Gradient Clock Synchronization with Practically Constant Local Skew

Christoph Lenzen
CISPA Helmholtz Center for Information Security
Saarbruecken, Germany
Reykjavik University
Reykjavik, Iceland
christophlen@ru.is

Abstract

Gradient Clock Synchronization (GCS) is the task of minimizing the *local skew,* i.e., the clock offset between neighboring clocks, in a larger network. While asymptotically optimal bounds are known, from a practical perspective they have crucial shortcomings:

- Local skew bounds are determined by upper bounds on offset estimation that need to be guaranteed throughout the entire lifetime of the system.
- Worst-case frequency deviations of local oscillators from their nominal rate are assumed, yet frequencies tend to be much more stable in the (relevant) short term.

State-of-the-art deployed synchronization methods adapt to the true offset measurement and frequency errors, but achieve no non-trivial guarantees on the local skew.

In this work, we provide a refined model and novel analysis of existing techniques for solving GCS in this model. By requiring only *stability* of measurement and frequency errors, we can circumvent existing lower bounds, leading to dramatic improvements under very general conditions. For example, if links exhibit a uniform worst-case estimation error of Δ and a *change* in estimation errors of $\delta \ll \Delta$ on relevant time scales, we bound the local skew by $\mathcal{O}(\Delta + \delta \log D)$ for networks of diameter D, effectively "breaking" the established $\Omega(\Delta \log D)$ lower bound, which holds when $\delta = \Delta$. Surprisingly, these results require only very limited knowledge of Δ. In particular, the likely dominant term of $\mathcal{O}(\Delta)$ in the local skew is determined by the *actual* link performance, not an upper bound covering worst-case conditions.

The full version of this paper [17] also shows how to achieve full self-stabilization, perform external synchronization, and limit the influence of local oscillators on δ to scale with the *change* of frequency of an individual oscillator on relevant time scales.

CCS Concepts

• **Theory of computation** → **Distributed algorithms**; • **Computer systems organization** → *Real-time systems*; • **Hardware** → Very large scale integration design.

PODC '26, Egham, United Kingdom

ACM ISBN 979-8-4007-2512-8/26/07
https://doi.org/10.1145/3796701.3815957

Keywords

bounded delay and drift, adaptive skew, hardware synchronization

ACM Reference Format:
Christoph Lenzen. 2026. Gradient Clock Synchronization with Practically Constant Local Skew. In *ACM Symposium on Principles of Distributed Computing (PODC '26), July 6–10, 2026, Egham, United Kingdom.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3796701.3815957

1 Introduction and Related Work

Clock synchronization, the task of minimizing the offset—also known as *skew*—between clocks in a distributed network, has been studied extensively. Being a distributed task by nature, it drew the attention of researchers in distributed computing early on, see e.g. [2, 9, 19, 23]. Biaz and Welch established that in a network of diameter D, the worst-case skew that can be guaranteed between all pairs of nodes—called *global skew*—grows linearly with D [3].

One of the main benefits of matching lower bound constructions is that they do not only prove what the "best" bound is, but also tend to shed light on the true obstacle to better performance. In this case, the issue is that nodes at a great distance from each other have no way of getting accurate information on each other's current clock value. However, many applications require only the clocks of nearby nodes to be well-synchronized. In their seminal work from 2004 [10], Fan and Lynch focused on this aspect, introducing the problem of Gradient Clock Synchronization (GCS). GCS asks to minimize the worst-case skew between neighbors, on which the lower bound construction from [3] has no non-trivial implication. The key insight provided by Fan and Lynch was that, nonetheless, this so-called *local skew* must grow as $\Omega(\log D/ \log\log D)$. They showed this based on unknown variations in message delivery and processing times, but essentially the same argument can be made based on unknown frequency variations in local oscillators [20].

Follow-up work then pinpointed the achievable local skew to $\Theta(\log D)$ [18], extended the results to non-uniform [16] and dynamic networks [14, 15], and showed how to add tolerance to Byzantine faults [7]. Moreover, it has been demonstrated that hardware implementations of the algorithm are feasible and promising [5, 6], strongly supporting that practical application of the algorithmic technique is of interest. However, all of these works assume a known (and thus worst-case!) bound Δ on the clock offset estimation error across each edge, and let node v increase its clock value to match the largest one it perceives in its neighborhood, so long as this does not cause a perceived offset of more than Δ to any of its neighbors. This results in skews of at least Δ, even if the *actual* measurement error is much smaller.

In contrast, "naively" synchronizing via a BFS tree rooted at r by tracking the parent's clock value would result in a global skew of at most $2\max_{v\in V}\{|d(r,v)|\}$, where $d(r,v)$ is the (possibly negative) distance induced by labeling each tree edge with the actual measurement error. Loosely speaking, this is in fact what deployed protocols like the Network Time Protocol (NTP) [21] and the Precision Time Protocol (PTP) [1] *do,* typically resulting in significantly better performance than the aforementioned lower bounds would suggest. The reason is that a worst-case bound Δ on measurement errors that needs to hold across the entire lifetime of all systems deploying the algorithm is extremely conservative.

At first glance, the lower bound constructions from [10, 18] seem to show that this price needs to be paid: as with all indistinguishability arguments, it is the *possibility* that the errors might be large that forces a worst-case optimal algorithm to account for this scenario, even if this contingency is not realized. Fortunately, a closer inspection of the lower bound constructions reveals that this is not so! A critical step in these constructions is to suddenly change offset estimation errors as much as possible, swinging by Δ on some edges. This provides fresh "slack" for building up even larger local skews without the algorithm being able to perceive where this takes place. However, in many settings, the amount by which the measurement error can fluctuate on the time scale relevant for the lower bound is much smaller.

Similarly, the lower bound construction introduces additional skew by modifying the speed of local oscillators. Assuming that these oscillators run at a rate of at least 1 and at most $\vartheta > 1$, skew can be introduced into the system at a rate of $\vartheta - 1$. Again, frequency varies much more over the system's lifetime than on the time scale of the construction. Not coincidentally, practically deployed protocols compensate for local oscillators' frequency errors by locking their frequency to that of an (external) time reference [1, 21]. This also substantially improves performance in practice. Thus, the main question motivating this work is the following.

> *Is it possible to exploit* stability *of measurement errors and clock frequencies to achieve stronger bounds on the local skew?*

1.1 Main Results

We provide a positive answer to the above question. Introducing the model assumption that measurement errors on each edge do not change by more than $\delta(T) \ll \Delta$ within T time, cf. Table 1, the lower bound from [18] becomes $\Omega(\delta(T)\log D)$ with T being the time to build up this amount of skew using the construction. We provide a matching upper bound.

Corollary 1. *Suppose that $\mu > 2\vartheta - 1$ and $T \geq C\Delta D/\mu$ for a sufficiently large constant $C > 0$, and denote $\sigma := \mu/(\vartheta - 1)$. Assuming that output clocks can be initialized such that $\mathcal{G}(0) = \mathcal{O}(D\Delta)$, there is a GCS algorithm outputting clocks running at rates between* 1 *and $\vartheta(1+\mu)$ that at times $t \geq T$ achieves global and local skew bounds of*

$$\mathcal{G}(t) \in \left(1 + \frac{3}{\sigma - 1}\right)(\Delta + \mathcal{O}(\delta(T)))D$$
$$\text{and} \quad \mathcal{L}(t) \in 3\Delta + 4\delta(T)(\log_\sigma D + \mathcal{O}(1)).$$

Note that $\delta(T)$ is increasing in T, i.e., it is desirable to choose T small. However, T needs to be large enough for the algorithm to adjust clocks to (perceived) skews. This results in the lower bound on T, which is proportional to (the bound on) $\mathcal{G}(t)$ divided by μ, the parameter of the algorithm controlling by how much output clocks may run faster than local oscillators. We remark that (i) one could further reduce the lower bound on T by considering by how quickly the dynamics of link performance *change* the (perceived) global skew within T time and (ii) the O-notation does not hide large constants. For ease of presentation, we opted against covering these optimizations. Likewise, we refrain from modeling how to initialize clocks, which for the required weak bound of $\mathcal{G}(0) = \mathcal{O}(D\Delta)$ is straightforward in any reasonable model.

It is worth noting that in contrast to other variants of the algorithm discussed in the full version [17], the results presented here are maximally communication-efficient: by working exclusively based on estimates of clock offsets to neighbors, they introduce no overhead in communication over solving the task to compute local, sufficiently accurate such estimates.[1]

Adaptivity w.r.t. Measurement Errors. Even for relatively small values of μ, one is unlikely to encounter a network large enough for the logarithmic term to dominate the local skew bound, unless δ is fairly close to Δ. However, the above statement is still expressed in terms of Δ, a worst-case bound on offset measurements. The bounds we show are actually stronger. Denoting for edge $\{v,w\}$ by $e_{v,w}(t)$ the error in the measurement v has of the offset between its own and w's output clock at time t, the local skew is bounded by

$$|e_{v,w}(t)| + \delta(T)\left(4s + \mathcal{O}\left(\log_\sigma \frac{\mathcal{W}^s}{\delta(T)}\right)\right),$$

where $s \in \mathbb{N}$ is minimal such that the network graph with edge weights $4(s-1/2)\delta(T) - e_{v,w}(t)$ has no negative cycles and $\mathcal{W}^s$ is its diameter w.r.t. the induced distances. Note that if $\delta(T)$ is small, s will be such that the $4(s-1/2)\delta(T)$ term is just enough to match the sum Σ of the $e_{v,w}(t)$ values on the path maximizing the sum of errors $e_{v,w}(t)$ along its edges. If $e_{v,w}(t) \approx -e_{w,v}(t)$, which we will ensure, this means that $\mathcal{W}^s \approx 2\Sigma$. Intuitively, this means that the obtained bound is "independent" of $\delta(T)$, so long as the logarithmic term is not dominant. Therefore, in this regime the algorithm is fully adaptive w.r.t. to the quality of offset estimation. For the general case, we observe that $e_{v,w}(t) \leq \Delta$, implying $s \leq \lceil 1/2 + \Delta/(4\delta(T))\rceil$ and $\mathcal{W}^s \leq (\Delta + 6\delta)D$, giving rise to Corollary 1. The situation is similar for the global skew bound.

