8 Gradient Clock Synchronization

Chapter Contents

Learning Goals

todo

8.1 Overview

In Chapter 7, we have studied the clock synchronization problem. We have see that a global skew of $\Theta(uD + (\vartheta - 1)d)$ is worst-case optimal, where D is the diameter of the network graph (with crashed nodes removed). However, if the logical clocks are intended to clock the system, the global skew is not determining the clock frequency the system can sustain under synchronous operation. Rather, the essential property is the clock skew between nodes (i.e., clock domains) that communicate with each other.

In this chapter, we will assume that the network graph corresponds to the communication graph, i.e., if the clock domains corresponding to nodes $v \in V$ and $w \in V$ communicate directly, then there is also an edge $\{v, w\} \in E$ in the network graph $G = (V, E)$ on which we solve the clock synchronization problem. In this setting, a highly relevant quality measure for synchronization is the *local skew.*

Definition 8.1 (Local Skew)**.** *Given an algorithm that computes logical clocks* $L_v(t)$*,* $t \in \mathbb{R}_{\geq 0}$*, at each node* $v \in V$*, define its* local skew *as*

$$\mathcal{L} := \sup_{t \in \mathbb{R}_{\geq 0}} \{\mathcal{L}(t)\},$$

over all executions $\mathcal{E}$*, where*

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

We study the local skew for clock synchronization algorithms in TMP.

One might hope that the local skew can be kept much smaller than the global skew. In fact, since the lower bound on the global skew given in Theorem 7.12 is based on "hiding" a large clock skew between nodes in distance D from each other, one might venture the guess that a local skew of $O(u + (\vartheta - 1)d)$ can be guarenteed. Our first main result in this chapter shows that such an ideal distribution of the global skew over the network cannot always be maintained with bounded logical clock rates.

Theorem 8.3. *Any clock synchronization algorithm satisfying that*

$$\frac{dH_v}{dt}(t) \leq \frac{dL_v}{dt}(t) \leq (1 + \mu)\frac{dH_v}{dt}(t)$$

for all nodes v *and times* t *has*

$$\mathcal{L} \geq \left(\frac{u}{4} - (\vartheta - 1)d\right) \log_{\lceil \sigma \rceil} D,$$

where $\sigma := \mu/(\vartheta - 1)$*.*

This lower bound constraints clock rates from below by $\frac{dH_v}{dt}(t) \geq 1$ and from above by $(1+\mu)\frac{dH_v}{dt}(t) \leq \vartheta(1+\mu)$. Recall that the lower bound is motivated by requiring that clocks make progress; however, as can be seen by analyzing a variant of Algorithm 2 in TMP, allowing for an *amortized* logical clock rate[10] of at least 1 enables us to keep the local skew constant.

Theorem 8.7. *There is a clock synchronization algorithm achieving a local skew of* $\max\{d, \vartheta u\}$*, amortized* 1*-progress with* $C = 0$*, and* $\frac{dL_v}{dt}(t) \leq \frac{dH_v}{dt}(t)$ *for all times* t *and nodes* $v \in V$.

The algorithm providing these guarantees locally halts the logical clock for up to uD time. Although a formal statement would be more convoluted,[11] the proof of Theorem 8.3 reveals that, roughly speaking, this is necessary to ensure a local skew of $O(u)$.

But what if halting or dramatically slowing down the clocks is a problem? If the system is required to respond to local events as quickly as possible, one would want the the logical clocks that drive the nodes' computations are guaranteed to make progress at all times. In this setting, the requirement that $\frac{dL_v}{dt}(t) \geq \frac{dH_v}{dt}(t)$ should be upheld. So what about the upper bound, i.e., that $\frac{dL_v}{dt}(t) \leq (1+\mu)\frac{dH_v}{dt}(t)$? It turns out that choosing $\mu \geq \log_{1/(\vartheta-1)} D$ is of no use, as an algorithm utilizing faster clocks will inadvertantely introduce a large skew due to neighors' not being able to keep track of each others clocks any more.

