Exercise 1: Global vs. Local

Task 1: Even more Globally Optimal (60 points)

In the lecture, we proved a lower bound of $\Omega(uD)$ for the global skew of any algorithm, while the uppper bound achieved by the max algorithm is $\mathcal{O}(((\vartheta-1)d+u)D)$. Thus, the upper bound may be significantly larger than our lower bound in the regime where $(\vartheta-1)d \gg u$. In this exercise, you will show that an upper bound of $\mathcal{O}(uD+(\vartheta-1)d)$ is always achievable.

(a) Let $v \in V$ be a node and $w \in N_v$ an arbitrary neighbor. Consider the following process: at time $t=0$, v sends a message to w. When w receives v's message, w immediately replies with a message. When v receives w's message, v again responds immediately and so on. Thus, for any exection, this process defines a sequence of times $0=t_0<t_1<t_2<\cdots$ when v receives a message from w. Define a "hardware clock" $\widehat{H}_v$ that (discontinuously) adjusts its value at times $t_1, t_2, \ldots$ and increases at the same rate as H_v at all other times such that for all times $t<t'$, $\widehat{H}_v$ satisfies

$$\widehat{H}_v(t')-\widehat{H}_v(t) \leq \left(1+\frac{u}{d-u}\right)(t'-t)+c(u+(\vartheta-1)d) \quad \text{and} \quad H_v(0)+t \leq \widehat{H}_v(t).$$

for some suitable constant $c>0$. (25 points) *Hint: What do you know about* $t_{i+1}-t_i$*?*

(b) Consider the "Min Algorithm," in which nodes reduce their logical clock value to $L+d$ when receiving a message $\langle L\rangle$, but use $\widehat{H}_v$ as "hardware" clock at v. (The messages sent are specified in c)). Define a "virtual clock" $L_{\min}$ that lower bounds all actual clock values, but also takes into account messages that are in transit, which might cause nodes to set their clocks back. This clock should satisfy that it increases at least at rate 1. Conclude that the algorithm guarantees amortized 1-progress. (10 points)

(c) Bound the global skew of the algorithm. We are not concerned with the number of sent messages, so nodes will send a message immediately after receiving one, as well as send one every d local time (w.r.t. H_v). (Remark: If it helps with notation, feel free to show the statement only on a graph that is a simple path.) (25 points)

Solution

(a) Fix a node v, set $t_0=0$, and let $t_1<t_2<\cdots$ be the consecutive (objective) times at which v receives messages from its "chosen" neighbor, w. Observe that for all i, we have

$$2(d-u) \leq t_i - t_{i-1} \leq 2d. \tag{1}$$

Define $\widehat{H}_v(0)=H_v(0)$. We inductively define the values of $\widehat{H}_v(t)$ in the interval $(t_{i-1}, t_i]$ as follows:

$$\widehat{H}_v(t)=\begin{cases}\widehat{H}_v(t_{i-1})+(H_v(t)-H_v(t_{i-1})) & \text{for } t<t_i\\ \widehat{H}_v(t_{i-1})+\min\{2d, H_v(t)-H_v(t_{i-1})\} & \text{for } t=t_i.\end{cases}$$

Now fix arbitrary times t', t with $t'>t$. First consider the case where there are no times t_i satisfying $t \leq t_i \leq t'$. Observe that this case implies that $t'-t \leq 2d$. Since

$\widehat{H}_v$ never increases at a rate faster than ϑ (the maximum rate of H_v), we have

$$\begin{aligned}\widehat{H}_v(t') - \widehat{H}_v(t) &\le \vartheta \cdot (t' - t)\\ &= 1 \cdot (t' - t) + (\vartheta - 1)(t' - t)\\ &\le 1 \cdot (t' - t) + (\vartheta - 1) \cdot (2d).\end{aligned}$$

This gives the first desired bound on the speed of $\widehat{H}_v(t)$ setting the constant $c = 2$. Now consider the case where there are time $t_k, t_{k+1}, \ldots, t_\ell$ satisfying $t < t_k \le t_\ell < t'$ with $t_{k-1} \le t$ and $t' \le t_{\ell+1}$. By the definition of $\widehat{H}_v$, for $i = k+1, k+2, \ldots, \ell$, we have $\widehat{H}_v(t_i) \le \widehat{H}_v(t_{i-1}) + 2d$. Therefore, applying Equation (1) gives

$$\frac{\widehat{H}_v(t_i) - \widehat{H}_v(t_{i-1})}{t_i - t_{i-1}} \le \frac{2d}{2(d-u)} = \frac{d - u + u}{d - u} = 1 + \frac{u}{d-u},$$

hence

$$\widehat{H}_v(t_i) - \widehat{H}_v(t_{i-1}) \le \left(1 + \frac{u}{d-u}\right)(t_i - t_{i-1}). \tag{2}$$