Without additional assumptions, each of the terms in the above refined local skew bound is necessary. The logarithmic term corresponds to the known lower bound on the local skew from [18]. To see that the other two terms are essentially tight, consider a cycle in which all measurements are perfectly accurate, except for a single edge e, where the error is always f in one direction and $-f$ in the other. We now initialize clock values so that all perceived offsets are identical, regardless of which edge is inaccurate.[2] While an

[1] Deriving such estimates is highly specific to the context, meaning that no general bounds on the number and size of messages can be given. However, our results offer a clean separation between the concerns of how to locally estimate offsets and how to use these measurements to adjust the clocks.

[2] This is valid when the initial state of the system can be chosen adversarially. Otherwise, start with all clock values equal and errors of 0, and slowly build up skew and

Table 1: Notation cheat sheet.

variable	meaning	comments
(V,E)	network graph	simple; assumed to be given
D	network diameter	unweighted
$L_v(t)$	output clock of v	also referred to as logical clock
$\mathcal{G}(t)$	global skew	$\max_{v,w\in V}\{L_v(t)-L_w(t)\}$
$\mathcal{L}_e(t)$	local skew on edge e	$\lvert L_v(t)-L_w(t)\rvert$, where $e=\{v,w\}$
$\mathcal{L}(t)$	local skew	$\max_{e\in E}\{\mathcal{L}_e(t)\}$
$o_{v,w}(t)$	v's offset estimate for w	for $\{v,w\}\in E$; $o_{v,w}(t)\approx L_v(t)-L_w(t)$
$e_{v,w}(t)$	error of $o_{v,w}(t)$	$e_{v,w}(t)=L_v(t)-L_w(t)-o_{v,w}(t)$; $e_{v,w}(t)\approx -e_{w,v}(t)$
Δ	worst-case bound for $o_{v,w}$	$\max_{\{v,w\}\in E}\{\lvert e_{v,w}(t)\rvert\}\le\Delta$
$\delta_{\{v,w\}}(T)$	change of $o_{v,w}$ over T time	for $\lvert t'-t\rvert\le T$, $\lvert e_{v,w}(t')-e_{v,w}(t)\rvert<\delta_{\{v,w\}}(T)$; T is fixed and typically omitted from notation
$\delta(T)$	upper bound on $\delta_{\{v,w\}}(T)$	special case of uniform speed of changes in errors
$H_v(t)$	local oscillator of v	also referred to as hardware clock
ϑ	maximum rate of H_v	given; $1\le\frac{dH_v}{dt}(t)\le\vartheta$
μ	speedup factor for L_v	chosen; usually $\frac{dH_v}{dt}(t)\le\frac{dL_v}{dt}(t)\le(1+\mu)\frac{dH_v}{dt}(t)$
α,β	rate bounds for L_v	$\alpha\le\frac{dL_v}{dt}(t)\le\beta$
σ	base of log in bound for $\mathcal{L}$	$\sigma=\mu/(\vartheta-1)$
s	level	algorithm described by triggers and conditions for $s\in\mathbb{N}$
$[a,a+T]$	analyzed time interval	fixed throughout analysis; general results follow by covering $\mathbb{R}_{\ge0}$ with overlapping intervals
$O_{v,w}$	nominal offset	$O_{v,w}=(e_{v,w}(a+T/2)-e_{w,v}(a+T/2))/2$
$\vec{E}$	oriented edges	$\vec{E}:=\bigcup_{\{v,w\}\in E}\{(v,w),(w,v)\}$
$\omega^s(v,w)$	level-s weights	$\omega^s(v,w):=4s\delta_{\{v,w\}}-O_{v,w}$ for $(v,w)\in\vec{E}$, $s\in 1/2\cdot\mathbb{N}$
$\omega^s(W)$	level-s weight of walk W	$\sum_{i=1}^{\ell}\omega^s(v_{i-1},v_i)$, where $W=v_0,v_1,\dots,v_\ell$
G^s	level-s graph	$G^s=(V,\vec{E},\omega^s)$
s_0	max. level with neg. cycles	no negative cycle in G_s for $s>s_0$
$d^s(v,w)$	distance in G^s	well-defined for $s>s_0$
$\mathcal{W}^s$	diameter of G^s	weighted; $\mathcal{W}^s=\max_{v,w\in V}\{d^s(v,w)\}$
$\Psi^s_v(t)$	level-s potential at v	$\Psi^s_v(t)=\max_{w\in V}\{L_w(t)-L_v(t)-d^s(v,w)\}$ for $s>s_0$
$\Psi^s(t)$	level-s potential	$\Psi^s(t)=\max_{v\in V}\{\Psi^s_v(t)\}$ for $s>s_0$

algorithm can adjust clock values, indistinguishability means that the end result in terms of *measured* skews is identical, regardless of our choice of $e=\{v,w\}$. Thus, the best an algorithm can do is to balance the observed offsets between neighbors, which along the n-node cycle sum to f or $-f$, respectively, depending on direction, yielding that the algorithm adjusts clocks so that observed skews are f/n resp. $-f/n$.

Up to sign, this results in a skew of $(n-1)f/n$ between v and w and f/n between any other pair of neighbors. Moreover, the cycle has weight $4n(s-1/2)\delta(T)\pm f$ (depending on direction), so $s\approx f/(4n\delta(T))$. Here, we can make $\delta(T)$ arbitrarily small, as we only need to change measurement errors to maintain indistinguishability while building up skew. Thus, choosing $\delta(T)$ sufficiently small ensures that $4s\delta(T)\approx f/n$, $\mathcal{W}^s\approx 2f$, and $\delta(T)\log_\sigma(\mathcal{W}^s/\delta(T))\approx\delta(T)\log_\sigma(2f/\delta(T))\ll f/n$.

We conclude that for any pair of neighbors v', w' different from v, w, due to $e_{v',w'}(t)=0$ the term of $4s\delta(T)$ is necessary to account for the unavoidable skew in this scenario. Likewise, for the skew between v and w, the term $\lvert e_{v,w}(t)\rvert=f$ is essential; the relative contribution of the other terms becomes arbitrarily small when n is made large and $\delta(T)$ small.

Of course, it is worth emphasizing that we could avoid the large skew across e and, in fact, elsewhere, if we *knew* that the alternative path from v to w offers more precise offset estimates. However, this would require *known* bounds on the estimation errors that allow to maintain smaller skews. In contrast, we see that it is impossible to bound the skew on edge $\{v,w\}$ solely in terms of $e_{v,w}$ and some notion of distance pertaining to actual measurement errors.

In analogy to [16], all of our results hold also for heterogeneous graphs, i.e., $\delta(T)$ can be different for each edge e. However, for the sake of exposition and easier comparison to related work, we confine ourselves to the uniform case in this overview of our results. Moreover, we stress that it is not possible to completely uncouple the algorithm from *some* knowledge of $\mathcal{W}^s$ and, by extension, Δ. The parameter $\delta(T)$ needs to be known to the algorithm, and the condition $T\ge C\mathcal{W}^s/\mu$ must hold. Therefore, $\delta(T)$ can only be chosen correctly based on an upper bound on $\mathcal{W}^s$. However, typically, a safe bound for $\mathcal{W}^s$ will be substantially smaller than $D\Delta$, and $\delta(T)$ is likely small enough for other terms to be dominant even when T is chosen conservatively.

adjust errors to maintain indistinguishability. Given enough time, this can be done with arbitrarily small $\delta(T)$.

1.2 When Does it Matter?

To illustrate when the presented improvements in the performance of GCS algorithms are of interest, let us briefly discuss three sample application scenarios.

1.2.1 Internet Synchronization. Our first and negative example is synchronization via the Internet. Here, NTP achieves worst-case errors in the tens to hundreds of milliseconds, while achieving "better than one millisecond accuracy in local area networks under ideal conditions" [22]. The main issue in the Internet are highly varying and asymmetric communication delays, rendering an approach based on stable clock offset measurements impractical. While at first glance the factor 10-100 gap between "global" and "local" performance sounds promising, it is important to remember that we require an upper bound on how much measurement errors can change on suitable time scales. Since there is no control over routing paths and queuing delays available to NTP in this setting, $\delta \approx \Delta$. Moreover, the primary use case for NTP over the Internet is coarse synchronization to UTC, rendering the utility of local skew bounds questionable.

1.2.2 Clock Distribution for Synchronous Hardware. As highlighted by the above example, relevant application scenarios satisfy two key criteria: (i) $\delta(T) \ll \Delta$ and (ii) the local skew being the key quality metric. In [5], the authors propose to leverage GCS for clocking of large-scale synchronous hardware. Here, the local skew in terms of physical proximity within the system is to be kept small: as the communicating subcircuits are physically adjacent, the local skew limits the frequency at which the system can reliably be operated. Combining a regular grid of GCS nodes with local clock distribution trees bears the promise of outperforming traditional clocking methods. However, clearly the break-even point in terms of system size heavily depends on the performance of the GCS algorithm.

The dominant sources of clock offset error estimation between the GCS nodes of such a system are delay variations due to process variations, changes in temperature and voltage, and aging of the components. These result in standard on-chip (ring) oscillators having impractically low accuracy. Accordingly, such oscillators are locked to external quartz references to result in more accurate and stable frequencies. This results in $\vartheta' - 1$ around 10^{-6} for a (cheap) quartz oscillator that would be used as common reference for all nodes [13, Sec. 7.1.6]. However, clock speeds in modern systems are beyond one gigahertz, meaning that local skews of at most tens of picoseconds can be tolerated. Thus, even for minimally chosen μ, a back of the envelope calculation suggests 10^{-3} s for $D \leq 100$ as a safe upper bound on $\mathcal{W}^s/\mu$. This is below the time scales relevant for temperature [8] and, certainly, aging. As the effect of process variations is not time-dependent, voltage fluctuations are the remaining factor. These need to be kept in check by modern systems to the best possible extent[3] anyway, as they pose a key constraint on the achievable trade-off between clock speed and energy consumption, see e.g. [12]. Accordingly, the assumption that $\delta \ll \Delta$ is justified, and the presented results could improve the achievable local skew by an order of magnitude or more.