Theorem 8.9. *Any clock synchronization algorithm satisfying that* $\frac{dH_v}{dt}(t) \leq \frac{dL_v}{dt}(t)$ *for all nodes* v *and times* t *has*

$$\mathcal{L} = \Omega\left(\left(\frac{u}{4} - (\vartheta-1)d\right)\log_{\lceil\sigma\rceil} D\right)$$

for $\sigma = \log_{1/(\vartheta-1)} D)/(\vartheta-1)$.

A painfully more elaborate argument shows that the same holds for $\sigma = \Theta(1/(\vartheta-1))$ [**?**], but due to being executed in a different model, it does not immediately provide a corresponding statement in our setting. In favor of simplicity and intuition, we stick to the weaker bound.

Note that for $(1+\mu) \leq \vartheta$, it is impossible for logical clocks of nodes with $\frac{dH_v}{dt} = 1$ to catch up with the logical clocks of nodes with $\frac{dH_v}{dt} = \vartheta$. Thus, the above results lead to the question whether for the range of $\mu > \vartheta - 1$ the lower

[10] The theorem states amortized 1-progress for $C = 0$, but recall that for notational convenience we assume that all nodes wake up at time 0. If the last node wakes up at time t_0, then $C = t_0$.

[11] The clocks do not need to be halted, but they must progress sufficiently slow to prevent the build-up of skew the lower bound accomplishes.

bound from Theorem 8.3 can be (up to constants) matched by a corresponding algorithm. The second main result of this chapter is that this is indeed the case, provided that nodes can estimate the logical clock values of their neighbors up to an error of $\delta = O(u)$ at all times and $\mathcal{G}/\delta = O(D)$.

Theorem ??. *Suppose that $\kappa > \delta$ and $H_v(0) - H_w(0) \leq \kappa$ for all edges $\{v, w\} \in E$. Then* **??** *maintains a local skew of*

$$\mathcal{L} \leq 2\kappa \left\lceil \log_\sigma \frac{\mathcal{G}}{\kappa} \right\rceil,$$

where $\sigma := \mu/(\vartheta - 1)$.

This means that, while we are not able to guarantee a local skew that is entirely independent of D, the dependence on D is only logarithmic. Moreover, not that the base of the logarithm can become very large, at the expense of a larger "drift" of logical clocks than of hardware clocks.

We then proceed to show that $\delta = O(u)$ is easily achieved, provided that $(\vartheta - 1)d = O(u)$, $\mu = O(u/d)$, and that **??** guarantees a global skew of $?\kappa D$. Together, under the mild constraint that $u/4 - (\vartheta - 1)d = \Omega(u)$ this implies that **??** is, up to constant factors, simultaneously optimal with respect to both local and global skew, for any choice of $\vartheta - 1 < \mu = O(u/d)$.

Corollary 8.2. *Suppose that $(\vartheta - 1)d = O(u)$, $\vartheta - 1 < \mu = O(u/d)$, and $H_v(0) - H_w(0) \in O(u)$ for all edges $\{v, w\} \in E$. Then we can guarantee that*

- $\frac{dH_v}{dt}(t) \leq \frac{dL_v}{dt}(t) \leq (1 + \mu)\frac{dH_v}{dt}(t)$ *for all nodes v and times t,*
- $\mathcal{G} = O(uD)$, *and*
- $\mathcal{L} = O(u \log_\sigma D)$, *where $\sigma = \mu/(\vartheta - 1)$.*

8.2 Lower Bound on the Local Skew with Bounded Clock Rates

In Chapter 7, we proved essentially matching upper and lower bounds on the worst-case global skew for the clock synchronization problem. We saw that during an execution of the Max algorithm (Algorithm 5), all logical clocks in all executions eventually agree up to an additive term of $O(uD)$ (ignoring other parameters). The lower bound we proved in Theorem 7.12 shows that a global skew of $\Omega(uD)$ is unavoidable for any algorithm in which clocks run at an amortized constant rate, at least in the worst case. In our lower bound construction, the two nodes v and w that achieved the maximal skew were distance D apart. However, the lower bound did not preclude neighboring nodes from remaining closely synchronized throughout an execution. As we will see in Theorem 8.7, this is indeed possible if one is willing to slow down clocks arbitrarily (or simply stop them), even if the amortized rate is constant.