Finally, we compute

$$\begin{aligned}\widehat{H}_v(t') - \widehat{H}_v(t) &= (\widehat{H}_v(t') - \widehat{H}_v(t_\ell)) + (\widehat{H}_v(t_\ell) - \widehat{H}_v(t_{\ell-1})) + \cdots + (\widehat{H}_v(t_k) - \widehat{H}_v(t))\\ &\le 1 \cdot (t' - t_\ell) + (\vartheta - 1) \cdot (2d)\\ &\quad + \sum_{i=k+1}^{\ell} \left(1 + \frac{u}{d-u}\right)(t_i - t_{i-1})\\ &\quad + 1 \cdot (t_k - t) + (\vartheta - 1) \cdot 2d\\ &\le \left(1 + \frac{u}{d-u}\right)(t' - t) + 4(\vartheta - 1) \cdot d,\end{aligned}$$

which gives the desired upper bound. For the lower bound $\widehat{H}_v(t) \ge H_v(0) + t$ first observe that for all $t \in [t_i, t_{i-1}]$, we have $\widehat{H}_v(t) \ge H_v(t_{i-1}) + (t - t_i)$ because H_v increases at a rate of at least 1. Next, we have

$$\widehat{H}_v(t_i) - \widehat{H}_v(t_{i-1}) \ge \min\{2d, H_v(t_i) - H_v(t_{i-1})\}$$

If we have $2d < H_v(t_i) - H_v(t_{i-1})$ then, $\widehat{H}_v(t_i) - \widehat{H}_v(t_{i-1}) = 2d$, while $t_i - t_{i-1} \le 2d$. On the other hand, $H_v(t_i) - H_v(t_{i-1}) \ge t_i - t_{i-1}$ because H_v's rate is at least 1. The desired result follows by writing

$$\widehat{H}_v(t) - H_v(0) = (\widehat{H}_v(t) - \widehat{H}_v(t_\ell)) + (\widehat{H}_v(t_\ell) - \widehat{H}_v(t_{\ell-1})) + \cdots + (\widehat{H}_v(t_1) - H_v(0)),$$

and applying the above observations to each term in this sum separately.

(b) Define $L_{\min}(t)$ to be the minimum clock value in the system at time t, where we take message $\langle L \rangle$ in transit into account as value $L + (t - t_s)$ if it has been sent at time t_s. It's easy to see that $L_{\min}$ increases at rate at least 1 at all times; no node will ever set its clock to a smaller value than $L_{\min}$. It follows that the algorithm satisfies amortized 1-progress, as $L_{\min}$ does (and the skew is bounded; this is technically only shown in the next part).

(c) Consider node $v \in V$ and time t. We claim that $L_v(t_r) \le L_{\min}(t_r) + u(D+1) + (\vartheta - 1)d$ for some $t_r \le t + d(D+1)$. First, if $L_{\min}(t) = L + (t - t_s)$ for some message $\langle L \rangle$, it will be received no later than time $t' = t_s + d$, cause the respective clock be set back to $L+d$ if necessary, and trigger a message $\langle L' \rangle$ with $L' \leq L+d \leq L_{\min}(t')+u$ (as the message is under way for at least $d-u$ time). On the other hand, if $L_{\min}(t) = L_w(t)$ for some w, this node will send a message $\langle L \rangle$ at a time $t' \in [t, t+d)$, for which $L \leq L_{\min}(t') + (\vartheta - 1)d$. Either way, this causes a chain of messages reaching w after at most D hops. On each hop, the communicated value increases by (at most) d, and the message travels for at least $d-u$ time. As $L_{\min}$ increases at least at rate 1, the claim follows. As $L_v(t) - L_v(t_r) \leq \widehat{H}_v(t) - \widehat{H}_v(t_r) \in (1+2u/d)(t-t') + \mathcal{O}((\vartheta-1)d)$ and $L_{\min}$ increases at least at rate 1, we conclude that for times $t \geq d(D+1)$, we have that

$$\begin{aligned}
L_v(t) &\in L_{\min}(t) + \left(1 + \frac{2u}{d} - 1\right) d(D+1) + \mathcal{O}((\vartheta-1)d) + u(D+1) + (\vartheta-1)d \\
&\subseteq \mathcal{O}(uD + (\vartheta-1)d)\,.
\end{aligned}$$