[3]Voltage droops due to sudden load changes cause significant timing variability and power integrity challenges, motivating adaptive voltage and frequency scaling techniques to maintain reliable operation; see, e.g., [4, 11]. In contrast, the clocking subsystem draws steady power and can be protected from such droops by using dedicated supplies, which ensures stable operation under varying load conditions.

1.2.3 Synchronization in Wireless and Cellular Networks. Further applications arise in wireless networks. Especially for low-latency communication, the participants of such a network need to maintain tight synchronization. This enables to align transmission time slots with small guardbands and avoids interference, which is of particular interest in cellular networks. Moreover, performing time offset measurements via the wireless medium is equivalent to measuring distances. This allows us to tightly synchronize base stations, which in turn enables passively listening devices to determine their location based on knowledge of the position of base stations and the relative arrival time of the signals of the base stations. Given the limitations of GPS coverage both due to line-of-sight requirements and the prevalence of jamming, such an approach could complement satellite-based time and navigation services to improve availability and reliability.

Observe that also here, the local skew is decisive, not the global one: communication happens with close-by devices, and interference falls off quickly at larger distances; positioning is based on nearby basestations, as their signals are strongest. While dynamic environments may cause fairly sudden delay changes, round-trip measurements are only susceptible to introduced asymmetry. Mitigation techniques like taking the median of several measurements, dropping outliers, and temporarily removing unreliable links—it is known how to handle dynamic graphs [14, 15], and the techniques from this work are compatible with our approach—are likely to filter out the vast majority of cases where delays are unstable. Noting that the GCS algorithm gracefully degrades in face of small, short-lived errors and can be made fully self-stabilizing, it is highly promising to exploit the typical case, i.e., relatively stable communication delays and hence offset measurements.

1.3 Technical Overview

We now give an overview of the structure of the remainder of this paper and highlight the main technical challenges and ideas. In Section 2, we introduce model and notation. For the sake of a streamlined exposition, we do not justify the modeling choices in Section 2. However, as this is crucial especially when proposing novel extensions, we offer an in-depth discussion to the interested reader in the full version of this paper [17].

Section 3 presents and analyses our algorithm. Interestingly, the algorithm itself is near-identical to prior GCS algorithms. We follow [16] in our presentation, expressing the algorithm in the form of conditions it implements, which are ensured by adhering to the *fast* and *slow triggers* that can be expressed in terms of estimated clock offsets to neighbors. The key difference is that the conditions and triggers scale with $\delta(T)$ instead of Δ.

The main challenge lies in analyzing this variant of the algorithm. In prior work, this was based on the potential functions

$$\Psi^s(t) := \max_{v \in V}\{\Psi^s_v(t)\}, \text{ where } \Psi^s_v(t) = \max_{w \in V}\{L_w(t) - L_v(t) - d^s(v,w)\},$$

$s \in \mathbb{N}$, where $d^s(v,w)$ is the distance between v and w in the graph $(V, E, 4s\Delta_e)$, i.e., edge e has weight corresponding to its worst-case offset estimation error Δ_e. The conditions mentioned above ensure that (i) $\Psi^s_v(t)$ can grow only at rate $\vartheta - 1$, as any maximizing node satisfies the slow trigger, and (ii) for $s \neq 0$, we are guaranteed that L_v will make at least $\Psi_v^s(t)$ more than minimum progress by time $t + \Psi^{s-1}(t)/\mu$. As $\Psi^0(t) = \mathcal{G}(t)$, induction over s then shows that we can bound $\Psi^s(t + \mathcal{O}(\mathcal{G}(t)/\mu)) \leq \mathcal{G}(t)/\sigma^s$, where $\sigma = \mu/(\vartheta - 1)$. Since for $\{v, w\} \in E$, we have that $d^s(v, w) \leq 4s\Delta_e$, choosing $s = \lceil \mathcal{G}(t)/\Delta_e \rceil$ yields a bound of $\mathcal{L}_e(t) \leq \mathcal{O}(\Delta_e \log_\sigma \mathcal{G}/\mu)$ on the local skew on edge $e = \{v, w\}$ for times $t \geq t_0 \in \mathcal{O}(\mathcal{G}(0)/\mu)$, where $\mathcal{G}$ is an upper bound on the global skew at times $t \geq t_0' \in \mathcal{O}(\mathcal{G}(0)/\mu)$ that is established by similar arguments.

The crucial insight is that this approach can be adapted to our setting by replacing the edge weights with a carefully chosen term. The above potential considers $L_v(t) - L_w(t) = 0$ as the "target" state for neighbors v, w, and by extension for all pairs $v, w \in V$. Instead, fix a time t_0 and consider the offset $O_{v,w}$ required such that $L_v(t_0) - L_w(t_0) = O_{v,w}$ would result in v *measuring* a skew of 0 to w; note that this aligns with the intuition provided by our earlier example, in which an algorithm could not do better than matching the observed skews on each edge of the cycle, because it cannot know which edge(s) are responsible for its inability to reduce all observed skews to 0. Putting weight $4s\delta_{\{v,w\}}(T) - O_{v,w}$ on edge $(v, w) \in \vec{E} := \bigcup_{\{v,w\} \in E}\{(v, w), (w, v)\}$, we can encode that we "expect" a skew of $O_{v,w}$ on the edge from the perspective of the analysis. The corresponding adjustment to the conditions and triggers shifts the thresholds causing clocks to run fast or slow by $O_{v,w}$ as well, meaning that the offset measurements nodes take are now "accurate" from the perspective of the modified potential up to a remaining error of $\delta(T)$ at times $t \in [t_0 - T, t_0 + T]$. Bounding the potential similarly to above, one would then hope to prove that $\mathcal{L}_{\{v,w\}}(t) \leq \max\{O_{v,w}, O_{w,v}\} + \mathcal{O}(\mathcal{G}(0)/\mu)$.

Naturally, there is a catch. Since the edge weights of the graph $(V, \vec{E}, 4s\delta_{\{v,w\}}(T) - O_{v,w})$ can be negative, it is possible that a negative cycle is created, and there are no well-defined distances. This corresponds to a situation in which the potentials $\Psi_v^s(t)$ are not meaningful; in particular, we cannot guarantee the key properties enabling the above induction. Fortunately, once s is large enough, negative cycles are prevented, and the desired properties are restored. Our earlier example illustrates why this is a necessity rather than an artifact of our analysis.

Note that this problem arises even in acyclic graphs, as we replaced each undirected edge by two directed ones. Addressing this issue, we require that $O_{v,w} = -O_{w,v}$, which is ensured by having both v and w derive their offset estimates from the same measurements. As clocks run at slightly different speeds, this requires to account for additional errors, but these can be subsumed by $\delta_{\{v,w\}}(T)$.[4] This eliminates cycles of length 2, strengthening the obtained bounds. Moreover, it means that graphs of high girth are likely to have well-defined potentials Ψ_v^s for small values of s; in particular, $s = 0$ is valid on trees, meaning that the algorithm's performance is provably close to that of tree-based protocols like NTP or PTP when using the same tree for synchronization.

With the modifications to the model, conditions, triggers, and potentials in place, the remaining technical contribution of Section 3 is to adapt the analysis to these new notions.

[4] As neighbors need to communicate in order to determine the estimates anyway, sending the computed value to the respective other node has small impact on $\delta_{\{v,w\}}$.

2 Model and Notation

We model the distributed system as an undirected connected graph $G = (V, E)$.

Hardware clocks. Each node $v \in V$ is equipped with a *hardware clock* of bounded frequency error $\vartheta - 1$. Formally, we model this clock as a differentiable function $H_v : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ satisfying that $1 \leq \frac{dH_v}{dt}(t) \leq \vartheta$, i.e., hardware clock rates vary between 1 and ϑ. Nodes have query access to their hardware clocks, i.e., can read them at any time, but cannot access the underlying "true" time t. Note that there are no guaranteed bounds on $H_v(t) - t$ at any time t, so without access to an external reference, the system cannot synchronize to the "true" time.

Logical clocks and output requirements. The goal of a synchronization algorithm is to output at each node a logical clock L_v that (i) always advances at a rate close to 1, (ii) always has a small offset to other nodes' logical clocks, and (iii) if some node has access to an external reference, is also close to t. However, we aim for a self-stabilizing solution. Therefore, we require these properties only at times $t \geq S$, where S is the *stabilization time* of the algorithm. Note that minimizing S is an optimization goal in its own right.

Formally, node $v \in V$ maintains logical clock values $L_v(t)$ such that $L_v : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ satisfies for all $t, t' \in \mathbb{R}_{\geq 0}$, $t' \geq t$, that

$$\alpha(t' - t) \leq L_v(t') - L_v(t) \leq \beta(t' - t),$$

where we seek to minimize both the maximum rate difference $\beta - \alpha$ and the deviation $|1 - (\alpha + \beta)/2|$ from a nominal rate of 1.[5]

In terms of quality of synchronization, we seek to minimize the *global skew*

$$\mathcal{G}(t) := \max_{v,w \in V}\{L_v(t) - L_w(t)\}$$

and the *local skew*, i.e.,

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

Clock offset estimates. Node $v \in V$ maintains for each neighbor w an estimate $o_{v,w}(t)$ of $L_v(t) - L_w(t)$, whose error $e_{v,w}(t) := L_v(t) - L_w(t) - o_{v,w}(t)$ does not satisfy a known bound. For convenience, we assume that $o_{v,w}(t)$ changes only at discrete points in time, i.e., $o_{v,w}(t)$ is piece-wise constant.[6] We require that this error is approximately antisymmetric, i.e., $e_{v,w}(t) \approx -e_{w,v}(t)$, and that it changes relatively slowly over time. For convenient use in our analysis, we merge both requirements into a single constraint. That is, for $T \in \mathbb{R}_{\geq 0}$ there is a known value $\delta_{\{v,w\}}(T)$ such that for all $t_0 \in \mathbb{R}_{\geq 0}$ and $t, t' \in [t_0, t_0 + T]$, it holds that

$$|e_{v,w}(t') - e_{v,w}(t)| < \delta_{\{v,w\}}(T) \text{ and } |e_{v,w}(t') + e_{w,v}(t)| < \delta_{\{v,w\}}(T).$$

In essence, any estimation with slowly changing error can be turned into one with the required approximate antisymmetry, by sending one node's estimate to the other, which flips the sign.