We now look into what happens if one requires that clocks progress at a constant rate at all times. That is, we constrain logical clocks to increase at rates between $\frac{dH}{dt}$ and $(1+\mu)\frac{dH}{dt}$ at all times.

Before proving Theorem 8.3, we provide some intuition. Assume that $(\vartheta-1)d \ll u$, so we can ignore terms with $(\vartheta-1)d$ for the moment and drop them from the notation. The basic strategy of the proof is to construct a sequence of executions $\mathcal{E}_0, \mathcal{E}_1, \ldots, \mathcal{E}_\ell$ and times $t_0 < t_1 < \cdots < t_\ell$ such that at each time t_i, there exist nodes v_i, w_i satisfying $L_{v_i}(t_i) - L_{w_i}(t_i) \geq i\alpha u \cdot \operatorname{dist}(v_i, w_i)$, for some suitable constant α. Our construction works up to $\ell = \Omega(\log_\sigma D)$ with $\operatorname{dist}(v_\ell, w_\ell) = 1$, which gives the desired result.

In more detail, the idea of the proof is to use the "shifting" technique of Theorem 7.12 applied ℓ times to closer and closer pairs of nodes. By Theorem 7.12, there is an execution $\mathcal{E}_0$ and a pair of nodes v_0, w_0 satisfying $\operatorname{dist}(v_0, w_0) = D$ such that at time $t_0 = d + (u/(2(\vartheta-1)) - d)D$, we have $L_{v_0}(t_0) - L_{w_0}(t_0) \gtrapprox u/2 \cdot \operatorname{dist}(v_0, w_0)$. Fix a shortest path P from v_0 to w_0. For any pair of nodes v, w along P, we define the *average skew* between v and w at time t to be $|L_v(t) - L_w(t)|/\operatorname{dist}(v, w)$. In particular, the average skew between v_0 and w_0 is at least (roughly) $u/2$.

We extend the execution $\mathcal{E}_0$ for $t > t_0$ by setting all hardware clock rates to 1 for $t > t_0$ and all message delays to $d - u/2$ (as in the execution $\mathcal{E}$ in Theorem 7.12). Thus, by assumption at times $t > t_0$ logical clock rates are always between $\frac{dH}{dt} = 1$ and $(1+\mu)\frac{dH}{dt} = 1+\mu$. Hence, for every $t > t_0$ in the extended execution, we have $L_{v_0}(t) - L_{w_0}(t) \geq u/2 \cdot \operatorname{dist}(v_0, w_0) - \mu \cdot (t - t_0)$. That is, the average skew between v_0 and w_0 decreases at a rate of at most $\mu/\operatorname{dist}(v_0, w_0)$. By taking

$$t_1 = t_0 + d + \left(\frac{u}{2} \cdot (\vartheta - 1) - d\right) \cdot k \approx t_0 + \frac{u}{2} \cdot (\vartheta - 1)$$

for some suitably chosen k, there exists a pair of nodes v_1, w_1 on P with $\operatorname{dist}(v_1, w_1) = k$ such that the average skew between v_1 and w_1 at time t_1 is (roughly) at least

$$\frac{u}{2} - \frac{\mu}{\operatorname{dist}(v_0, w_0)} \cdot (t_1 - t_0) = \frac{u}{2} - \frac{u}{2} \cdot \frac{\mu}{\vartheta - 1} \cdot \frac{k}{\operatorname{dist}(v_0, w_0)}$$

