How to Speed-up Fault-Tolerant Clock Generation in VLSI Systems-on-Chip via Pipelining

Andreas Dielacher[1], Matthias Függer[⋆], and Ulrich Schmid[2]

[1] RUAG Aerospace Austria, Stachegasse 16, A-1120 Vienna (Austria). andreas.dielacher@ruag.com
[2] Technische Universität Wien, Institut für Technische Informatik, Embedded Computing Systems Group E182/2, Treitlstrasse 1-3, A-1040 Vienna (Austria). {fuegger,s}@ecs.tuwien.ac.at

Abstract. In this paper[1], we demonstrate that pipelining is a viable approach for speeding up the distributed fault-tolerant DARTS clock generation approach introduced in (Függer, Schmid, Fuchs, Kempf, EDCC'06), where a distributed Byzantine fault-tolerant tick generation algorithm has been used to replace the traditional quartz oscillator and highly balanced clock tree in VLSI Systems-on-Chip (SoCs). We provide a pipelined version of the original DARTS algorithm, termed pDARTS, together with a novel modeling and analysis framework for hardware-implemented asynchronous fault-tolerant distributed algorithms, which is employed for rigorously analyzing its correctness & performance. Our results, which have also been confirmed by an experimental evaluation of an FPGA prototype implementation, reveal that pipelining indeed allows to entirely remove the adverse effect of large interconnect delays on the achievable clock frequency, and demonstrate again that methods and results from distributed algorithms research can successfully be applied in the VLSI context.

Key words: Fault-tolerant distributed algorithms, VLSI, clock synchronization, pipelining, modeling approaches.

1 Motivation

Modern *very-large scale integration* (VLSI) circuits, in particular, *systems-on-chip* (SoCs), have much in common with the loosely-coupled distributed systems that have been studied by the fault-tolerant distributed algorithms community for decades [1, 2]. It is hence tempting to try and employ distributed algorithms results and methods in this new application domain. Recent work e.g. on scheduling of DRAM memory requests [3] and hardware-implemented transactional memory in multicores [4], fault-tolerant clock generation in SoCs [5], and self-stabilizing microprocessors [6] confirm that this is indeed feasible and quite promising. Conversely, results and methods from VLSI design have also been applied successfully in the distributed algorithms context. Examples are error-correcting codes, which allow to efficiently cope with Byzantine adversaries [7] and bear interesting relations to fault-tolerant consensus [8], and *pipelining*, the most important paradigm for concurrency in VLSI design (see Appendix A for a short introduction), which is also a well-known technique for speeding up synchronous distributed algorithms.

This paper[2] extends and integrates techniques from both distributed algorithms and VLSI design for developing and proving correct a pipelined version of the DARTS fault-tolerant clock generation approach for SoCs introduced in [5]: Instead of using a quartz oscillator and a clock tree for disseminating the clock signal throughout the chip, DARTS employs a Byzantine fault-tolerant distributed tick generation algorithm. The latter is a variant of Srikanth & Toueg's consistent broadcasting primitive [10] introduced in [11], which has been adapted to the particular needs of a VLSI implementation [5, 12]. Unfortunately, since the frequency of an ensemble of DARTS clocks is solely determined by the end-to-end delays along certain paths (which depend on the physical dimensions of the chip and hence cannot be made arbitrarily small), the maximum clock frequency is limited. For example, our first FPGA prototype implementation ran at about 24 MHz; our recent space-hardened 180 nm CMOS DARTS ASIC runs at about 55 MHz. Fortunately, pipelining comes as a rescue for further speeding-up the clock frequency here; first estimates predict a clock frequency of 200 MHz for our 180 nm technology, for example.

The purpose of our paper is actually two-fold: First, the resulting pDARTS algorithm demonstrates that pipelining is not only effective for speeding-up synchronous distributed algorithms, but also fault-tolerant asynchronous ones in systems with large bandwidth$\times$delay products. Note that the latter shows a clear rising trend in modern distributed systems, since the bandwidth provided by state-of-the-art computer networks and processors — as well as by VLSI data

[⋆] Corresponding author. T:+43(1)58801-18203, F:+43(1)58801-18297
[1] This work is supported by the Austrian bm:vit FIT-IT project *DARTS* (proj. no. 809456-SCK/SAI) and the Austrian Science Foundation (FWF) projects P17757 (Theta) and P20529 (PSRTS).
[2] A Brief Announcement [9] of this work will be presented at PODC'09.

paths and circuits — has tremendously increased, while the spatial distances between nodes — including inter-chip and on-chip communication — remain essentially the same. In the asynchronous context, pipelining exploits the fact that any FIFO data transmission/processing path in a distributed system, ranging from asynchronously "computing" logic gates interconnected by simple wires to multiprocessors nodes (CPU, network processors, DMA controllers, etc.) interconnected via FIFO network links, has an inherently pipelined architecture. For example, a simple message channel [even a wire] with bandwidth 10^9 messages/s and delay 1 μs is able to "store" $10^9 \cdot 10^{-6} = 1000$ messages. A fault-tolerant distributed algorithm may hence immediately start phase k (rather than wait for the acknowledgments of the previous data processing phase $k-1$ in a "stop-and-go fashion"), provided that the acknowledgments for phase $k-x$ (for some integer $x > 1$) have already been received from sufficiently many correct processes.