[5] The reader is invited to assume that $\alpha = 1$. We stick to the more general notation to remain consistent with the full version.

[6] Since the logical clocks L_v increase at bounded rate, this can be achieved by sampling any $o_{v,w}(t)$ without this constraint at least once every $\varepsilon \in \mathbb{R}_{>0}$ time and increasing $\Delta_{v,w}$ by $(\beta - \alpha)\varepsilon$.

Computational model. To avoid the notion of continuous computations, the algorithm is only required to output L_v when queried at discrete points in time, which may be chosen adversarially. The computation of $L_v(t)$ when queried at time t has no further effect on the execution, i.e., a query only "reads" the logical clock. Note that since v *could* be queried at any time and L_v is continuous, this implies that an algorithm specifies $L_v(t)$ at all times $t \in \mathbb{R}_{\geq 0}$, consistent with the above definitions.

Each node executes a state machine, where two types of events may trigger a computational step of $v \in V$ at time t: (i) $o_{v,w}$ changes for some $\{v, w\} \in E$ or (ii) $H_v(t) = h$ for a previously computed (or hard-coded) hardware clock value. Algorithms are not allowed to (ab)use (ii) to cause an unbounded number of computational steps in finite time. Since $o_{v,w}$ changes only at discrete points in time, this means that the number of computational steps a state machine takes in a finite time interval is finitely bounded.

In a computational step taking place at time t, the state machine receives as input $H_v(t)$ and $o_{v,w}(t)$ for each $\{v, w\} \in E$. It then updates its state; an algorithm is specified by (i) its state variables, (ii) determining how these computations are performed in line with the stated constraints, and (iii) specifying how $L_v(t)$ is computed when queried at time t, based on $H_v(t)$ and v's state.

Initialization of state variables, which includes the initial clock values, is a concern that requires further modeling. As the full version [17] shows how to achieve full self-stabilization and even the variant of the algorithm presented here converges to small skews for any initial values of L_v, we assume an arbitrary state at time 0.

Notation. We denote by $\mathbb{N}$ the non-negative integers. For $k \in \mathbb{N}$, we denote $[k]_0 := [0, k] \cap \mathbb{N}$ and $[k] := [k]_0 \setminus \{0\}$. When we write $x = y \pm z$, this is equivalent to $|x - y| \leq z$. A *walk* is any sequence of nodes $W = v_0, \dots, v_\ell$ such that for each $i \in [\ell]$ it holds that $\{v_{i-1}, v_i\} \in E$. If the graph is weighted with edge weights ω, then the walk has weight $\omega(W) := \sum_{i=1}^{\ell} \omega(v_{i-1}, v_i)$. The set of walks from $v \in V$ to $w \in V$ is denoted by $W_{v,w}$. Analogous definitions are used for directed graphs. If for a directed edge $e = (v, w)$ a value val_e is not defined, but $\mathrm{val}_{\{v,w\}}$ is, val_e is interpreted as $\mathrm{val}_{\{v,w\}}$.

Throughout the paper, unless stated otherwise, we assume that all considered times lie within an interval $[t, t+T]$. We will prove a bound that T must satisfy in order to ensure that the stabilization time $S \leq T/2$, implying that we can cover $\mathbb{R}_{\geq 0}$ by the overlapping intervals $[iT/2+S, iT/2+T]$, $i \in \mathbb{N}$, so that for each interval we can apply the derived bounds on skews etc. during $[iT/2 + S, iT/2 + T] \subseteq [(i + 1)T/2, iT/2 + T]$. This covers the interval $[S, \infty)$, i.e., establishes all bounds at times $t \geq S$, as required for an algorithm with stabilization time S.

3 GCS Algorithm

In the following, we fix $T \in \mathbb{R}_{\geq 0}$ and will specify the algorithm $\mathcal{A}(T)$. To do so, we follow the approach of first specifying the key properties the algorithm needs to satisfy to achieve strong bounds on the local skew $\mathcal{L}$. Once specified, implementing these properties is extremely straightforward.

3.1 Conditions and Triggers

We start by expressing the desired conditions in terms of the logical clock offsets to neighbors, which nodes do not have access to. In prior work, it was assumed that there is a known bound $\Delta_{\{v,w\}}$ such that $|o_{v,w}(t)| \leq \Delta_{\{v,w\}}$ for *all* $t \in \mathbb{R}_{\geq 0}$. The imposed conditions then would require a node to advance its logical clock at rate $\frac{dH_v}{dt}(t)$ if for some $s \in \mathbb{N}$ (i) there is some neighbor whose logical clock lags by $2s\Delta_{\{v,w\}}$ behind and (ii) no neighbor's clock is by $2s\Delta_{\{v,w\}}$ ahead of $L_v(t)$. Similarly, the logical clock rate should be $(1 + \mu)\frac{dH_v}{dt}(t)$ for a choosable parameter $\mu > \vartheta - 1$ if (i) there is some neighbor whose logical clock is by more than $(2s+1)\Delta_{\{v,w\}}$ ahead and (ii) no neighbor's clock is by more than $(2s + 1)\Delta_{\{v,w\}}$ behind $L_v(t)$. The "separation" of these rules by an offset of $\Delta_{\{v,w\}}$ makes it possible to simultaneously implement both of them despite nodes having access only to inaccurate estimates $o_{\{v,w\}}(t)$ of $L_v(t) - L_w(t)$ for each neighbor w.

We seek to replace $\Delta_{\{v,w\}}$ by the smaller $\delta_{\{v,w\}} := \delta_{\{v,w\}}(T)$, but doing this naively entails that the estimates cannot be used to implement the conditions. Instead, we must accept that the conditions that are realized incur a large bias of up to $\Delta_{\{v,w\}}$ (where $\Delta_{\{v,w\}}$ is not known). However, as we analyze a time interval of length T, we have the guarantee that the error of the estimates *changes* little.

With this in mind, for an arbitrary $a \in \mathbb{R}_{\geq 0}$, let us fix a time interval $[a, a + T]$ throughout this section. All times will be from this interval only, so that for any considered times t and t' and $\{v, w\} \in E$, it holds that $|e_{v,w}(t') - e_{v,w}(t)| < \delta_{\{v,w\}}(T)$ and $|e_{v,w}(t') + e_{w,v}(t)| < \delta_{\{v,w\}}(T)$. For this interval, for each $\{v, w\} \in E$ we can fix a "nominal" offset around which the estimation error $e_{\{v,w\}}(t)$ varies.

Definition 2 (Nominal Offset)**.** *For each $\{v, w\} \in E$, set $O_{v,w} := (e_{v,w}(a + T/2) - e_{w,v}(a + T/2))/2$.*

Note that this definition ensures not only that $|e_{v,w}(t) - O_{v,w}| < \delta_{\{v,w\}}$ for all $t \in [a, a + T]$, but also that $O_{v,w} = -O_{w,v}$. Thus, if $L_v(t) - L_w(t) - O_{v,w} = 0$, then also $L_w(t) - L_v(t) - O_{w,v} = -(L_v(t) - L_w(t) - O_{v,w}) = 0$. In this sense, $O_{v,w}$ is going to be the "nominal" offset: our conditions will be such that $L_v(t) - L_w(t) = O_{v,w}$ is considered the "desired" state of edge $\{v, w\}$. This leads to the following variation of the slow and fast conditions introduced by Kuhn and Oshman [16].

Definition 3 (Slow Condition)**.** *The* slow condition *holds at $v \in V$ at time $t \in \mathbb{R}_{\geq 0}$ if and only if there is some* level $s \in \mathbb{N}$ *such that*

$$\exists \{v, w\} \in E : \; L_v(t) - L_w(t) - O_{v,w} \geq 4s\delta_{\{v,w\}}$$
$$\forall \{v, w\} \in E : \; L_v(t) - L_w(t) - O_{v,w} \geq -4s\delta_{\{v,w\}}.$$

Definition 4 (Fast Condition)**.** *The* fast condition *holds at $v \in V$ at time $t \in \mathbb{R}_{\geq 0}$ if and only if there is some* level $s \in \mathbb{N}$ *such that*

$$\exists \{v, w\} \in E : \; L_v(t) - L_w(t) - O_{v,w} \leq -(4s + 2)\delta_{\{v,w\}}$$
$$\forall \{v, w\} \in E : \; L_v(t) - L_w(t) - O_{v,w} \leq (4s + 2)\delta_{\{v,w\}}.$$

Intuitively, the difference of the above conditions to [16] is that we re-adjust the goal from seeking to minimize the skew on edge $\{v, w\} \in E$ towards minimizing the *apparent* skew when considering $O_{v,w}$ the baseline. From the point of view of the measurement on this particular edge, this is the best we can hope for without knowledge of $O_{v,w}$, as with $L_v(t) - L_w(t) = O_{v,w}$ we might have $o_{v,w}(t) = o_{w,v}(t) = 0$, i.e., neither node observes skew on the edge.

Next, we translate these conditions into "triggers" based on the nodes' estimates of clock offsets. The triggers are chosen such that if a condition holds, the corresponding trigger holds, too.

