Lecture 3

Lower Bound on the Local Skew

In Chapter 1, we proved tight upper and lower bounds of $\Theta(D)$ for the global skew of any clock synchronization algorithm. However, the algorithms achieving optimal global skew had the undesireable feature that the maximal global skew could be attained between any pair of nodes in the network—even adjacent nodes. In Chapter 2, we developed a more refined algorithm that further controlled the gradient skew—the maximum skew between any pair of *adjacent* nodes. Specifically, the gradient clock synchronization (GCS) algorithm of Chapter 2 achieved a local skew of $\mathcal{O}(\delta \log D)$.

In this chapter, we address the question of whether the $\mathcal{O}(\delta \log D)$ skew upper bound for GCS can be improved. Since gradient clock synchronization is a local property (in the sense that the definition of gradient skew only references logical clocks of neighboring nodes), one may expect that a distributed algorithm may be able to achieve $\mathcal{O}(\delta)$ local skew. However, we will show that this is impossible: any GCS algorithm must incur local skew of $\Omega(u \log D)$ for some executions. Thus, the GSC algorithm of Chapter 2 is asymptotically optimal.

3.1 Lower Bound with Bounded Clock Rates

In this section, we first prove a lower bound assuming that each logical clock increases at a rate of at most $(1+\mu)h_v > 1$. That is, for all $v \in V$ and $t, t' \in \mathbb{R}_0^+$ with $t < t'$, we assume $L_v(t') - L_v(t) \leq (1+\mu)(H_v(t') - H_v(t))$.[1] We use the model of Chapter 1. Moreover, all logical clocks have a minimum rate of 1: for all $v \in V$ and $t, t' \in \mathbb{R}_0^+$ with $t < t'$, we have $L_v(t') - L_v(t) \geq t' - t$. Under these assumptions, we will prove the following theorem.

Theorem 3.1. *Any algorithm for the gradient clock synchronization problem with logical clock rates between* 1 *and* $(1+\mu)h_v$ *incurs a worst-case gradient skew of* $\mathcal{L} \geq (u/4 - (\vartheta - 1)d) \log_{\lceil \sigma \rceil} D$*, where* $\sigma \coloneqq \mu/(\vartheta - 1)$*.*

[1] Note that this assumption does not allow for algorithms that increase their clocks discontinuously. For example, the argument does not apply to the max algorithm presented in Chapter 1.

To gain some intuition, assume that $(\vartheta - 1)d \ll u$, so we can neglect this term. In order to prove Theorem 3.1, we first show that the adversary can build up a *hardware* clock skew of $\Omega(uk)$ between any pair of nodes in distance k in $\mathcal{O}(uk/(\vartheta - 1))$ time, in an indistinguishable way. Specifically, for v and w in distance k, we get that $H_v^{(\mathcal{E}_v)}(t) - H_v^{(\mathcal{E}_1)}(t) \in \Omega(uk)$ for some time t, while $H_w^{(\mathcal{E}_v)}(t) = H_w^{(\mathcal{E}_1)}(t)$. By the minimum progress condition, this implies that the logical clock of v differs by at least $\Omega(uk)$ between the two executions. This is, in fact, a straightforward generalization of Lemma 1.5. The key difference is that we sacrifice a factor of 2 in the amount of skew we sneak in, so we can choose the pair of nodes between which we build the skew *after* examining what happens in $\mathcal{E}_1$, i.e., $\mathcal{E}_1$ provides no information regarding where the skew will "appear."

We can use this inductively as follows. Assuming that we know how to build up a skew of αuk between nodes in distance k (initially, $k \approx D$ and $\alpha = 0$), we run a given GCS algorithm for $\mathcal{O}(uk'/(\vartheta - 1))$ time with all hardware clock rates being 1 (that's the case in $\mathcal{E}_1$), where $k' \in \Theta(k/\sigma)$ (with constants chosen suitably). As logical clock rates are between 1 and $1+\mu$ in $\mathcal{E}_1$, the skew between the original nodes is still $\alpha uk - \mathcal{O}(u\mu k'/(\vartheta - 1)) = (\alpha - \mathcal{O}(1))uk$. Thus, there must be two nodes in distance k' with skew at least $(\alpha - \mathcal{O}(1))uk'$. If v is the node with the larger clock value, we now consider $\mathcal{E}_v$, in which the skew is by $\Omega(uk')$ larger. For the right choice of k', we end up with a path of length k' that has skew $(\alpha + \Omega(1))uk'$! We can repeat this up to $\Theta(\log_\sigma D))$ many times, yielding the desired lower bound.

