Exercise 3: Control Issues

Task 1: Controlling the Global Skew

In the lecture, we proved that the GCS algorithm achieves local skew $\mathcal{O}(\log \mathcal{G})$, where $\mathcal{G}$ is maximum global skew in the network. However, we did show that GCS maintains any bound on $\mathcal{G}$ itself. Here you will analyze a variant of the GCS algorithm that maintains $\mathcal{G} = \mathcal{O}(D)$, where D is the network diameter. Thus the performance of this variant matches the lower bound proven in the previous lecture.

Recall that our analysis of the GCS algorithm only requires that nodes satisfying **FC** (respectively **SC**) run in fast (respectively slow) mode. The modification of the algorithm you describe here will modify the behavior of GCS only when neither **FC** nor **SC** applies in such a way that the algorithm maintains a bound on the global skew.

(a) Add the condition that any node $v \in V$ satisfying $L_v(t) = \max_{w \in V}\{L_w(t)\}$ is in slow mode at time t and determine a suitable trigger condition. Show that your trigger condition does not conflict with **FT**.

(b) Apply the techniques used in the (refined) Max Algorithm to maintain an estimate $M_v(t)$ of the largest clock value throughout the system at each $v \in V$. Show that $\max_{v \in V}\{L_v(t)\} \geq M_v(t) \geq \max_{v \in V}\{L_v(t)\} - \mathcal{G}_{\max}$ for some $\mathcal{G}_{\max}$. (Hints: Make minimal modifications to the Max Algorithm, so that the reasoning changes very little. This way, you can argue that the proof of the bound is analogous. Note that you need to be slightly more careful regarding the rate at which nodes increase the estimates when $L_v(t) < M_v(t)$: use rate $h_v/\vartheta \leq 1$. You should obtain $\mathcal{G}_{\max} = ((\vartheta - 1/\vartheta)T + (\vartheta - 1)d + u)D$.)

(c) Show that $L_v(t) = \min_{w \in V}\{L_w(t)\}$ implies that v does not satisfy **ST** at time t.

(d) Assume that $\sigma = \mu/(\vartheta - 1) > 1$ and that $\max_{v \in V}\{H_v(0)\} \leq \mathcal{G}_{\max}$. Add the condition that any node $v \in V$ satisfying $L_v(t) < M_v(t)$ such that **ST** does not hold at time t is in fast mode. Conclude that the modified algorithm has global skew $\mathcal{G} \leq \mathcal{G}_{\max}$ and still obeys **FC** and **SC**. What is the resulting local skew, provided that $\max_{\{v,w\} \in E}\{H_v(0) - H_w(0)\} \leq \delta$?

Task 2: Controlling Uncertainty

In the lecture, we assumed that $v \in V$ has an estimate $\tilde{L}_w$ of the logical clock L_w of each of its neighbors $w \in N_v$, satisfying that $L_w(t) - \delta < \tilde{L}_w(t) \leq L_w(t)$ at all times t. In this exercise, you determine δ for a straightforward way of deriving such an estimate. You may assume that $\max_{\{v,w\} \in E}\{H_v(0) - H_w(0)\} \leq \delta - (\vartheta(1+\mu) - 1/\vartheta)d$ throughout this exercise and that $\vartheta \in \mathcal{O}(1)$.

a) Suppose $w \in V$ sends a message with its current logical clock value whenever $H_v(t) = kT$ for some $k \in \mathbb{N}$, and also at time 0. Determine a (good) estimate $\tilde{L}_w(t)$ that $v \in V$ can compute based on this information. Bound the resulting δ. (Hint: It's ok to be a bit sloppy with lower order terms or constant factors, as long as you get the asymptotics right.)

b) For fixed values of all other parameters, determine a choice of μ asymptotically minimizing our upper bound on the local skew (i.e., up to constant factors). (Hint: Argue that $\delta \in \Omega(\mathcal{G})$ implies that the upper bound is trivial and that it doesn't matter (asymptotically) to choose μ to be at least $\max\{u/(T+d), 8(\vartheta - 1)\}$. Having ruled out these corner cases, check how the bound changes if a value of μ satisfying these constraints is doubled.)

c) For this method of determining estimates, the asymptotically optimal choice of μ you computed, and the global skew bound you obtained in the first exercise, determine the bound on the local skew as function of T (use the same value of T for global and local estimates).

Task 3*: Control Right from the Beginning

So far, we have largely ignored the issue of network initialization. It is unrealistic to assume that all nodes start executing the algorithm precisely at time 0. Indeed, this would require perfect synchronization! Instead, we now assume that nodes can spontanteously wake up and execute the algorithm at any time, and that a node wakes up when it receives its first message. The hardware clock of a node is 0 at the time when it wakes up. W.l.o.g., assume that at least one node wakes up at time 0.

a) Initialize the network by flooding, i.e., on wake-up, a node broadcasts a message to all its neighbors. Adapt the clock estimation technique from Task 2 to account for the modified initialization.

b) Extend the hardware clock functions, logical clock functions, and clock estimates to be defined from time 0 on such that (i) $1 \leq h_v(t) \leq \vartheta$ for all t, (ii) $L_v(t) = H_v(t)$ at times t when v has not yet woken up, (iii) $L_w(t) - \delta < \tilde{L}_w(t) \leq L_w(t)$ at all times t, and (iv) node v is in slow mode at times t when it has not woken up yet according to the (modified) GCS algorithm.

c) Convince Will and your fellow students that this approach yields the same skew bounds you computed in Tasks 1 and 2!