Definition 5 (Slow Trigger). *The* slow trigger *holds at $v \in V$ at time $t \in \mathbb{R}_{\geq 0}$ if and only if there is some* level $s \in \mathbb{N}$ *such that*

$$\exists\{v,w\} \in E: \ o_{v,w}(t) > (4s-1)\delta_{\{v,w\}}$$
$$\forall\{v,w\} \in E: \ o_{v,w}(t) > -(4s+1)\delta_{\{v,w\}}.$$

Definition 6 (Fast Trigger). *The* fast trigger *holds at $v \in V$ at time $t \in \mathbb{R}_{\geq 0}$ if and only if there is some* level $s \in \mathbb{N}$ *such that*

$$\exists\{v,w\} \in E: \ o_{v,w}(t) < -(4s+1)\delta_{\{v,w\}}$$
$$\forall\{v,w\} \in E: \ o_{v,w}(t) < (4s+3)\delta_{\{v,w\}}.$$

Lemma 7. *If the slow (fast) condition holds at $v \in V$ at time $t \in \mathbb{R}_{\geq 0}$, then the slow (fast) trigger holds at $v \in V$ at time $t \in \mathbb{R}_{\geq 0}$.*

Proof. Let s be a level for which the slow condition holds at $v \in V$ at time t. Then there is some $\{v,w\} \in E$ such that

$$\begin{aligned} o_{v,w}(t) &= L_v(t) - L_w(t) - e_{v,w}(t) \\ &\geq L_v(t) - L_w(t) - O_{v,w} - |O_{v,w} - e_{v,w}(t)| \\ &> L_v(t) - L_w(t) - O_{v,w} - \delta_{\{v,w\}} \geq (4s-1)\delta_{\{v,w\}}. \end{aligned}$$

Similarly, for all $\{v,w\} \in E$, the condition implies that

$$o_{v,w}(t) > L_v(t) - L_w(t) - O_{v,w} - \delta_{\{v,w\}} \geq -(4s+1)\delta_{\{v,w\}},$$

i.e., the slow trigger holds at v at time t.

Now let s be a level for which the fast condition holds at $v \in V$ at time t. Then there is some $\{v,w\} \in E$ such that

$$o_{v,w}(t) < L_v(t) - L_w(t) - O_{v,w} + \delta_{\{v,w\}} < -(4s+1)\delta_{\{v,w\}}.$$

Similarly, for all $\{v,w\} \in E$, the condition implies that

$$o_{v,w}(t) < L_v(t) - L_w(t) - O_{v,w} + \delta_{\{v,w\}} < (4s+3)\delta_{\{v,w\}},$$

i.e., the fast trigger holds at v at time t. □

Lemma 8. *The slow and fast trigger are mutually exclusive.*

Proof. Let s be a level for which the slow trigger holds at $v \in V$ at time t and consider level $s' \in \mathbb{N}$. If $s' \geq s$, the slow trigger entails for all $\{v,w\} \in E$ that $o_{v,w}(t) > -(4s+1)\delta_{\{v,w\}} \geq -(4s'+1)\delta_{\{v,w\}}$, so the first requirement of the fast trigger cannot be satisfied on level s'. If $s' < s$, the slow trigger implies that there is some $\{v,w\} \in E$ such that $o_{v,w}(t) > (4s-1)\delta_{\{v,w\}} \geq (4s'+3)\delta_{\{v,w\}}$, so the second requirement of the fast trigger cannot be satisfied on level s'. □

3.2 Algorithm

As stated above, we wish for our algorithm to run "fast," i.e., increase L_v at rate $(1+\mu)\frac{dH_v}{dt}$ whenever the fast trigger holds, and run "slow," i.e., increase L_v at rate $\frac{dH_v}{dt}$ whenever the slow trigger holds. If neither applies, any rate between the two extremes is a valid choice. As the estimates $o_{v,w}$ are piece-wise constant and any change of an estimate causes v to execute a computational step of its state machine, this is straightforward to realize, see Algorithm 1. Clearly, Algorithm 1 specifies an algorithm in the sense of Section 2. For simplicity, the algorithm checks whether r_v is consistent with the offset estimates once every time unit (according to H_v). However, any suitable frequency for such consistency checks is valid. In the following, we assume that $a \geq 1$, i.e., each node checked for state consistency at least once and $r_v = 1+\mu$ if and only if the fast trigger holds at v.

Algorithm 1: Basic GCS algorithm, code at $v \in V$. It will converge to small skews for any initial values of the variables. However, the time to converge scales with $\mathcal{G}(0)/\mu$.

variables: $L_v \in \mathbb{R}_{\geq 0}$, $H_v \in \mathbb{R}_{\geq 0}$, and $r_v \in \{1, 1+\mu\}$

When executing a computational step at time t:
$L_v \leftarrow L_v + r_v \cdot (H_v(t) - H_v)$
$H_v \leftarrow H_v(t)$
if *fast trigger holds at* v **then** $r_v \leftarrow 1+\mu$
else $r_v \leftarrow 1$

When $H_v(t) \bmod \mathbb{N} = 0$:
execute a computational step

When queried for L_v at time t:
return $L_v + r_v \cdot (H_v(t) - H_v)$

Lemma 9. *Defining $r_v(t) := 1+\mu$ if the fast trigger holds at $v \in V$ at time t and $r_v(t) := 1$ otherwise, with Algorithm 1 we have that $L_v(t) = L_v(1) + \int_1^t r_v(\tau)\frac{dH_v}{dt}(\tau)\,d\tau$. In particular, for any time t at which r_v does not change, Algorithm 1 ensures that*

$$\begin{aligned} \frac{dL_v}{dt}(t) &= r_v(t) \cdot \frac{dH_v}{dt}(t) \\ &= \begin{cases} (1+\mu) \cdot \frac{dH_v}{dt}(t) & \text{if the fast trigger holds at } v \text{ at time } t \\ \frac{dH_v}{dt}(t) & \text{else.} \end{cases} \end{aligned}$$

Proof. Whether the fast trigger holds at $v \in V$ can only change when $o_{v,w}$ changes. Since $o_{v,w}$ is piece-wise constant and a change results in execution of the code given in Algorithm 1, after the first computational step of L_v the variable r_v equals $1+\mu$ if and only if the fast trigger holds, and $r_v = 1$ else. The first such step is taken at a time $t_v \leq 1$. Hence, by induction on the computational steps of v we see that $L_v(t) = L_v(t_v) + \int_{t_v}^t r_v(\tau)\frac{dH}{dt}(\tau)\,d\tau$, where we use that $\int_{t_0}^{t_1} r \cdot \frac{dH_v}{dt}(\tau)\,d\tau = r \cdot (H_v(t_1) - H_v(t_0))$ for any $t_1 > t_0 \geq t_v$ and constant r. In particular, for $t \geq 1 \geq t_v$, we get that indeed $L_v(t) = L_v(1) + \int_1^t r_v(\tau)\frac{dH}{dt}(\tau)\,d\tau$. As taking the derivative of an integral after its upper integration boundary equals the integrand evaluated at this boundary, the remaining claim follows. □

Rate bounds on L_v are immediate from the above relation between L_v and H_v.

Corollary 10. *Algorithm 1 guarantees $\alpha = 1$ and $\beta = (1+\mu)\vartheta$.*

3.3 Potentials

We analyze Algorithm 1 based on the following potentials.

Definition 11. *For $s \in \mathbb{N}$, the* level-s potential at $v \in V$ *is*

$$\Psi_v^s(t) := \sup_{w \in V} \sup_{v_0,\ldots,v_\ell \in W_{v,w}} L_w(t) - L_v(t) - \sum_{i=1}^{\ell}(4s\delta_{\{v_{i-1},v_i\}} - O_{v_{i-1},v_i})$$

and the level-s potential *is $\Psi^s(t) := \sup_{v \in V} \Psi_v^s(t)$.*

The key difference between this potential and prior work is the inclusion of the term O_{v_{i-1},v_i}, which is necessary to account for the inclusion of the same term in the slow and fast conditions. However, this calls into question whether this potential can yield useful results. In fact, in general it is not even bounded, as there might be a cycle in which $4s\delta_{\{v_{i-1},v_i\}} - O_{v_{i-1},v_i}$ sums to a negative value.

Definition 12. *For $s \in 1/2 \cdot \mathbb{N}$, the* level-$s$ graph *is the weighted digraph $G^s := (V, \vec{E}, \omega^s)$, where $\vec{E} := \bigcup_{\{v,w\}\in E}\{(v,w),(w,v)\}$ and $\omega^s(v,w) := 4s\delta_{\{v,w\}} - O_{v,w}$ for all $(v,w) \in \vec{E}$. Let $s_0 \in \mathbb{N}$ be minimal such that for $s \geq s_0 + 1/2$, G^s contains no negative (directed) cycle. For $s \geq s_0 + 1/2$, we denote by $d^s(v,w)$ the distance from v to w in G^s and by $\mathcal{W}^s := \max_{v,w\in V}\{d^s(v,w)\}$ the diameter of G^s.*

We will reason about levels $s > s_0$. Before doing so, let us first point out that s_0 is bounded. We use the following shorthand.

Definition 13 (Maximum Absolute Estimate Error). *For each edge $\{v,w\} \in E$, define $\Delta_{\{v,w\}} := \max_{t\in[a,a+T]}\{|e_{v,w}(t)|, |e_{w,v}(t)|\}$.*

Lemma 14. *It holds that $s_0 \leq \max_{\{v,w\}\in E}\{\lceil \Delta_{\{v,w\}}/(4\delta_{\{v,w\}})\rceil - 1/2\}$.*

Proof. Let $s \geq \max_{\{v,w\}\in E}\{\lceil \Delta_{\{v,w\}}/(4\delta_{\{v,w\}})\rceil\}$. Then for each $(v,w) \in \vec{E}$, we have that $\omega^s(v,w) \geq \Delta_{\{v,w\}} - O_{v,w} \geq \Delta_{\{v,w\}} - (|e_{v,w}(a+T/2)| + |e_{w,v}(a+T/2)|)/2 \geq 0$, so G^s cannot contain a negative cycle. □

For $s > s_0$, we have a well-defined distance function, allowing to simplify how Ψ^s_v is expressed.