Lemma 3.2. *Assume that $(\vartheta-1)d < u/2$ and set $t_0 \coloneqq d(v,w)(u/(2(\vartheta-1))-d)$. For any algorithm, there is an execution $\mathcal{E}_1$ such that for any $v, w \in V$, there is an indistinguishable execution $\mathcal{E}_v$ satisfying that*

- $H_x^{(\mathcal{E}_1)}(t) = t$ *for all* $x \in V$ *and* t,
- $H_v^{(\mathcal{E}_v)}(t) = H_v^{(\mathcal{E}_1)}(t) + d(v,w)(u/2 - (\vartheta - 1)d)$ *for all* $t \geq t_0$, *and*
- $H_w^{(\mathcal{E}_v)}(t) = t$ *for all* t.

Proof. The proof is very similar to the one of Lemma 1.5. In both executions and for all $x \in V$, we set $H_x(0) \coloneqq 0$. Execution $\mathcal{E}_1$ is given by running the algorithm with all hardware clock rates being 1 at all times and the message delay from x to y being $d - u/2$.

Set

$$d(x) \coloneqq \begin{cases} -d(v,w) & \text{if } d(x,w) - d(x,v) < -d(v,w) \\ d(v,w) & \text{if } d(x,w) - d(x,v) > d(v,w) \\ d(x,w) - d(x,v) & \text{else.} \end{cases}$$

Note that $|d(x) - d(y)| \leq 2$ for any $\{x, y\} \in E$. Moreover, $d(v) = d(v,w)$ and $d(w) = -d(v,w)$. In $\mathcal{E}_v$, we set the hardware clock rate of node $x \in V$ to $1 + (\vartheta - 1)(d(x) + d(v,w))/(2d(v,w))$ at all times $t \leq t_0$ and to 1 at all times $t > t_0$. This implies that

$$H_v^{(\mathcal{E}_v)}(t_0) = \vartheta t_0 = H_v^{(\mathcal{E}_1)}(t_0) + d(v,w)\left(\frac{u}{2} - (\vartheta - 1)d\right) \quad \text{and}$$
$$H_w^{(\mathcal{E}_v)}(t_0) = t_0\,.$$

As clock rates are 1 from time t_0 on, this means that the hardware clocks satisfy all stated constraints.

It remains to specify message delays and show that the two executions are indistinguishable. We achieve this by simply ruling that a message sent from some $x \in V$ to a neighbor $y \in N_x$ in $\mathcal{E}_v$ arrives at the same local time at y as it does in $\mathcal{E}_1$. By induction over the arrival sending times of messages, then indeed all nodes also send identical messages at identical local times in both executions, i.e., the executions are indistinguishable. However, it remains to prove that this results in all message delays being in the range $(d-u, d)$.

To see this, recall that for any $\{x, y\} \in E$, we have that $|d(x) - d(y)| \leq 2$. As clock rates are 1 after time t_0 and constant before, and all hardware clocks are 0 at time 0, the maximum difference between any two local times between neighbors is attained at time t_0. We compute

$$H_x^{(\mathcal{E}_v)}(t_0) - H_y^{(\mathcal{E}_v)}(t_0) = \frac{d(y) - d(x)}{2d(v,w)} \cdot (\vartheta - 1)t_0 = \frac{d(y) - d(x)}{2}\left(\frac{u}{2} - (\vartheta - 1)d\right) .$$

In execution $\mathcal{E}_1$, a message sent from x to y at local time $H_x^{(\mathcal{E}_1)}(t) = t$ is received at local time $H_y^{(\mathcal{E}_1)}(t) = H_x^{(\mathcal{E}_1)}(t) + d - u/2$. If a message is sent at time t in $\mathcal{E}_v$, we have that

$$\begin{aligned} H_y^{(\mathcal{E}_v)}(t+d) &\geq H_y^{(\mathcal{E}_v)}(t) + d \\ &= H_x^{(\mathcal{E}_v)}(t) + d + \frac{d(x) - d(y)}{2}\left(\frac{u}{2} - (\vartheta - 1)d\right) \\ &> H_x^{(\mathcal{E}_v)}(t) + d - \frac{u}{2} \end{aligned}$$

where the last inequality uses that $d(x) - d(y) \geq -2$ and that $u/2 > (\vartheta - 1)d$ by assumption. On the other hand,

