Exercise 5: Aligning our Clocks

Task 1: Converging to Agreement

a) Given a skew bound $\mathcal{S}_r$ for pulse r, determine T_r and δ_r so that performing a respective iteration of the loop of Algorithm 5.2 results in correct execution of round r.

b) Determine a skew bound $\mathcal{S}_{r+1}$ for pulse $r+1$ as function of $\mathcal{S}_r$ for the (minimal) choices of T_r and δ_r from a). What is $\mathcal{S}_\infty := \lim_{r\to\infty} \mathcal{S}_r$?

c) Assume that $\max_{v\in V_g}\{H_v(0)\} \leq H$ for some known $H > \mathcal{S}_\infty$. Given ε, determine the round r_ε so that $\mathcal{S}_r \leq \mathcal{S}_\infty + \varepsilon$ for all $r \geq r_\varepsilon$. How long does it take in terms of real time until this skew bound is reached? (Hint: an asymptotic bound suffices, where we consider ϑ (and thus all values depending only on ϑ) to be a constant.)

d) Is this bound good/bad/optimal?

Task 2: We're not Synchronized!

a) Fix any T and $\mathcal{S}$ in accordance with Theorem 5.10, and compute Δ_w^v as in Lemma 5.9. Assume that node v uses default value 0 for Δ_w^v if no (or conflicting) messages are received from w during a round. Under these conditions, give an execution of Algorithm 5.2 in which skews remain larger than $T/2$ forever. You may assume that ϑ is sufficiently small to simplify matters, and negative hardware clock values are permitted (these represent late initialization).

b) Now assume that there are $n-f \leq n-2$ correct nodes $v \in V_g$ satisfying $0 \leq H_v(0) \leq \mathcal{S}$ and you are given an execution in which the skew is $\mathcal{S}$ for each pulse, and each correct node generates a pulse exactly every $P \in \mathbb{R}^+$ time. Moreover, faulty nodes never send messages and you may assume that the algorithm's parameters are such that, potentially, Δ_w^v could become much larger than $\mathcal{S}$ (in a correct execution of the algorithm). Show that if one of the faulty nodes is merely a "confused" correct node whose initial hardware clock value is off, there is an execution in which this node never synchronizes with the others. (Hint: Don't crunch numbers, find a way of giving the faulty nodes control over the confused node's clock adjustments, and use this to keep it away from the others!)

c*) Can you fix this by modifying the algorithm? That is, make sure that in the scenario of b), but even with up to $f-1 < n/3$ Byzantine nodes, eventually all correct nodes have skew at most $\mathcal{S}$? Again, you may assume that ϑ is close to 1. (Hint: Modify the behavior of nodes when they have *proof* that something is amiss so that they either catch up with the main field or slow down enough for the main field to catch up with them.)