Lemma 15. *For all $s_0 < s \in \mathbb{N}$ and $v \in V$, it holds that*

$$\Psi^s_v(t) = \max_{w\in V}\{L_w(t) - L_v(t) - d^s(v,w)\}.$$

Proof. Consider any walk $W = v_0, \dots, v_\ell \in W_{v,w}$ of weight $\omega^s(W) = \sum_{i=1}^{\ell}\omega^s(v_{i-1},v_i)$. As G^s contains no negative cycles, we know that this sum is bounded from below, and that the minimum is attained by a shortest path from v to w. The claim follows, as by definition $\sup_{w\in V}\sup_{W\in W_{v,w}} L_w(t) - L_v(t) - \omega^s(W)$. □

As distances in G^s can be negative even for $s > s_0$, $\Psi^s(t)$ can exceed $\mathcal{G}(t)$. However, its deviation from $\mathcal{G}(t)$ is bounded by $\mathcal{W}^s$, which follows from the following helper statement.

Lemma 16. *For $v, w \in V$ and $s_0 < s \in \mathbb{N}$, $-d^s(v,w) \leq d^s(w,v)$.*

Proof. Let $P \in W_{v,w}$ and $Q \in W_{w,v}$ be shortest paths w.r.t. edge weights ω^s_e. As the concatenation of Q and P forms a cycle, the definition of s_0 implies that

$$d^s(v,w) + d^s(w,v) = \omega^s(P) + \omega^s(Q) \geq 0. \qquad \square$$

Our goal is to bound the potentials $\Psi^s(t)$ for $s_0 < s \in \mathbb{N}$ and suitable times t, as they readily imply bounds on $\mathcal{G}(t)$ and $\mathcal{L}(t)$.

Corollary 17. *For $s_0 < s \in 1/2 \cdot \mathbb{N}$, we have $|\mathcal{G}(t) - \Psi^s(t)| \leq \mathcal{W}^s$.*

Proof. Let $v, w \in V$ such that $\mathcal{G}(t) = L_w(t) - L_v(t)$. We have

$$\mathcal{G}(t) = L_w(t) - L_v(t) \leq \Psi^s_v(t) + d^s(v,w) \leq \Psi^s(t) + \mathcal{W}^s$$

by Lemma 15. Now consider $v, w \in V$ satisfying $\Psi^s(t) = L_w(t) - L_v(t) - d^s(v,w)$. By Lemma 16,

$$\begin{aligned}\Psi^s(t) &= L_w(t) - L_v(t) - d^s(v,w)\\ &\leq L_w(t) - L_v(t) + d^s(w,v) \leq \mathcal{G}(t) + \mathcal{W}^s. \qquad \square\end{aligned}$$

Corollary 18. *For $\{v,w\} \in E$, it holds that*

$$\mathcal{L}_{\{v,w\}}(t) \leq \min_{s_0<s\in\mathbb{N}}\{\Psi^s(t) + 4s\delta_{\{v,w\}} + |O_{v,w}|\}.$$

Proof. Let $s_0 < s \in \mathbb{N}$, $\{v,w\} \in E$, and w.l.o.g. $\mathcal{L}_{\{v,w\}}(t) = L_w(t) - L_v(t)$. By Lemma 15,

$$\begin{aligned}\mathcal{L}_{\{v,w\}}(t) &= L_w(t) - L_v(t) \leq \Psi^s_v(t) + d^s(v,w)\\ &\leq \Psi^s(t) + \omega^s(v,w) \leq \Psi^s(t) + 4s\delta_{\{v,w\}} + |O_{v,w}|. \qquad \square\end{aligned}$$

When increasing the level s, the potential function can only decrease, as we substract more.

Lemma 19. *For each $s_0 < s < s'$, $s, s' \in 1/2 \cdot \mathbb{N}$, $v \in V$, and time t, we have that $\Psi^{s'}_v(t) \leq \Psi^s_v(t)$.*

Proof. By definition, for each edge $(v,w) \in \vec{E}$, we have that $\omega^{s'}(v,w) \geq \omega^s(v,w)$. Hence, $d^{s'}(v,w) \geq d^s(v,w)$ for each $v, w \in V$ and the claim readily follows from Definition 11. □

3.4 Bounding the Potentials

As our first step, we bound how quickly potentials can grow.

Lemma 20. *Let $s_0 \leq s \in \mathbb{N}$, $v \in V$, and $t' > t$. If $\Psi^s_v(\tau) > 0$ for all $\tau \in (t,t')$, then $\Psi^s_v(t') \leq \Psi^s_v(t) + \vartheta(t'-t) - (L_v(t') - L_v(t))$.*

Proof. By Lemma 9, with the exception of the discrete points in time when r_v changes at some node, Ψ^s_v is differentiable with derivative

$$\frac{d\Psi^s_v}{dt}(t') = \max_{V\ni w \text{ maximizes } \Psi^s_v(t')}\left\{\frac{dL_w}{dt}(t') - \frac{dL_v}{dt}(t')\right\}.$$

Hence, if we can show that $\frac{dL_w}{dt}(\tau) \leq \frac{dH_w}{dt}(\tau) \leq \vartheta$ for any w maximizing $\Psi^s_v(\tau) > 0$ at a given time $\tau \in (t,t') \subseteq I_k$ when L_w is differentiable, the claim follows from integration. By Lemmas 8 and 9, this follows if such a w satisfies the slow trigger. Therefore, by Lemma 7 it is sufficient to prove that for any w maximizing $\Psi^s_v(\tau)$ at such a time τ, w satisfies the slow condition, as it implies the slow trigger.

Accordingly, suppose that

$$\Psi^s_v(\tau) = L_w(\tau) - L_v(\tau) - d^s(v,w) > 0,$$

i.e., w maximizes $\Psi^s_v(\tau) > 0$. For any $\{w,x\} \in E$, we have that $L_x(\tau) - L_v(\tau) - d^s(v,x) \leq L_w(\tau) - L_v(\tau) - d^s(v,w)$, yielding by the substractive triangle inequality that

$$\begin{aligned}L_x(\tau) - L_w(\tau) &\leq d^s(v,x) - d^s(v,w) \leq d^s(w,x)\\ &\leq \omega^s(w,x) = 4s\delta_{\{w,x\}} - O_{w,x},\end{aligned}$$

which can be rearranged to show the second requirement of the slow condition for node w at time τ.

To show the first, we note that $w \neq v$, because $\Psi^s_v(\tau) > 0 = L_v(\tau) - L_v(\tau) - d^s(v,v)$. Therefore, we may consider a neighbor x of w on a shortest path from v to w in G^s, satisfying $d^s(v,w) = d^s(v,x) + \omega^s(x,w)$. Using that w maximizes $\Psi^s_v(\tau)$ once more, it follows that

$$\begin{aligned}L_w(\tau) - L_x(\tau) &\geq d^s(v,w) - d^s(v,x) = \omega^s(x,w)\\ &= 4s\delta_{\{w,x\}} - O_{x,w} = 4s\delta_{\{w,x\}} + O_{w,x},\end{aligned}$$

which can be rearranged to show the first requirement. □

Corollary 21. *For all $s_0 \leq s \in \mathbb{N}$, $v \in V$, and $t' > t$, it holds that*

$$\Psi_v^s(t') \leq \Psi_v^s(t) + (\vartheta - 1)(t' - t).$$

Proof. Follows from Lemma 20, as $\frac{dL_v}{dt} \geq 1$ by Corollary 10 (except at the discrete points in time when it does not exist). □

Lemma 22. *For each $s_0 < s \in \mathbb{N}$, node $v \in V$, and times $t < t'$, we have that $L_v(t') - L_v(t)$ is at least*

$$t' - t + \min\left\{\Psi_v^{s-1/2}(t), \mu(t'-t) - \Psi^{s-1/2}(t) + \Psi_v^{s-1/2}(t)\right\}.$$

Proof. Choose $w \in V$ such that $\Psi_v^{s-1/2}(t) = L_w(t) - L_v(t) - d^{s-1/2}(v, w)$. For $x \in V$, define

$$\phi_x(\tau) := L_w(t) + (\tau - t) - L_x(\tau) - d^{s-1/2}(x, w)$$

and let $\Phi(\tau) := \max_{x\in V}\{\phi_x(\tau)\}$. Note that

$$L_v(t') - L_v(t) - (t' - t) = \phi_v(t) - \phi_v(t') \geq \phi_v(t) - \Phi(t').$$

Hence, if $\Phi(t') \leq 0$, we get that

$$\begin{aligned} L_v(t') - L_v(t) &\geq t' - t + \phi_v(t) \\ &= t' - t + L_w(t) - L_v(t) - d^{s-1/2}(v, w) \\ &= t' - t + \Psi_v^{s-1/2}(t), \end{aligned}$$

as desired.

Next, observe that by Corollary 10

$$\phi_x(\tau) - \phi_x(t) = \tau - t - (L_x(\tau) - L_x(t)) \leq 0,$$

i.e., ϕ_x is decreasing. Thus, if $\Phi(\tau) \leq 0$ for *any* $\tau \in [t, t']$, the statement of the lemma follows.

Hence, it remains to consider the case that $\Phi(\tau) > 0$ for all $\tau \in [t, t']$. We claim that this entails that $\Phi(t') \leq \Phi(t) - \mu(t' - t)$, from which the statement follows due to

$$\begin{aligned} &L_v(t') - L_v(t) - (t' - t) \\ &\geq \phi_v(t) - \Phi(t') \geq \phi_v(t) - \Phi(t) + \mu(t' - t) \\ &= \Psi_v^{s-1/2}(t) - \max_{x\in V}\{L_w(t) - L_x(t) - d^{s-1/2}(x, w)\} + \mu(t' - t) \\ &\geq \Psi_v^{s-1/2}(t) - \Psi^{s-1/2}(t) + \mu(t' - t). \end{aligned}$$

To show this claim, note that by Lemma 9 $\Phi(\tau)$ is differentiable with derivative

$$\frac{d\Phi}{d\tau}(\tau) = \max_{V\ni x \text{ maximizes } \Phi(\tau)}\left\{1 - \frac{dL_x}{dt}(\tau)\right\}$$

except for a discrete set of points in time when not all derivatives exist. Accordingly, consider a node $x \in V$ maximizing $\Phi(\tau)$ for $\tau \in [t, t']$ (when the derivatives exist), i.e., $\phi_x(\tau) = \Phi(\tau) > 0$. We will show that $\frac{dL_x}{dt}(\tau) \geq 1+\mu$, which will establish that $\frac{d\Phi}{d\tau}(\tau) \leq -\mu$. The claim (and hence the lemma) then readily follow from integration over the interval $[t, t']$.

