Optimal Gradient Clock Synchronization in Dynamic Networks (Technical Report)

Fabian Kuhn[1] Christoph Lenzen[2] Thomas Locher[3] Rotem Oshman[4]
[1]Faculty of Informatics, University of Lugano, Switzerland
fabian.kuhn@usi.ch
[2]Computer Engineering and Networks Laboratory, ETH Zurich, Switzerland
lenzen@tik.ee.ethz.ch
[3]IBM Zurich Research Laboratory, Switzerland
thl@zurich.ibm.com
[4]Computer Science and Artificial Intelligence Lab, MIT, USA
rotem@csail.mit.edu

Abstract

We study the problem of clock synchronization in highly dynamic networks, where communication links can appear or disappear at any time. The nodes in the network are equipped with hardware clocks, but the rate of the hardware clocks can vary arbitrarily within specific bounds, and the estimates that nodes can obtain about the clock values of other nodes are inherently inaccurate. Our goal in this setting is to output a logical clock at each node, such that the logical clocks of any two nodes are not too far apart, and nodes that remain close to each other in the network for a long time are better synchronized than distant nodes. This property is called *gradient clock synchronization.*

Gradient clock synchronization has been widely studied in the static setting, where the network topology does not change. We show that the asymptotically optimal bounds obtained for the static case also apply to our highly dynamic setting: if two nodes remain at distance d from each other for sufficiently long, it is possible to upper bound the difference between their clock values by $\mathcal{O}(d \log(D/d))$, where D is the diameter of the network. This is known to be optimal for static networks, and since a static network is a special case of a dynamic network, it is optimal for dynamic networks as well. Furthermore, we show that our algorithm has optimal *stabilization time*: when a path of length d appears between two nodes, the time required until the clock skew between the two nodes is reduced to $\mathcal{O}(d \log(D/d))$ is $\mathcal{O}(D)$, which we prove is optimal.

1 Introduction

A core algorithmic problem in distributed computing is to establish coordination among the participants of a distributed system, which is often achieved through a common notion of time. Typically, every node in a network has its own local hardware clock, which can be used for this purpose; however, hardware clocks of different nodes run at slightly different rates, and the rates can change over time. This *clock drift* causes clocks to drift out of synch, requiring periodic communication to restore synchronization. However, communication is typically subject to delay, and although an upper bound on the delay may be known, specific message delays are unpredictable. Consequently, estimates for the current local time at other nodes are inherently inaccurate.

A distributed clock synchronization algorithm computes *logical clocks* at every node, and the goal is to synchronize these clocks as tightly as possible. Traditionally, distributed clock synchronization algorithms focus on minimizing the *clock skew* between the logical clocks of any two nodes in the network. The clock skew between two clocks is simply the difference between the two clock values. The maximum clock skew that may occur in the worst case between any two nodes at any time is called the *global skew* of a clock synchronization algorithm. A well-known result states that no algorithm can guarantee a global skew better than $\Omega(D)$, where D denotes the diameter of the network [1]. However, in many cases it is more important to tightly synchronize the logical clocks of nearby nodes in the network than it is to minimize the global skew. For example, if a time division multiple access (TDMA) protocol is used to coordinate access to a shared communication medium in a wireless sensor network, it suffices to synchronize the clocks of nodes that interfere with each other when transmitting. The problem of providing better guarantees on the synchronization quality between nodes that are closer is called *gradient clock synchronization.* The problem was introduced in a seminal paper by Fan and Lynch [5], where the authors show that a clock skew of $\Omega(\log D/\log\log D)$ cannot be prevented between immediate neighbors in the network. The largest possible clock skew that may occur between the logical clocks of any two adjacent nodes at any time is called the *local skew* of a clock synchronization algorithm. For static networks, it has been proved that the best possible local skew that an algorithm can achieve is bounded by $\Theta(\log D)$ [11, 12].

While tight bounds have been shown for the static model, the dynamic case has not been as well understood. A dynamic network arises in many natural contexts: for example, when nodes are mobile, or when communication links are unreliable and may fail and recover. The dynamic network model we consider in this paper is general: it allows communication links to appear and disappear arbitrarily, subject only to a global connectivity constraint (which is required to synchronize all nodes to each other). Hence the model is suitable for modeling various types of dynamic networks which remain connected over time.

In a dynamic network the distances between nodes change over time as communication links appear and disappear. Consequently, we divide the synchronization guarantee into two parts: a *global skew guarantee* bounds the skew between any two nodes in the network at any time, and a *dynamic gradient skew guarantee* that bounds the skew between two nodes as a function of the distance between them and how long they remain at that distance.

In [8], three of the authors showed that a clock synchronization algorithm cannot react immediately to the formation of new links, and that a certain *stabilization time* is required before the clocks of newly-adjacent nodes can be brought into synch. The stabilization time is inversely related to the synchronization guarantee: the tighter the synchronization required in stable state, the longer the time to reach that state. Intuitively, this is because when strict synchronization guarantees are imposed, the algorithm cannot change clock values quickly without violating the guarantee, and hence it takes longer to react. The algorithm given in [8] achieved the optimal trade-off between skew bound and stabilization time; however, the local skew bound it achieved was $\mathcal{O}(\sqrt{D})$, which is far from optimal.

In this paper, we describe an anlgorithm that achieves the same asymptotically optimal skew bounds as in the static model: if two nodes remain at distance d for sufficiently long, the skew between them is reduced to $\mathcal{O}(d\log(D/d))$, where D is the dynamic diameter of the network (corresponding roughly to the time it takes for information to reach from one end of the network to the other). The stabilization time of the algorithm, that is, the time to reach this guarantee is $\mathcal{O}(D)$. We also improve the trade-off lower bound from [8] to show that a stabilization time of $\Omega(D)$ is necessary for an algorithm with the $\mathcal{O}(d \log(D/d))$-gradient skew property. This shows that our algorithm is optimal in its stabilization time as well as its skew guarantee. In addition, in a conference version of this paper, we decribe a more elegant variant of our algorithm with a slightly worse stabilization time of $\mathcal{O}(D \log D)$ [7].

2 Related Work

The fundamental problem of synchronizing clocks in distributed systems has been studied extensively and many results have been published for various models over the course of the last approximately 30 years (see, e.g, [16, 18, 19, 20]). Until recently, the main focus has been on bounding the clock skew that may occur between any two nodes in the network, and a tight bound of $\Theta(D)$ has been proved [1, 3, 12, 20].

The problem of synchronizing clocks of nodes that are close-by as accurately as possible has been introduced by Fan and Lynch [5]. In their work, the authors show that a clock skew of $\Omega(\log D / \log\log D)$ between neighboring nodes cannot be avoided if the clock values must increase at a constant minimum progress rate. Subsequently, this result has been improved to $\Omega(\log D)$ [12]. If we take the minimum clock rate α, the maximum clock rate β, and the maximum clock drift rate ρ into account, the more general statement of the lower bound is that a clock skew of $\Omega(\log_b D)$, where $b := \min\{1/\rho, (\beta-\alpha)/(\alpha\rho)\}$ cannot be avoided. The first algorithm guaranteeing a sublinear bound on the worst-case clock skew between neighbors achieves a bound of $\mathcal{O}(\sqrt{\rho D})$ [13, 14]. Recently, this result has been improved to $\mathcal{O}(\log D)$ [11] (the base of the logarithm is a constant) and subsequently to $\mathcal{O}(\log_b D)$ [12]. Thus, tight bounds have been achieved for static networks in which neither nodes nor edges fail.

The problem of synchronizing clocks in the presence of faults has also received considerable attention (see, e.g., [2, 6, 10, 15, 17]). Some of the proposed algorithms are able to handle not only simple node or edges failures but also Byzantine behavior, which is outside the scope of this paper. However, while these algorithms can tolerate a broader range of failures, their network model is not fully dynamic as their results rely on the assumption that a large part of the network remains non-faulty and stable at all times. For the fully dynamic setting, it has been shown that there is an inherent trade-off between the clock skew $\mathcal{S}$ guaranteed between neighboring nodes that have been connected for a long time and the time it takes to guarantee a small clock skew over newly added edges. In particular, the time it takes to reduce the clock skew over new edges to $\mathcal{O}(\mathcal{S})$ is $\Omega(D/\mathcal{S})$, where n denotes the number of nodes in the network [8]. In the same work, it is shown that for $\mathcal{S} \in \Omega(\sqrt{\rho D})$, there is an algorithm that reduces the clock skew between any two nodes to $\mathcal{O}(\mathcal{S})$ in $\Theta(D/\mathcal{S})$ time. In this paper, we show that $\mathcal{S}$ can be reduced to $\Omega(\log_b D)$, i.e., the same optimal bound as for static networks can be achieved.

3 Model and Definitions

Clock synchronization. In the clock synchronization problem, each node u is equipped with a continuous *hardware clock* $H_u : \mathbb{R}_0^+ \to \mathbb{R}_0^+$, which is initialized to $H_u(0) := 0$. The hardware clocks do not necessarily progress at the rate of real time; they are subject to a (relative) clock drift bounded by $\rho \in (0,1)$. At all times t we assume that $\frac{d}{dt}H_u(t) \in [1-\rho, 1+\rho]$ for all nodes u.

The objective of a clock synchronization algorithm (CSA) is to output a *logical clock* $L_u : \mathbb{R}_0^+ \to \mathbb{R}_0^+$ (also initialized to $L_u(0) := 0$), such that at all times, the logical clock values of different nodes are close to each other. Logical clocks must also have a bounded drift: there must exist constants $\alpha, \beta > 0$ such that $\frac{d}{dt} L_u(t) \in [\alpha, \beta]$ for all times t and for all nodes u.

The estimate graph. In [9] two of the authors introduced an abstraction called *the estimate layer*, which simplifies reasoning about CSAs. Synchronization typically involves periodic exchanges of clock values between nodes, either through messages or by other means (e.g., RBS [4]). The estimate layer encapsulates all means by which nodes can estimate the clock values of other nodes, and eliminates the need to reason explicitly about delay bounds and other parameters of the system.

The estimate layer provides an *estimate graph*, where each edge $\{u, v\}$ represents the fact that node u has some means of estimating v's current clock value and vice versa. The edges of the estimate graph are not necessarily direct communication links between nodes. Node u is provided with a *local estimate* $\tilde{L}_u^v$ of L_v, whose accuracy is guaranteed by the estimate layer:

$$\forall t \; \forall u \in V, v \in N_u(t) : |L_v(t) - \tilde{L}_u^v(t)| \le \epsilon_{\{u,v\}}, \tag{1}$$

where $\epsilon_{\{u,v\}}$ is called the *uncertainty*, or the *weight*, of the edge $\{u, v\}$. Each edge is also associated with a propagation delay $\mathcal{T}_{\{u,v\}}$ that bounds the time needed to send a message from u to v and vice versa. In the sequel, we refer to estimate edges of the sort described above simply as *edges*; similarly, when we say "the graph" we mean the estimate graph. We do not reason explicitly about the communication graph, as the salient aspects of communication are encapsulated by the estimate layer.

Dynamic networks. We consider dynamic networks over a fixed set of nodes V, where we denote $n := |V|$. Edge insertions and removals are modeled as discrete events controlled by a worst-case adversary. In keeping with the abstract representation from [9], we say that there is an *estimate edge* $\{u, v\}$ between two nodes $u, v \in V$ at time $t \ge 0$ iff u and v have a means of obtaining clock value estimates about each other at time t. As explained above, this does not necessarily mean that there is a direct communication link between u and v at time t.

We assume that nodes do not necessarily detect link formations and failures immediately, or even at the same time as the other endpoint of the link. For each edge $\{u, v\}$, we assume that there is a parameter $\tau_{\{u,v\}}$ such that both nodes u and v find out about the appearance or disappearance of edge $\{u, v\}$ within $\tau_{\{u,v\}}$ time units of the event itself. Hence, although we are interested in undirected networks, we model the estimate layer as a directed graph, where edge (u, v) exists whenever u thinks v is its neighbor. The appearance and disappearance of edges induces a dynamic graph $G = (V, E)$, where $E : \mathbb{R}_0^+ \to 2^{V \times V}$ maps non-negative times $t > 0$ to a set of directed estimate edges that exist at time t. The graph is subject to the following constraints, which approximate symmetry up to the delay ($\tau_{\{u,v\}}$) in finding out about link changes: (a) if for all $t' \in [t - \tau_{\{u,v\}}, t + \tau_{\{u,v\}}]$ we have $(u, v) \in E(t')$, then $(v, u) \in E(t)$; (b) if for all $t' \in [t - \tau_{\{u,v\}}, t + \tau_{\{u,v\}}]$ we have $(u, v) \notin E(t')$, then $(v, u) \notin E(t)$. Throughout each execution, every node u maintains a dynamic set of neighbors $N_u : \mathbb{R}_0^+ \to 2^V$, where $N_u(t)$ contains all nodes v such that $(u, v) \in E(t)$. In the sequel, we frequently refer to *undirected* edges $\{u, v\}$. When we write $\{u, v\} \in E(t)$ we mean that both $(u, v) \in E(t)$ and $(v, u) \in E(t)$.

We say that edge $\{u, v\}$ *exists* throughout a time interval $[t_1, t_2]$ if for all $t \in [t_1, t_2]$ we have $\{u, v\} \in E(t)$. By extension, a path p is said to exist throughout $[t_1, t_2]$ if all its edges exist throughout the interval.

Definition 3.1 (Weighted Paths)**.** *Let $G = (V, E)$ be a dynamic graph with edge weights ϵ_e, $e \in E(t)$. A* path *$p = (u_0, \dots, u_k)$ of length $k \geq 0$ in G at time t is a tuple of nodes such that for all $i \in \{1, \dots, k\}$ it holds that $\{u_{i-1}, u_i\} \in E(t)$. The* weight *of p is $\epsilon_p := \sum_{i=1}^{k} \epsilon_{\{u_{i-1}, u_i\}}$.*

We frequently refer to *the skew on a path* $p = (u_0, \dots, u_k)$ at time t, by which we mean $|L_{u_0}(t) - L_{u_k}(t)|$.

Dynamic diameter. A fundamental lower bound [1] shows that the performance of a CSA in a static network depends on the diameter of the network. In dynamic networks there is no immediate equivalent to a diameter. Informally, the diameter corresponds to the length of time it takes (at most) for information to spread from one end of the network to the other. To formalize this idea we adopt the following definitions.

Definition 3.2 (Flooding)**.** *A* flood *that originates at node u is a process initiated when node u sends a* Flood *message to all its neighbors. Each node that receives the message for the first time forwards it immediately to all its neighbors. We say that the flood is* complete *when all nodes have received a* Flood *message.*

Definition 3.3 (Dynamic Diameter)**.** *We say that dynamic graph G has a* dynamic diameter *of D (or simply "diameter" for short) if a flood originating at any node in the graph at any time in the execution always completes in at most D time units.*

Clock skew. To measure the quality of a CSA we consider two kinds of requirements: a *global skew constraint* which gives a bound on the difference between any two logical clock values in the system, and a *dynamic gradient skew constraint* which becomes stronger the closer two nodes u, v are to each other and the longer u, v stay close to each other. In particular, for nodes that remain neighbors for a long time, the dynamic gradient skew constraint requires a much smaller skew than the global skew constraint.

Definition 3.4 (Global Skew)**.** *A CSA guarantees a* global skew *of $\bar{\mathcal{G}}$ if at all times t, for any two nodes $u, v \in V$, it holds that $L_u(t) - L_v(t) \leq \bar{\mathcal{G}}$.*

In contrast, the dynamic gradient skew constraint does depend on the dynamic graph: the older the shortest path between u and v, the better synchronized u and v are required to be.

Definition 3.5 (Dynamic Gradient Skew)**.** *Given a function $\mathcal{S} : \mathbb{R}_0^+ \times \mathbb{R}_0^+ \to \mathbb{R}_0^+$ that is non-decreasing in the first parameter (distance) and non-increasing in the second (time), we say that a CSA $\mathcal{A}$ guarantees a* dynamic gradient skew *of $\mathcal{S}$ if for all time intervals $[t_1, t_2]$ and each path $p = (u_0, \dots, u_k)$ that exists throughout the interval $[t_1, t_2]$, we have that*

$$L_{u_0}(t_2) - L_{u_k}(t_2) \leq \mathcal{S}(\epsilon_p, t_2 - t_1).$$

If a CSA $\mathcal{A}$ guarantees a dynamic gradient skew of $\mathcal{S}$, we say that it ensures a stable gradient skew *of $\mathcal{S}^\infty(d) := \lim_{\Delta t \to \infty} \mathcal{S}(d, \Delta t)$, where $\mathcal{S}^\infty : \mathbb{R}_0^+ \to \mathbb{R}_0^+$.*

If a CSA $\mathcal{A}$ guarantees a dynamic gradient skew of $\mathcal{S}$, then we call $\mathcal{A}$ an "$\mathcal{S}$-dynamic gradient CSA". The literature on gradient clock synchronization (e.g., [5, 8, 12, 14]) is typically concerned with the *local skew* of a CSA, which bounds the skew on any single edge. The local skew can be considered equivalent to the stable gradient skew $\mathcal{S}^{\infty}(1)$, provided that all edges are of uniform weight 1.

We will further discuss the *stabilization time* of a CSA, which we define as follows.

Definition 3.6 (Stabilization Time)**.** *Let $\mathcal{A}$ be a dynamic gradient CSA with a dynamic gradient skew of $\mathcal{S}$. The* stabilization time *of $\mathcal{A}$ is defined as*

$$\mathcal{T}_S := \inf\left\{\Delta t \mid \forall d \forall \Delta t' \geq \Delta t : \mathcal{S}(d, \Delta t') \leq 2\mathcal{S}^{\infty}(d)\right\}.$$

The dynamic and stable gradient skew and the stabilization time are parametrized by D, the diameter of the network, and potentially other parameters such as the bound on the clock drift ρ or the minimum edge weight. Usually we omit these dependencies to simplify the notation. Note that the choice of 2 as the constant in the definition above is arbitrary; we are interested in the asymptotic behavior of the clock skew as $D \to \infty$.

4 A Class of Synchronization Algorithms

Instead of providing a concrete algorithm, we define a class of algorithms, denoted by $\mathfrak{A}$, and then proceed to show that each algorithm $\mathcal{A}$ that belongs to the class $\mathfrak{A}$ guarantees (asymptotically) optimal skew bounds and integration times of new edges in Section 5. The advantage of this approach is that the focus is strictly on the essential properties of an algorithm and there are no details of implementation that distract from these properties.