And second, the new modeling framework used for the correctness proof and performance analysis of pDARTS shows how to cope with the quite special situation of fault-tolerant distributed algorithms developed for VLSI circuits: Appropriate algorithms are made up of a typically large number of simple building blocks (a multiplication is already non-trivial here) consisting of logic gates, interconnected by simple wires carrying boolean signals, which "compute" asynchronously, continuously and concurrently. Our modeling framework, which is based on continuous time and zero-bit FIFO channels with delay, facilitates "switching" between different — but consistent — views (state, transition and counting view) of the same signal. In sharp contrast to existing modeling frameworks capable of expressing timed executions,[3] these features allow to express the properties of fault-tolerant distributed algorithms designed for a direct implementation in (partially) synchronous/asynchronous digital logic in a very natural and simple way.

Detailed contributions: (1) We adapt the Byzantine fault-tolerant distributed tick generation algorithm introduced in [11] for pipelined execution, and make it suitable for a direct implementation in asynchronous digital logic. Like the original DARTS [5], the resulting pipelined pDARTS algorithm achieves this by enforcing certain atomic actions ("interlocking") via implicit handshaking, and by replacing multi-bit messages by zero-bit messages, i.e., by anonymous up/down signal transitions. (2) We introduce the relevant parts of our novel modeling and analysis framework for fault-tolerant distributed algorithms designed for a direct implementation in VLSI. (3) We present the cornerstones[4] of the correctness proof of the pDARTS algorithm, and the worst case bounds for performance metrics like synchronization precision and minimum/maximum clock frequency. Since our "system-level proof" rests on fundamental properties of simple basic building blocks only, it effectively reduces the complex problem of guaranteeing system correctness to the simple problem of assuring the correctness of the implementations of the basic blocks. (4) We provide a glimpse of the results of the experimental evaluation of pDARTS in an FPGA prototype system, which confirm the feasibility and efficiency of our approach.

2 Informal Overview

The basic idea of DARTS[5] is to replace the common quartz oscillator and the clock tree that disseminates the clock in a SoC by a fully distributed GALS-like approach (globally asynchronous, locally synchronous [15]): Every functional unit Fu_i in the SoC has attached a dedicated fault-tolerant tick generation unit (TG-Alg) here, which generates Fu_i's local clock signal. To accomplish this, all TG-Algs communicate with each other over a simple "network" of clock signals (TG-Net). In contrast to GALS, however, DARTS ensures that the local clock signals of different Fu's are synchronized to each other to within a few clock cycles. Since we have proved elsewhere [16] that even such loose synchrony suffices for implementing metastability-free high-speed communication between different Fu's, SoCs built atop of DARTS do not need asynchronous communication mechanisms, i.e., handshaking. Besides fault-tolerance, DARTS clocks (patented in [17]) provide a number of additional advantages, which make them particularly promising for critical applications e.g. in the aerospace domain.

Like DARTS, the pipelined pDARTS TG-Alg developed and analyzed in this paper derives from a simple synchronizer for the Θ-Model [18, 19] introduced in [11]. Its pseudo-code description is given in Fig. 1; X is a system

[3] Our framework is substantially different from existing modeling frameworks for asynchronous VLSI circuits ("trace theory"), which are time-free and hence cannot deal with failures [13]. See Section 6 for an overview of related work.

[4] Lacking space does not allow us to include the complete proof in the paper. To assist reviewing, all the details are provided in an appendix and in the technical report [14].

[5] See our project web page *http://ti.tuwien.ac.at/darts* for further details.

parameter determined by the inherent pipeline depth of the system: If X is chosen well (see Section 5 for dimensioning issues), then pDARTS will generate clock ticks with a frequency that is independent of the TG-Net delay. Note that, at this level of description, pDARTS for $X = 0$ is the same as the original DARTS.