in the execution $\mathcal{E}_0$. Recalling that $\sigma = \mu/(\vartheta - 1)$ and choosing $k = \operatorname{dist}(v_0, w_0)/\lceil 2\sigma \rceil$, this is at least $u/4$. We then apply the shifting technique again to the nodes v_1 and w_1 on the interval $[t_0, t_1]$. In this way we define an execution $\mathcal{E}_1$ in which there is a time when the skew between v_1 and w_1 is by roughly $uk/2$ larger than the skew in $\mathcal{E}_0$ at time t_1. Therefore, in $\mathcal{E}_1$, the average skew bewtween v_1 and w_1 reaches about $3/4u$.

We then iterate the procedure above $\ell\lfloor\log_{\lceil 2\sigma\rceil} D\rfloor$ times. In the i-th iteration, we obtain a pair of nodes v_i, w_i at distance $D/\lfloor 2\sigma\rfloor^i$ such that the average skew between v_i and w_i is at least $(1/2 + i/4)\cdot u$. Thus, after ℓ iterations, the skew between adjacent nodes v_ℓ and w_ℓ is roughly $u/2 \cdot \log_\sigma D$, which gives the desired result.

Theorem 8.3. *Any clock synchronization algorithm satisfying that*

$$\frac{dH_v}{dt}(t) \le \frac{dL_v}{dt}(t) \le (1+\mu)\frac{dH_v}{dt}(t)$$

for all nodes v and times t has

$$\mathcal{L} \ge \left(\frac{u}{4} - (\vartheta - 1)d\right)\log_{\lceil\sigma\rceil} D,$$

where $\sigma := \mu/(\vartheta - 1)$.

Proof. Note that the claim is vacuous if $(\vartheta-1)d \ge u/4$, so we can assume the opposite in the following. Set $b := \lceil 2\sigma\rceil$ and $i_{\max} := \lfloor\log_b D\rfloor$. By induction over $i \in [i_{\max}+1]$, we show that we can build up a skew of $(i+2)(u/4 - (\vartheta - 1)d)\,\mathrm{dist}(v,w)$ between nodes $v, w \in V$ in distance $\mathrm{dist}(v,w) = b^{i_{\max}-i}$ 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 **??**, choosing t_0 as in the lemma. We pick two nodes $v, w \in V$ in distance $b^{i_{\max}} \le D$ of each other such that $L_v^{(\mathcal{E}_1)}(t_0) \ge L_w^{(\mathcal{E}_1)}(t_0)$. Now consider $\mathcal{E}_v$ for this choice of $v, w \in V$, which satisfies $H_v^{(\mathcal{E}_v)}(t_0) = H_v^{(\mathcal{E}_1)}(t_0) + (u/2 - (\vartheta-1)d)\,\mathrm{dist}(v,w)$ and $H_w^{(\mathcal{E}_v)}(t_0) = H_w^{(\mathcal{E}_1)}(t_0)$. Denote by $t < t_0$ the time such that $H_v^{(\mathcal{E}_v)}(t) = H_v^{(\mathcal{E}_1)}(t_0)$. We get that

$$\begin{aligned}
L_v^{(\mathcal{E}_v)}(t_0) &= L_v^{(\mathcal{E}_v)}(t) + L_v^{(\mathcal{E}_v)}(t_0) - L_v^{(\mathcal{E}_v)}(t) && \text{adding } 0\\
&\ge L_v^{(\mathcal{E}_v)}(t) + H_v^{(\mathcal{E}_v)}(t_0) - H_v^{(\mathcal{E}_v)}(t) && \tfrac{dL_v}{dt} \ge \tfrac{dH_v}{dt}\\
&= L_v^{(\mathcal{E}_v)}(t) + \left(\frac{u}{2} - (\vartheta-1)d\right)\mathrm{dist}(v,w) && \text{by definition}\\
&= L_v^{(\mathcal{E}_1)}(t_0) + \left(\frac{u}{2} - (\vartheta-1)d\right)\mathrm{dist}(v,w). \qquad (8.1) && \text{by indist.}
\end{aligned}$$