4.1 Conditions of $\mathfrak{A}$

To execute an algorithm $\mathcal{A} \in \mathfrak{A}$, the nodes must maintain an upper bound $\hat{\mathcal{G}}$ on the global skew. We assume that all nodes have access to this bound. For a trivial bound $\hat{\mathcal{G}}$, it is sufficient for all nodes to know the number n of nodes in the network, and use n as a conservative estimate for the diameter D of the graph in the global skew bound established in Section 5.1. If additional information is known about the network or if nodes run an additional protocol to estimate the dynamic diameter D of the network, the complexity of the algorithm can be improved by using a better bound $\hat{\mathcal{G}}$.

Additionally, the algorithm is provided a parameter $\mu > 4\rho/(1-\rho)$ that defines the permitted relative speedup of the logical clock in comparison to the hardware clock. This parameter will be discussed in greater detail shortly. Furthermore, we assume that the algorithm knows (an upper bound on) ρ. Given these parameters, we define the value

$$\sigma := (1-\rho)\mu/(4\rho) > 1 \tag{2}$$

whose purpose will be clarified in the analysis. Note that the requirement of $\sigma > 1$ necessitates the lower bound on the parameter μ given above. Lastly, we assume that the algorithm is aware of (upper bounds on) the values $\tau_{\{u,v\}}$ and $\epsilon_{\{u,v\}}$ for any edge $\{u, v\}$ that u believes to exist at a given time.

The algorithm operates in loosely synchronized rounds $r \in \mathbb{N}$. In the following, we add the superscript (r) to variables that depend on the round number r. Round r locally starts and ends according to the logical clock values, i.e., two nodes might be in different rounds at the same time. The estimate $\hat{\mathcal{G}}^{(r)}$ that a node $u \in V$ determines at time $T_1^{(r)}$ (when it enters round r) is used to compute $s_{\max} = s_{\max}^{(r)}(u)$, the times $T_2^{(r)}, \ldots, T_{s_{\max}}^{(r)}$, and also the time $T_1^{(r+1)}$ when round $r+1$ starts. The duration of a round is $T_1^{(r+1)} - T_1^{(r)} \in \mathcal{O}(\hat{\mathcal{G}}^{(r)}/\mu)$.

The logical times $T_2^{(r)}, \ldots, T_{s_{\max}}^{(r)}$ serve the purpose of gradually incorporating new edges into the synchronization protocol without jeopardizing the gradient property on edges that already existed for a sufficiently long period of time. To this end, every node u stores sets $N_u^s \subseteq N_u$ for $s \in \{1, \ldots, s_{\max}^{(r)}(u)\}$. An algorithm $\mathcal{A} \in \mathfrak{A}$ must update each set N_u^s according to a specific subroutine that works as follows.

At the beginning of the first round, all N_u^s are empty. At the beginning of each round $r \geq 2$, we have that $N_u^1 = N_u^2 = \ldots = N_u^{s_{\max}^{(r-1)}}$. At this time, after determining $s_{\max}^{(r)}(u)$, the new sets $N_u^{s_{\max}^{(r-1)}+1}, \ldots, N_u^{s_{\max}^{(r)}}$ are initialized to the same set ($N_u^1 = N_u^2 = \ldots$) of neighbors if $s_{\max}^{(r)}(u) > s_{\max}^{(r-1)}(u)$. Afterwards, for any newly appeared edge $\{u, v\}$ the node v is added to N_u^1 and v is informed about this action. Each set N_u^s, $s \in \{2, \ldots, s_{\max}^{(r)}(u)\}$, is updated by setting it to N_u^{s-1} whenever u's logical clock reaches a value $T_s^{(r)}$. A node v is removed from these sets in two cases, in both of which it is removed from all N_u^s at once: (i) node v is removed from N_u, i.e., u believes that the edge $\{u, v\}$ has failed, at any time in (its local) round r, or (ii) node u does not receive a notification that v has added u to its set N_v^1 until u's logical clock value reaches $T_2^{(r)}$. The second case makes sure that we do not end up with "unidirectional" edges where only one of the nodes treats the edge as existent during round r, which again would risk violation of the gradient property on edges that have existed for a long time.

Naturally, for our edge integration scheme to work properly, the estimate $\hat{\mathcal{G}}^{(r)}$ must satisfy certain properties.

Condition 4.1. *For all $r \in \mathbb{N}$ we demand the following.*

- *Any node $u \in V$ can access the value $\hat{\mathcal{G}}^{(r)}$ at the latest at the time $t_u^{(r)}$ when $T_1^{(r)} = L_u(t_u^{(r)})$ and all nodes agree on that value, i.e., it does not depend on u.*
- *$\hat{\mathcal{G}}^{(r)} \geq \mathcal{G}(t)$ for all t when $L_u(t) \in [T_1^{(r)}, T_2^{(r+1)})$ for some $u \in V$, i.e., $\hat{\mathcal{G}}^{(r)}$ is a valid upper bound on the global skew for a sufficiently long time.*
- *$\hat{\mathcal{G}}^{(r)}/((1+\mu)\mu) \geq \max_{u \in V} \max_{e \in E(t_u(r))}\{\mathcal{T}_e + \tau_e\}$, i.e., we exclude pathological cases where the time it takes to send even a single message over some edges is larger than the duration of the whole round.*

We will now discuss this procedure, which is summarized in Algorithm 1, in greater detail. In each round, the following steps are carried out. At the beginning of the round, i.e., at time T_1, the upper bound $\hat{\mathcal{G}}$ on the global skew is updated whose new value has been computed during the course of the previous round using a separate mechanism. Then, the new $s_{\max}$ is computed based on $\hat{\mathcal{G}}$, μ, and ρ. Afterwards, the new times $T_2 < \ldots < T_{s_{\max}}$ are computed (see Lines 9-13). The reason why the times $T_2, \ldots, T_{s_{\max}}$ form a geometric progression will become clear in Section 5. Subsequently, the new T_1, i.e., the time when the subsequent round will start, is determined and $\mathbf{T}$ is set to $\{T_1, \ldots, T_{s_{\max}}\}$.

Algorithm 1 The logical clock value of node u reaches $T_s \in \mathbf{T}$ for some $s \in \{1, \ldots, s_{\max}\}$.

```
1:  $\mathbf{T} := \mathbf{T} \setminus \{T_s\}$;
2:  if $\mathbf{T} = \{\}$ then
3:      $r := r + 1$;   // next round, as $T_1$ is always the largest time in $\mathbf{T}$
4:      $\hat{\mathcal{G}}$ := updateGlobalSkew();
5:      $s_{\max} := \min\{s \in \mathbb{N} \,|\, \forall v \in N_u :\ 2\hat{\mathcal{G}}/\sigma^{s-2} < \kappa_{\{u,v\}}\}$;
6:      for $s = 2, \ldots, s_{\max}$ do
7:          $N_u^s := N_u^1$;
8:      end for
9:      $T_2 := L_u + \hat{\mathcal{G}} + (1+\rho)\hat{\mathcal{G}}/\mu$;
10:     $T_3 := T_2 + \left(\frac{8(1+\mu)}{(1-\rho)\mu} + \frac{4+(12s+16)\mu}{\mu}\right)\hat{\mathcal{G}}$;
11:     for $s = 4, \ldots, s_{\max}$ do
12:         $T_s := T_{s-1} + \left(\frac{8(1+\mu)}{(1-\rho)\mu\sigma^{s-4}} + \frac{(2+(6s+2)\mu)(\sigma+1)}{\mu\sigma^{s-3}}\right)\hat{\mathcal{G}}$;
13:     end for
14:     $T_1 := T_3 + \frac{\sigma}{\sigma-1}\left(\frac{8(1+\mu)}{(1-\rho)\mu} + \frac{(2+(6s+2)\mu)(\sigma+1)}{\mu\sigma}\right)\hat{\mathcal{G}} + \hat{\mathcal{G}}$;   // set $T_1^{(r+1)}$
15:     $\mathbf{T} := \{T_1, \ldots, T_{s_{\max}}\}$;
16: end if
17: if $s = 1$ then
18:     $N_{new} := N_u \setminus N_u^1$;
19:     $N_u^1 := N_u^1 \cup N_{new}$;
20: else if $s = 2$ then
21:     $N_u^1 := N_u^1 \setminus \{v \in N_{new} \,|\,$ (neighbor_removed,$v$,$r$) or no (new_neighbor,$v$,$r$) message received $\}$
22:     $N_u^2 := N_u^1$;
23: else
24:     $N_u^s := N_u^{s-1}$;
25: end if
```

During each round, only new neighbors that are in N_u at the beginning of the round are added to the sets $N_u^1, \ldots, N_u^{s_{\max}}$, i.e., new neighbors that appear after T_1 are ignored until the next round starts. Thus, at time T_1 new neighbors are identified and stored in a set N_{new} (see Line 18). A new neighbor v is only accepted permanently if v also accepts u as a neighbor in the same round. In order to guarantee this, u sends a (new_neighbor,u,r) message. From the second and third property required of $\hat{\mathcal{G}}$ we know that at local time T_2 node u will have received a (new_neighbor,v,r) message from v exactly if v added u to N_v^1 at the time when round r started at v.[1] This strategy is however not sufficient to guarantee consistency of neighbors' decisions. If an edge $\{u, v\}$ fails after a (new_neighbor,v,r) message was sent by v, but the edge reappears before u reaches round r and the message is still delivered, u might believe that v added u to N_v^1 *after* the edge reappeared, and therefore adds it as well. In order to identify this situation, node v also sends a (neighbor_removed,v,r) message whenever v believes that the edge reappears after it failed in round r.

Having sorted this out, each new neighbor is added to N_u^2 at time T_2 (see Line 22), provided that u received (*i*) a (new_neighbor,v,r) message and (*ii*) no (neighbor_removed,v,r). Any other

---

[1] Note that it is possible that a node u receives a (new_neighbor,v,r) message before it starts round r.

set N_u^s, $s \in \{3, \dots, s_{\max}\}$, is updated by simply setting it to N_u^{s-1} at time T_s (see Line 24).

Since merely new neighbors that have been present at the beginning of the round are added, it is possible that some set N_u^s does not contain nodes in N_u for more than one round. On the contrary, as mentioned before, whenever u detects that an edge $\{u, v\}$ has *disappeared*, node v is removed from all sets N_u^s immediately. Thus, it is possible that when N_u^s is set to N_u^{s-1} at time T_s, the set N_u^{s-1} may have changed since time T_{s-1}; however, it always holds that $N_u^s \subseteq N_u^{s-1} \subseteq N_u$ for all $s \in \{2, \dots, s_{\max}\}$.

While the sets N_u^s and the corresponding update times are clearly defined, an algorithm has more freedom when it comes to determining the progress rate of its logical clock. As stated in the previous section, an algorithm must ensure that the logical clock value increases continuously. An algorithm $\mathcal{A}$ that belongs to the class $\mathfrak{A}$ imposes a somewhat stronger constraint on the logical clock rate that depends on the parameter $\mu > \rho/(1-\rho)$: The logical clock rate must always be at least the progress rate of the hardware clock, but at most a factor of $1+\mu$ larger. In other words, the logical clock rate is the hardware clock rate times a *clock rate multiplier* r_v, where $r_v(t) \in [1, 1+\mu]$ at any time t. Moreover, an algorithm is forced to set its clock rate multiplier to 1 or to $1+\mu$ under certain circumstances. These additional conditions depend on a parameter $\kappa_{\{u,v\}}$, which is defined for each $v \in N_u$. For all $v \in N_u$, it must hold that

$$\kappa_{\{u,v\}} \geq 4\epsilon_{\{u,v\}} + 8\mu\tau_{\{u,v\}}. \tag{3}$$

We further demand that $\kappa_{\{u,v\}} = \kappa_{\{v,u\}}$, which is easy to accomplish if the uncertainty of the estimates is assumed to be the same in both directions and the two nodes use the same parameters.[2] Given these parameters, we can now state the conditions that restrict the clock rate multiplier.

The fast and slow conditions. The *fast mode condition* determines when an algorithm must set the clock rate multiplier to $1+\mu$.

Definition 4.1 (Fast Mode Condition). *Given a graph $G = (V, E)$ and κ_e for all $e \in E$, an algorithm satisfies the* fast mode condition *if and only if for all $s \in \mathbb{N}$, for all $u \in V$, and all times t it holds that*

$$\mathbf{FC}: \quad \left.\begin{array}{l} \exists w \in N_u^s(t):\ L_w(t) - L_u(t) \geq s\kappa_{\{u,w\}} \\ \forall v \in N_u^s(t):\ L_u(t) - L_v(t) \leq s\kappa_{\{u,v\}} + 2\mu\tau_{\{u,v\}} \end{array}\right\} \Rightarrow r_u(t) = 1+\mu.$$

The so-called *slow mode condition* specifies when the clock rate multiplier must be 1.

Definition 4.2 (Slow Mode Condition). *Given a graph $G = (V, E)$ and κ_e for all $e \in E$, an algorithm satisfies the* slow mode condition *if and only if for all $s \in \mathbb{N}$, for all $u \in V$, and all times t it holds that*

$$\mathbf{SC}: \quad \left.\begin{array}{l} \forall v \in N_u^s(t):\ L_v(t) - L_u(t) \leq \left(s + \frac{1}{2}\right)\kappa_{\{u,v\}} + \delta \\ \exists w \in N_u^s(t):\ L_u(t) - L_w(t) \geq \left(s + \frac{1}{2}\right)\kappa_{\{u,w\}} - \delta \end{array}\right\} \Rightarrow r_u(t) = 1,$$

where $\delta > 0$ is an arbitrarily small constant.

---

[2] Otherwise, u and v could establish a common $\kappa_{\{u,v\}}$ by exchanging $\epsilon_{\{u,v\}}$ and $\epsilon_{\{v,u\}}$ and compute $\kappa_{\{u,v\}}$ based on the larger of the two values.

Max estimates. As in [8, 12, 13, 14], each node maintains a local estimate M_u of the maximum logical clock value in the network, and makes sure never to exceed it. As this is a standard technique we omit the implementation details, and note only that max estimates satisfy the following constraint: if the dynamic graph has a diameter of D, then for all $t \geq 0$ and for all nodes u we have

$$M_u(t) \leq \max_{v \in V} \{L_v(t)\}, \tag{4}$$

$$\forall t \geq 2D: \; M_u(t) > \max_{v \in V} \{L_v(t - 2D)\}, \tag{5}$$

$$M_u(t) \geq L_u(t), \tag{6}$$

$$M_u(t) \geq \max_{v \in N_u(t)} \left\{ \tilde{L}_u^v(t) - \epsilon_{\{u,v\}} \right\}, \tag{7}$$

That is, the max estimate of any node is never more than the true maximum, and it represents the true maximum from $2D$ time units ago, where D is the time required to complete a flood in the dynamic graph. (The factor 2 arises from the fact that nodes do not constantly flood the network with max estimates. It can be reduced to $(1 + \iota)$, where ι is an arbitrarily small constant, by starting floods sufficiently often.) In addition, (6) asserts that nodes cannot set their logical clock ahead of their max estimate, and (7) asserts that the max estimate always reflects the logical clock values of immediate neighbors. The max estimate $M_u(t)$ of a node is only used to determine the clock rate multiplier $r_u(t)$ if neither **FC** nor **SC** are satisfied. In this case, if $L_u(t) = M_u(t)$ and for all neighbors v, $L_v(t) \leq L_u(t)$, it must hold $r_u(t) = 1$. Further, if $L_u = M_u(t)$ and for all neighbors v, $L_v(t) \geq L_u(t)$, it must hold $r_u = 1 + \mu$.

These are all the conditions that an algorithm $\mathcal{A} \in \mathfrak{A}$ must satisfy. We will now show that such an algorithm can be implemented.

4.2 Implementation of an Algorithm $\mathcal{A} \in \mathfrak{A}$

It is clear that an algorithm $\mathcal{A}$ can maintain sets $N_u^1, \ldots, N_u^{s_{\max}}$ and update them at the correct times by running Algorithm 1. The conditions given by the max estimates $M_u(t)$ can easily be satisfied by using the estimate layer. We have to show that an algorithm can implement rules guaranteeing that **FC** and **SC** are always satisfied.

The following rule, called the *fast mode rule*, ensures that **FC** holds.

Definition 4.3 (Fast Mode Rule)**.** *An algorithm implements the* fast mode rule *if and only if for all $s \in \mathbb{N}$, for all $u \in V$, and all times t it holds that*

$$\left.\begin{array}{l} \exists w \in N_u^s(t): \; \tilde{L}_u^w(t) - L_u(t) \geq s\kappa_{\{u,w\}} - \epsilon_{\{u,w\}} \\ \forall v \in N_u^s(t): \; L_u(t) - \tilde{L}_u^v(t) \leq s\kappa_{\{u,v\}} + \epsilon_{\{u,v\}} + 2\mu\tau_{\{u,v\}} \end{array}\right\} \Rightarrow r_u(t) = 1 + \mu$$

The corresponding rule for **SC**, the *slow mode rule*, is defined as follows.

Definition 4.4 (Slow Mode Rule)**.** *An algorithm implements the* slow mode rule *if and only if for all $s \in \mathbb{N}$, for all $u \in V$, and all times t it holds that*

$$\left.\begin{array}{l} \forall v \in N_u^s(t): \; \tilde{L}_u^v(t) - L_u(t) \leq \left(s + \frac{1}{2}\right)\kappa_{\{u,v\}} + \delta_{\{u,v\}} + \epsilon_{\{u,v\}} \\ \exists w \in N_u^s(t): \; L_u(t) - \tilde{L}_u^w(t) \geq \left(s + \frac{1}{2}\right)\kappa_{\{u,w\}} - \delta_{\{u,w\}} - \epsilon_{\{u,w\}} \end{array}\right\} \Rightarrow r_u(t) = 1,$$

where $\delta_e \in (0, (\kappa_e - 4\epsilon_e - 4\mu\tau_e)/6]$ for all $e = \{u, v\}$, $v \in N_u^s(t)$.

The following lemma states that implementing these rules indeed guarantee that **FC** and **SC** are satisfied.