The algorithm assumes a message-driven system (where nodes make atomic receive-compute-send steps whenever they receive a message) of $n = 3f + 1$ nodes (= TG-Alg instances), at most f of which may behave arbitrarily faulty, i.e., Byzantine. The nodes are connected by a reliable[6] point-to-point message-passing network (= TG-Net): No spurious messages are ever generated by the network, no messages are lost or altered, and all messages sent at time t are received within the interval $t + [\tau^-, \tau^+]$, where τ^- (resp. τ^+) denotes the (possibly unknown) lower (resp. upper) bound on the end-to-end delay of messages exchanged between correct nodes. Let $\varepsilon = \tau^+ - \tau^-$ be the maximum uncertainty of the message delay, and $\Theta = \tau^+/\tau^-$ the maximum delay ratio.

1: VAR k: integer /* *Local clock value* */
2: send tick($-X$) ... tick(0) to all /* *At booting time* */
3: k:=0
4: **if** received tick(ℓ) from at least $f + 1$ distinct nodes with $\ell > k$ **then**
5: send tick($k + 1$) ... tick(ℓ) to all [once]
6: $k := \ell$
7: **if** received tick(ℓ) from at least $2f + 1$ distinct nodes, with $\ell \geq k - X$ **then**
8: send tick($k + 1$) to all [once]
9: $k := k + 1$

Fig. 1. TG-Alg for pDARTS

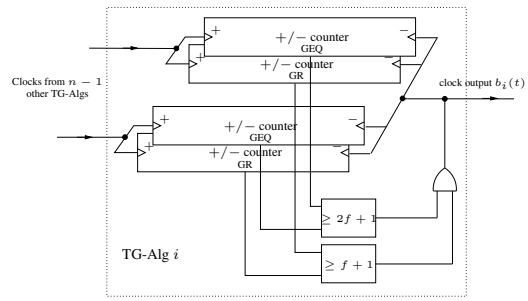

Fig. 2. Basic architecture of a pDARTS TG-Alg.

The algorithm of Fig. 1 works as follows: Initially, every node broadcasts tick($-X$), ..., tick(0). Note that all correct nodes are assumed to initialize (almost) at the same time, in the sense that they don't lose any message due to late booting. If node p receives $f + 1$ tick(ℓ) messages (line 4), it can be sure that at least one of those was broadcast by a correct node. Therefore, p can safely catch up and send tick($k + 1$), ..., tick(ℓ). If some correct node p receives $2f + 1$ tick($k - X$) messages (line 7) and thus broadcasts tick($k + 1$), one can be sure that all messages among the $2f + 1$ ones that were broadcast by correct nodes, i.e., at least $f + 1$, will be received within ε by every other node. Hence, every correct node will execute line 4 and send tick($k - X$) by that time at latest. It follows that all tick(ℓ), $k - X \leq \ell \leq k$, occur quite close to each other at every correct node.

Our detailed analysis will reveal that this indeed suffices to prove that correct nodes generate a sequence of consecutive messages tick(k), $k \geq 1$, in a synchronized way (see Section 3.4): If $b_p(t)$ denotes the value of the variable k (cf. Fig. 1) of the TG-Alg at node p at real-time t, which gives the number of tick(k) messages broadcast so far, it turns out that $(t_2 - t_1)\alpha_{\min} \leq b_p(t_2) - b_p(t_1) \leq (t_2 - t_1)\alpha_{\max}$ for any correct node p and $t_2 - t_1$ sufficiently large ("accuracy"); the constants $\alpha_{\min}$ and $\alpha_{\max}$ depend on τ^-, τ^+ and X. Moreover, every two correct nodes p, q maintain $|b_p(t) - b_q(t)| \leq \pi$ ("precision"), for a small constant π that depends on Θ and X only.

Comparison with the original DARTS algorithm [5] reveals that only the progress rule (line 7) had to be changed in order to implement pipelining:[7] Rather than just waiting for tick(ℓ) with $\ell \geq k$, as in the original DARTS algorithm, the pDARTS algorithm waits for $\ell \geq k - X$. Note that our first intuition was to incorporate X also in the catchup rule (line 4), which would have considerably reduced the complexity of the low-level hardware implementation of pDARTS. It turned out, however, that doing this causes the algorithm to fail.

Since the pDARTS algorithm in Fig. 1 looks very simple, it is tempting to conclude that it is easily implemented in hardware: Node p's TG-Alg just needs to drive a boolean-valued clock signal, where it outputs the k-th signal transition when it sends its tick(k) message; the TG-Net is formed by feeding all clock signals to all TG-Algs (e.g., by a bus). In [5], however, it turned out that several challenging issues (see Appendix B) must be solved in order to arrive at a low-level version of the pDARTS TG-Alg as depicted in Fig. 2. The major building blocks of a single TG-Alg are $2(n - 1)$ custom $+/-$ *counters*, two for each of the $n - 1$ other TG-Algs q in the system. The two $+/-$ counters for remote TG-Alg q at local TG-Alg p are responsible for maintaining $\ell_q - k_q > 0$ (resp. $\ell_q - k_q \geq -X$), which are