$$\begin{aligned} H_y^{(\mathcal{E}_v)}(t+d-u) &< H_y^{(\mathcal{E}_v)}(t) + \vartheta d - u \\ &= H_x^{(\mathcal{E}_v)}(t) + \vartheta d - u + \frac{d(x) - d(y)}{2}\left(\frac{u}{2} - (\vartheta - 1)d\right) \\ &\leq H_x^{(\mathcal{E}_v)}(t) + d - \frac{u}{2} , \end{aligned}$$

where the final inequality holds with equality if $d(x) - d(y) = 2$ and thus also for $d(x) - d(y) < 2$, as $u/2 > (\vartheta - 1)d$. □

Proof of Theorem 3.1. Note that the claim is vacuous if $(\vartheta - 1)d \geq u/4$, so we can assume the opposite in the following. Set $b \coloneqq \lceil 2\sigma \rceil$ and $i_{\max} \coloneqq \lfloor \log_b D \rfloor$. By induction over $i \in [i_{\max} + 1]$, we show that we can build up a skew of $(i+2)(u/4-(\vartheta-1)d)d(v,w)$ between nodes $v, w \in V$ in distance $d(v,w) = b^{i_{\max}-i}$ at a time t_i in execution $\mathcal{E}^{(i)}$, such that after time t_i all hardware clock rates are 1 and all sent messages have delays of $d - u/2$.

We anchor the induction at $i = 0$ by applying Lemma 3.2, choosing t_0 as in the lemma. We pick two nodes $v, w \in V$ in distance $b^{i_{\max}} \leq D$ of each other such that $L_v^{(\mathcal{E}_1)}(t_0) \geq L_w^{(\mathcal{E}_1)}(t_0)$. Now consider $\mathcal{E}_v$ for this choice of $v, w \in V$, which satisfies that $H_v^{(\mathcal{E}_v)}(t_0) = H_v^{(\mathcal{E}_1)}(t_0) + (u/2 - (\vartheta - 1)d)d(v,w)$ and $H_w^{(\mathcal{E}_v)}(t_0) = H_w^{(\mathcal{E}_1)}(t_0)$. By indistinguishability of the two executions and the

minimum logical clock rate of 1, we get that

$$\begin{aligned} L_v^{(\mathcal{E}_v)}(t_0) - L_w^{(\mathcal{E}_v)}(t_0) &= L_v^{(\mathcal{E}_1)}\left(t_0 + \left(\frac{u}{2} - (\vartheta-1)d\right)d(v,w)\right) - L_w^{(\mathcal{E}_1)}(t_0) \\ &\geq L_v^{(\mathcal{E}_1)}(t_0) + \left(\frac{u}{2} - (\vartheta-1)d\right)d(v,w) - L_w^{(\mathcal{E}_1)}(t_0) \\ &\geq \left(\frac{u}{2} - (\vartheta-1)d\right)d(v,w)\,. \end{aligned}$$

We obtain $\mathcal{E}^{(0)}$ by changing all hardware clock rates in $\mathcal{E}_v$ to 1 at time t_0 and all message delays of messages sent at or after time t_0 to $d - u/2$. As this does not affect the logical clock values at time t_0 — $\mathcal{E}^{(0)}$ is indistinguishable from $\mathcal{E}_v$ at $x \in V$ until local time $H_x^{(\mathcal{E}^{(0)})}(t_0)$ — this shows the claim for $i = 0$.

For the induction step from i to $i+1$, let $v, w \in V$, $\mathcal{E}^{(i)}$, and t_i be given by the induction hypothesis, i.e.,

$$L_v^{(\mathcal{E}^{(i)})}(t_i) - L_w^{(\mathcal{E}^{(i)})}(t_i) \geq (i+2)\left(\frac{u}{4} - (\vartheta-1)d\right)d(v,w)\,,$$

and from time t_i on all hardware clock rates are 1 and sent messages have delay $d - u/2$. Note that the latter conditions mean that $\mathcal{E}^{(i)}$ behaves exactly like $\mathcal{E}_1$ from Lemma 3.2 from time t_i on, except that some messages sent at times $t < t_i$ may arrive during $[t_i, t_i + d)$. Hence, if we apply the same modifications to $\mathcal{E}^{(i)}$ as to $\mathcal{E}_1$, but starting from time $t_i + d$ instead of time 0, we can, for any $v', w' \in V$, construct an execution $\mathcal{E}_{v'}$ indistinguishable from $\mathcal{E}^{(i)}$, where