Lemma 4.2. *If an algorithm implements the fast and the slow mode rule, it satisfies the fast and the slow mode condition, respectively.*

Proof. Assume that the precondition of **FC** holds for some $s \in \mathbb{N}$, i.e., for all t we have that

$$\begin{aligned} \exists w \in N_u^s(t) \quad &: \quad L_w(t) - L_u(t) \geq s\kappa_{\{u,w\}} \\ \forall v \in N_u^s(t) \quad &: \quad L_u(t) - L_v(t) \leq s\kappa_{\{u,v\}} + 2\mu\tau_{\{u,v\}}. \end{aligned}$$

Since $L_w(t) \leq L_u^w(t) + \epsilon_{\{u,w\}}$ for all $w \in N_u^s(t)$, we immediately get that

$$\begin{aligned} \exists w \in N_u^s(t) \quad &: \quad \left(\tilde{L}_u^w(t) + \epsilon_{\{u,w\}}\right) - L_u(t) \geq s\kappa_{\{u,w\}} \\ \forall v \in N_u^s(t) \quad &: \quad L_u(t) - \left(\tilde{L}_u^v(t) + \epsilon_{\{u,v\}}\right) \leq s\kappa_{\{u,v\}} + 2\mu\tau_{\{u,v\}}. \end{aligned}$$

As the algorithm implements the fast mode rule, it must hold that $r_u(t) = 1 + \mu$ as desired.

Now, define $\delta := \min_{e \in E}\{\delta_e\} > 0$ and assume that the precondition of **SC** holds for some $s \in \mathbb{N}$, i.e., for all t we have that

$$\begin{aligned} \forall v \in N_u^s(t) \quad &: \quad L_v(t) - L_u(t) \leq \left(s + \frac{1}{2}\right)\kappa_{\{u,v\}} + \delta \\ \exists w \in N_u^s(t) \quad &: \quad L_u(t) - L_w(t) \geq \left(s + \frac{1}{2}\right)\kappa_{\{u,w\}} - \delta. \end{aligned}$$

We can now use that $L_w(t) \geq L_u^w(t) - \epsilon_{\{u,w\}}$ for all $w \in N_u^s(t)$ to see that

$$\begin{aligned} \forall v \in N_u^s(t) \quad &: \quad \left(\tilde{L}_u^v(t) - \epsilon_{\{u,v\}}\right) - L_u(t) \leq \left(s + \frac{1}{2}\right)\kappa_{\{u,v\}} + \delta \leq \left(s + \frac{1}{2}\right)\kappa_{\{u,v\}} + \delta_{\{u,v\}} \\ \exists w \in N_u^s(t) \quad &: \quad L_u(t) - \left(\tilde{L}_u^w(t) - \epsilon_{\{u,w\}}\right) \geq \left(s + \frac{1}{2}\right)\kappa_{\{u,w\}} - \delta \geq \left(s + \frac{1}{2}\right)\kappa_{\{u,w\}} - \delta_{\{u,v\}}. \end{aligned}$$

In this case, the assumption that the algorithm implements the slow mode rule implies that $r_u(t) = 1$. □

It remains to show that the clock rate multiplier is always well-defined, i.e., the two rules never cause a conflict.

Lemma 4.3. *If $\kappa_e > 4\epsilon_e + 4\mu\tau_e$ for all $e \in E$, the fast mode rule and the slow mode rule are mutually exclusive.*

Proof. Assume that at node u at time t the prerequisites of the fast mode rule are met for some $s \in \mathbb{N}$, and the prerequisites of the slow mode rule are met for $s' \in \mathbb{N}$. We will first assume that $s \leq s'$, which implies that $N_u^{s'} \subseteq N_u^s$. In this case, according to the slow mode rule, there is a node $w \in N_u^{s'}(t)$ such that

$$L_u(t) - \tilde{L}_u^w(t) \geq \left(s' + \frac{1}{2}\right)\kappa_{\{u,w\}} - \delta_{\{u,w\}} - \epsilon_{\{u,w\}}.$$

However, we know that $w \in N_u^s(t)$ because $N_u^{s'}(t) \subseteq N_u^s(t)$. Hence, according to the prerequisites of the fast mode rule, we know that

$$L_u(t) - \tilde{L}_u^w(t) \leq s\kappa + \epsilon_{\{u,w\}} + 2\mu\tau_{\{u,w\}}.$$

It follows that

$$\begin{aligned}\left(s' + \frac{1}{2}\right)\kappa_{\{u,w\}} - \delta_{\{u,w\}} - \epsilon_{\{u,w\}} &\leq s\kappa + \epsilon_{\{u,w\}} + 2\mu\tau_{\{u,w\}} \\ &\leq s'\kappa + \epsilon_{\{u,w\}} + 2\mu\tau_{\{u,w\}},\end{aligned}$$

implying that $\kappa_{\{u,w\}} \leq 4\epsilon_{\{u,w\}} + 4\mu\tau + 2\delta_{\{u,w\}}$. Since $0 < \delta_{\{u,w\}} \leq (\kappa_{\{u,w\}} - 4\epsilon_{\{u,w\}} - 4\mu\tau_{\{u,w\}})/6$, we get the contradiction that $4\epsilon_{\{u,w\}} + 4\mu\tau_{\{u,w\}} + 2\delta_{\{u,w\}} < \kappa_{\{u,w\}}$.

Assume now that $s \geq s' + 1$, implying that $N_u^s(t) \subseteq N_u^{s'}(t)$. The prerequisites of the fast mode rule say that there is a node $w \in N_u^s(t)$ such that

$$L_u(t) - \tilde{L}_u^w(t) \geq s\kappa_{\{u,w\}} - \epsilon_{\{u,w\}}.$$

Since $N_u^s(t) \subseteq N_u^{s'}(t)$, we know that $w \in N_u^{s'}(t)$. The prerequisites of the slow mode rule imply that

$$L_u(t) - \tilde{L}_u^w(t) \leq \left(s' + \frac{1}{2}\right)\kappa_{\{u,w\}} + \delta_{\{u,w\}} + \epsilon_{\{u,w\}}.$$

In this case, it holds that

$$\left(s' + 1\right)\kappa_{\{u,w\}} - \epsilon_{\{u,w\}} \leq s\kappa_{\{u,w\}} - \epsilon_{\{u,w\}} \leq \left(s' + \frac{1}{2}\right)\kappa_{\{u,w\}} + \delta_{\{u,w\}} + \epsilon_{\{u,w\}}.$$

Hence, we have that $\kappa_{\{u,w\}} \leq 4\epsilon_{\{u,w\}} + 2\delta_{\{u,w\}}$. By using that $\delta_{\{u,w\}} \leq (\kappa_{\{u,w\}} - 4\epsilon_{\{u,w\}} - 4\mu\tau_{\{u,w\}})/6$, we get that $4\epsilon_{\{u,w\}} + 2\delta_{\{u,w\}} < \kappa_{\{u,w\}}$, again a contradiction. $\square$

Assuming that $\kappa_e > 4\epsilon_e + 4\mu\tau_e$ for all $e \in E$, Lemma 4.2 states that adhering to the fast mode and the slow mode rule implies that both the fast mode condition and the slow mode condition are met, while Lemma 4.3 proves that there is never a conflict between the two rules. Note that these two results together imply that the fast and the slow mode condition are also mutually exclusive. We will now analyze the properties of the class $\mathfrak{A}$.

As discussed in Section 3, nodes u and v are not necessarily informed that the edge $\{u,v\}$ appeared or disappeared at the same time. However, it is possible to characterize for which times $v \in N_u^s$ implies that $u \in N_v^s$ as the following lemma shows.

Lemma 4.4. *Assume that $v \in N_u^s(t_1)$ for some $s \in \mathbb{N}$, $s \neq 1$, and a time t_1, and choose $r \in \mathbb{N}$ such that $T_s^{(r)} \leq L_u(t_1) < T_s^{(r+1)}$. For all $s' \in \{1, \ldots, s\}$, let $t_u^{s'}$ and $t_v^{s'}$ denote the times when $L_u(t_u^{s'}) = T_{s'}^{(r)}$ and $L_v(t_v^{s'}) = T_{s'}^{(r)}$, respectively. It holds that $v \in N_u^{s'}(t)$ for all times $t \in [t_u^{s'}, t_1]$ and $u \in N_v^{s'}(t)$ for all times $t \in [t_v^{s'}, t_1 - \tau_{\{u,v\}}]$.*

Proof. We know that v is not removed from N_u at any time in the interval $[t_u^1, t_1]$, i.e., $(u,v) \in E(t)$ for all $t \in [t_u^1, t_1]$. Otherwise we would have that $v \notin N_u^s(t_1)$ because, according to Algorithm 1, a removed neighbor cannot be added to N_u^s again until the time t when $L_u(t) = T_s^{(r+1)}$, which is larger than t_1. Consider any s' for which $t_u^{s'} \leq t_1$. For each such s' we know that v is added to

N_u^s at time $t_u^{s'}$ unless v has already been in $N_u^{s'}$ before the round started. In either case, since v is continuously considered a neighbor until time $t_1 \geq t_u^{s'}$, it follows that $v \in N_u^{s'}(t)$ for all times $t \in [t_u^{s'}, t_1]$.

We will now prove the second claim. First, assume that $u \notin N_v^1(t_v^1)$, implying that $u \notin N_v(t_v^1)$. Thus, node u cannot have received a (new_neighbor,v,r) message from v, which is a contradiction as it would have removed v from N_u^1 at logical time $T_2^{(r)} \leq T_s^{(r)}$, i.e., before time t_1.

If $(v,u) \in E(t)$ at all times $t \in [t_v^1, t_1 - \tau_{\{u,v\}}]$, the claim is obviously true unless v did not receive a (new_neighbor,u,r) message until time t_v^2. According to the second and third assumption on $\hat{\mathcal{G}}^{(r)}$, we have that

$$\begin{aligned} L_v(t_u^1 + \mathcal{T}_{\{u,v\}}) &\leq L_v(t_u^1) + (1+\rho)(1+\mu)\mathcal{T}_{\{u,v\}} \\ &\leq L_u(t_u^1) + \mathcal{G}(t_u^1) + (1+\rho)(1+\mu)\mathcal{T}_{\{u,v\}} \\ &\leq T_1^{(r)} + \hat{\mathcal{G}}^{(r)} + \frac{(1+\rho)\hat{\mathcal{G}}^{(r)}}{\mu} \\ &\leq T_2^{(r)} = L_v(t_v^2), \end{aligned}$$

i.e., indeed v received the (new_neighbor,u,r) message from u in time, added u to the sets $N_v^{s'}$ at the times $t_v^{s'}$, and did not remove it again until time $t_1 - \tau_{\{u,v\}}$.

Therefore, we assume that the edge $\{u,v\}$ failed from the perspective of v at a time $t \in [t_v^1, t_1 - \tau_{\{u,v\}}]$, i.e., $(v,u) \notin E(t)$ and $\{u,v\} \notin E(t')$ for some time $t' \in [t - \tau_{\{u,v\}}, t]$. In this case, u registered that $\{u,v\}$ failed at the latest at time t_1 and removed v from all neighborhood sets $N_u^{s'}$. Thus, as already observed, if the edge failed at time t', it must hold that $t' < t_u^1$ because (u,v) exists throughout the interval $[t_u^1, t_1]$. On the other hand, as we know that (u,v) is in the edge set during $[t_u^1, t_1]$, we also have that $(v,u) \in E(t)$ at times $t \in [t_u^1 + \tau_{\{u,v\}}, t_1 - \tau_{\{u,v\}}]$. Therefore, at the latest at time $t_u^1 + \tau_{\{u,v\}}$ (when it notices the reappearance of the edge), v will send a (neighbor_removed,v,r) message to u that is successfully delivered not later than at time $t_u^1 + \tau_{\{u,v\}} + \mathcal{T}_{\{u,v\}}$. According to the third assumption on $\hat{\mathcal{G}}^{(r)}$, we have that

$$L_u(t_u^1 + \mathcal{T}_{\{u,v\}}) \leq L_u(t_u^1) + (1+\rho)(1+\mu)(\tau_{\{u,v\}} + \mathcal{T}_{\{u,v\}}) \leq T_1^{(r)} + \frac{(1+\rho)\hat{\mathcal{G}}^{(r)}}{\mu} < T_2^{(r)} = L_u(t_u^2).$$

Therefore, we get the contradiction that u did not add v to N_u^2 at time t_u^2 (instead, v is removed also from N_u^1), which completes the proof. □

The claimed bounds on the clock skew and the stabilization time are proved in the subsequent section.

5 Analysis

In this section, we analyze the algorithm described in Section 4 and bound its worst-case global and dynamic gradient skew.

5.1 The Global Skew

Like its predecessors in [9, 12], Algorithm $\mathcal{A}^{\mathrm{DW}}$ achieves an asymptotically optimal global skew of $\mathcal{O}(D)$, where D is now defined as the dynamic diameter of the graph (see Section 3).

Theorem 5.1. *Algorithm* $\mathcal{A}^{\mathrm{DW}}$ *achieves a global skew of* $2(1+\rho)D \in \mathcal{O}(D)$ *in networks of diameter* D*.*

Proof. The proof is similar to the ones in [8, 12]. First, consider the node u with the largest clock in the network, i.e., $L_u(t) = \max_{v\in V}\{L_v(t)\}$. From (4) and (6) we have $M_u(t) = L_u(t)$, that is, node u *knows* that it has the largest clock value. In addition, by (3) and (7) we have $M_u(t) \geq \max_{v\in N_u(t)}\{L_v(t) - \epsilon_{\{u,v\}}\}$ and for any neighbor v, $(1-\lambda)\kappa_u^v(t) \geq (1-\lambda)\kappa_{\{u,v\}}^{\infty} > \epsilon_{\{u,v\}}$. Hence, u's estimates of its neighbors' clock values cannot be large enough for **FC** to hold. Together with the fact that $L_u(t) = M_u(t)$, this forces u to be in slow mode.

For this reason, the maximum clock value in the network increases at most at rate $1+\rho$, the maximum hardware clock rate. From (5) it follows that for any node v we have $M_v(t) > \max_{w\in V}\{L_w(t)\} - 2(1+\rho)D$; that is, the max estimates of all nodes are "not too far off" the true maximum.

Now consider a node u with the *smallest* clock in the network, $L_u(t) = \min_{v\in V}\{L_v(t)\}$, and suppose that $L_u(t) = \max_{v\in V}\{L_v(t)\} - 2D(1+\rho)$, in other words, that there is a large gap between the smallest and largest clocks. Because $M_v(t) > \max_{w\in V}\{L_w(t)\} - 2(1+\rho)D$, we immediately obtain $L_u(t) < M_u(t)$, so node u knows it is behind. In addition, since node u has the smallest clock, for each $v \in N_u(t)$ we have $L_v(t) \geq L_u(t)$, and hence $\tilde{L}_u^v(t) \geq L_u(t) - \epsilon_{\{u,v\}} > L_u(t) - (1/2-\lambda)\kappa_u^v(t)$, which means that **SC** does not hold (all of u's neighbors are too far ahead). Together with the fact that $L_u(t) < M_u(t)$, this forces node u to be in fast mode.

We have shown that whenever there is a large enough skew such that $\min_{v\in V}\{L_v(t)\} = \max_{v\in V}\{L_v(t)\} - 2(1+\rho)D$, all nodes with the smallest logical clock value, $\min_{v\in V}\{L_v(t)\}$, will be in fast mode, and all nodes with the largest logical clock value will be in slow mode. A node in fast mode increases its logical clock at a rate of at least $(1+\mu)(1-\rho)$, and a node in slow mode increases its logical clock at a rate of at most $(1+\rho)$. Because $\mu > \rho/(1-\rho)$, $(1+\mu)(1-\rho) > 1+\rho$, so the nodes that are the most behind cannot fall behind any further. The continuity of logical clocks thus ensures that the global skew never exceeds $2(1+\rho)D$. □

As explained in Section 4.1, we assume that each node maintains an upper bound $\bar{\mathcal{G}}$ on the global skew of the network. This can be done dynamically, by running an estimation protocol alongside the clock synchronization algorithm, or the bound $\bar{\mathcal{G}}$ can be computed statically based on known properties of the network. For simplicity, we use a single global skew estimate $\bar{\mathcal{G}}$ throughout the paper. All algorithms and proofs can however be adapted to a scenario where each node maintains an individual and possibly dynamic upper bound on the global skew.

5.2 Analysis of the Gradient Skew

The gradient property that we will show depends on a certain *gradient sequence.* A gradient sequence is defined as follows.

Definition 5.1 (Gradient Sequences)**.** *A non-increasing sequence of positive values* $C = \{C_s\}_{s\in\mathbb{N}}$ *is a* gradient sequence *at time* t*, if* $C_1 > 2\mathcal{G}(t)$*.*

Throughout our analysis, we will assume that the condition $C_1 > 2\mathcal{G}(t)$ holds at all considered times. We will discuss this time-dependent condition separately in Corollary 5.11. Moreover, we assume that $s_{\max}(u) = \infty$ for all nodes $u \in V$. We will see later, in Corollary 5.12, that there is no need for the algorithm to implement the fast and slow mode rules on any level higher than $s_{\max}^{(r)}(u)$

at logical times in the interval $[T_1^{(r)}, T_1^{(r+1)})$ for any $r \in \mathbb{N}$. We will also make frequent use of the following definitions.

Definition 5.2 (Level-s Paths)**.** *Given the gradient sequence C, we define for all $u_0 \in V$, $s \in \mathbb{N}$, and all times t the set of* level-s paths starting at node u_0 at time t *to be*

$$P_{u_0}^s(t) := \{p = (u_0, \dots, u_k) \,|\, \kappa_p \leq C_s \wedge \forall i \in \{0, \dots, k-1\} : \ \{u_i, u_{i+1}\} \in E(t)\}.$$

Definition 5.3. *For all paths $p = (u, \dots, v)$, all $s \in \mathbb{N}$, and all times t we define*

$$\xi_p^s(t) := L_u(t) - L_v(t) - s\kappa_p.$$

Given a gradient sequence C, we further define for all $u \in V$ that

$$\Xi_u^s(t) := \max_{p \in P_u^s(t)} \{\xi_p^s(t)\}.$$

Definition 5.4. *For all paths $p = (u, \dots, v)$, all $s \in \mathbb{N}$, and all times t we define*

$$\psi_p^s(t) := L_v(t) - L_u(t) - \left(s + \frac{1}{2}\right)\kappa_p.$$

Furthermore, given a gradient sequence C, we set for all $u \in V$

$$\Psi_u^s(t) := \max_{p \in P_u^s(t)} \{\psi_p^s(t)\}.$$

A certain clock skew that depends on the given gradient sequence C is allowed. If this bound is not violated, we say that the system is *legal.* Theconcept of legality is formally defined as follows.