[6] This reliable network assumption is not unduly restrictive, since communication failures can be mapped to failures of the sending node.
[7] Note that our "top-down approach" for incorporating pipelining is quite different from the way it is usually done in VLSI design.

required for implementing line 4 (resp. line 7) in Fig. 1. Herein, ℓ_q denotes the number of tick-messages seen from q at the $+/-$ counter so far, whereas k_q denotes the number of already perceived tick-messages from the own TG-Alg p. The status signals GR (resp. GEQ) signal when the corresponding inequality holds. In addition, a "$\geq f+1$" (resp. "$\geq 2f+1$") *threshold circuit* implements the rules in line 4 (resp. line 7) in Fig. 1. Finally, there is a device (shown as an OR-gate in Fig. 2), which is responsible for generating every local clock tick exactly once from the outputs of the threshold gates. These local ticks are not only broadcast to all $n-1$ remote TG-Algs via remote links, but are also fed back to each of the $2(n-1)$ $+/-$ counters locally.

Note that, in order to circumvent ambiguities, we had to split the signal GR into two signals GR^e and GR^o, tied to even and odd ticks, respectively: GR^e is true iff the inequality $\ell_q - k_q > 0$ holds and $k_q \in \mathbb{N}_{even} := 2\mathbb{N}$, whereas GR^o is true iff the inequality $\ell_q - k_q > 0$ holds and $k_q \in \mathbb{N}_{odd} := 2\mathbb{N}+1$. The associated threshold circuit is also duplicated, providing one for the GR^e signals and one for the GR^o signals. The same splitting is done for the signals GEQ and their threshold circuits. Finally, all their outputs are combined to generate p's odd and even ticks.

3 Formalization

It has been highlighted in the previous section that even the simple distributed algorithm presented in Fig. 1 makes use of design elements that are not available or too costly at the hardware implementation level. Correctness proofs and performance analyses using standard distributed computing models would hence suffer from a substantial "proof gap" w.r.t. the actual implementation, which substantially diminishes their value. Consequently, we will base our formal analysis on a more low-level model for specifying the building blocks and analyzing the execution of fault-tolerant distributed algorithms that are implemented directly in asynchronous digital logic. Note that our framework is substantially different from existing modeling frameworks for asynchronous VLSI circuits like trace theory, which are time-free and hence cannot deal with failures [13], and is also different from the framework used for the correctness proof and performance analysis [5] of the original DARTS.

VLSI distributed algorithms typically have a hierarchical structure: For example, the top-level of pDARTS is made up of n TG-Algs interconnected via the signal wires making up the TG-Net. Every TG-Alg can be further partitioned into several building blocks (like the $+/-$ counters), which are interconnected in some non-regular way. Our model will abstract from these internals by considering *modules*, which possess input and output *ports* (boolean signals). A module's *behavior* specifies how the signals on the input and output ports are related. Note that modules differ from timed automatons [20] primarily in that they *continuously* compute their outputs, based on the history of their inputs.

Compound modules consist of multiple sub-modules and their interconnect, which specifies how sub-module ports are connected to each other and to the module's input/output ports. Note that the interconnect specification itself assumes zero delays; modeling non-zero interconnect delays, e.g., for real wires, requires intermediate channels: A channel possesses a single input port and a single output port, and its behavior specifies delayed FIFO delivery of input port signal transitions at the output port. Modules that are not further refined are called *basic modules*. Elementary basic modules are zero-delay boolean functions (AND, OR, ...) and channels.

Clearly, the behavior of a (non-faulty) composite module is determined by the behavior of its constituent sub-modules; the behavior of a basic module must be given *a priori*. An execution of a system is specified by the behaviors of each of its signals, and is typically modeled as a set of event traces (see below). Correctness proofs establish properties of the behaviors of higher-level modules, based on the assumption that (1) the system and failure model holds, and (2) that the implementations of the basic modules indeed satisfy their behavioral specification.

3.1 Signals and zero-bit message channels

Since we target implementations using asynchronous circuits, our formal framework will be based on a continuous notion of real-time $t \in \mathbb{R}_0^+$. We assume that the system initialization (reset) occurs at time $t = 0$.

A *signal* S may be either represented (i) by its event trace, (ii) by its status, or (iii) by its counting function. These representations are consistent, in a well-defined way, and can hence be used interchangeably.