- $H_x^{(\mathcal{E}^{(i)})}(t) = H_x^{(\mathcal{E}^{(i)})}(t_i) + t - t_i$ for all $x \in V$ and $t \geq t_i$,
- $H_{v'}^{(\mathcal{E}_{v'})}(t) = H_{v'}^{(\mathcal{E}^{(i)})}(t) + d(v',w')(u/2 - (\vartheta-1)d)$ for all times $t \geq t_i + d + (u/(2(\vartheta-1)) - d)d(v',w')$, and
- $H_{w'}^{(\mathcal{E}_{v'})}(t) = H_{w'}^{(\mathcal{E}^{(i)})}(t_i) + t - t_i$ for all $t \geq t_i$.

Consider the logical clock values of v and w in $\mathcal{E}^{(i)}$ at time

$$t_{i+1} \coloneqq t_i + d + \left(\frac{u}{2(\vartheta-1)} - d\right)\frac{d(v,w)}{b}\,.$$

Recall that $l_v(t) \geq h_v(t) \geq 1$ and $l_w(t) \leq (1+\mu)h_w(t)$ at all times t. As $h_w^{(\mathcal{E}^{(i)})}(t) = 1$ at times $t \geq t_i$, we get that

$$L_v^{(\mathcal{E}^{(i)})}(t_{i+1}) - L_w^{(\mathcal{E}^{(i)})}(t_{i+1}) \geq L_v^{(\mathcal{E}^{(i)})}(t_i) - L_w^{(\mathcal{E}^{(i)})}(t_i) - \mu(t_{i+1} - t_i)\,. \tag{3.1}$$

Recall that $d(v,w) = b^{i_{\max}-i}$ and that $b = \lceil 2\sigma \rceil$. We split up a shortest path from v to w in b subpaths of length $b^{i_{\max}-(i+1)}$. By the pidgeon hole principle, at least one of these paths must exhibit at least a $1/b$ fraction of the skew between v and w, i.e., there are $v', w' \in V$ with $d(v',w') = b^{i_{\max}-(i+1)} = d(v,w)/b$ so

that

$$
\begin{aligned}
& L_{v'}^{(\mathcal{E}^{(i)})}(t_{i+1}) - L_{w'}^{(\mathcal{E}^{(i)})}(t_{i+1}) \\
\geq\ & \frac{L_{v}^{(\mathcal{E}^{(i)})}(t_{i+1}) - L_{w}^{(\mathcal{E}^{(i)})}(t_{i+1})}{b} \text{ by (3.1) we have:} \\
\geq\ & \frac{L_{v}^{(\mathcal{E}^{(i)})}(t_i) - L_{w}^{(\mathcal{E}^{(i)})}(t_i) - \mu(t_{i+1} - t_i)}{b} \\
=\ & \frac{L_{v}^{(\mathcal{E}^{(i)})}(t_i) - L_{w}^{(\mathcal{E}^{(i)})}(t_i) - \mu(d + (u/(2(\vartheta-1)) - d)d(v', w'))}{b} \\
\geq\ & \frac{L_{v}^{(\mathcal{E}^{(i)})}(t_i) - L_{w}^{(\mathcal{E}^{(i)})}(t_i) - \mu u d(v', w')/(2(\vartheta-1))}{b} \\
\geq\ & \frac{L_{v}^{(\mathcal{E}^{(i)})}(t_i) - L_{w}^{(\mathcal{E}^{(i)})}(t_i)}{b} - \frac{\mu}{2\sigma(\vartheta-1)} \cdot \frac{u}{2} \cdot d(v', w') \\
=\ & \frac{L_{v}^{(\mathcal{E}^{(i)})}(t_i) - L_{w}^{(\mathcal{E}^{(i)})}(t_i)}{b} - \frac{u}{4} \cdot d(v', w') \\
\geq\ & \frac{(i+2)(u/4 - (\vartheta-1)d)d(v, w)}{b} - \frac{u}{4} \cdot d(v', w') \\
=\ & \left((i+2)\left(\frac{u}{4} - (\vartheta-1)d\right) - \frac{u}{4}\right) d(v', w')\,.
\end{aligned}
$$