Definition 5.5 (Legality)**.** *Given a weighted, dynamic graph G and a gradient sequence C, for each $s \in \mathbb{N}$ the system is s-*legal with respect to C at time t and node $u \in V$, *if and only if it holds that*

$$\Psi_u^s(t) < \frac{C_s}{2}$$

and

$$\Xi_u^s(t) < C_s.$$

*The system is C-*legal at t and u *if it is s-legal for all $s \in \mathbb{N}$ at node u and time t with respect to C.*

We start our analysis by showing a few straightforward properties.

Lemma 5.2. *The following statements hold at all times t and all nodes $u \in V$.*

(i) The system is 1-legal.

(ii) $\forall s, s' \in \mathbb{N}$, $s' \leq s$, $P_u^s(t) \subseteq P_u^{s'}(t)$.

(iii) $\forall s, s' \in \mathbb{N}$, $s' \leq s$, $\Psi_u^s(t) \leq \Psi_u^{s'}(t)$

(iv) $\forall s, s' \in \mathbb{N}$, $s' \leq s$, $\Xi_u^s(t) \leq \Xi_u^{s'}(t)$

(v) If for some $s \in \mathbb{N}$ the system is s-legal and $C_s = C_{s+1}$, then the system is also $(s+1)$-legal.

Proof. Statement (i) holds by definition as for any path $p = (u, \dots, v)$ and any time t we have

$$\max\{\xi_p^1(t), \psi_p^1(t)\} \leq |L_u(t) - L_v(t)| \leq \mathcal{G}(t) < \frac{C_1}{2}.$$

Statement (ii) is a direct consequence of Algorithm 1 and Definition 5.1 since for all nodes v and all times t it holds that $N_v^s(t) \subseteq N_v^{s'}(t)$ and $C_s \leq C_{s'}$. Statements (iii) and (iv) thus follow from Statement (ii). Finally, Statement (v) is deduced from Statements (iii) and (iv). □

The s-legality can be used to prove a bound on the clock skew between two nodes u and v.

Lemma 5.3. *Assume that for some $s \in \mathbb{N}$, the system is s-legal at node $u \in V$ and time t. If $p = (u, \dots, v) \in P_u^s(t)$, then $|L_u(t) - L_v(t)| < (s+1)C_s$.*

Proof. Since the system is s-legal at u at time t, we have that

$$L_u(t) - L_v(t) - sC_s \leq L_u(t) - L_v(t) - s\kappa_p = \xi_p^s(t) \leq \Xi_u^s(t) < C_s$$

and

$$L_v(t) - L_u(t) - \left(s + \frac{1}{2}\right) C_s \leq L_v(t) - L_u(t) - \left(s + \frac{1}{2}\right) \kappa_p = \psi_p^s(t) \leq \Psi_p^s(t) < \frac{C_s}{2}.$$

These inequalities imply the claimed bound on the clock skew. □

In order to prove the claimed bound on the stabilization time, we require a *stabilization condition*, which depends on the s-legality of the system for a certain $s \in \mathbb{N}$.

Definition 5.6 (Stabilization Condition)**.** *Given $u \in V$, $s \in \mathbb{N}$, $s \neq 1$, and a time interval $[t_0, t_1]$, we say that u satisfies the* stabilization condition on level s *during $[t_0, t_1]$ if and only if*

- *For all $s' \in \{2, \dots, s-1\}$, the system is s'-legal at u and all $t \in [t_0 - C_{s'-1}/((1+\rho)\mu), t_1]$.*
- *We have for all $t \in [t_0, t_1]$ and all $r \in \mathbb{N}$ that*

$$|L_u(t) - T_s^{(r)}| \geq \frac{(1 + (s+1)\mu)C_{s-1}}{\mu}.$$

Given the stabilization condition, we can conclude that certain paths must exist for a certain period of time.

Lemma 5.4. *Assume that node $u \in V$ satisfies the stabilization condition on level $s \in \mathbb{N}$, $s \neq 1$, during the time interval $[t_0, t_1]$ and let $p = (u, \dots, v) \in P_u^s(t_1)$. Then $p \in P_u^s(t)$ for all*

$$t \in \left[t_0 - \frac{C_{s-1}}{(1+\rho)\mu}, t_1\right].$$

Furthermore, if $w \in N_v^s(t_1)$ and $\xi^s_{(u,\dots,v,w)}(t_1 - \tau_{\{v,w\}}) \geq 0$ or $\xi^s_{(w,v,\dots,u)}(t_1 - \tau_{\{v,w\}}) \geq 0$, then $(u, \dots, v, w) \in P_u^s(t)$ for all

$$t \in \left[t_0 - \frac{C_{s-1}}{(1+\rho)\mu}, t_1 - \tau_{\{v,w\}}\right],$$

and this time interval is not empty.

Proof. Let $r \in \mathbb{N}$ be maximal such that $L_u(t) \geq T_s^{(r)}$ for all $t \in [t_0, t_1]$. Since clock values are continuous, the stabilization condition yields that

$$L_u(t) \in \left[T_s^{(r)} + \frac{(1+(s+1)\mu)C_{s-1}}{\mu}, T_s^{(r+1)} - \frac{(1+(s+1)\mu)C_{s-1}}{\mu}\right] \tag{8}$$

for all $t \in [t_0, t_1]$. Set $X := \{(x,y) \,|\, \{x,y\} \in p\} \cup \{(v,w)\}$, i.e., the set of (directed) edges containing the edges of p with arbitrary orientation and, in order to prove the second statement, the edge (v,w). Now let

$$t' \in \left[t_0 - \frac{C_{s-1}}{(1+\rho)\mu}, t_1\right]$$

be minimal with the property that $X \subseteq E(t)$ for all $t \in [t', t_1]$. Because $p \in P_u^s(t_1)$, the definition of t' guarantees that for any edge $(x,y) \in X$ it holds that u and x are connected by a subpath of p that exists at time t'. Hence, we can apply Lemma 5.3 for $s-1$, as the system is $(s-1)$-legal at u and time t' (recall that the system is always 1-legal). We get that

$$L_x(t') < L_u(t') + sC_{s-1} \overset{(8)}{<} T_s^{(r+1)}$$

and

$$L_x(t') > L_u(t') - sC_{s-1} \overset{(8)}{>} T_s^{(r)}.$$

Therefore, if

$$t' > t_0 - \frac{C_{s-1}}{(1+\rho)\mu},$$

we can apply Lemma 4.4 (for $s' = s$) to all pairs $(x,y) \in X$ in order to see that a time $\tilde{t}$ exists with

$$t' > \tilde{t} \in \left[t_0 - \frac{C_{s-1}}{(1+\rho)\mu}, t_1\right]$$

such that $X \subseteq E(t)$ for all $t \in [\tilde{t}, t_1]$, contradicting the minimality property of t'. Thus,

$$t' = t_0 - \frac{C_{s-1}}{(1+\rho)\mu}.$$

In particular, $p \in P_u^s(t)$ for all

$$t \in \left[t_0 - \frac{C_{s-1}}{(1+\rho)\mu}, t_1\right]$$

as claimed.

Define $p' := (u, \ldots, v, w)$. To prove the second statement of the lemma, we first show that $L_w(t_1 - \tau_{\{v,w\}}) \geq T_1^{(r)}$. Recall that $\kappa_{\{v,w\}} > (1+\rho)\mu\tau_{\{v,w\}}$, implying that $\kappa_{p'} > (1+\rho)\mu\tau_{\{v,w\}}$. Since by assumption $\xi_{p'}^s(t_1 - \tau_{\{v,w\}}) \geq 0$ or $\xi_{(w,v,\ldots,u)}^s(t_1 - \tau_{\{v,w\}}) \geq 0$, we have that

$$\begin{aligned}
\frac{C_1}{2} &> \mathcal{G}(t_1 - \tau_{\{v,w\}}) \\
&\geq |L_u(t_1 - \tau_{\{v,w\}}) - L_w(t_1 - \tau_{\{v,w\}})| \\
&= \max\{\xi_{p'}^s(t_1 - \tau_{\{v,w\}}), \xi_{(w,v,\ldots,u)}^s(t_1 - \tau_{\{v,w\}})\} + s\kappa_{p'} \\
&> \kappa_{p'} \\
&> (1+\rho)\mu\tau_{\{v,w\}},
\end{aligned} \tag{9}$$

i.e., $\tau_{\{v,w\}} < C_1/(2(1+\rho)\mu)$. Therefore, if $L_w(t-\tau_{\{v,w\}}) < T_1^{(r)}$, we had that

$$\begin{array}{rcl} L_u(t_1) & \leq & L_u(t_1-\tau_{\{v,w\}}) + (1+\rho)(1+\mu)\tau_{\{v,w\}} \\ & \leq & L_w(t_1-\tau_{\{v,w\}}) + \mathcal{G}(t_1-\tau_{\{v,w\}}) + (1+\mu)C_1/(2\mu) \\ & < & T_1^{(r)} + (1+2\mu)C_1/(2\mu) \\ & \overset{(8)}{<} & L_u(t_1), \end{array}$$

a contradiction. Thus, if t_w denotes the time when $L_w(t_w) = T_1^{(r)}$, Lemma 4.4 shows that $v \in N_w^1(t)$ at all times $t \in [t_w, t_1 - \tau_{\{v,w\}}] \neq \emptyset$.

Now we claim that $p' \in P_u^s(t_1 - \tau_{\{v,w\}})$. Assuming for the sake of contradiction that the statement is false, let $s' \in \{1, \ldots, s-1\}$ be maximal such that $v \in N_w^{s'}(t_1 - \tau_{\{v,w\}})$, i.e., $p' \in P_u^{s'}(t_1 - \tau_{\{v,w\}})$. Since the system is s'-legal at u and time $t_1 - \tau_{\{v,w\}}$, by assumption we either have

$$\begin{array}{rcl} 0 & \leq & \xi_{p'}^{s}(t_1-\tau_{\{v,w\}}) \\ & = & \xi_{p'}^{s'}(t_1-\tau_{\{v,w\}}) - (s-s')\kappa_{p'} \\ & < & C_{s'} - \kappa_{p'}, \end{array} \tag{10}$$

or

$$\begin{array}{rcl} 0 & \leq & \xi_{(w,v,\ldots,u)}^{s}(t_1-\tau_{\{v,w\}}) \\ & = & \xi_{(w,v,\ldots,u)}^{s'}(t_1-\tau_{\{v,w\}}) - (s-s')\kappa_{p'} \\ & < & C_{s'} - \kappa_{p'}. \end{array} \tag{11}$$

Both cases imply that $\kappa_{p'} < C_{s'}$. We get that

$$\tau_{\{v,w\}} \overset{(3)}{<} \frac{\kappa_{\{v,w\}}}{(1+\rho)\mu} < \frac{C_{s'}}{(1+\rho)\mu}.$$

Moreover, $L_w(t_1 - \tau_{\{v,w\}}) < T_{s'+1}^{(r)}$, since otherwise Lemma 4.4 yielded that $v \in N_w^{s'+1}(t_1 - \tau_{\{v,w\}})$, which is impossible due to the maximality of s' and the assumption that $v \notin N_w^s(t_1 - \tau_{\{v,w\}})$. Thus, invoking Lemma 5.3 leads to the contradiction

$$\begin{array}{rcl} L_u(t_1) & \leq & L_u(t_1-\tau_{\{v,w\}}) + (1+\rho)(1+\mu)\tau_{\{v,w\}} \\ & < & L_w(t_1-\tau_{\{v,w\}}) + (s'+1)C_{s'} + \dfrac{(1+\mu)C_{s'}}{\mu} \\ & < & T_{s'+1}^{(r)} + \dfrac{(1+(s'+2)\mu)C_{s'}}{\mu} \\ & \overset{(8)}{\leq} & L_u(t_1). \end{array}$$

We conclude that it must hold that $p' \in P_u^s(t_1 - \tau_{\{v,w\}})$. Furthermore, Inequalities (10) and (11) for the value $s-1$ show that $t_1 - \tau_{\{v,w\}} \geq t_0 - C_{s-1}/((1+\rho)\mu)$, i.e., $[t_0 - C_{s-1}/((1+\rho)\mu), t_1 - \tau_{\{v,w\}}] \neq \emptyset$ as claimed. Finally, using the same reasoning as for the statement that $p \in P_u^s(t)$ for all $t \in [t_0 - C_{s-1}/((1+\rho)\mu), t_1]$, we see that $p' \in P_u^s(t)$ for all $t \in [t_0 - C_{s-1}/((1+\rho)\mu), t_1 - \tau_{\{v,w\}}]$, concluding the proof. □

Assuming that the stabilization condition holds on a certain level s at a node u and that Ξ_u^s is positive throughout a certain interval, we can now give an upper bound on the change of Ξ_u^s.

Lemma 5.5. *Assume that a node $u \in V$ satisfies the stabilization condition on level $s \in \mathbb{N}$, $s \neq 1$, during the time interval $[t_0, t_1]$. If $\Xi_u^s(t) > 0$ for all $t \in [t_0, t_1]$, then a time*

$$t' \in \left[t_0 - \frac{C_{s-1}}{(1+\rho)\mu}, t_0\right]$$

exists that satisfies

$$\Xi_u^s(t_1) - \Xi_u^s(t') \leq L_u(t_1) - L_u(t') - (1-\rho)(1+\mu)(t_1 - t') - (1+\rho)\mu(t_0 - t').$$

Proof. Set $u_0 := u$ and let $p = (u_0, \ldots, u_k)$ be a path maximizing $\Xi_u^s(t)$ at a time $t \in [t_0, t_1]$. Since $\Xi_u^s(t) = \xi_p^s(t)$ is positive, we know that $u_0 \neq u_k$, i.e., the path is non-trivial. As $\kappa_{(u_0,\ldots,u_{k-1})} < \kappa_p \leq C_s$, it follows that $(u_0, \ldots, u_{k-1}) \in P_u^s(t)$. Hence, we have that

$$L_{u_{k-1}}(t) - L_{u_k}(t) \geq s\kappa_{\{u_{k-1},u_k\}},$$

since otherwise

$$\Xi_u^s(t) \geq \xi^s_{(u_0,\ldots,u_{k-1})}(t) > \xi_p^s(t) = \Xi_u^s(t).$$

If for all $v \in N_{u_k}^s(t)$ we have that

$$L_{u_k}(t) - L_v(t) \leq s\kappa_{\{u_k,v\}} + 2\mu\tau_{\{u_k,v\}}, \tag{12}$$

FC states that $r_{u_k}(t) = 1 + \mu$. Now let $\theta \in [t_0, t_1]$ be the minimal time such that for all $t \in (\theta, t_1]$ and any path $p = (u_0, \ldots, u_k)$ maximizing $\Xi_u^s(t)$ Inequality (12) is satisfied for all $v \in N_{u_k}^s(t)$. We get that

$$\Xi_u^s(t_1) - \Xi_u^s(\theta) \leq L_u(t_1) - L_u(\theta) - (1-\rho)(1+\mu)(t_1 - \theta), \tag{13}$$

since, by definition of θ, the nodes at the ends of paths maximizing Ξ_u^s are increasing their logical clocks at least at rate $(1-\rho)(1+\mu)$ throughout the time interval $(\theta, t_1]$.

If $\theta = t_0$, we set $t' := t_0$ and are done. Otherwise, we use the fact that at time θ for some path $p = (u_0, \ldots, u_k)$ maximizing $\Xi_u^s(\theta)$ we have that

$$\exists v \in N_{u_k}^s(\theta): \ L_{u_k}(\theta) - L_v(\theta) > s\kappa_{\{u_k,v\}} + 2\mu\tau_{\{u_k,v\}}. \tag{14}$$

Set $p' := (u_0, \ldots, u_k, v)$. We estimate

$$\begin{aligned}
\xi_{p'}^s(\theta - \tau_{\{u_k,v\}}) &= \xi_{p'}^s(\theta) - (L_u(\theta) - L_u(\theta - \tau_{\{u_k,v\}})) + L_v(\theta) - L_v(\theta - \tau_{\{u_k,v\}}) \\
&\geq \xi_p^s(\theta) + (L_{u_k}(\theta) - L_v(\theta) - s\kappa_{\{u_k,v\}}) \\
&\quad -(L_u(\theta) - L_u(\theta - \tau_{\{u_k,v\}})) + (1-\rho)\tau_{\{u_k,v\}} \\
&\overset{(14)}{>} \xi_p^s(\theta) - (L_u(\theta) - L_u(\theta - \tau_{\{u_k,v\}})) + (1-\rho+2\mu)\tau_{\{u_k,v\}} \\
&= \Xi_u^s(\theta) - (L_u(\theta) - L_u(\theta - \tau_{\{u_k,v\}})) + (1-\rho+2\mu)\tau_{\{u_k,v\}} \qquad (15) \\
&> 0,
\end{aligned}$$

where in the last step we used that $(1-\rho)\mu > 2\rho$. Hence, Lemma 5.4 yields that $p' \in N_u^s(\theta - \tau_{\{u_k,v\}})$, giving

$$\begin{aligned}\Xi_u^s(\theta) - \Xi_u^s(\theta - \tau_{\{u_k,v\}}) &\leq \Xi_u^s(\theta) - \xi_{p'}^s(\theta - \tau_{\{u_k,v\}}) \\ &\overset{(15)}{<} L_u(\theta) - L_u(\theta - \tau_{\{u_k,v\}}) - (1-\rho+2\mu)\tau_{\{u_k,v\}}.\end{aligned}$$

We conclude that

$$\Xi_u^s(t_1) - \Xi_u^s(\theta - \tau_{\{u_k,v\}}) \overset{(13)}{\leq} L_u(t_1) - L_u(\theta - \tau_{\{u_k,v\}}) - (1-\rho)(1+\mu)(t_1 - (\theta - \tau_{\{u_k,v\}})) - (1+\rho)\mu\tau_{\{u_k,v\}}.$$

Recall that $\theta \geq t_0$. According to Lemma 5.4, it also holds that

$$\theta - \tau_{\{u_k,v\}} \in \left[t_0 - \frac{C_{s-1}}{(1+\rho)\mu}, t_1 - \tau_{\{u_k,v\}}\right].$$

Thus, if $\theta - \tau_{\{u_k,v\}} \leq t_0$, the statement follows by setting $t' := \theta - \tau_{\{u_k,v\}}$. Otherwise, we repeat this argumentation inductively, in the next step considering the time interval $[t_0, \theta - \tau_{\{u_k,v\}}]$. Since in each step we go back in time by at least $\min_{x \neq y \in V}\{\tau_{\{x,y\}}\} > 0$ and at most until we reach $t_0 - C_{s-1}/((1+\rho)\mu)$, the induction halts after a finite number of steps at a time

$$t' \in \left[t_0 - \frac{C_{s-1}}{(1+\rho)\mu}, t_0\right],$$

for which it holds that

$$\Xi_u(t_1) - \Xi_u(t') \leq L_u(t_1) - L_u(t') - (1-\rho)(1+\mu)(t_1 - t') - (1+\rho)\mu(t_0 - t')$$

as claimed. □

We need another simple lemma that roughly states that if a node is in fast mode, the slow mode condition **SC** is false. This lemma is basically an alternative formulation of Lemma 4.3.