We conclude that

$$\begin{aligned}
L_v^{(\mathcal{E}_v)}(t_0) - L_w^{(\mathcal{E}_v)}(t_0) &= L_v^{(\mathcal{E}_v)}(t_0) - L_w^{(\mathcal{E}_1)}(t_0) && \text{by indist.}\\
&\ge L_v^{(\mathcal{E}_1)}(t_0) + \left(\frac{u}{2} - (\vartheta-1)d\right)\mathrm{dist}(v,w) - L_w^{(\mathcal{E}_1)}(t_0) \qquad (8.2) && (8.1)\\
&\ge \left(\frac{u}{2} - (\vartheta-1)d\right)\mathrm{dist}(v,w). && \text{choice of } v, w
\end{aligned}$$

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.,

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

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 **??** 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, analogously to the lemma we show the following. For any $v', w' \in V$, we can construct an execution $\mathcal{E}_{v'}$ indistinguishable from $\mathcal{E}^{(i)}$, such that

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

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

Recall that $\frac{dL_v}{dt}(t) \geq \frac{dH_v}{dt}(t) \geq 1$ and $\frac{dL_w}{dt}(t) \leq (1+\mu)\frac{dH_w}{dt}(t)$ at all times t. As $\frac{dH_w^{(\mathcal{E}^{(i)})}}{dt}(t) = 1$ at times $t \geq t_i$, we obtain

$$L_v^{(\mathcal{E}^{(i)})}(t_{i+1}) - L_w^{(\mathcal{E}^{(i)})}(t_{i+1}) \geq L_v^{(\mathcal{E}^{(i)})}(t_i) - L_w^{(\mathcal{E}^{(i)})}(t_i) - \mu(t_{i+1} - t_i)\,. \quad (8.3)$$

Recall that $\operatorname{dist}(v, w) = b^{i_{\max} - i}$ 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 $\operatorname{dist}(v', w') = b^{i_{\max} - (i+1)} = \operatorname{dist}(v, w)/b$

so that

$$\begin{aligned}
& L_{v'}^{(\mathcal{E}^{(i)})}(t_{i+1}) - L_{w'}^{(\mathcal{E}^{(i)})}(t_{i+1}) \\
\geq\ & \frac{L_{v}^{(\mathcal{E}^{(i)})}(t_{i+1}) - L_{w}^{(\mathcal{E}^{(i)})}(t_{i+1})}{b} && \text{pidgeon hole} \\
\geq\ & \frac{L_{v}^{(\mathcal{E}^{(i)})}(t_{i}) - L_{w}^{(\mathcal{E}^{(i)})}(t_{i}) - \mu(t_{i+1}-t_i)}{b} && (8.3) \\
=\ & \frac{L_{v}^{(\mathcal{E}^{(i)})}(t_{i}) - L_{w}^{(\mathcal{E}^{(i)})}(t_{i}) - \mu(d + (u/(2(\vartheta-1)) - d)\operatorname{dist}(v',w'))}{b} && \text{by definition} \\
\geq\ & \frac{L_{v}^{(\mathcal{E}^{(i)})}(t_{i}) - L_{w}^{(\mathcal{E}^{(i)})}(t_{i}) - \mu u \operatorname{dist}(v',w')/(2(\vartheta-1))}{b} && \operatorname{dist}(v',w') \geq 1 \\
\geq\ & \frac{L_{v}^{(\mathcal{E}^{(i)})}(t_{i}) - L_{w}^{(\mathcal{E}^{(i)})}(t_{i})}{b} - \frac{\mu}{2\sigma(\vartheta-1)} \cdot \frac{u}{2} \cdot \operatorname{dist}(v',w') && b = \lceil 2\sigma \rceil \\
=\ & \frac{L_{v}^{(\mathcal{E}^{(i)})}(t_{i}) - L_{w}^{(\mathcal{E}^{(i)})}(t_{i})}{b} - \frac{u}{4} \cdot \operatorname{dist}(v',w') && \sigma = \mu/(\vartheta-1) \\
\geq\ & \frac{(i+2)(u/4 - (\vartheta-1)d)\operatorname{dist}(v,w)}{b} - \frac{u}{4} \cdot \operatorname{dist}(v',w') && \text{induction hyp.} \\
=\ & \left((i+2)\left(\frac{u}{4} - (\vartheta-1)d\right) - \frac{u}{4}\right)\operatorname{dist}(v',w'). \quad (8.4) && \operatorname{dist}(v',w') = \operatorname{dist}(v,w)/b
\end{aligned}$$

