\%$. Be as precise as you can in exhibiting the tradeoff.

ii. From this analysis, can you deduce which points on the Figure 1 curve correspond to bigger block sizes? Which points correspond to higher mining difficulty?

3. (20%) In this question, we will consider the Streamlet protocol run by n permissioned nodes, with a tunable quorum size q for notarization.

   a) (8%) Suppose we want to maximize the resilience of the protocol, i.e., maximize the number of adversarial nodes f that can be tolerated such that the protocol is both safe and live. Derive the optimal quorum size q and show that the resulting optimal resilience is $f = n/3$.

   b) (12%) Suppose you believe that an attack against safety is more likely than an attack against liveness (since a double-spend can provide significant rewards to the attacker). Hence, you want to tune Streamlet to increase the resilience against safety attacks, even at the expense of decreasing the resilience against a liveness attack. Can this be done? If so, exhibit and plot the tradeoff between the two resiliences. If not, explain why not.

4. (20%) A consensus protocol is said to be m-accountable if *whenever* safety is violated, at least m nodes can be irrefutably proven to have violated the protocol. In the lecture, we claimed that Streamlet (with quorum size $2n/3$) is $n/3$-accountable, where n is the total number of nodes. But we have only gone through one possible safety violation scenario, not all. Complete the analysis by considering all safety violation scenarios.