Lemma 5.6. *Assume that for some node $u \in V$, $s \in \mathbb{N}$, and times $t_0 < t_1$ we have that $T_s^{(r)} \leq L_u(t_0) < L_u(t_1) < T_s^{(r+1)}$, for some $r \in \mathbb{N}$. If*

$$t' := \min\{t \in [t_0, t_1] \mid L_u(t_1) - L_u(t) \leq (1+\rho)(t_1 - t)\} > t_0,$$

then

$$\exists w \in N_u^s(t') : \; L_w(t') - L_u(t') > \left(s + \frac{1}{2}\right)\kappa_{\{u,w\}} \tag{16}$$

or

$$\forall v \in N_u^s(t') : \; L_u(t') - L_v(t') < \left(s + \frac{1}{2}\right)\kappa_{\{u,v\}}. \tag{17}$$

Proof. Assuming the contrary, the logical negation of (16) ∨ (17) is

$$\begin{aligned}&\forall v \in N_u^s(t') : \; L_v(t') - L_u(t') \leq \left(s + \frac{1}{2}\right)\kappa_{\{u,v\}} \\ \wedge \quad &\exists w \in N_u^s(t') : \; L_u(t') - L_w(t') \geq \left(s + \frac{1}{2}\right)\kappa_{\{u,w\}}.\end{aligned}$$

Due to the assumed bounds on the logical clock value of u at any time $t \in [t_0, t_1]$, we know from Algorithm 1 that no neighbors are added to N_u^s during this time interval. Because edges that are removed at time t' are not in $N_u^s(t')$ and N_u^s is finite at all times, there must be a time interval of non-zero length ending at time t' during which N_u^s does not change. Thus, as logical clocks are continuous and $t' > t_0$, a time $\tilde{t} \in [t_0, t')$ exists such that for all $\theta \in [\tilde{t}, t']$ it holds that

$$\begin{aligned} & \forall v \in N_u^s(t') = N_u^s(\theta) : \; L_v(\theta) - L_u(\theta) \leq \left(s + \frac{1}{2}\right) \kappa_{\{u,v\}} + \delta \\ \wedge \quad & \exists w \in N_u^s(t') = N_u^s(\theta) : \; L_u(\theta) - L_w(\theta) \geq \left(s + \frac{1}{2}\right) \kappa_{\{u,w\}} - \delta. \end{aligned}$$

Due to Condition **SC**, we get that $r_u(\theta) = 1$ for all $\theta \in [\tilde{t}, t']$ and thus

$$L_u(t_1) - L_u(\tilde{t}\,) = L_u(t_1) - L_u(t') + L_u(t') - L_u(\tilde{t}\,) \leq (1+\rho)(t_1 - \tilde{t}\,),$$

contradicting the definition of t'. □

The next lemma is essential and it takes quite some work to prove it. It states that if for all level-s paths starting from node u and ending at some node v the stabilization condition is satisfied at node v throughout a certain time interval, then $\Psi_u^s \in \mathcal{O}(\frac{\rho}{\mu} C_{s-1})$ holds in a certain sub-interval.

Lemma 5.7. *Let $u \in V$ and assume at all times $t \in [\underline{t}, \bar{t}\,]$ for all paths $p = (u, \ldots, v) \in P_u^s(t)$ that the stabilization condition on level $s \in \mathbb{N}$, $s \neq 1$, is satisfied at node v. Let $\Delta t := C_{s-1}/((1-\rho)\mu)$. Then, it holds for all*

$$t \in \left[\underline{t} + \Delta t + \frac{C_{s-1}}{(1+\rho)\mu}, \bar{t}\,\right]$$

that

$$\Psi_u^s(t) < 2\rho\Delta t = \frac{2\rho C_{s-1}}{(1-\rho)\mu}.$$

Proof. Set $u_k := u$ and assume for the sake of contradiction that at some time $t_0 \in [\underline{t} + \Delta t + C_{s-1}/((1+\rho)\mu), \bar{t}\,]$ a path $p = (u_k, \ldots, u_0) \in P_u^s(t_0)$ maximizes $\Psi_u^s(t_0) \geq 2\rho\Delta t$, i.e.,

$$\psi_p^s(t_0) = \Psi_u^s(t_0) \geq 2\rho\Delta t. \tag{18}$$

Part I. Getting closer to u_k. We define a sequence of decreasing times $t_0 \geq t_1 \geq \ldots \geq t_{\ell+1} \geq t_0 - \Delta t \geq \underline{t} + C_{s-1}/((1+\rho)\mu)$ inductively. Given t_i, we define

$$t_{i+1} := \min \left\{ t \in [t_0 - \Delta t, t_i] \mid L_{u_i}(t_i) - L_{u_i}(t) \leq (1+\rho)(t_i - t) \right\}. \tag{19}$$

For $i \leq k$, time t_{i+1} is well-defined, as clock values are continuous and $L_{u_i}(t_i) - L_{u_i}(t_i) = 0 = (1+\rho)(t_i - t_i)$. We will see later that the construction can never reach node u_k. We halt the construction at index ℓ if $t_{\ell+1} = t_0 - \Delta t$ or if

$$\exists w \in N_{u_\ell}^s(t_{\ell+1}) : \; L_w(t_{\ell+1}) - L_{u_\ell}(t_{\ell+1}) > \left(s + \frac{1}{2}\right) \kappa_{\{u_\ell, w\}}. \tag{20}$$

Otherwise, note that Lemma 5.4 states that $p \in P_u^s(t)$ for all $t \in [\underline{t}, t_0]$, implying that u_i satisfies the stabilization condition on level s throughout $[\underline{t}, t_0]$. Thus, Lemma 5.6 yields that

$$\forall v \in N_{u_i}^s(t_{i+1}) : \; L_{u_i}(t) - L_v(t) < \left(s + \frac{1}{2}\right) \kappa_{\{u_i, v\}}. \tag{21}$$

We show by induction that for all $i \in \{0, \dots, \ell\}$ and for all $t \in [t_{i+1}, t_i]$ it holds that

$$\xi^s_{(u_i,\dots,u_k)}(t) \geq \Psi^s_u(t_0) + \frac{\kappa_{(u_i,\dots,u_k)}}{2} - (1+\rho)(t_0 - t) + L_{u_k}(t_0) - L_{u_k}(t). \tag{22}$$

Let $t \in [t_1, t_0]$. For the base case we compute

$$\begin{aligned}
\xi^s_{(u_0,\dots,u_k)}(t) &= \xi^s_{(u_0,\dots,u_k)}(t_0) - (L_{u_0}(t_0) - L_{u_0}(t)) + (L_{u_k}(t_0) - L_{u_k}(t)) \\
&\overset{(19)}{\geq} \xi^s_{(u_0,\dots,u_k)}(t_0) - (1+\rho)(t_0 - t) + L_{u_k}(t_0) - L_{u_k}(t) \\
&= \psi^s_p(t_0) + \frac{\kappa_p}{2} - (1+\rho)(t_0 - t) + L_{u_k}(t_0) - L_{u_k}(t) \\
&\overset{(18)}{=} \Psi^s_u(t_0) + \frac{\kappa_p}{2} - (1+\rho)(t_0 - t) + L_{u_k}(t_0) - L_{u_k}(t).
\end{aligned}$$

As for the induction step, assume that the claim holds for $i < \ell$. Note that $i < \ell$ implies that $t_{i+1} > \underline{t}$ as otherwise the construction would halt at index i, a contradiction to the definition of ℓ. From Inequality (21) we know that

$$L_{u_i}(t_{i+1}) - L_{u_{i+1}}(t_{i+1}) < \left(s + \frac{1}{2}\right) \kappa_{\{u_i, u_{i+1}\}}. \tag{23}$$

Thus, we can write

$$\begin{aligned}
\xi^s_{(u_{i+1},\dots,u_k)}(t_{i+1}) &= \xi^s_{(u_i,\dots,u_k)}(t_{i+1}) - L_{u_i}(t_{i+1}) + L_{u_{i+1}}(t_{i+1}) + s\kappa_{\{u_i,u_{i+1}\}} \\
&\overset{(23)}{>} \xi^s_{(u_i,\dots,u_k)}(t_{i+1}) - \frac{\kappa_{\{u_i,u_{i+1}\}}}{2} \\
&\overset{I.H.}{\geq} \Psi^s_u(t_0) + \frac{\kappa_{(u_{i+1},\dots,u_k)}}{2} \\
&\quad -(1+\rho)(t_0 - t_{i+1}) + L_{u_k}(t_0) - L_{u_k}(t_{i+1}).
\end{aligned} \tag{24}$$

We need to show that $i+1 \neq k$ as claimed. Assuming the contrary, Inequality (24) leads to the contradiction

$$\begin{aligned}
0 = \xi^s_{(u_k)}(t_k) &\overset{(24)}{>} \Psi^s_u(t_0) - (1+\rho)(t_0 - t_k) + L_{u_k}(t_0) - L_{u_k}(t_k) \\
&\geq \Psi^s_u(t_0) - 2\rho(t_0 - t_k) \\
&\overset{(19)}{\geq} \Psi^s_u(t_0) - 2\rho\Delta t \\
&\overset{(18)}{\geq} 0.
\end{aligned}$$

Hence it follows that $i+1 < k$.

Now let $t \in [t_{i+1}, t_{i+2}]$ (recall that $t_{i+1} > t_0 - \Delta t$ and $i+1 \neq k$, i.e., time t_{i+2} is defined). We obtain

$$\begin{aligned}
\xi^s_{(u_{i+1},\dots,u_k)}(t) &\geq \xi^s_{(u_{i+1},\dots,u_k)}(t_{i+1}) - (L_{u_{i+1}}(t_{i+1}) - L_{u_{i+1}}(t)) + (L_{u_k}(t_{i+1}) - L_{u_k}(t)) \\
&\overset{(19)}{\geq} \xi^s_{(u_{i+1},\dots,u_k)}(t_{i+1}) - (1+\rho)(t_{i+1} - t) + L_{u_k}(t_{i+1}) - L_{u_k}(t) \\
&\overset{(24)}{\geq} \Psi^s_u(t_0) + \frac{\kappa_{(u_{i+1},\dots,u_k)}}{2} - (1+\rho)(t_{i+1} - t) + L_{u_k}(t_0) - L_{u_k}(t),
\end{aligned}$$

i.e., the induction step succeeds.

Evaluating the statement at time $t_{\ell+1}$, we obtain for any $t \in [t_{\ell+1}, t_0]$ that

$$\begin{aligned}
\xi^s_{(u_\ell,\dots,u_k)}(t) &= \xi^s_{(u_\ell,\dots,u_k)}(t_{\ell+1}) + L_{u_\ell}(t) - L_{u_\ell}(t_{\ell+1}) - (L_{u_k}(t) - L_{u_k}(t_{\ell+1}))\\
&\overset{(22)}{>} \Psi^s_u(t_0) - (1+\rho)(t_0 - t_{\ell+1}) + L_{u_\ell}(t) - L_{u_\ell}(t_{\ell+1}) + L_{u_k}(t_0) - L_{u_k}(t) \qquad (25)\\
&\geq \Psi^s_u(t_0) - 2\rho(t_0 - t_{\ell+1}) \qquad (26)\\
&\geq \Psi^s_u(t_0) - 2\rho\Delta t\\
&\overset{(18)}{\geq} 0. \qquad (27)
\end{aligned}$$

Part II. Getting further away from u_k. We define a finite chain of nodes $w_\ell, \dots, w_m$ and times $t_{\ell+1} \geq t_{\ell+2} \geq \dots \geq t_{m+1} = t_0 - \Delta t$ (where $w_\ell := u_\ell$ and $t_{\ell+1}$ are the node and the time, respectively, at which the previous construction left off). The construction is inductive and maintains the following properties: For all $i \in \{\ell+1, \dots, m\}$ it holds that

$$L_{w_i}(t_i) - L_{w_{i-1}}(t_i) > \left(s + \frac{1}{2}\right)\kappa_{\{w_i, w_{i-1}\}}, \qquad (28)$$

and for all $t \in [t_{i+1}, t_i]$ we have

$$\xi^s_{(w_i,\dots,w_\ell=u_\ell,\dots,u_k)}(t) \geq \xi^s_{(u_\ell,\dots,u_k)}(t_{\ell+1}) + \frac{\kappa_{(w_i,\dots,w_\ell)}}{2} - 2\rho(t_{\ell+1} - t). \qquad (29)$$

If $t_{\ell+1} = t_0 - \Delta t$, we have $\ell + 1 = m + 1$ and there is nothing to show. Otherwise, the previous construction must have halted because Statement (20) was satisfied at time $t_{\ell+1} > t_{m+1} = t_0 - \Delta t$. We choose $w_{\ell+1}$ to be the node w from that statement, implying that Inequality (28) holds true for $i = \ell + 1$. Thus,

$$\begin{aligned}
\xi^s_{(w_{\ell+1},w_\ell=u_\ell,\dots,u_k)}(t_{\ell+1}) &\geq \xi^s_{(u_\ell,\dots,u_k)}(t_{\ell+1}) + L_{w_{\ell+1}}(t_{\ell+1}) - L_{w_\ell}(t_{\ell+1}) - s\kappa_{\{w_{\ell+1},w_\ell\}}\\
&\overset{(28)}{>} \xi^s_{(u_\ell,\dots,u_k)}(t_{\ell+1}) + \frac{\kappa_{(w_{\ell+1},w_\ell)}}{2},
\end{aligned}$$

i.e., Inequality (29) holds for $i = \ell + 1$ at time $t_{\ell+1}$.

Now assume that for some $i \in \{\ell+1, \dots, m\}$ we have already constructed the chain up to node w_i, Inequality (28) is true, and Inequality (29) holds at time $t_i > t_0 - \Delta t$. We define

$$t_{i+1} := \min\{t \in [t_0 - \Delta t, t_i] \mid L_{w_i}(t_i) - L_{w_i}(t) \leq (1+\rho)(t_i - t)\}, \qquad (30)$$

i.e., in the same way as we did for the previous construction.

Hence, for $t \in [t_{i+1}, t_i]$,

$$\begin{aligned}
\xi^s_{(w_i,\dots,w_\ell=u_\ell,\dots,u_k)}(t) &= \xi^s_{(w_i,\dots,w_\ell=u_\ell,\dots,u_k)}(t_i) - (L_{w_i}(t_i) - L_{w_i}(t)) + L_{u_k}(t_i) - L_{u_k}(t)\\
&\overset{(30)}{\geq} \xi^s_{(w_i,\dots,w_\ell=u_\ell,\dots,u_k)}(t_i) - 2\rho(t_i - t)\\
&= \xi^s_{(w_{i-1},\dots,w_\ell=u_\ell,\dots,u_k)}(t_i) + L_{w_i}(t_i) - L_{w_{i-1}}(t_{i-1}) - s\kappa_{\{w_i,w_{i-1}\}} - 2\rho(t_i - t)\\
&\overset{I.H.}{>} \xi^s_{(w_{i-1},\dots,w_\ell=u_\ell,\dots,u_k)}(t_i) + \frac{\kappa_{\{w_i,w_{i-1}\}}}{2} - 2\rho(t_i - t)\\
&\overset{I.H.}{\geq} \xi^s_{(u_\ell,\dots,u_k)}(t_{\ell+1}) + \frac{\kappa_{(w_i,\dots,w_\ell)}}{2} - 2\rho(t_{\ell+1} - t). \qquad (31)
\end{aligned}$$

If $t_{i+1} = t_0 - \Delta t$, the construction halts. Note that this happens after a finite number of steps because clock rates are bounded from above by $(1+\rho)(1+\mu)$, and clock values increase by at least $\min_{x \neq y \in V}\{(s+1/2)\kappa_{\{x,y\}}\} > 0$ whenever we move from a node w_i to a node w_{i+1}.