In other words, as the average skew on a shortest path from v to w did not decrease by more than $u/4$, there most be some subpath of length $d(v,w)/b$ with at least the same average skew. Now we sneak in additional skew by advancing the (hardware and thus also logical) clock of v' using the indistinguishable execution $\mathcal{E}_{v'}$:

$$
\begin{aligned}
& L_{v'}^{(\mathcal{E}_v)}(t_{i+1}) - L_{w'}^{(\mathcal{E}_v)}(t_{i+1}) \\
=\ & L_{v'}^{(\mathcal{E}^{(i)})}\left(t_{i+1} + \left(\frac{u}{2} - (\vartheta-1)d\right) d(v', w')\right) - L_{w'}^{(\mathcal{E}^{(i)})}(t_{i+1}) \\
\geq\ & L_{v'}^{(\mathcal{E}^{(i)})}(t_{i+1}) + \left(\frac{u}{2} - (\vartheta-1)d\right) d(v', w') - L_{w'}^{(\mathcal{E}^{(i)})}(t_{i+1}) \\
\geq\ & (i+3)\left(\frac{u}{4} - (\vartheta-1)d\right) d(v', w')\,.
\end{aligned}
$$

This completes the induction. Plugging in $i = i_{\max}$ and noting that $\log b = \log\lceil 2\sigma\rceil \leq 1 + \log\lceil\sigma\rceil$, we get an execution in which two nodes at distance $b^0 = 1$ exhibit a skew of at least

$$
\begin{aligned}
(i_{\max}+2)\left(\frac{u}{4} - (\vartheta-1)d\right) &\geq \left(\frac{u}{4} - (\vartheta-1)d\right)(1 + \log_b D) \\
&\geq \left(\frac{u}{4} - (\vartheta-1)d\right)\log_{\lceil\sigma\rceil} D\,. \qquad \square
\end{aligned}
$$

Remarks:

- It is somewhat "bad form" to adapt Lemma 3.2 on the fly, as we did in the proof. However, the alternative of carefully defining partial executions, how to stitch them together, and proving indistinguishability results in this setting would mean to crack a nut with a sledgehammer.

- By making the base of the logarithm larger (i.e., making paths shorter more quickly), we can reduce the "loss" of skew in each step. Thus, we get a skew of $(u/2 - (\vartheta - 1)d - \varepsilon)$ per iteration, at the cost of reducing the number of iterations by a factor of $\log \sigma / (\log \sigma - \log \varepsilon^{-1})$.
- We can gain another factor of two by introducing skew more carefully. If we constract $\mathcal{E}_1$ so that messages "in direction of w" have delay (roughly) $d - u$ and messages "in direction of v" have delay d, we can hide u skew per hop, just like in Lemma 1.5. We favored the simpler construction to avoid additional bookkeeping.
- Overall, if $(\vartheta - 1)d \ll u$, $\sigma \gg 1$, and $\log_\sigma D \gg 1$, we can show a lower bound of $(u - \varepsilon) \log_\sigma D$ for some small $\varepsilon > 0$.
- Assuming a similar bunch of reasonable things and that $T \in \mathcal{O}(d)$ (i.e., message frequency is not the bottleneck in determining estimates), the asymptotically optimal choice of μ we computed in the exercises yields a skew of roughly $2u \log_\sigma D$ for our GCS algorithm. Thus, this lower bound shows that the algorithm is optimal up to a factor of roughly 2, provided $\sigma \gg 1$ and $(\vartheta - 1)d \ll u$. Dropping that $\sigma \gg 1$, we still get optimality up to a constant factor.
- So what of the case that $(\vartheta - 1)d$ is comparable to u or even larger? Recall that we have shown how to generate a better "logical hardware clock" in this case by bouncing messages back and forth between nodes. Using this idea (with some modifications and the occasional atrocity), one *could*, up to an additive $\mathcal{O}((\vartheta - 1)d)$, eliminate the dependence of the upper bound on $(\vartheta - 1)d$.
- As for a lower bound construction we can always pretend that clock drifts are actually smaller, e.g., $\vartheta' := \min\{\vartheta, 1 + u/(4d)\}$, the lower bound is asymptotically optimal in all cases...
- ...except for unbounded clock rates, which we will deal with next.

3.2 Lower Bound with Arbitrary Clock Rates

It can be shown that clock rates $l_v(t) \in \omega(1)$ do not help. That is, if $(\vartheta - 1)d < u/4$, we have that $\mathcal{L} \in \Omega(u \log_{1/(\vartheta-1)} D)$. However, the only (currently known) proof for this is tedious, to the point where it conveys little insight regarding what's going on. Hence, we will settle for a (much) simpler argument by Fan and Lynch showing a slightly weaker lower bound, followed by some intution as to why the stronger result is true as well.