In other words, as the average skew on a shortest path from v to w did not decrease by more than $u/4$, there must be a subpath of length $\operatorname{dist}(v,w)/b$ 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 $\mathcal{E}_{v'}$. By an analogous derivation to that of (8.2), we get that

$$\begin{aligned}
& L_{v'}^{(\mathcal{E}_v)}(t_{i+1}) - L_{w'}^{(\mathcal{E}_v)}(t_{i+1}) \\
\geq\ & L_{v'}^{(\mathcal{E}^{(i)})}(t_{i+1}) + \left(\frac{u}{2} - (\vartheta-1)d\right)\operatorname{dist}(v',w') - L_{w'}^{(\mathcal{E}^{(i)})}(t_{i+1}) && \text{anal. to } (8.2) \\
\geq\ & (i+3)\left(\frac{u}{4} - (\vartheta-1)d\right)\operatorname{dist}(v',w'). && (8.4)
\end{aligned}$$

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

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

- It is somewhat "bad form" to adapt Lemma **??** 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 $\log \sigma / (\log \sigma - \log \varepsilon^{-1})$. As typically $\sigma \gg 1$, this means that we gain roughly a factor of 2.
- We can gain another factor of 2 by introducing skew more carefully. If we construct $\mathcal{E}_1$ 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. 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 $\varepsilon > 0$.
- What if $(\vartheta - 1)d$ is comparable to u or even larger? 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 does not get weaker if the hardware clocks get worse. On the other hand, we will see that larger ϑ is not really an issue (up to a "one-time" additive term of $O((\vartheta - 1)d)$), as we can then bounce messages back and forth between nodes to keep track of time with greater accuracy than the "base clocks" permit.

8.3 Constant Local Skew with Halting Clocks

From Theorem 8.3, we know that we cannot concurrently have

- $\frac{dL_v}{dt}(t) \geq \frac{dH_v}{dt}(t)$ for all v and t,
- $\frac{dL_v}{dt}(t) \leq (1 + \mu)\frac{dH_v}{dt}(t)$ for all v and t, and
- $\mathcal{L} \leq f(d, u, \mu, \vartheta)$ for some function f, i.e., a local skew that does not depend on the network size.

In the next sections, we will address these points one by one.

In this section, we start with the first point. It entails that all logical clocks increase at rate at least 1 at all times. We now show how the other two requirements can be satisfied, by relaxing the first one to *amortized* 1-progress (see Definition 7.10, i.e., we demand that for each execution there is some $C \in \mathbb{R}_{\geq 0}$ such that for all $t' \geq t$ and all $v \in V$ it holds that

$$L_v(t') - L_v(t) \geq t' - t - C.$$

In order to prove this claim, we analyze Algorithm 6, which can be viewed as a TMP version of Algorithm 2 that maintains logical clocks. First, we prove amortized 1-progress.