If $t_{i+1} > t - \Delta t$, we claim that

$$\exists w \in N^s_{w_i}(t_{i+1}) : \; L_w(t_{i+1}) - L_{w_i}(t_{i+1}) > \left(s + \frac{1}{2}\right)\kappa_{\{w_i,w\}}. \tag{32}$$

We will now prove this claim. Observe that, by using Lemma 5.4 inductively, for

$$\tau := \sum_{j=\ell}^{i+1} \tau_{\{w_i,w_{i+1}\}}$$

we get that $(w_i, \ldots, w_\ell = u_\ell, \ldots, u_k) \in P^s_{w_i}(t)$ for all $t \in [\underline{t}, t_i - \tau]$. In particular,

$$\tau \overset{(3)}{<} \frac{\kappa_{(w_{i+1},\ldots,w_\ell)}}{(1+\rho)\mu} \leq \frac{C_s}{(1+\rho)\mu}. \tag{33}$$

Let r be maximal such that $L_{w_i}(t_i - \tau) \geq T^{(r)}_s$. The continuity of logical clock values and the assumption that w_i satisfies the stabilization condition on level s during the interval $[\underline{t}, t_i - \tau]$ imply that

$$\begin{aligned} T^{(r)}_s < L_{w_i}(\underline{t}) < L_{w_i}(t_{i+1}) \leq L_{w_i}(t_i) \quad &\leq \quad L_{w_i}(t_i - \tau) + (1+\rho)(1+\mu)\tau \\ &\overset{(33)}{\leq} \quad L_{w_i}(t_i - \tau) + \frac{(1+\mu)C_s}{\mu} \leq T^{(r+1)}_s. \end{aligned}$$

Hence, according to the definition of t_{i+1} and Lemma 5.6, it follows from the assumption that Statement (32) is false that

$$L_{w_i}(t_{i+1}) - L_{w_{i-1}}(t_{i+1}) < \left(s + \frac{1}{2}\right)\kappa_{\{w_i,w_{i-1}\}}. \tag{34}$$

We infer that

$$\begin{aligned} L_{w_{i-1}}(t_i) - L_{w_{i-1}}(t_{i+1}) \quad &\overset{(28,34)}{<} \quad \left(L_{w_i}(t_i) - \left(s + \frac{1}{2}\right)\kappa_{\{w_i,w_{i-1}\}}\right) \\ &\qquad - \left(L_{w_i}(t_{i+1}) - \left(s + \frac{1}{2}\right)\kappa_{\{w_i,w_{i-1}\}}\right) \\ &= \quad L_{w_i}(t_i) - L_{w_i}(t_{i+1}) \\ &\overset{(30)}{\leq} \quad (1+\rho)(t_i - t_{i+1}), \end{aligned}$$

which yields

$$\begin{aligned} L_{w_{i-1}}(t_{i-1}) - L_{w_{i-1}}(t_{i+1}) \quad &= \quad L_{w_{i-1}}(t_{i-1}) - L_{w_{i-1}}(t_i) + L_{w_{i-1}}(t_i) - L_{w_{i-1}}(t_{i+1}) \\ &\overset{(30)}{<} \quad (1+\rho)(t_{i-1} - t_{i+1}). \end{aligned}$$

This contradicts Definition (30), implying that indeed Statement (32) must hold true. Thus, we can define $w_{i+1} := w$ from Statement (32), complying with Inequality (28). Moreover,

$$\begin{aligned}
\xi^s_{(w_{i+1},\dots,w_\ell=u_\ell,\dots,u_k)}(t_{i+1}) &\geq \xi^s_{(w_i,\dots,w_\ell=u_\ell,\dots,u_k)}(t_{i+1}) + L_{w_{i+1}}(t_{i+1}) - L_{w_i}(t_{i+1}) - s\kappa_{\{w_i,w_{i+1}\}} \\
&\overset{(28)}{>} \xi^s_{(w_i,\dots,w_\ell=u_\ell,\dots,u_k)}(t_{i+1}) + \frac{\kappa_{\{w_i,w_{i+1}\}}}{2} \\
&\overset{(31)}{\geq} \xi^s_{(u_\ell,\dots,u_k)}(t_{\ell+1}) + \frac{\kappa_{(w_{i+1},\dots,w_\ell)}}{2} - 2\rho(t_{i+1}-t),
\end{aligned}$$

showing that Inequality (29) is true for the index $i+1$ and time t_{i+1}.

This completes the induction step, as we started from the assumption that node w_i satisfying Inequality (28) has been chosen and Inequality (29) holds at time $t_i > t_0 - \Delta t$.

Part III. Bounding Ξ^s. We proceed by constructing a third and last finite sequence of times $t_0 = \theta'_0 \geq \theta_1 \geq \theta'_1 \geq \theta_2 \geq \theta'_2 \geq \dots$ While going back in time, we will trace a subsequence of the nodes $w_\ell, \dots, w_m$. The construction halts at time θ_h if we have $\theta_h \leq t_0 - \Delta t = t_{m+1}$, and it halts at time θ'_h if we have $\theta'_h \leq t_0 - \Delta t$. Let $i_j \in \{\ell, \dots, m\}$, $j \in \{1, \dots, h\}$, denote the minimal index satisfying $\theta_j \in (t_{i_j+1}, \theta'_{j-1}]$, or $m+1$ if $\theta_j \leq t_{m+1}$. We will maintain the properties that (i) for all $j \in \{0, \dots, h-1\}$ and all $t \in [t_0 - \Delta t, \theta'_j]$ we have $p_j := (w_{i_j}, \dots, w_\ell = u_\ell, \dots, u_k) \in P^s_{w_{i_j}}(t)$, (ii) for all θ'_j it holds that

$$\Xi^s_{w_{i_j}}(\theta'_j) \geq \xi^s_{p_0}(t_0) - (L_{u_\ell}(t_0) - L_{w_{i_j}}(\theta'_j)) + (1-\rho)(1+\mu)(t_0 - \theta'_j) - \left(s + \frac{1}{2}\right)\kappa_{(w_{i_j},\dots,w_\ell)}, \quad (35)$$

and (iii) for all θ_j we have that

$$\Xi^s_{w_{i_{j-1}}}(\theta_j) \geq \xi^s_{p_0}(t_0) - (L_{u_\ell}(t_0) - L_{w_{i_{j-1}}}(\theta_j)) + (1-\rho)(1+\mu)(t_0 - \theta_j) - \left(s + \frac{1}{2}\right)\kappa_{(w_{i_{j-1}},\dots,w_\ell)}. \quad (36)$$

Recall that $u = u_k$ and $w_\ell = u_\ell$. To anchor the induction, observe first that since $p \in P^s_u(t_0)$, also $(u_k, \dots, u_\ell) \in P^s_u(t_0)$, implying that $p_0 \in P^s_{w_\ell}(t_0)$. Hence, $\Xi^s_{w_\ell}(t_0) \geq \xi_{p_0}(t_0)$ and Statement (ii) is trivially satisfied. Furthermore, since by assumption for node $u_k = u$ the stabilization condition holds during the time interval $[t_0 - \Delta t, t_0] \subseteq [\underline{t} + C_{s-1}/((1+\rho)\mu), \bar{t}]$, Lemma 5.4 thus states that $(u_k, \dots, u_\ell) \in P^s_u(t)$ for all $t \in [t_0 - \Delta t, t_0]$, showing Statement (i) for $j = 0$.

We need to prove Statement (iii) for $j = 1$. As $p_0 \in P^s_{w_\ell}(t)$ for all $t \in [t_{\ell+1}, t_0]$, it follows that

$$\Xi^s_{w_\ell}(t) \geq \xi^s_{p_0}(t) \overset{(27)}{>} 0.$$

Furthermore, $p_0 \in P^s_{w_\ell}(t)$ entails that the stabilization condition holds at node u_ℓ at time t. Hence, we can apply Lemma 5.5 to the time interval $[t_{\ell+1}, t_0]$, yielding time $\theta_1 \in [t_{\ell+1} - C_{s-1}/((1+\rho)\mu), t_{\ell+1}]$ with the property that

$$\begin{aligned}
\Xi^s_{w_\ell}(\theta_j) &\geq \Xi^s_{w_\ell}(t_0) - (L_{u_\ell}(t_0) - L_{w_\ell}(\theta_1)) + (1-\rho)(1+\mu)(t_0 - \theta_1) + (1+\rho)\mu(t_{\ell+1} - \theta_1) \quad (37) \\
&\geq \xi^s_{p_0}(t_0) - (L_{u_\ell}(t_0) - L_{w_\ell}(\theta_1)) + (1-\rho)(1+\mu)(t_0 - \theta_1),
\end{aligned}$$

i.e., Statement (iii) holds for $j = 1$.

Assume now that Statements (i) and (ii) hold for some value $j > 0$ and the construction does not halt at time θ'_j, i.e., $\theta'_j > t_{m+1}$. We make a case distinction. If $\theta'_j \leq t_{i_j+1}$, we set $\theta_{j+1} := \theta'_j$, and Statement *(iii)* for $j+1$ trivially follows from Statement *(ii)* for j. Otherwise, if $\theta'_j > t_{i_j+1}$, due to Statement *(i)* for j we have for all $t \in [t_{i_j+1}, \theta'_j] \subset [t_0 - \Delta t, \theta'_j]$ that $p_j \in P^s_{w_{i_j}}(t)$. Moreover, the minimality of i_j guarantees that $\theta'_j \leq t_{i_j}$, implying that $t \in [t_{i_j+1}, t_{i_j}]$. Therefore,

$$\begin{aligned} \Xi^s_{w_{i_j}}(t) &\geq \xi^s_{p_j}(t) \\ &\overset{(29)}{>} \xi^s_{p_0}(t_{\ell+1}) - 2\rho(t_{\ell+1} - t) \\ &\overset{(26)}{\geq} \Psi^s_u(t_0) - 2\rho(t_0 - t) \\ &> \Psi^s_u(t_0) - 2\rho\Delta t \\ &\overset{(18)}{\geq} 0. \end{aligned}$$

Since $p_j \in P^s_{w_{i_j}}(t)$ is equivalent to $(u_k, \dots, u_\ell = w_\ell, \dots, w_{i_j}) \in P^s_u(t)$, by assumption w_{i_j} satisfies the stabilization condition during the time interval $[t_{i_j+1}, \theta'_j]$. We apply Lemma 5.5, yielding the time $\theta_{j+1} \in [\underline{t}, t_{i_j+1}]$ that satisfies

$$\begin{aligned} \Xi^s_{w_{i_j}}(\theta_{j+1}) &\geq \Xi^s_{w_{i_j}}(\theta'_j) - (L_{w_{i_j}}(\theta'_j) - L_{w_{i_j}}(\theta_{j+1})) + (1-\rho)(1+\mu)(\theta'_j - \theta_{j+1}) \\ &\quad +(1+\rho)\mu(t_{i_j+1} - \theta_{j+1}) \qquad (38) \\ &\overset{(35)}{\geq} \xi^s_{p_0}(t_0) - (L_{u_\ell}(t_0) - L_{w_{i_j}}(\theta_{j+1})) + (1-\rho)(1+\mu)(t_0 - \theta_{j+1}) \\ &\quad - \left(s + \frac{1}{2}\right)\kappa_{(w_{i_j}, \dots, w_\ell)}. \end{aligned}$$

Therefore, in both cases Claim *(iii)* is established for $j+1$.

Next, assume that Statement *(iii)* holds for some value j and Statements *(i)* and *(ii)* are true for $j-1$, and also that the construction does not halt at time θ_j, i.e., $\theta_j > t_{m+1}$. According to Statement *(i)*, $p_{j-1} \in P^s_{w_{i_{j-1}}}(t)$ for all $t \in [t_0 - \Delta t, \theta_j] \subseteq [t_0 - \Delta t, \theta'_{j-1}]$, thus the inverse path is in $P^s_u(t)$. Moreover, the minimality of i_j ensures that $\theta_j \leq t_i$ for all $i \geq i_j$. Define

$$\tau := \sum_{i=i_{j-1}+1}^{i_j} \tau_{\{w_i, w_{i-1}\}} \overset{(3)}{\leq} \frac{\kappa_{(w_{i_j}, \dots, w_{i_{j-1}})}}{8\mu} < \frac{\kappa_{(w_{i_j}, \dots, w_{i_{j-1}})}}{2(2\rho + 2(1+\rho)\mu)} \qquad (39)$$

and $\theta'_j := \theta_j - \tau$ (note that $\tau = 0$ if $w_{i_j} = w_{i_{j-1}}$). Since the stabilization condition is satisfied by node u throughout the time interval $[t_0 - \Delta t, \theta_j]$, inductively applying Lemma 5.4 gives that $p_j \in P^s_{w_{i_j}}(t)$ for all $t \in [t_0 - \Delta t, \theta'_j]$, proving Statement *(i)* for the index j.

Let $(w_{i_{j-1}}, \dots, x)$ denote a path maximizing $\Xi^s_{w_{i_{j-1}}}(\theta_j)$. Since $p_{j-1} \in P^s_{w_{i_{j-1}}}(\theta'_{j-1})$ by Statement (i), we know that $\xi^s_{p_{j-1}}(\theta_j) \le \xi^s_{(w_{i_{j-1}},\dots,x)}(\theta_j)$. Thus, as $\theta_j \in [t_{i_j+1}, t_{i_j}]$, we can bound

$$\begin{aligned}
\xi^s_{(w_{i_j},\dots,w_{i_{j-1}},\dots,x)}(\theta_j) &= \xi^s_{(w_{i_j},\dots,w_{i_{j-1}})}(\theta_j) + \xi^s_{(w_{i_{j-1}},\dots,x)}(\theta_j) \\
&\ge \xi^s_{p_j}(\theta_j) \\
&\overset{(29)}{\ge} \xi^s_{p_0}(t_{\ell+1}) - 2\rho(t_{\ell+1} - \theta_j) + \frac{\kappa_{(w_{i_j},\dots,w_\ell)}}{2} \\
&\overset{(26)}{\ge} \Psi^s_u(t_0) - 2\rho(t_0 - \theta_j) + \frac{\kappa_{(w_{i_j},\dots,w_\ell)}}{2} \\
&\ge \Psi^s_u(t_0) - 2\rho\Delta t + \frac{\kappa_{(w_{i_j},\dots,w_\ell)}}{2} \\
&\overset{(18)}{\ge} \frac{\kappa_{(w_{i_j},\dots,w_\ell)}}{2}.
\end{aligned}$$

Therefore,

$$\begin{aligned}
\xi^s_{(w_{i_j},\dots,w_{i_j},\dots,x)}(\theta'_j) &= \xi^s_{(w_{i_j},\dots,w_{i_j},\dots,x)}(\theta_j) - (L_{w_{i_j}}(\theta_j) - L_{w_{i_j}}(\theta'_j)) + L_x(\theta_j) - L_x(\theta'_j) \\
&\ge \frac{\kappa_{(w_{i_j},\dots,w_\ell)}}{2} - (2\rho + (1+\rho)\mu)(\theta_j - \theta'_j) \\
&= \frac{\kappa_{(w_{i_j},\dots,w_\ell)}}{2} - (2\rho + (1+\rho)\mu)\tau \\
&\overset{(39)}{>} 0.
\end{aligned}$$

By virtue of Lemma 5.4, we obtain that $(w_{i_j}, \dots, w_{i_{j-1}}, \dots, x) \in P^s_{w_{i_j}}(\theta'_j)$.

We conclude that

$$\begin{aligned}
\Xi^s_{w_{i_j}}(\theta'_j) &\ge \xi_{(w_{i_j},\dots,w_{i_{j-1}},\dots,x)}(\theta'_j) \\
&= L_{w_{i_j}}(\theta'_j) - L_{w_{i_{j-1}}}(\theta_j) + L_{w_{i_{j-1}}}(\theta_j) - L_x(\theta_j) + L_x(\theta_j) - L_x(\theta'_j) - s\kappa_{(w_{i_j},\dots,w_{i_{j-1}},\dots,x)} \\
&\ge L_{w_{i_j}}(\theta'_j) - L_{w_{i_{j-1}}}(\theta_j) - s\kappa_{(w_{i_j},\dots,w_{i_{j-1}})} + (1-\rho)(\theta_j - \theta'_j) + \xi^s_{(w_{i_{j-1}},\dots,x)}(\theta_j) \\
&\overset{(39)}{>} L_{w_{i_j}}(\theta'_j) - L_{w_{i_{j-1}}}(\theta_j) - \left(s + \frac{1}{2}\right)\kappa_{(w_{i_j},\dots,w_{i_{j-1}})} \\
&\quad +(1-\rho+2\mu)(\theta_j - \theta'_j) + \Xi^s_{w_{i_{j-1}}}(\theta_j) \qquad (40) \\
&\overset{(36)}{\ge} \xi^s_{p_0}(t_0) - (L_{u_\ell}(t_0) - L_{w_{i_j}}(\theta'_j)) + (1-\rho)(1+\mu)(t_0 - \theta'_j) - \left(s + \frac{1}{2}\right)\kappa_{(w_{i_j},\dots,w_\ell)}.
\end{aligned}$$

This proves statement (ii) for the index j, completing the induction step.