(i) Event trace: The representation of S by an event trace, denoted by $\widehat{S}$, is specified by a relation $\widehat{S} \subseteq \mathbb{R}_0^+ \times \{0,1\}$, where event $(t,1) \in \widehat{S}$ (resp. $(t,0) \in \widehat{S}$) means that S takes on value 1 (resp. 0) at real-time t. In order to enforce a unique initial value, we require either $(0,1) \in \widehat{S}$ or $(0,0) \in \widehat{S}$. We further demand non-simultaneity of contradicting

events for any single signal, i.e., $((t,x) \in \widehat{S}) \wedge ((t,y) \in \widehat{S}) \Rightarrow (x = y)$. Let $\text{pre}(\widehat{S},t) := \{(t',v') \in \widehat{S} \mid t' \leq t\}$ denote the prefix of $\widehat{S}$ until time t. In this paper, we will restrict our attention to event traces where only finitely many alternating events (i.e., with different value) can occur in any finite time interval. Due to this restriction,

$$\text{last-val}(\widehat{S},t) := v' \text{ s.t. } \exists (t',v') \in \text{pre}(\widehat{S},t) : \forall (t'',v'') \in \text{pre}(\widehat{S},t) : (t'' \geq t') \Rightarrow (v'' = v')$$

is always well-defined. Note that this definition still allows arbitrarily many idempotent events to occur in a prefix.
(ii) Status: Since idempotent events do not change the state of a signal, they are often irrelevant. Idempotent events can be abstracted away by considering the signal S's status representation, denoted by $\widetilde{S}$, which is a function $\widetilde{S} : \mathbb{R}_0^+ \to \{0,1\}$ from time t to the boolean value of S at time t. Obviously, signals may be composed out of already defined signals by using arbitrary boolean predicates, e.g., $\widetilde{A} := \widetilde{B} \wedge \widetilde{C}$, with signals $\widetilde{B}, \widetilde{C}$ defined as $\widetilde{A}(t) := \widetilde{B}(t) \wedge \widetilde{C}(t)$.

One can easily switch between the two representations of S: If given the event trace $\widehat{S}$ of S, the equivalent status representation $\widetilde{S}$ of S can be obtained by $\widetilde{S}(t) := \text{last-val}(\widehat{S},t)$. Given $\widetilde{S}$, one cannot regenerate the "original" event trace $\widehat{S}$, since all idempotent events have been lost. Still, most of the time, it suffices to obtain *some* event trace $\widehat{S}'$ that has the same status representation as $\widehat{S}$. Such an event trace can simply be obtained from $\widetilde{S}$ by $\widehat{S} := \{(t, \widetilde{S}(t)) \mid t \geq 0\}$; note that $\widehat{S}$ contains continuum-many events.
(iii) Counting function: Finally, a signal S can be represented by the number of non-idempotent events that occur during $(0,t]$, denoted as the counting function $S(t)$. For example, if $\widehat{S} = \{(0,0),(1,1),(1.5,1),(2,0)\}$, then $S(0) = 0$ and $S(2) = 2$. Sometimes, we will also employ generalized counting functions $S'(t)$ that have an initial value other than 0: We define $S'(t) := S(t) + S_0$, where $S(t)$ is the standard counting function of S and S_0 an arbitrary offset.

It is again easy to switch between the counting function representation of S and the other representations: $S(t)$ can be obtained from $\widehat{S}$ by just counting the non-idempotent events in $\widehat{S}$ (excluding the initial event), and getting $\widehat{S}$ from $S(t)$ [for some given initial event $(0,I)$, $I \in \{0,1\}$] is accomplished via $\widehat{S} := \{(t, [I + S(t)] \bmod 2) \mid t \geq 0\}$. Switching between the status representation and the counting function can be done transitively via the corresponding event trace representation, or directly via $\widetilde{S}(0) := 0$, $\widetilde{S}(t) := [I + S(t)] \mod 2$ for $t > 0$.

A *channel* models a reliable FIFO channel for signal transitions with finite delay. Since signal transitions must be alternating, only the occurrence time but no data can be conveyed over a single channel ("zero-bit messages"). Formally, the semantics of a channel X is as follows: Let X^s be the channel's single input port [which will be connected to an output port of a single sender module], and X^r be its single output port [which will be connected to the input ports of some receiver module(s)]. There exists a continuous and strongly monotonically increasing *delivery function* $f : \mathbb{R}_0^+ \to \mathbb{R}_0^+$ for X, which maps sending time t to delivery time $f(t)$. We assume that the channel delay is within $\left[\tau_X^-, \tau_X^+\right]$, i.e., $f(t) - t \in \left[\tau_X^-, \tau_X^+\right]$. From the properties of f, it follows immediately that f is a bijection from $\mathbb{R}_0^+$ to its codomain $f(\mathbb{R}_0^+)$. More specifically, f maps every closed interval $[t_1, t_2]$ bijectively to the closed interval $[f(t_1), f(t_2)]$. Clearly, the inverse function f^{-1} of f also exists and has the same properties. In addition, we will assume that the channel output has some well-defined initial state (is initialized to) $I \in \{0,1\}$, which is $I = 0$ if not specified otherwise. Given f, the channel's behavior in terms of event traces is

$$\begin{aligned} &(0,I) \in \widehat{X}^r \wedge \nexists (t,v) \in \widehat{X}^r \text{ with } v \in \{0,1\}, t \in [0, f(0))\,, \text{ and} \\ &(f(t),x) \in \widehat{X}^r \Leftrightarrow (t,x) \in \widehat{X}^s. \end{aligned} \tag{1}$$