We need a technical lemma stating that, provided that we leave some slack in terms of clock drifts and message delays, we can introduce $\Omega(u)$ hardware clock skew between any pair of neighbors in an indistinguishable manner. As this follows from repetition of previous arguments, we skip the proof.

Lemma 3.3. *Let $\mathcal{E}$ be any execution in which clock rates are at most $1+(\vartheta-1)/2$ and message delays are in the range $(d-3u/4, d-u/4)$. Then, for any $\{v, w\} \in E$ and sufficiently large times t, there is an indistinguishable execution $\mathcal{E}_v$ such that $L_v^{(\mathcal{E}_v)}(t) = L_v^{(\mathcal{E})}(t + u/4)$ and $L_w^{(\mathcal{E}_v)}(t) = L_w^{(\mathcal{E})}(t)$.*

Proof Sketch. The general idea is to use the remaining slack of $u/2$ to hide the additional skew, and the slack in the clock rates to introduce it. We can do this as slowly as needed, just as in the proof of Lemma 1.5. Again, we can choose the clock rates according to the function $d(x)$ defined in Lemma 3.2; as v and w are neighbors here, it can only take on values of -1, 0, or 1. □

This is all we need to generalize our lower bound to arbitrarily large logical clock rates.

Theorem 3.4. *Assume that $\vartheta \leq 2$. Any algorithm for the gradient clock synchronization problem with logical clock rates of at least 1 incurs a worst-case gradient skew of*

$$\mathcal{L} \in \Omega\left(\left(\frac{u}{4} - (\vartheta - 1)d\right) \log_{(\log D)/(\vartheta-1)} D\right).$$

Proof. Set $u' \coloneqq u/2$, $d' \coloneqq d - u/4$, and $\vartheta' \coloneqq 1 + (\vartheta - 1)/2$. We perform the exact same construction as in Theorem 3.1, with three modifications. First, u, d, and ϑ are replaced by u', d', and ϑ'. Second, before starting the construction, we wait for sufficiently long so that Lemma 3.3 is applicable to all times when we actually "work," i.e., we let the algorithm run for the required time with hardware clock rates of 1 and message delays of $d' - u'/2$. Third, we assume that $\mu = \log_{1/(\vartheta-1)} D$ in the construction; if ever we attempt to use this (assumed) bound on the clock rates in an inequality and it does not hold, the construction fails.

Now two things can happen. The first is that the construction succeeds. Note that we may assume that $u'/4 > (\vartheta' - 1)d'$, as otherwise $u/4 < (\vartheta - 1)d$, i.e., nothing is to show. Thus, the construction shows a lower bound of

$$\begin{aligned}\left(\frac{u'}{4} - (\vartheta' - 1)d'\right) \log_{\lceil\sigma\rceil} D &> \left(\frac{u}{8} - \frac{(\vartheta - 1)d}{2}\right) \log_{\lceil\mu/(\vartheta'-1)\rceil} D \\ &\in \Omega\left(\left(\frac{u}{4} - (\vartheta - 1)d\right) \log_{\mu/(\vartheta-1)} D\right).\end{aligned}$$

As

$$\begin{aligned}\log_{\mu/(\vartheta-1)} D &= \frac{\log D}{\log \mu - \log(\vartheta - 1)} \\ &= \frac{\log D}{\log(\log D - \log(\vartheta - 1)) - \log(\vartheta - 1)} \\ &\in \Omega\left(\frac{\log D}{\log\log D - \log(\vartheta - 1}\right) \\ &= \Omega\left(\log_{(\log D)/(\vartheta-1)} D\right),\end{aligned}$$

the claim follows in this case.