To prove that $\frac{dL_x}{dt}(\tau) \geq 1 + \mu$, we show that x satisfies the fast condition at time τ. By Lemma 7, the fast trigger then holds at x, which by Lemma 9 yields that indeed $\frac{dL_x}{dt}(\tau) = (1 + \mu)\frac{dH_x}{dt}(\tau) \geq 1 + \mu$.

To this end, first consider any $\{x, y\} \in E$. As $\phi_y(\tau) \leq \phi_x(\tau)$, by the triangle inequality we have that

$$\begin{aligned} L_x(\tau) - L_y(\tau) &\leq d^{s-1/2}(y, w) - d^{s-1/2}(x, w) \leq d^{s-1/2}(y, x) \\ &\leq \omega^{s-1/2}(y, x) = (4s - 2)\delta_{\{y,x\}} - O_{y,x} \\ &= (4(s - 1) + 2)\delta_{\{x,y\}} + O_{x,y}, \end{aligned}$$

which can be rearranged into the second requirement of the fast condition (on level $s - 1$) for x at time τ.

To show the first requirement, note that $x \neq w$, as

$$\phi_w(\tau) \leq \phi_w(t) = 0 < \Phi(\tau) = \phi_x(\tau).$$

Hence, we may consider the first node $y \in V$ that is on a shortest path from x to w, entailing that $d^{s-1/2}(x, w) = \omega^{s-1/2}(x, y) + d^{s-1/2}(y, w)$. We get that

$$\begin{aligned} L_x(\tau) - L_y(\tau) &\leq d^{s-1/2}(y, w) - d^{s-1/2}(x, w) \\ &= -\omega^{s-1/2}(x, y) = -(4(s - 1) + 2)\delta_{\{x,y\}} + O_{x,y}, \end{aligned}$$

which can be rearranged into the first requirement of the fast condition on level $s - 1$. □

Corollary 23. *For each $s_0 < s \in \mathbb{N}$, node $v \in V$, and times t, t' with $t' \geq t+\Psi^{s-1/2}(t)/\mu$, it holds that $L_v(t')-L_v(t) \geq t'-t+\Psi_v^s(t)$.*

Proof. By Lemma 19 and the lower bound on t',

$$\Psi_v^s(t) \leq \Psi_v^{s-1/2}(t) \leq \mu(t' - t) - \Psi^{s-1/2}(t) + \Psi_v^{s-1/2}(t),$$

so the statement follows by application of Lemma 22. □

Lemma 20 and Lemma 22 are key to obtaining bounds on the potentials that decrease exponentially in s. First, we cover the base case for anchoring the induction at level $s > s_0$.

Lemma 24. *Suppose that $s_0 < s \in \mathbb{N}$, $\sigma := \mu/(\vartheta-1) \geq 2$ and that $\varepsilon > 0$. Then at all times $t \in [a+(2\Psi^s(a)+\lceil\log_\sigma 2/\varepsilon\rceil\cdot 4\mathcal{W}^s)/\mu, a+T]$, it holds that $\Psi^s(t) \leq 2\mathcal{W}^s/(\sigma - 1) + \varepsilon\Psi^{s-1/2}(0)$.*

Proof. Set $t_0 := a$, $t_1 := a + (\Psi^s(a) + 2\mathcal{W}^s)/\mu$, and $B_0 := 2\Psi^s(a) + 2\mathcal{W}^s/\sigma$. We apply Corollary 23 to see that for $t \in [t_0, t_1]$, it holds that

$$\Psi^s(t) \leq \Psi^s(a) + (\vartheta - 1)(t_1 - a) \leq \Psi^s(a) + \frac{\Psi^s(a) + 2\mathcal{W}^s}{\sigma} \leq B_0.$$

We now inductively define

$$t_{i+1} := t_i + \frac{B_i + 2\mathcal{W}^s}{\mu} \quad \text{and} \quad B_{i+1} := \frac{B_i + 2\mathcal{W}^s}{\sigma}$$

for $i \in \mathbb{N}_{>0}$ and $i \in \mathbb{N}$, respectively. We claim that $\Psi^s(t) \leq B_i$ for all $t \in [t_i, \min\{t_{i+1}, T\}]$ and $i \in \mathbb{N}$, which we show by induction on i. We already covered the base case of $i = 0$, so assume that the claim holds for index $i \in \mathbb{N}$. For $t \in [t_{i+1}, t_{i+2}]$, choose $v \in V$ such that $\Psi^s(t) = \Psi_v^s(t)$. Define $t' := \max\{t - (B_i + 2\mathcal{W}^s)/\mu, t_0\} \in [t_i, t_{i+1}]$. By Corollary 17 and the induction hypothesis,

$$\Psi^{s-1/2}(t') \leq \Psi^s(t') + \mathcal{W}^{s-1/2} + \mathcal{W}^s \leq B_i + 2\mathcal{W}^s,$$

where the last step uses that $\mathcal{W}^s$ is increasing in s, because the same is true for ω^s. Hence, if $t' = t - (B_i + 2\mathcal{W}^s)/\mu$, Lemma 20 and

Corollary 23 state that

$$\Psi_v^s(t) \leq \Psi_v^s(t') + \vartheta(t-t') - (L_v(t) - L_v(t'))$$
$$\leq (\vartheta-1)(t_{i+1}-t_i) = \frac{B_i + 2\mathcal{W}^s}{\sigma} = B_{i+1}.$$

Note that if $i \geq 1$, then from $t \geq t_{i+1}$ it follows that $t - (B_i + 2\mathcal{W}^s)/\mu \geq t_i$ and thus $t' := t - (B_i + 2\mathcal{W}^s)/\mu$. Therefore, the remaining case is that $t' = t_0 = a$ and $i = 0$. Then Corollary 17 entails that

$$\frac{\Psi^{s-1/2}(t')}{\mu} \leq \frac{\Psi^s(a) + \mathcal{W}^{s-1/2} + \mathcal{W}^s}{\mu} \leq \frac{\Psi^s(a) + 2\mathcal{W}^s}{\mu} \leq t - t_0.$$

Therefore, this case analogously yields that $\Psi_v^s(t) \leq B_1 = B_{i+1}$, as desired.

With the induction complete, another straightforward induction on i shows that

$$B_i = \frac{2\Psi^s(a)}{\sigma^i} + 2\mathcal{W}^s \cdot \sum_{j=1}^{i+1} \sigma^{-j} < \frac{2\Psi^s(a)}{\sigma^i} + \frac{2\mathcal{W}^s}{\sigma - 1}$$

for all $i \in \mathbb{N}$. As for $i \geq i_0 := \lceil \log_\sigma 2/\varepsilon \rceil$, we have that $2\Psi^s(a)/\sigma^i \leq \varepsilon \Psi^s(a)$, it remains to show that t_{i_0} is sufficiently small. To see this, observe that

$$t_{i_0} - a = \sum_{i=0}^{i_0-1} t_{i+1} - t_i = \sum_{i=0}^{i_0-1} \frac{B_i + 2\mathcal{W}^s}{\mu}$$
$$< \sum_{i=0}^{i_0-1} \left(\frac{\Psi^s(a)}{\mu\sigma^i} + \frac{2\mathcal{W}^s}{\mu} \cdot \sum_{j=0}^{i} \sigma^{-j} \right)$$
$$< \frac{\sigma}{\sigma-1} \cdot \frac{\Psi^s(a) + i_0 \cdot 2\mathcal{W}^s}{\mu} \leq \frac{2\Psi^s(a) + \lceil \log_\sigma 2/\varepsilon \rceil \cdot 4\mathcal{W}^s}{\mu},$$

where the final step again uses that $\sigma \geq 2$. □

Corollary 25. *Suppose that $s_0 < s \in \mathbb{N}$, $\sigma := \mu/(\vartheta - 1) \geq 2$, $\varepsilon > 0$ is a constant, and $T \geq C(\Psi^s(a) + \mathcal{W}^s)/\mu$ for a suitable constant $C = C(\varepsilon)$. Then at all times $t \in [a + T/4, a + T]$, it holds that $\Psi^s(t) \leq 2\mathcal{W}^s/(\sigma - 1) + \varepsilon \Psi^{s-1/2}(a)$.*

Using the preceding lemmas, bounding Ψ^{s+1} given a bound on Ψ^s is straightforward.

Lemma 26. *Suppose that for some $s_0 < s \in \mathbb{N}$, it holds that $\Psi^s(t') \leq B$ at all times $t' \geq t$. Then $\Psi^{s+1}(t') \leq B/\sigma$ at all times $t' \geq t + B/\mu$, where $\sigma = \mu/(\vartheta - 1)$.*

Proof. For any $t' \geq t + B/\mu$, consider $v \in V$ so that $\Psi^{s+1}(t') = \Psi_v^{s+1}(t')$. By assumption and Lemma 19, we have that

$$t' \geq t' - \frac{B - \Psi^{s+1}(t' - B/\mu)}{\mu} \geq t' - \frac{B - \Psi^{s+1/2}(t' - B/\mu)}{\mu}$$
$$= t' - \frac{B}{\mu} + \frac{\Psi^{s+1/2}(t' - B/\mu)}{\mu}.$$

Thus, Lemma 20 and Corollary 23 yield that

$$\Psi^{s+1}(t') = \Psi_v^{s+1}(t')$$
$$\leq \Psi_v^{s+1}\left(t' - \frac{B}{\mu}\right) + \vartheta \cdot \frac{B}{\mu} - \left(L_v(t') - L_v\left(t' - \frac{B}{\mu}\right)\right)$$
$$\leq (\vartheta - 1) \cdot \frac{B}{\mu} = \frac{B}{\sigma}.$$ □

Putting the induction together, we can bound the potentials and, by extension, skews.