Since f carries over the total order of the events (t,x) in $\widehat{X}^s$ to the events $(f(t),x)$ in $\widehat{X}^r$ [called *matching* events in the sequel], it follows that $\widehat{X}^r$ is an event trace. In terms of states, the channel behavior can be defined as

$$\forall t \in [0, f(0)) : \widetilde{X}^r(t) := \widetilde{X}^r(0) = I \text{ and } \forall t \geq 0 : \widetilde{X}^r(f(t)) := \widetilde{X}^s(t).$$

3.2 pDARTS System Model

The pDARTS system consists of a set P of $n := |P|$ top-level modules, where $n \in \mathbb{N}$. These top-level modules will interchangeably be called node/TG-Alg and are usually denoted by letters $p, q, \cdots \in P$. Every node p has exactly one output port with the counting function $b_p(t)$, and one input port per remote node $q \in P \setminus \{p\}$ with the counting function $r_{p,q}^{rem}(t)$. We assume a fully connected system, i.e., from every node p to every node $q \in P \setminus \{p\}$, there is a

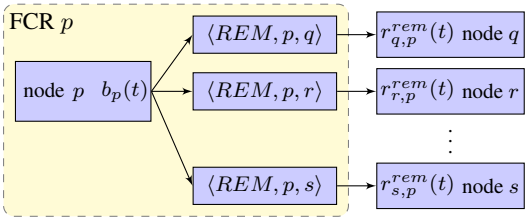

Fig. 3. Fault-containment region FCR p of node p

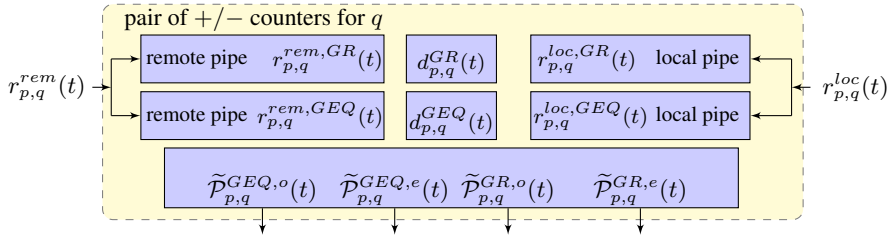

Fig. 4. Architecture of a pair of $+/-$ counters at p corresponding to q

channel $\langle REM, p, q\rangle$ with input $b_p(t)$, output $r^{rem}_{q,p}(t)$, and delay in $[\tau^-_{rem}, \tau^+_{rem}]$. Fig. 3 shows the resulting outbound channels of node p. Let $t_{p,boot}$ denote the time when correct node p completes booting, i.e., starts executing its TG-Alg. We require that $t_{p,boot} \in [0, B]$ for some constant B, where $B \leq \tau^-_{rem}$. Due to this assumption, messages sent by p may not get lost at any correct node q because of late booting.

The following notation will be used throughout the paper: For any $k \geq 1$, we say that node p sends tick k, at time $t_{p,k}$, if the kth event (without counting idempotent events) occurs at $t_{p,k}$. The time when the first (resp. the last) correct node sends tick k is denoted by $t_{first,k}$ (resp. $t_{last,k}$); note that the node who is the first (resp. last) one to send tick k may be different for different k. Analogously, we say that p receives tick k from q at time t, if $r^{rem}_{p,q}(t) = r^{rem}_{p,q}(t^{\rightarrow}) + 1 = k$, where $t^{\rightarrow}$ it the time immediately before the reception takes place.