On the other hand, if the construction fails, there is an index $i < i_{\max}$ for which (3.1) does not hold — this is the only place where we make use of the fact that logical clocks do not run faster than rate μ. Thus,

$$L_w^{(\mathcal{E}^{(i)})}(t_{i+1}) - L_w^{(\mathcal{E}^{(i)})}(t_i) > \mu(t_{i+1} - t_i)$$

for some $i < i_{\max}$. Recall that in the construction, $d(v,w) = b^{i_{\max}-i} \geq b$ and

$$t_{i+1} - t_i = d + \left(\frac{u}{2(\vartheta-1)} - d\right)\frac{d(v,w)}{b} > \frac{u}{2(\vartheta-1)} - d > \frac{u}{4(\vartheta-1)} \geq \frac{u}{4}\,.$$

Hence, there must be a time $t \geq t_i$ so that

$$L_w^{(\mathcal{E}^{(i)})}\left(t + \frac{u}{4}\right) - L_w^{(\mathcal{E}^{(i)})}(t) > \frac{\mu u}{4}\,.$$

Let $x \in N_w$ be arbitrary. By Lemma 3.3, we can construct an execution $\mathcal{E}_w$ so that

$$L_w^{(\mathcal{E}_w)}(t) = L_w^{(\mathcal{E}^{(i)})}\left(t + \frac{u}{4}\right) > L_w^{(\mathcal{E}^{(i)})}(t) + \frac{\mu u}{4}$$

and $L_x^{(\mathcal{E}_w)}(t) = L_x^{(\mathcal{E}^{(i)})}(t)$. Thus, in at least one of the executions, the local skew exceeds

$$\frac{\mu u}{8} = \frac{u}{8}\log_{1/(\vartheta-1)} D\,. \qquad \square$$

We conclude this chapter with the promised intuition regarding the influence of D on the base of the logarithm. Consider a path of length k with a skew of exactly α per hop, for a total of αk between its endpoints. Now suppose that an algorithm cleverly uses a large logical clock rate, perfectly reducing the skew at the same rate between any pair of neighbors. Consider the point in time when the skew has been reduced to, say, $\alpha - u/8$ per hop. The node in the middle of the path has increased its logical clock at half the rate of the endpoint that's catching up — and the nodes in between have been even faster! Denoting this rate by r, slipping in hardware clock skew at rate $\vartheta - 1$ means adding logical clock skew at rate at least $r(\vartheta-1)/2$. So, even if it takes factor r less time to reduce the skew to, say $\alpha - u/8$ per hop than it would for $\mu = 1$, it also takes factor $r/2$ less time to build up additional skew. We would end up with the same result!

Remarks:

- Unfortunately, molding this idea into a proof is challenging, and the result is not pretty.
- The D in the base of the logarithm is of little importance unless clocks are of poor quality. A standard quartz oscillator guarantees that $\vartheta - 1 \leq 10^{-5}$. Even a gigantic diameter of 10^5 would not affect the bound by more than a factor 2 for such clocks!
- The assumption that $\vartheta \leq 2$ in Theorem 3.4 is an artifact of the proof. However, hardware clocks that are this inaccurate hardly deserve the name "clock," so this corner case is not of interest.
- Overall, the GCS algorithm from the previous lecture appears to be optimal or very close to optimal for essentially all choices of parameters.
- Don't fall into the trap of forgetting that relaxing the model enables better solutions! For instance, if it is not important that clocks make progress at all times (or most of the time), constant local skew can be achieved (buzzword: α-synchronizer)!

Bibliographic Notes

There is not much to add to the notes for the previous lecture. The seminal paper by Fan and Lynch [FL06] introducing the problem provided Theorem 3.4. Meier and Thiele show that essentially the same lower bound arises from bounded communication rates, without uncertainty (i.e., $u = 0$) [**?**]. Theorem 3.1 follows [LLW10], which also tightens the lower bound for unbounded clock rates by removing the D from the base of the logarithm. In the dynamic setting, one can show bounds on how quickly an edge can be incorporated into the subgraph of edges that satisfy the skew bounds, and asymptotic optimality can be achieved simultaneously with other guarantees [KLO11, KLLO10].

Bibliography

[FL06] Rui Fan and Nancy Lynch. Gradient Clock Synchronization. *Distributed Computing*, 18(4):255–266, 2006.

[KLLO10] Fabian Kuhn, Christoph Lenzen, Thomas Locher, and Rotem Oshman. Optimal Gradient Clock Synchronization in Dynamic Networks. *CoRR*, abs/1005.2894, 2010.

[KLO11] Fabian Kuhn, Thomas Locher, and Rotem Oshman. Gradient Clock Synchronization in Dynamic Networks. *Theory Comput. Syst.*, 49(4):781–816, 2011.

[LLW10] Christoph Lenzen, Thomas Locher, and Roger Wattenhofer. Tight Bounds for Clock Synchronization. *J. ACM*, 57(2):8:1–8:42, 2010.