Exercise 5: Aligning our Clocks

Task 1: Converging to Agreement

a) Given a skew bound $S_r$ for pulse $r$, determine $T_r$ and $\delta_r$ such that performing a respective iteration of the loop of Algorithm 5.2 results in correct execution of round $r$.

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

c) Assume that $\max_{v \in V} \{H_v(0)\} \leq H$ for some known $H > S_\infty$. Given $\varepsilon$, determine the round $r_\varepsilon$ so that $S_r \leq S_\infty + \varepsilon$ 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 $\vartheta$ (and thus all values depending only on $\vartheta$) to be a constant.)

d) Is this bound good/bad/optimal?

Task 2: We’re not Synchronized!

a) Fix any $T$ and $S$ in accordance with Theorem 5.10, and compute $\Delta^v_w$ as in Lemma 5.9. Assume that node $v$ uses default value 0 for $\Delta^v_w$ 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 $\vartheta$ 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 $0 \leq H_v(0) \leq S$ and you are given an execution in which the skew is $S$ 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, $\Delta^v_w$ could become much larger than $S$ (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 $S$? Again, you may assume that $\vartheta$ 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.)