We partition our system into multiple *fault-containment regions* (FCRs), i.e., sets of modules that are potentially affected by a single fault like a particle hit and thus cannot be assumed to fail independently. More specifically, we define FCR p to consist of the single node p together with all its outgoing channels, as depicted in Fig. 3. If FCR p is correct, then each of its sub-modules behaves as specified in Section 3.3. If FCR p is faulty, any of its sub-modules may behave arbitrarily (Byzantine). [8] Since every FCR is associated with exactly one node, we will use these terms interchangeably as well. Throughout the paper, let C be the set of correct FCRs, and F, with $f := |F|$, the set of faulty FCRs. Clearly $P = C \cup F$ and $C \cap F = \emptyset$, i.e., C and F partition P. We will prove in Section 4 that correct nodes behave as specified in Section 3.4 in the presence of up to f Byzantine faulty FCRs, provided that the total number of nodes is $n \geq 3f + 2$. Note that this is slightly more than the required lower bound of $n \geq 3f + 1$ for clock synchronization [28], but facilitates a considerably better precision and accuracy.[9]

3.3 Specifications of TG-Alg basic modules

The internal architecture of a single TG-Alg, as described in Section 2, is obtained by expressing and refining Fig. 2 in terms of our formal model. Due to space limitations, we will only present the formalization of the $+/-$ counters here, which turned out to be the most intricate component anyway.[10]

As shown in Fig. 4, the implementation of every $+/-$ counter consists of two elastic pipelines [29], which can be seen as shift registers/FIFO buffers for signal transitions. One is attached to the remote clock signal, the other one is fed by the local clock signal. They are fitted together at their ends via a special *Diff-Gate*, which removes "matching" transitions as soon as they traveled through the pipelines. The status signals GEQ^o, GEQ^e, GR^o, and GR^e are provided by the *pipe compare signal generator (PCSG)* circuits, which monitor the last few stages of both pipes. Note that we did not include the channels arising in the (sub-)modules' interconnect in Fig. 4 for simplicity.

Pairs of elastic pipes: Every node p incorporates two pairs of elastic pipelines for every remote TG-Alg $q \in P \setminus \{p\}$. The pipepair responsible for the GEQ-rule is denoted $(p, q)_{GEQ}$, the one for the GR-rule $(p, q)_{GR}$. Every pair con-

[8] Since hardware faults easily lead to Byzantine failures [21], we assume this failure semantics here: The adverse power of Byzantine failures in our context lies in the ability of a faulty node to generate wrong clock ticks (early timing failures or even spurious) that are perceived inconsistently at different remote nodes. Such failures could be the consequence of manufacturing defects or electrostatic breakdown [22], particle hits [23, 24], or electromagnetic noise [25], which may affect any module in a TG-Alg. Due to different wire lengths and signal-level detection thresholds, such faults typically propagate differently to different receivers. Note that we allow faulty nodes to create even metastability [26], but we must assume that metastability cannot propagate beyond FCRs; we have already some convincing evidence [27] that this is ensured by the elastic pipelines in the $+/-$ counters with high probability.

[9] This follows from counting only remote messages when calculating the $f + 1$ and $2f + 1$ thresholds; including self-reception would lead to $\tau^-_{rem} = \tau^-_{loc}$ in Theorem 2, which spoils the achievable worst-case precision considerably.

[10] The specifications of the other parts have been relegated to Section C in the appendix.

sists of a remote pipeline that can store up to $S_{rem,geq}$ (resp. $S_{rem,gr}$) ticks sent by q, and a local pipeline that can hold up to $S_{loc,geq}$ (resp. $S_{loc,gr}$) ticks generated by p locally. The numbers $S_{rem,geq}, S_{rem,gr}, S_{loc,geq}$ and $S_{loc,gr}$ are implementation parameters that have to be chosen in accordance with the bounds of Section 3.4; in the specifications of this section, they are just assumed to be unbounded (arbitrary large).

The local pipe of $(p,q)_{GEQ}$ has a single input port represented by the counting function $r^{loc}_{p,q}(t)$, which is fed by TG-Alg p's local clock ticks $b_p(t)$ supplied via a channel $\langle LOC, p, q\rangle$, and a single output port represented by the counting function $r^{loc,GEQ}_{p,q}(t)$, which denotes the number of ticks that arrived at the output of the local pipe by time t. Similarly, the remote pipe of $(p,q)_{GEQ}$ has a single input port represented by the counting function $r^{rem}_{p,q}(t)$, which is fed by remote TG-Alg q's local clock ticks supplied via the channel $\langle REM, q, p\rangle$, cp. Fig. 3, and a single output port represented by the counting function $r^{rem,GEQ}_{p,q}(t)$, which denotes the number of ticks that arrived at the output of the remote pipe by time t. The same description applies to the pipepair $(p,q)_{GR}$, except that $r^{rem,GEQ}_{p,q}(t)$ and $r^{loc,GEQ}_{p,q}(t)$ is replaced by $r^{rem,GR}_{p,q}(t)$ and $r^{loc,GR}_{p,q}(t)$, respectively. Upon initialization, the remote pipe of every $(p,q)_{GEQ}$ is pre-filled with virtual ticks $-X,\ldots,0$, whereas every other pipe is initialized with virtual tick 0; these ticks are called virtual since they were never transmitted.