Theorem 27. *Suppose that $s_0 < s \in \mathbb{N}$, $\sigma := \mu/(\vartheta - 1) \geq 2$, $\varepsilon > 0$ is a constant, and $T \geq C(\Psi^s(a) + \mathcal{W}^s)/\mu$ for a suitable constant $C = C(\varepsilon)$. Then for each $s \leq s' \in \mathbb{N}$ and time $t \in [a + T/2, a + T]$, it holds that*

$$\Psi^{s'}(t) \leq \frac{2\mathcal{W}^s/(\sigma - 1) + \varepsilon \Psi^{s-1/2}(a)}{\sigma^{s'-s}}.$$

Proof. Abbreviate $\psi := 2\mathcal{W}^s/(\sigma - 1) + \varepsilon \Psi^{s-1/2}(a)$. By Corollary 25, $\Psi^s(t) \leq \psi$ at times $t \in [a+T/4, a+T]$. Inductive application of Lemma 26 establishes the claimed bound for each s' and times larger than

$$a + \frac{T}{4} + \sum_{s''=s+1}^{s'} \frac{\psi}{\mu\sigma^{s''-s}} < a + \frac{T}{4} + \frac{\sigma\psi}{\mu(\sigma - 1)} \leq a + \frac{T}{4} + \frac{2\psi}{\mu} \leq a + \frac{T}{2},$$

where the last two steps use that $\sigma \geq 2$ and that C is sufficiently large. □

Corollary 28. *Suppose that $s_0 < s \in \mathbb{N}$, $\sigma := \mu/(\vartheta - 1) \geq 2$, $\varepsilon > 0$ is constant, and $T \geq C(\Psi^s(a) + \mathcal{W}^s)/\mu$ for a suitable constant C. Then for each $s \leq s' \in \mathbb{N}$ and time $t \in [a + T/2, a + T]$, it holds that*

$$\Psi^{s'}(t) \leq \frac{(2+\varepsilon)\mathcal{W}^s}{\sigma^{s'-s}(\sigma - 1)},$$
$$\mathcal{G}(t) \leq \mathcal{W}^s + \frac{(2+\varepsilon)\mathcal{W}^s}{\sigma - 1}, \text{ and, for each } \{v,w\} \in E,$$
$$\mathcal{L}_{\{v,w\}}(t) \in |O_{v,w}| + 4\left(s + \log_\sigma\left(\frac{\mathcal{W}^s}{\delta_{\{v,w\}}}\right) + \mathcal{O}(1)\right)\delta_{\{v,w\}}.$$

Proof. Follows from Theorem 27 with $s' = s + \lceil \log_\sigma \mathcal{W}^s/\delta_{\{v,w\}} \rceil$, Corollary 17, and Corollary 18. □

Proof of Corollary 1. Set $s := \lceil \Delta/(4\delta(T)) \rceil$. As the graph is uniform with edge parameters Δ and $\delta := \delta(T)$, Lemma 14 states that $s_0 \in \Delta/(4\delta) + \mathcal{O}(1)$. Hence, $\mathcal{W}^s \in D(\Delta + \mathcal{O}(\delta))$. By Corollaries 17 and 21, $\Psi^s(C\mathcal{W}^s/(2\mu)) \leq \Psi^s(0) + (\vartheta - 1)C\mathcal{W}^s/(2\mu) \leq \mathcal{G}(0) + \mathcal{O}(\mathcal{W}^s) \in \mathcal{O}(D\Delta)$.

We inductively apply Corollary 28 with[7] $a = C\mathcal{W}^s/(2\mu) + iT/2$ for $i \in \mathbb{N}$, $\varepsilon = 1$, and $s = s_0 + 1$, yielding the desired bound on the global skew at times $t \geq T \geq C\mathcal{W}^s/(2\mu) + T/2$. Using that $\sigma \geq 2$, the corollary then bounds the local skew at such times by

$$2\Delta + 4\delta\left(\log_\sigma \frac{D\Delta}{\delta} + \mathcal{O}(1)\right)$$
$$= 2\Delta + 4\delta\left(\log_\sigma D + \log_\sigma \frac{\Delta}{4\delta} + \mathcal{O}(1)\right) \leq 3\Delta + 4\delta(\log_\sigma D + \mathcal{O}(1)).$$ □

[7] This requires that $C\mathcal{W}^s/(2\mu) \geq 1$. However, if this is not the case, we may modify Algorithm 1 to trigger computational steps whenever $H_v(t) \bmod C\mathcal{W}^s/(2\mu) \cdot \mathbb{N} = 0$.

References

[1] 2008. IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems. *IEEE Std 1588-2008 (Revision of IEEE Std 1588-2002)* (2008), 1–269.

[2] H. Attiya, A. Herzberg, and S. Rajsbaum. 1996. Optimal Clock Synchronization under Different Delay Assumptions. *SIAM J. Comput.* 25, 2 (1996), 369–389.

[3] Saâd Biaz and Jennifer Lundelius Welch. 2001. Closed Form Bounds for Clock Synchronization under Simple Uncertainty Assumptions. *Inform. Process. Lett.* 80 (2001), 151–157.

[4] Keith A. Bowman, Carlos Tokunaga, Tanay Karnik, Vivek K. De, and James W. Tschanz. 2013. A 22 nm All-Digital Dynamically Adaptive Clock Distribution for Supply Voltage Droop Tolerance. *IEEE Journal of Solid-State Circuits* 48, 4 (2013), 907–916.

[5] Johannes Bund, Matthias Függer, Christoph Lenzen, Moti Medina, and Will Rosenbaum. 2020. PALS: Plesiochronous and Locally Synchronous Systems. In *International Symposium on Asynchronous Circuits and Systems (ASYNC)*. 36–43.

[6] Johannes Bund, Matthias Függer, and Moti Medina. 2023. PALS: Distributed Gradient Clocking on Chip. *IEEE Trans. Very Large Scale Integr. Syst.* 31, 11 (2023), 1740–1753. doi:10.1109/TVLSI.2023.3311178

[7] Johannes Bund, Christoph Lenzen, and Will Rosenbaum. 2019. Fault Tolerant Gradient Clock Synchronization. In *Symposium on Principles of Distributed (PODC)*. 357–365.

[8] Ayse Kivilcim Coskun, Tajana Simunic Rosing, and Kenny C. Gross. 2008. Proactive Temperature Management in MPSoCs. In *International Symposium on Low Power Electronics and Design (ISLPED)*. 165–170.

[9] D. Dolev, J. Halpern, B. Simons, and R. Strong. 1995. Dynamic Fault-tolerant Clock Synchronization. *Journal of the ACM (JACM)* 42, 1 (1995), 143–185.

[10] Rui Fan and Nancy Lynch. 2004. Gradient Clock Synchronization. In *Symposium on Principles of Distributed Computing (PODC)*. 320–327.

[11] Matthias Függer, Attila Kinali, Christoph Lenzen, and Ben Wiederhake. 2022. Fast All-Digital Clock Frequency Adaptation Circuit for Voltage Droop Tolerance. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 41, 8 (2022), 2518–2531.

[12] Meeta S. Gupta, Jarod L. Oatley, Russ Joseph, Gu-Yeon Wei, and David M. Brooks. 2007. Understanding Voltage Variations in Chip Multiprocessors using a Distributed Power-Delivery Network. In *Design, Automation & Test in Europe Conference & Exhibition (DATE)*. 1–6.

[13] Paul Horowitz and Winfield Hill. 2015. *The Art of Electronics* (3rd ed.). Cambridge University Press, Cambridge, UK.

[14] Fabian Kuhn, Christoph Lenzen, Thomas Locher, and Rotem Oshman. 2010. Optimal Gradient Clock Synchronization in Dynamic Networks. In *Symposium on Principles of Distributed Computing (PODC)*.

[15] Fabian Kuhn, Thomas Locher, and Rotem Oshman. 2009. Gradient Clock Synchronization in Dynamic Networks. In *Proc. 21st ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*. 270–279.

[16] Fabian Kuhn and Rotem Oshman. 2009. Gradient Clock Synchronization Using Reference Broadcasts. In *Conference on Principles of Distributed Systems (OPODIS)*. 204–218.

[17] Christoph Lenzen. 2025. *Gradient Clock Synchronization with Practically Constant Local Skew.* Technical Report 2511.01420. arXiv. doi:10.48550/arXiv.2511.01420 Full version of this paper.

[18] Christoph Lenzen, Thomas Locher, and Roger Wattenhofer. 2010. Tight Bounds for Clock Synchronization. *J. ACM* 57, 2 (2010).

[19] Jennifer Lundelius and Nancy Lynch. 1984. An Upper and Lower Bound for Clock Synchronization. *Information and Control* 62, 2/3 (1984), 190–204.

[20] Lennart Meier and Lothar Thiele. 2005. Brief Announcement: Gradient Clock Synchronization in Sensor Networks. In *Proc. 24th ACM Symposium on Principles of Distributed Computing (PODC)*. 238.

[21] David Mills. 1991. Internet Time Synchronization: the Network Time Protocol. *IEEE Transactions on Communications* 39 (1991), 1482–1493.

[22] David L. Mills. 2010. Network Time Protocol Version 4: Protocol and Algorithms Specification. RFC 5905. https://datatracker.ietf.org/doc/html/rfc5905

[23] Rafail Ostrovsky and Boaz Patt-Shamir. 1999. Optimal and Efficient Clock Synchronization under Drifting Clocks. In *Proc. 18th ACM Symposium on Principles of Distributed Computing (PODC)*. 400–414.