Part IV. Putting everything together. We make a case distinction. If the previous induction stopped at a time $\theta_h \le t_0 - \Delta$, we preserve the term $(1+\rho)\mu(t_{i_{h-1}+1} - \theta_h)$ from Inequality (38) (or from Inequality (37) if $h = 1$) in the last step, which yields that

$$\begin{aligned}
\Xi^s_{w_{i_{h-1}}}(\theta_h) &\geq \xi^s_{p_0}(t_0) - (L_{u_\ell}(t_0) - L_{w_{i_{h-1}}}(\theta_h)) + (1-\rho)(1+\mu)(t_0 - \theta_h) \\
&\quad - \left(s + \frac{1}{2}\right) \kappa_{(w_{i_{h-1}},\ldots,w_\ell)} + (1+\rho)\mu(t_{i_{h-1}+1} - \theta_h) \\
&\geq \xi^s_{p_0}(t_0) - (L_{u_\ell}(t_0) - L_{w_{i_{h-1}}}(t_{i_{h-1}+1})) - (1+\rho)(t_{i_{h-1}+1} - \theta_h) \\
&\quad +(1-\rho)(1+\mu)(t_0 - \theta_h) - \left(s + \frac{1}{2}\right) \kappa_{(w_{i_{h-1}},\ldots,w_\ell)} \\
&= \xi^s_{p_0}(t_0) - (L_{u_\ell}(t_0) - L_{u_\ell(t_{\ell+1})}) - (1+\rho)(t_{i_{h-1}+1} - \theta_h) \\
&\quad + \sum_{i=\ell+1}^{i_{h-1}} \left(L_{w_i}(t_{i+1}) - L_{w_{i-1}}(t_i) - \left(s + \frac{1}{2}\right) \kappa_{\{w_i,w_{i-1}\}} \right) \\
&\quad +(1-\rho)(1+\mu)(t_0 - \theta_h) \\
&\overset{(28)}{\geq} \xi^s_{p_0}(t_0) - (L_{u_\ell}(t_0) - L_{u_\ell(t_{\ell+1})}) - (1+\rho)(t_{i_{h-1}+1} - \theta_h) \\
&\quad + \sum_{i=\ell+1}^{i_{h-1}} (L_{w_i}(t_{i+1}) - L_{w_i}(t_i)) + (1-\rho)(1+\mu)(t_0 - \theta_h) \\
&\overset{(30)}{\geq} \xi^s_{p_0}(t_0) - (L_{u_\ell}(t_0) - L_{u_\ell(t_{\ell+1})}) - (1+\rho)(t_{\ell+1} - \theta_h) \\
&\quad +(1-\rho)(1+\mu)(t_0 - \theta_h).
\end{aligned}$$

Furthermore, setting $t = t_0$ in Inequality (25),

$$\xi^s_{p_0}(t_0) \overset{(25)}{>} \Psi^s_u(t_0) - (1+\rho)(t_0 - t_{\ell+1}) + L_{u_\ell}(t_0) - L_{u_\ell}(t_{\ell+1}),$$

leading to

$$\begin{aligned}
\Xi^s_{w_{i_{h-1}}}(\theta_h) &> \Psi^s_u(t_0) - (1+\rho)(t_0 - \theta_h) + (1-\rho)(1+\mu)(t_0 - \theta_h) \\
&\geq \Psi^s_u(t_0) + ((1-\rho)\mu - 2\rho)\Delta t \\
&\overset{(18)}{\geq} (1-\rho)\mu\Delta t \\
&= C_{s-1}
\end{aligned}$$

Statement (*i*) for $h-1$ yields that $p_{h-1} \in P_{w_{i_{h-1}}}(\theta_h)$, therefore it follows that $w_{i_{h-1}}$ satisfies the stabilization condition at time $\theta_h \geq \underline{t}$ and thus the system is $(s-1)$-legal at $w_{i_{h-1}}$ at time θ_h. This is a contradiction to the previous inequality because

$$\Xi^s_{w_{i_{h-1}}}(\theta_h) \leq \Xi^{s-1}_{w_{i_{h-1}}}(\theta_h) \leq C_{s-1}.$$

The other case is that the induction halts at a time θ'_h, i.e., $i_h = m$. In this case, we preserve the additional term $(1+\rho)\mu(\theta_h-\theta'_h)$ from Inequality (40) in the last step, yielding

$$\begin{aligned}\Xi^s_{w_m}(\theta'_h) &\geq \xi^s_{p_0}(t_0)-(L_{u_\ell}(t_0)-L_{w_m}(\theta'_h))+(1-\rho)(1+\mu)(t_0-\theta_h)\\ &\quad -\left(s+\frac{1}{2}\right)\kappa_{(w_m,\dots,w_\ell)}+(1+\rho)\mu(\theta_h-\theta'_h)\\ &> \xi^s_{p_0}(t_0)-(L_{u_\ell}(t_0)-L_{w_m}(\theta'_h))+(1-\rho)(1+\mu)(t_0-\theta_h)\\ &\quad -\left(s+\frac{1}{2}\right)\kappa_{(w_m,\dots,w_\ell)}+(1+\rho)\mu(t_{m+1}-\theta'_h).\end{aligned}$$

Thus, with θ'_h taking the role of θ_h, w_m replacing $w_{i_{h-1}}$, and substituting t_{m+1} for $t_{i_{h-1}+1}$, analogously to the previous case a contradiction occurs, concluding the proof. ☐

The final lemma states that if the system is $(s'-1)$-legal for all $s'\in\{2,\dots,s-1\}$ in certain time intervals at all nodes whose clock skew to a node u is at most $\mathcal{O}(sC_{s-1})$, then there is a (sub-)interval during which the system is s-legal at u.

Lemma 5.8. *Let $s\in\mathbb{N}$, $s\neq 1$, and $u\in V$ such that for all times $t\in[\underline{t},\bar{t}\,]$ and all $r\in\mathbb{N}$,*

$$|L_u(t)-T_s^{(r)}|\geq\frac{(1+(3s+1)\mu)C_{s-1}}{\mu}.$$

Assume that for all times $t\in[\underline{t},\bar{t}\,]$ and nodes $v\in V$ with the property that

$$|L_v(t)-L_u(t)|\leq 2sC_{s-1}$$

the system is s'-legal at v for all $s'\in\{2,\dots,s-1\}$ and all times in $[t-C_{s'-1}/((1+\rho)\mu),t]$. Then, if $C_s\geq C_{s-1}/\sigma$, the system is s-legal at u at all times

$$t\in[(\underline{t}+2C_{s-1}/((1-\rho^2)\mu),\bar{t}\,].$$

Proof. Let

$$t'\in[(\underline{t}+2C_{s-1}/((1-\rho^2)\mu),\bar{t}\,]$$

and $p=(u,\dots,v)\in P^s_u(t')$. By assumption, the stabilization condition on level s holds at u at time t'. We apply Lemma 5.4 to see that $p\in P^s_u(t)$ for all $t\in[\underline{t},t'\,]$. Hence, the $(s-1)$-legality of the system at u at time t permits to apply Lemma 5.3, which yields that

$$|L_v(t)-L_u(t)|\leq sC_{s-1}.$$

Now let $(v,\dots,w)\in P^s_v(t)$. Thus, as by assumption the system is also $(s-1)$-legal at v at time t, the triangle inequality and Lemma 5.3 yield that

$$|L_w(t)-L_u(t)|\leq 2sC_{s-1}.$$

We conclude that (i) for all $s'\in\{2,\dots,s-1\}$ the system is s'-legal at node w and all times $t\in[\underline{t}-C_{s'-1}/((1+\rho)\mu),t']$ and (ii), using the triangle inequality once more,

$$|L_w(t)-T_s^{(r)}|\geq\frac{(1+(s+1)\mu)C_{s-1}}{\mu}$$

for all $r \in \mathbb{N}$.

Consequently, the stabilization condition holds at node w during the time interval $[\underline{t}, t']$. Thus, the preconditions of Lemma 5.7 are met at node v for the time interval $[\underline{t}, t']$, giving that for all $t \in [\underline{t} + 2C_{s-1}/((1-\rho^2)\mu), t'\,]$,

$$\Psi_v^s(t) < \frac{2\rho C_{s-1}}{(1-\rho)\mu} \overset{(2)}{=} \frac{C_{s-1}}{2\sigma} \leq \frac{C_s}{2}.$$

Since this statement holds for all paths $p \in P_u^s(t')$, we have that

$$\begin{aligned}
\Xi_u^s(t') &= \max_{p=(u,\dots,v)\in P_u^s(t')} \left\{ \psi_{(v,\dots,u)}^s(t') + \frac{\kappa_p}{2} \right\} \\
&\leq \max_{p=(u,\dots,v)\in P_u^s(t')} \left\{ \Psi_v^s(t') + \frac{C_s}{2} \right\} \\
&< C_s.
\end{aligned}$$

Moreover, from the trivial path (u) we get that

$$\Psi_u^s(t') < \frac{C_s}{2}.$$

Thus, the system is s-legal at u and all times

$$t' \in [(\underline{t} + 2C_{s-1}/(1-\rho^2)\mu), \bar{t}\,]$$

as claimed, concluding the proof. □

We are now in the position to prove the main theorem, which shows that the system is always legal with respect to some specific gradient sequence.

Theorem 5.9. *The system is legal at all nodes and all times with respect to the gradient sequence*

$$C := (2\hat{\mathcal{G}}, 2\hat{\mathcal{G}}, 2\hat{\mathcal{G}}/\sigma, 2\hat{\mathcal{G}}/\sigma^2, 2\hat{\mathcal{G}}/\sigma^3, \dots).$$

Proof. Define for $s \in \mathbb{N}$ that

$$\Delta_s := \frac{(2 + (6s+8)\mu)\hat{\mathcal{G}}}{\mu\sigma^{\max\{s-2,0\}}}$$

and

$$\nabla_s := \frac{4(1+\mu)\hat{\mathcal{G}}}{(1-\rho)\mu\sigma^{\max\{s-2,0\}}},$$

i.e., according to Algorithm 1 we have for all $r, s \in \mathbb{N}$, $s \neq 1$, that

$$T_{s+1}^{(r)} > T_s^{(r)} + 2\nabla_{s-1} + \Delta_{s-1} + \Delta_s. \tag{41}$$

For all $s \in \mathbb{N} \cup \{\infty\}$, $s \neq 1$, let $C(s)$ denote the gradient sequence $C(s) := \{C_{s'}(s)\}_{s' \in \mathbb{N}}$, where

$$C_{s'}(s) := \begin{cases} 2\hat{\mathcal{G}}\sigma^{1-s'} & \text{if } s' < s \\ 2\hat{\mathcal{G}}\sigma^{2-s'} & \text{else} \end{cases}$$

We prove the following stronger claim. For all $r \in \mathbb{N}$, $s \in \mathbb{N}$, $s \geq 3$, the system is

$$C(2)\text{−legal} \qquad \text{if } L_u(t) \in \Big[T_1^{(r)}, T_3^{(r)} - \Delta_2 - \nabla_1\Big) \tag{42}$$

$$C(s)\text{−legal} \qquad \text{if } L_u(t) \in \Big[T_s^{(r)} - \Delta_{s-1} - \nabla_{s-2}, T_{s+1}^{(r)} - \Delta_s - \nabla_{s-1}\Big) \tag{43}$$

$$C(\infty)\text{−legal} \qquad \text{if } L_u(t) \in \Big[T_\infty^{(r)}, T_2^{(r+1)} - \hat{\mathcal{G}}\Big) \tag{44}$$

at any node $u \in V$ at all times t.

Note that since $\sigma > 1$, the limit $T_\infty^{(r)}$ is well-defined. Moreover, $\mathbb{R}_0^+$ is the union of the given time intervals for varying r and s. Thus, showing the claim implies that the system is at all times and all nodes legal with respect to the sequence C because $C_s = \max_{s' \in \mathbb{N} \cup \{\infty\}} \{C_s(s')\}$.

Assume for the sake of contradiction that the claim is wrong. Denote by $\bar{t}$ the smallest time when the system is not legal at some node $u \in V$ and by $\bar{s} \in \mathbb{N}$ the smallest integer such that the system is not $\bar{s}$-legal at some node $\bar{u} \in V$ at time $\bar{t}$. Choose the smallest r such that $\bar{t} \geq T_1^{(r)}$.

Trivially, we have that $\bar{s} \neq 1$ from Statement (i) of Lemma 5.2. Define

$$\underline{t} := \bar{t} - \frac{\nabla_{\bar{s}-1}}{(1+\rho)(1+\mu)} = \bar{t} - \frac{2C_{\bar{s}-1}}{(1-\rho^2)\mu} \leq \bar{t} - \frac{2C_{\bar{s}-1}(\bar{s})}{(1-\rho^2)\mu}.$$

Since logical clock rates are upper bounded by $(1+\rho)(1+\mu)$, we have for all $t \in [\underline{t}, \bar{t}\,]$ that

$$L_{\bar{u}}(t) \in [L_{\bar{u}}(\bar{t}\,) - \nabla_{\bar{s}-1}, L_{\bar{u}}(\bar{t}\,)].$$

By Statement (v) of Lemma 5.2, we can exclude Case (43) for $s = \bar{s}$ and $L_{\bar{u}}(\bar{t}\,)$ because $C_{\bar{s}}(\bar{s}) = C_{\bar{s}-1}(\bar{s})$. Thus, either $L_{\bar{u}}(\bar{t}\,) \geq T_{\bar{s}+1}^{(r)} - \Delta_{\bar{s}} - \nabla_{\bar{s}-1}$ or $L_{\bar{u}}(\bar{t}\,) < T_{\bar{s}}^{(r)} - \Delta_{\bar{s}-1} - \nabla_{\bar{s}-2}$.

We make a case distinction. Assume first that $L_{\bar{u}}(\bar{t}\,) < T_{\bar{s}}^{(r)} - \Delta_{\bar{s}-1} - \nabla_{\bar{s}-2}$, i.e., Case (42) or (43) for some $s < \bar{s}$ apply to $L_{\bar{u}}(\bar{t}\,)$. Both cases imply that the system is not $C_{\bar{s}}(2)$-legal at node $\bar{u}$ and time $\bar{t}$. Algorithm 1 states that if $L_{\bar{u}}(\bar{t}\,) \leq T_{\bar{s}}^{(1)}$, $N_{\bar{u}}^{\bar{s}}(\bar{t}\,) = \emptyset$. Hence, it must hold that $r \neq 1$, i.e.,

$$L_{\bar{u}}\left(\underline{t} - \frac{C_1}{(1+\rho)\mu}\right) \geq L_{\bar{u}}(\underline{t}) - \frac{(1+\mu)C_1}{\mu} \geq T_1^{(2)} - \nabla_1 - \Delta_1 > 0,$$

yielding that $\underline{t} - C_1/((1+\rho)\mu) > 0$. As $\bar{t}$ is minimal, the system is legal with respect to $C(2) = C$ at all times $t < \bar{t}$ and nodes $v \in V$. Furthermore, as already observed, we have for all $t \in [\underline{t}, \bar{t}\,]$ and all $r' \in \mathbb{N}$ that

$$\left|L_{\bar{u}}(t) - T_{\bar{s}}^{(r')}\right| \geq \Delta_{\bar{s}-1} = \frac{(1+(3\bar{s}+1)\mu)C_{\bar{s}-1}(2)}{\mu}.$$

Consequently, the preconditions of Lemma 5.8 are met for $\bar{s}$, $\bar{u}$, and the time interval $[\underline{t}, \bar{t}\,]$, yielding that the system is $\bar{s}$-legal with respect to $C_{\bar{s}}(2)$ at time $\bar{t}$, a contradiction.

The second case is that $L_{\bar{u}}(\bar{t}\,) \geq T_{\bar{s}+1}^{(r)} - \Delta_{\bar{s}} - \nabla_{\bar{s}-1}$, i.e., Case (44) or (43) for some $s > \bar{s}$ apply to $L_{\bar{u}}(\bar{t}\,)$. Both cases imply that the system is not $C_{\bar{s}}(\infty)$-legal at node $\bar{u}$ and time $\bar{t}$. Again, we have that $L_{\bar{u}}(\underline{t}) \geq T_{\bar{s}+1}^{(r)} - \Delta_{\bar{s}} - \nabla_{\bar{s}-1} - \nabla_{\bar{s}-1} \geq T_3^{(r)} - \Delta_2 - 2\nabla_1 > \Delta_1$ and therefore $\underline{t} - C_1/((1+\rho)\mu) > 0$.

For all $s' \in \{2,\ldots,\bar{s}-1\}$, $t \in [\underline{t},\bar{t}\,]$, and $v \in V$ with $|L_v(t) - L_u(t)| \leq 2\bar{s}C_{\bar{s}-1}(\infty)$ we have that

$$\begin{aligned} L_v\left(t - \frac{C_{s'-1}(\infty)}{(1+\rho)\mu}\right) &\geq L_v(\underline{t}) - \frac{(1+\mu)C_{s'-1}(\infty)}{\mu} \\ &\geq L_{\bar{u}}(\underline{t}) - 2\bar{s}C_{\bar{s}-1}(\infty) - \frac{2(1+\mu)\hat{\mathcal{G}}}{\mu\sigma^{s'-2}} \\ &\geq L_{\bar{u}}(\bar{t}) - \nabla_{\bar{s}-1} - 4\bar{s}\hat{\mathcal{G}}\sigma^{2-\bar{s}} - \frac{2(1+\mu)\hat{\mathcal{G}}}{\mu\sigma^{s'-2}} \\ &> T^{(r)}_{\bar{s}+1} - \Delta_{\bar{s}} - 2\nabla_{\bar{s}-1} - \Delta_{\bar{s}-1} - \nabla_{s'-1} \\ &\geq T^{(r)}_{s'+2} - \Delta_{s'+1} - 2\nabla_{s'} - \Delta_{s'} - \nabla_{s'-1} \\ &> T^{(r)}_{s'+1} - \Delta_{s'} - \nabla_{s'-1}. \end{aligned}$$

Assume for the moment that $T_2^{(r+1)} = \infty$. Thus, the previous inequality implies that for all $s' \in \{2,\ldots,\bar{s}-1\}$, times $t \in [\underline{t} - C_{s'-1}(\infty)/((1+\rho)\mu), \bar{t}\,]$, and $v \in V$ with $|L_v(t) - L_u(t)| \leq 2\bar{s}C_{\bar{s}-1}(\infty)$ the system is s'-legal at v and t. Thus, Lemma 5.8 states that the system is $\bar{s}$-legal at time $\bar{t}$.

Note that in this setting we get the stronger result that the system is legal with respect to $C(\infty)$ at any node u and any time t when $L_u(t) \geq T_\infty^{(r)}$. Certainly, if we reduce $T_2^{(r+1)}$, this cannot affect the execution of the algorithm at any point in time where *all* logical clock values are smaller than $T_2^{(r+1)}$. Thus, we conclude that the system must be $C(\infty)$-legal at all nodes u with $L_u(t) \geq T_\infty^{(r)}$ provided that $L_v(t) < T_2^{(r+1)}$ for all $v \in V$, which is true at logical times $L_u(t) \in [T_\infty^{(r)}, T_2^{(r+1)} - \hat{\mathcal{G}})$. Therefore, also the second case leads to a contradiction, implying that the initial assumption that our claim is false must be wrong, finishing the proof. □

So far we assumed that the bound $\hat{\mathcal{G}}$ on the global skew is the same in each round. It is not hard to see that $\hat{\mathcal{G}}$ can be replaced by $\hat{\mathcal{G}}^{(r)}$, and the system is legal at any node u after the system stabilized, i.e., at all times t when $L_u(t) \in [T_\infty^{(r)}, T_1^{(r)}]$ in each round r.

Corollary 5.10. *Assume that $\hat{\mathcal{G}}^{(r)} \geq \mathcal{G}(t)$ at any time t when some node $u \in V$ satisfies that $L_u(t) \in [T_1^{(r)}, T_2^{(r+1)} - \hat{\mathcal{G}}^{(r)})$. Then, the system is legal with respect to the gradient sequence*

$$C^{(r)} := (2\hat{\mathcal{G}}^{(r)}, 2\hat{\mathcal{G}}^{(r)}, 2\hat{\mathcal{G}}^{(r)}/\sigma, 2\hat{\mathcal{G}}^{(r)}/\sigma^2, \ldots)$$

at any node $u \in V$ and times t when $L_u(t) \in [T_\infty^{(r)}, T_2^{(r+1)} - \hat{\mathcal{G}}^{(r)})$.