Algorithm 6 This clock synchronization algorithm can be viewed as a TMP variant of Algorithm 2; replacing the "tick" messages by the messages of some synchronous algorithm (labeled by round number), this algorithm could be simulated.

```
if t = 0 (i.e., v just woke up) then
    ℓ ← 0                                          ▷ initialize L_v(0) to 0
    h ← getH()
    r ← 1                                          ▷ logical clock is initially running
    for each p ∈ {1, . . . , deg(v) do
        COUNT_p ← 0
        send "tick" to p
    end for
end if
if r = 1 and getH() − h = d then
    ℓ ← ℓ + d                                      ▷ memorize progress
    h ← getH()
    for p ∈ {1, . . . , deg(v)} do
        send "tick" on p                           ▷ inform neighbors on progress
        COUNT_p ← COUNT_p − 1                      ▷ consume neighbors' ticks
        if COUNT_p = −1 then                       ▷ made d progress since last tick on p
            r ← 0                                  ▷ stop clock until tick on p arrives
        end if
    end for
end if
if received "tick" on port p then
    COUNT_p ← COUNT_p + 1
    if r = 0 and COUNT_q ≥ 0 for all q ∈ {1, . . . , deg(v) then
        h ← getH()
        r ← 1                                      ▷ restart clock
    end if
end if
procedure getL()                                   ▷ returns L_v(t)
    if r = 1 then
        return ℓ + getH() − h                      ▷ logical clock increases at rate dH_v/dt
    else
        return ℓ                                   ▷ logical clock is halted
    end if
end procedure
```

Lemma 8.4. *In graphs of diameter D, Algorithm 6 satisfies amortized* 1-*progress with $C = 0$.*

E8.1 Prove the lemma. Hint: Review the proof of Theorem 6.18 and recall that we assume that all nodes wake up at time 0.

Lemma 8.5. *Algorithm 6 satisfies for each $v \in V$ that L_v is continuous and that $\frac{dL_v}{dt}(t) \leq \frac{dH_v}{dt}(t)$ at all times t when r_v does not change.*

E8.2 Prove the lemma.

Lemma 8.6. *Algorithm 6 satisfies $\mathcal{L} \leq \max\{d, \vartheta u\}$.*

Proof. Fix neighbors $v, w \in V$ and a time t. W.l.o.g., assume that $L_v(t) \geq L_w(t)$ (otherwise, flip v and w). Let i be the number of tick messages v has sent by time t, and j the number of tick messages from w it has received. By a simple induction, we have that $j \geq i - 1$ (for every "tick" after its first, v waits for a "tick" from w), $(i-1)d \leq L_v(t) \leq id$, and $L_w(t) \geq (j-1)d$ (w must reach this clock value to send j "tick" messages). If $j \geq i$ or $i = 1$, we have that $L_v(t) - L_w(t) \leq d$.

Thus, it remains to consider the case that $j = i - 1$ for $i \geq 2$. Denonte by t_r the time when v received the j-th "tick" message from w and by t_s the time it was sent. Observe that $L_w(t_s) = (j-1)d$ and $L_v(t_r) \leq (i-d)$. Because w sets r to 1 when sending the message and does not set r back to 0 before d time has passed on its hardware clock, we have that

$$\begin{aligned}
\text{Line 30} \qquad L_w(t) &\geq L_w(t_s) + \min\{d, H_w(t) - H_w(t_s)\} \\
&= (j-1)d + \min\{d, H_w(t) - H_w(t_s)\} \\
j = i - 1 \qquad &= (i-2)d + \min\{d, H_w(t) - H_w(t_s)\} \\
\frac{dH_w}{dt} \geq 1 \qquad &\geq (i-2)d + \min\{d, t - t_s\} \qquad (8.5)
\end{aligned}$$