As this holds for any $v \in V$ and $L_{\min}(t) \geq \min_{w \in V}\{L_w(t)\}$, this is a bound on $\mathcal{G}(t)$ for $t \geq d(D+1)$. Assuming that $H \in \mathcal{O}(uD + (\vartheta-1)d)$, using the properties of $\widehat{H}$ we readily get that $\mathcal{G}(t) \in \mathcal{O}(uD + (\vartheta-1)d)$ for times $t < d(D+1)$ as well.

Task 2: The Least Possible Locally Optimal (40 points)

The *local skew* is defined in a similar way as the global skew except that we consider the logical clock difference between the neighboring nodes only:

$$\mathcal{L} := \sup_{t \in \mathbb{R}_0^+} \{\mathcal{L}(t)\}\,,$$

where

$$\mathcal{L}(t) := \max_{\{v,w\} \in E} \{|L_v(t) - L_w(t)|\}$$

In this exercise, we analyze the local skew of the Refined Max Algorithm.

a) Let $P = (v_0, \ldots, v_D)$ be a path of length D. Construct an execution in which (i) $H_v(0) = 0$ for all $v \in V$, (ii) $H_{v_{i+1}}(t_0) - H_{v_i}(t_0) = u$ for all $i \in [D]$, (iii) $L_v(t) = H_v(t)$ for all $v \in V$ and times t, and (iv) each hardware clock runs at rate 1 after time t_0. (Hint: Simply set all message delays to d and ramp hardware clock speeds. Then show that $L_v(t) = H_v(t)$ for all $v \in V$ and t, because no node v receives a message with $L > H_v(t) - d + u$ at any time t.) (20 points)

b) Take the above execution and modify it such that a skew of $u(D-1)$ occurs at some time between nodes v_0 and v_1. (Hint: Only change message delays after time t_0, nothing else. "Pull" the nodes to the current maximum clock value one by one, starting with v_{D-1}.) (20 points)

Solution

a) Set $t_0 := uD/(\vartheta - 1)$. From time t_0 on, all hardware clock rates are 1. For times $t \in [0, t_0]$, node v_i has hardware clock rate $1 + \frac{i(\vartheta-1)}{D}$. All message delays are d.

   Observe that, for neighbors v_i, v_{i+1}, we have that $L_{v_{i+1}}(t) - L_{v_i}(t) \leq \frac{\vartheta-1}{D} \cdot t_0 = u$ at all times t. As message delays are d, a message sent at local time H is received at local time $H - u + d$ (or later). As initially $H_v(0) = L_v(0)$ at all nodes and both increase at the same rate, there can be no first received message causing any node to set its logical clock to a larger value than its hardware clock. We conclude that $L_v(t) = H_v(t)$ for all times t.

b) At time t_0, we switch the delay of any message sent by v_D to $d-u$. Inductively, we define times t_i, $i \in [D]$ where we apply the same change to node v_{D-i}. Given t_{i-1}, t_i is defined as the reception time of the first fast message from node $D-(i-1)$. We observe that (i) at time t_i, $i > 0$, node $D-i$ sets $L_{v_{D-i}}(t_i) = L_{v_{D-(i-1)}}(t_i - (d-u)) + d - u = L_{v_{D-(i-1)}}(t_i)$, (ii) no other changes to clock values occur, as nodes $j \geq D-i$ have identical clock values from time t_i on and nodes $j < D-i$ have not observed a difference to the original execution yet. Thus, at time t_{D-1}, we have that

$$L_{v_1}(t_{D-1}) - L_{v_0}(t_{D-1}) = L_{v_D}(t_{D-1}) - L_{v_0}(t_{D-1}) = L_{v_D}(t_0) - L_{v_0}(t_0) = uD\,.$$

Task 3*: Global and Local Happiness

a) Find out what the term synchronizers refers to in distributed computing!

b) Use basic techniques for synchronizers to devise an algorithm that satisfies constant amortized progress, has asymptotically optimal global skew, and has local skew $\mathcal{O}(uD)$. You may assume that $(\vartheta - 1)d \leq u$, so you don't have to worry about removing $(\vartheta - 1)d$ terms.

c) Synchronize (i.e., communicate your findings) with the other students in the TA session!