Proof. The proof is carried out analogously to the proof of Theorem 5.9, with the modifications that we replace $\hat{\mathcal{G}}$ by $\hat{\mathcal{G}}^{(r)}$ and define $C(s)$, $1 \neq s \in \mathbb{N} \cup \{\infty\}$, as

$$C_{s'}(s) := \begin{cases} 2\hat{\mathcal{G}}^{(r)}\sigma^{1-s'} & \text{if } s' < s \\ 2\hat{\mathcal{G}}^{(r)}\sigma^{2-s} & \text{else} \end{cases}$$

Thus, legality on level s and higher is trivially satisfied by Statement (v) of Lemma 5.2. Therefore, the proof does not require any assumptions on the initial state of the system except the bound on the global skew, and legality with regard to $C^{(r)}$ is established inductively level by level. □

If we use the maximum of $\hat{\mathcal{G}}^{(r-1)}$ and $\hat{\mathcal{G}}^{(r)}$ to define the gradient sequence for round r, then we get that the system is legal with respect to this gradient sequence at all times in this round.

Corollary 5.11. *Define* $\hat{\mathcal{G}}^{(r-1,r)} := \max\{\hat{\mathcal{G}}^{(r-1)}, \hat{\mathcal{G}}^{(r)}\}$ *for all* $r \in \mathbb{N}$ *(where we take* $\hat{\mathcal{G}}^{(0)}$ *to be* 0*). Then the system is legal with respect to the gradient sequence*

$$C^{(r-1,r)} := (2\hat{\mathcal{G}}^{(r-1,r)}, 2\hat{\mathcal{G}}^{(r-1,r)}, 2\hat{\mathcal{G}}^{(r-1,r)}/\sigma, \hat{\mathcal{G}}^{(r-1,r)}/\sigma^2, \ldots)$$

at any node $u \in V$ *and times* t *when* $L_u(t) \in [T_1^{(r)}, T_1^{(r+1)})$*.*

Proof. By assumption $\hat{\mathcal{G}}^{(r-1,r)}$ is a valid upper bound on the global skew at any time t when some node $u \in V$ satisfies that $L_u(t) \in [T_1^{(r-1)}, T_1^{(r+1)})$. This time we define $C(s)$ as

$$C_{s'}(s) := \begin{cases} 2\hat{\mathcal{G}}^{(r)}\sigma^{1-s'} & \text{if } s' < s \\ \min\{2\hat{\mathcal{G}}^{(r)}\sigma^{2-s}, 2\hat{\mathcal{G}}^{(r-1,r)}\}\sigma^{2-s'} & \text{else} \end{cases}$$

Now, at respective (logical) times and nodes, for values $s' < s$ we can infer s'-legality with respect to $C(s)$ as in the proof of Corollary 5.10. For values $s' \geq s$ we either use Statement (v) of Lemma 5.2 or argue analogously to the proof of Theorem 5.9. The latter exploits that, according to Corollary 5.10, the system is $C^{(r,r-1)}$-legal at any node $u \in V$ provided that $L_u(t) \in [T_\infty^{(r-1)}, T_2^{(r+1)} - \hat{\mathcal{G}}^{(r)})$. As the argumentation in the proof of Theorem 5.9 relies only on this range of logical clock values smaller than $T_2^{(r+1)} - \hat{\mathcal{G}}^{(r)}$ if $L_{\bar{u}}(\bar{t}) \geq T_1^{(r)}$, this completes the proof. $\square$

Another simplification we made is that a neighborhood set $\mathcal{N}_u^s$ is stored for all $s \in \mathbb{N}$. We show now that for all nodes u it is indeed not necessary to store any such set for $s > s_{\max}^{(r)}(u)$ in any round r.

Corollary 5.12. *Assume that* $r, s \in \mathbb{N}$ *and* $u \in V$*, where* $s > s_{\max}^{(r)}(u)$*. Theorem 5.9 and Corollaries 5.10 and 5.11 hold true even if the algorithm does not implement the fast and slow mode rules on level* s *at* u *and times* t *when* $L_u(t) \in [T_1^{(r)}, T_1^{(r+1)})$*.*

Proof. Consider Theorem 5.9 first. Assume for the sake of contradiction that we have a minimal time t, a node $u \in V$ and some $s > s_{\max}^{(r)}(u)$ such that the fast or slow mode rule is satisfied with regard to s at u and t. Thus, by Lemma 4.2, the fast or the slow condition on level s is satisfied at time t at node u. In all cases, there is some node $v \in N_u^s(t)$ such that

$$|L_u(t) - L_v(t)| \geq s\kappa_{\{u,v\}}.$$

Denote by $s' \in \mathbb{N}$ the maximal value such that $\kappa_{\{u,v\}} \leq C_{s'}$ (with respect to the gradient sequence C from the theorem), i.e., $\kappa_{\{u,v\}} > C_{s'+1}$. Since $N_u^s(t) \subseteq N_u^1(t)$, according to Algorithm 1 we have that $s' + 1 \leq s_{\max}^{(r)}(u)$, implying $s' + 1 < s$. Hence,

$$\max\left\{\xi_{(u,v)}^{s'+1}(t - \tau_{\{u,v\}}), \xi_{(v,u)}^{s'+1}(t - \tau_{\{u,v\}})\right\} \geq (s - (s'+1))\kappa_{\{u,v\}} - (2\rho + (1+\rho)\mu)\tau_{\{u,v\}} \overset{(3)}{>} 0.$$

Since t is minimal, the claim of Theorem 5.9 must hold until time t, which implies that the stabilization condition on level $s'+1$ is satisfied at node u and time t. Thus, Lemma 5.4 states that $(u,v) \in P_u^{s'+1}(t - \tau_{\{u,v\}})$, implying in particular that $\kappa_{\{u,v\}} \leq C_{s'+1}$, a contradiction. Therefore, there cannot be a first time t when the fast or slow rule applies for $s > s_{\max}^{(r)}(u)$ at any node and any time. We conclude that the statement of the theorem thus must also remain true at all times.

Regarding Corollary 5.10, observe that legality of the system is established inductively, starting from the lowest level; no (non-trivial) skew guarantees are given on higher levels. For times t when $L_u(t) \leq T_{s+1}^{(r)} - \Delta_s - \nabla_{s-1}$, the legality of the system at u and time t follows solely from the rules up to level $s-1$. Thus, also Corollary 5.10 is not affected by the fact that the algorithm does not implement the fast and slow rules for levels larger than $s_{\max}^{(r)}(u)$ at node u and time t with $L_u(t) \in [T_1^{(r)}, T_1^{(r+1)})$.

Finally, we conclude that the statement of Corollary 5.11 remains valid because it is a consequence of Theorem 5.9 and Corollary 5.10. □

It remains to show the optimality of the achieved skew bounds.

Corollary 5.13. *Assume that Condition 4.1 holds, $\hat{\mathcal{G}}^{(r)} \in \mathcal{O}(D)$ for all $r \in \mathbb{N}$, and $\mu\tau_e \in \mathcal{O}(\epsilon_e)$ for all edges e that ever exist during the execution of the algorithm. Then, choosing $\kappa_e \in \mathcal{O}(\epsilon_e)$ for all such edges, an algorithm $A \in \mathcal{A}$ exhibits a stable gradient skew of $\mathcal{S}^\infty(d) \in \mathcal{O}(d \log_{\mu/\rho}(D/d))$, with a stabilization time of $\mathcal{O}(D/\mu)$. Specifically, if a path $p = (u, \dots, v)$ exists at time t and it has already existed for $\mathcal{O}(D/\mu)$ time, then $|L_u(t) - L_v(t)| \in \mathcal{O}(\varepsilon_p \log_{\mu/\rho}(D/\varepsilon_p))$. These bounds are asymptotically optimal.*

Proof. Let $G_{\mathcal{S}} = (V, E_{\mathcal{S}})$ denote the (dynamic) subgraph of the estimate graph G induced by all edges $\{u, v\}$ with $u \in N_v^{s_{\max}(v)}(t)$ and $v \in N_u^{s_{\max}(u)}(t)$. Note that since $\hat{\mathcal{G}}^{(r)} \in \mathcal{O}(D)$ for all $r \in \mathbb{N}$, Lemma 4.4 states that any edge that existed at least for the last $\Delta t \in \mathcal{O}(D/\mu)$ time units meets this criterion. Hence, for the first claim it is sufficient to show that in $G_{\mathcal{S}}$ we have a gradient skew of $\mathcal{S}^\infty(d) \in \mathcal{O}(d \log_{\mu/\rho}(D/d))$ at all times.

To this end, assume that at time t, $u, v \in V$ are two nodes in weighted distance $d \leq D$ in the graph $G_{\mathcal{S}}$, i.e., a shortest path p from u to v in $G_{\mathcal{S}}$ has weight $\epsilon_e = d$. Let $r, s \in \mathbb{N}$ be the unique integers such that $L_u(t) \in [T_1^{(r)}, T_1^{(r+1)})$ and $\kappa_e \in [C_s^{(r-1,r)}, C_{s+1}^{(r-1,r)})$, where $C^{(r-1,r)}$ is the gradient sequence from Corollary 5.11. Recall that $\sigma \in \Theta(\mu/\rho)$. The preconditions that $\hat{\mathcal{G}}^{(r)} \in \mathcal{O}(D)$ for all $r \in \mathbb{N}$ and $\kappa_e \in \mathcal{O}(\epsilon_e)$ for all edges e that exist at time t yield that $s \in \Theta(\log_\sigma(D/\kappa_p)) = \Theta(\log_{\mu/\rho}(D/d))$. Hence, as legality holds at node u at time t with respect to $C^{(r-1,r)}$ according to Corollaries 5.11 and 5.12, Lemma 5.3 states that

$$|L_u(t) - L_v(t)| \leq (s+1)C_s \in \mathcal{O}\left(d \log_{\mu/\rho}(D/d)\right)$$

as claimed. Since this bound is increasing with regard to d, it extends to arbitrary (i.e., non-shortest) paths from u to v in $G_{\mathcal{S}}$.

The asymptotic optimality of the stabilization time follows from Theorem 6.1 of the paper. The asymptotic optimality of the gradient property is shown in [12]. □

6 Lower Bound

In this section, we strengthen the lower bound in [8] to match the stabilization time of $\mathcal{A}^{\text{OPT}}$. The original lower bound stated, roughly speaking, that the stabilization time of any $\mathcal{S}$-dynamic gradient CSA with a stable gradient skew of $\mathcal{S}^\infty$ cannot be better than $\Omega(D/\mathcal{S}^\infty(1))$ in graphs of diameter D. For CSA with $\mathcal{O}(\log_{1/\rho} D)$-local skew, this bound implies that the stabilization time must be $\Omega(D/\log_{1/\rho} D)$. Algorithm $\mathcal{A}^{\text{OPT}}$ has a stabilization time of $\mathcal{O}(D)$, which does not match the bound in [8]; however, by refining the analysis in the lower bound we can show that the algorithm is in fact asymptotically optimal in its stabilization time. In the stronger bound we reason about the *full* gradient property, which bounds the skew on paths of all distances, rather than just the local skew property, which bounds the skew on single edges.

Let us call a dynamic gradient CSA *non-trivial* if it has a stable gradient skew satisfying $\mathcal{S}^{\infty}(1) \in o(D)$. This essentially means that the algorithm guarantees a *local* skew (e.g., along single edges) that is better than the global skew.

Theorem 6.1. *Let* $\mathcal{F} = \left\{ f_D : \mathbb{R}_0^+ \to \mathbb{R}_0^+ \mid D \in \mathbb{R} \right\}$ *be a family of functions, and let* $c_1, c_2 \in (0, 1/16)$ *be constants such that for all* $f_D \in \mathcal{F}$ *we have* $f_D(c_1 D) \leq c_2 D$. *Let* $\mathcal{A}$ *be a non-trivial stabilizing CSA guaranteeing a dynamic gradient skew of* f_D *in graphs of weighted diameter* D. *Then the stabilization time of* $\mathcal{A}$ *is at least* $\Omega(D)$.

Proof Sketch. We show that for sufficiently large diameters D, we can add a new edge and cause the skew on it to be larger than $\mathcal{S}$ after $\Omega(D)$ time. For simplicity we consider only line networks, where $D \in \Theta(n)$, but the proof can easily be modified to hold in general networks.

Consider a static line graph over $n+1$ nodes $v_0, \ldots, v_n$, where the estimate graph is the same as the communication graph and the weights of all edges are T. The diameter of the graph is $D = nT$. Let c_1, c_2 be the constants from the statement of the theorem, and let $u := v_{\lceil c_1 n \rceil}, v := v_{\lfloor n - c_1 n \rfloor}$. Finally, let $t_s \geq \mathcal{T}_S$ be some time after the stabilization time of the algorithm. By definition of $\mathcal{T}_S$, at any time after t_s, the skew on any path of weight d cannot exceed $2 f_D(d)$.

The distance between v_0 and u and between v and v_n is at least $c_1 n$; thus, for all $t \geq t_s$ we have

$$\begin{aligned} L_{v_0}(t) - L_u(t) &\leq f_D(c_1 n) \leq c_2 n, \\ L_v(t) - L_{v_n}(t) &\leq f_D(c_1 n) \leq c_2 n. \end{aligned}$$

Also, $\mathrm{dist}(u, v) \geq n - c_1 n - 1 - (c_1 n + 1) = n - 2c_1 n - 2$.

In [8], we show that we can create an execution E in which

(a) There exists a time $t_2 \geq t_s$ such that $L_u(t_2) - L_v(t_2) \geq \frac{1}{4}\,\mathrm{dist}(u, v) \geq n - 2c_1 n - 2$, and

(b) The message delays on all edges between v_0 and u and between v and v_n are always at least $T/(1+\rho)$.

Next we create a new execution E', which is identical to E until time $t_1 := t_2 - c_1 n \cdot T/(1+\rho)$. At time t_1 in E', an edge between v_0 and v_n appears. Our goal is to maintain a large skew on $\{v_0, v_n\}$ until time t_2 to show that the algorithm has not stabilized by then.

Due to the large message delays, nodes u, v do not find out about the new edge until time t_2. Consequently their skew in execution E' is the same as in E. The paths $v_0, v_1, \ldots, u$ and $v, \ldots, v_{n-1}, v_n$, which have weight at least $c_1 n$ by definition, are stable in both E and E'. Thus, the skew on each path cannot exceed $2 f_D(c_1 n)$, that is, it cannot exceed $2 c_2 n$. It follows that the skew between v_0 and v_n at time t_2 in E' is at least

$$\begin{aligned} L_{v_0}(t_2) - L_{v_n}(t_2) &= L_{v_0}(t_2) - L_u(t_2) + L_v(t_2) - L_{v_n}(t_2) + L_u(t_2) - L_v(t_2) \\ &\geq n - 2c_1 n - 2 - 4c_2 n > n/2 - 2. \end{aligned}$$

For sufficiently large n, this value exceeds $\mathcal{S}^{\infty}(1)$, since we assumed that $\mathcal{S}^{\infty}(1) \in o(D)$. Thus, we have showed that after $c_1 n \cdot T/(1+\rho) \in \Omega(D)$ time since edge $\{v_0, v_n\}$ appeared, the algorithm has not yet stabilized. □

References

[1] S. Biaz and J. Lundelius Welch. Closed Form Bounds for Clock Synchronization Under Simple Uncertainty Assumptions. *Information Processing Letters*, 80(3):151–157, 2001.

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

[3] D. Dolev, J. Halpern, and R. Strong. On the Possibility and Impossibility of Achieving Clock Synchronization. In *Proc. 16th ACM Symposium on Theory of Computing (STOC)*, pages 504–511, 1984.

[4] J. Elson, L. Girod, and D. Estrin. Fine-grained Network Time Synchronization Using Reference Broadcasts. *ACM SIGOPS Operating Systems Review*, 36:147–163, 2002.

[5] R. Fan and N. Lynch. Gradient Clock Synchronization. In *Proc. 23rd ACM Symposium on Principles of Distributed Computing (PODC)*, pages 320–327, 2004.

[6] J. Halpern, B. Simons, R. Strong, and D. Dolev. Fault-tolerant Clock Synchronization. In *Proc. 3rd ACM Symposium on Principles of Distributed Computing (PODC)*, pages 89–102, 1984.

[7] F. Kuhn, C. Lenzen, T. Locher, and R. Oshman. Optimal Gradient Clock Synchronization in Dynamic Networks. In *Proc. 29th ACM Symposium on Principles of Distributed Computing (PODC)*, 2010.

[8] F. Kuhn, T. Locher, and R. Oshman. Gradient Clock Synchronization in Dynamic Networks. In *Proc. 21st ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*, pages 270–279, 2009.

[9] F. Kuhn and R. Oshman. Gradient Clock Synchronization Using Reference Broadcasts. In *Proc. 13th International Conference on Principles of Distributed Systems (OPODIS)*, pages 204–218, 2009.

[10] L. Lamport and P. Melliar-Smith. Synchronizing Clocks in the Presence of Faults. *Journal of the ACM (JACM)*, 32(1):52–78, 1985.

[11] C. Lenzen, T. Locher, and R. Wattenhofer. Clock Synchronization with Bounded Global and Local Skew. In *Proc. 49th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 500–510, 2008.

[12] C. Lenzen, T. Locher, and R. Wattenhofer. Tight Bounds for Clock Synchronization. *Journal of the ACM (JACM)*, 57(2):1–42, 2010.

[13] T. Locher. *Foundations of Aggregation and Synchronization in Distributed Systems.* PhD thesis, ETH Zurich, 2009.

[14] T. Locher and R. Wattenhofer. Oblivious Gradient Clock Synchronization. In *Proc. 20th International Symposium on Distributed Computing (DISC)*, pages 520–533, 2006.

[15] J. Lundelius and N. Lynch. A New Fault-tolerant Algorithm for Clock Synchronization. In *Proc. 3rd ACM Symposium on Principles of Distributed Computing*, pages 75–88. ACM, 1984.

[16] J. Lundelius and N. Lynch. An Upper and Lower Bound for Clock Synchronization. *Information and Control*, 62(2/3):190–204, 1984.

[17] K. Marzullo and S. Owicki. Maintaining the Time in a Distributed System. In *Proc. 2nd ACM Symposium on Principles of Distributed Computing (PODC)*, pages 44–54, 1983.

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

[19] B. Patt-Shamir and S. Rajsbaum. A Theory of Clock Synchronization. In *Proc. 26th ACM Symposium on Theory of Computing (STOC)*, pages 810–819, 1994.

[20] T. K. Srikanth and S. Toueg. Optimal Clock Synchronization. *Journal of the ACM*, 34(3):626–645, 1987.