If $t - t_s \geq d$, we again get that $L_w(t) \geq (i-1)d$ and hence $L_v(t) - L_w(t) \leq d$, so assume that $t - t_s < d$. Then

$$\begin{aligned}
L_v(t) - L_w(t) &= L_v(t_r) + L_v(t) - L_v(t_r) - L_w(t) && \text{adding 0}\\
&\leq (i-1)d + L_v(t) - L_v(t_r) - L_w(t)\\
&\leq (i-1)d + H_v(t) - H_v(t_r) - L_w(t) && \text{Lemma 8.5}\\
&\leq (i-1)d + \vartheta(t - t_r) - L_w(t) && \tfrac{dH_v}{dt} \leq \vartheta\\
&\leq d + (\vartheta - 1)(t - t_r) - (t_r - t_s) && t - t_s < d, (8.5)\\
&\leq d + (\vartheta - 1)(t - t_s) - \vartheta(d - u) && t_r - t_s \geq d - u\\
&< \vartheta u. \qquad \square && t - t_s < d
\end{aligned}$$

Theorem 8.7. *There is a clock synchronization algorithm achieving a local skew of* $\max\{d, \vartheta u\}$*, amortized* 1*-progress with* $C = 0$*, and* $\frac{dL_v}{dt}(t) \leq \frac{dH_v}{dt}(t)$ *for all times* t *and nodes* $v \in V$.

Proof. Follows from Lemmas 8.4 to 8.6. □

E8.3 Show that a node may stop its logical clock (i.e., continuously have $r_v = 0$) for uD time. Hint: Maximize the global skew while all message delays are $u - d$. Then have a chain of messages starting at the node that is most behind all have delay d.

E8.4 Show that this is the worst case, i.e., no node halts its logical clock for more than uD time. Hint: Consider the time when some node halts its clock after generating tick i and argue that nodes in distance r have generated their tick $i - r$ at the latest $r(d - u)$ time earlier. This requires to use that all nodes wake up at time 0 (otherwise it holds only for sufficiently large times).

8.4 Lower Bound with Arbitrary Clock Rates

We will now show that clock rates $\frac{dL_v}{dt}(t) \in \omega(\log_{1/(\vartheta-1)} D)$ do not help. That is, if $(\vartheta - 1)d < u/4$, we have that $\mathcal{L} \in \Omega(u \log_{(\log_{1/(\vartheta-1)} D)/(\vartheta-1)} D)$.

To this end, 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 8.8. *Let* $\mathcal{E}$ *be any execution in which hardware 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 **??**. Again, we can choose the clock rates according to the function $d(x)$ defined in Lemma **??**; 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 8.9. *Any clock synchronization algorithm satisfying that* $\frac{dH_v}{dt}(t) \leq \frac{dL_v}{dt}(t)$ *for all nodes* v *and times* t *has*

$$\mathcal{L} = \Omega\left(\left(\frac{u}{4} - (\vartheta - 1)d\right)\log_{\lceil\sigma\rceil} D\right)$$

for $\sigma = \log_{1/(\vartheta-1)} D)/(\vartheta - 1)$.

Proof. Set $u' := u/2$, $d' := d - u/4$, and $\vartheta' := 1 + (\vartheta - 1)/2$. We perform the exact same construction as in Theorem 8.3, 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 8.8 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, resulting in the base of the logarithm being $\sigma' = 2\mu/(\vartheta - 1) = \Theta(\sigma)$; if we ever 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. Hence, the construction shows a lower bound of

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

i.e., the claim follows in this case.

On the other hand, if the construction fails, there is an index $i < i_{\max}$ for which (8.3) 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, $\text{dist}(v, w) = b^{i_{\max}-i} \geq b$ and

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

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 be an arbitrary neighbor of w. By Lemma 8.8, 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

$$\frac{\mu u}{8} = \frac{u}{8}\log_{1/(\vartheta-1)} D > \frac{u}{8}\log_{\mu/(\vartheta-1)} D = \Omega\left(\left(\frac{u}{4} - (\vartheta - 1)d\right)\log_{\lceil\sigma\rceil} D\right). \quad \square$$

Bibliography