Behavioral Description: Every pipe has the behavior of a zero-delay[11] channel: $r^{rem,GR}_{p,q}(t) := r^{rem}_{p,q}(t)$, $r^{rem,GEQ}_{p,q}(t) := r^{rem}_{p,q}(t) + X$, $r^{loc,GR}_{p,q}(t) = r^{loc,GEQ}_{p,q}(t) := r^{loc}_{p,q}(t)$, where $X \geq 0$ is the pipeline depth parameter. The initial values are $r^{rem,GEQ}_{p,q}(t_{p,boot}) = X$ and $r^{rem,GR}_{p,q}(t_{p,boot}) = r^{loc,GR}_{p,q}(t_{p,boot}) = r^{loc,GEQ}_{p,q}(t_{p,boot}) := 0$. Since the values of $r^{loc,GR}_{p,q}(t)$ and $r^{loc,GEQ}_{p,q}(t)$ are equal to $r^{loc}_{p,q}(t)$, we will only deal with $r^{loc}_{p,q}(t)$ explicitly from now on.

Diff-Gate: To avoid pipes with infinite capacity, each pair of pipes is equipped with a special Diff-Gate circuit that removes matching clock ticks from their outputs, i.e., clock ticks contained in both pipes: The Diff-Gate for $(p,q)_{GEQ}$ has two input ports connected to $r^{rem,GEQ}_{p,q}(t)$ and $r^{loc,GEQ}_{p,q}(t)$, and a single output port represented by the counting function $d^{GEQ}_{p,q}(t)$, which gives the largest tick number that has been removed from both the remote and local pipe of $(p,q)_{GEQ}$ by time t. [The description of the Diff-Gate of $(p,q)_{GR}$ is very similar.]

Behavioral Description: Let $t^{GEQ}_{rmv,k}$, $k \geq 0$, be the time when tick k is removed from both the remote outputs $r^{rem,GEQ}_{p,q}(t)$ and the local output $r^{loc,GEQ}_{p,q}(t)$ of the pipepair $(p,q)_{GEQ}$, i.e., $d^{GEQ}_{p,q}(t^{GEQ}_{rmv,k}) = k$. Note that tick k showing up at the output $r^{rem,GEQ}_{p,q}(t)$ is actually tick $k - X$ as stored in the remote pipe.

(i) For all $0 \leq k \leq X$, tick k shows up at $r^{rem,GEQ}_{p,q}(t^{GEQ}_{rem,k})$ of $(p,q)_{GEQ}$ at time $t^{GEQ}_{rem,k} := t_{p,boot}$. *(ii)* The (non-existing) tick -1 is removed at $t^{GEQ}_{rmv,-1} := t_{p,boot}$, i.e., $d^{GEQ}_{p,q}(t_{p,boot}) = -1$. *(iii)* For all $k \geq 0$, if tick $k+1$ shows up at the output $r^{rem,GEQ}_{p,q}(t^{GEQ}_{rem,k+1})$ of the remote pipe of $(p,q)_{GEQ}$ at time $t^{GEQ}_{rem,k+1}$ and tick $k+1$ shows up at the output $r^{loc,GEQ}_{p,q}(t^{GEQ}_{loc,k+1})$ of the local pipe of $(p,q)_{GEQ}$ at time $t^{GEQ}_{loc,k+1}$ and tick $k-1$ is removed at time $t^{GEQ}_{rmv,k-1}$, then tick k is removed at $t^{GEQ}_{rmv,k}$, with $t^{GEQ}_{rmv,k} \in \max\{t^{GEQ}_{rem,k+1}, t^{GEQ}_{loc,k+1}, t^{GEQ}_{rmv,k-1}\} + [\tau^{-}_{Diff}, \tau^{+}_{Diff}]$. On top of the above defined signals $r^{loc}_{p,q}(t)$ and $d^{GEQ}_{p,q}(t)$, the size of the local pipe of $(p,q)_{GEQ}$ at time t is defined as $s^{loc,geq}_{p,q}(t) := r^{loc}_{p,q}(t) - d^{GEQ}_{p,q}(t)$. The other pipes' sizes $s^{rem,geq}_{p,q}(t)$, $s^{loc,gr}_{p,q}(t)$ and $s^{rem,gr}_{p,q}(t)$ are defined analogously.

Pipe Compare Signal Generators (PCSGs): The signals provided by the pipepairs and their Diff-Gates are connected to the PCSG, which generates four status signals $\widetilde{\mathcal{P}}^{GEQ,o}_{p,q}(t)$, $\widetilde{\mathcal{P}}^{GEQ,e}_{p,q}(t)$, $\widetilde{\mathcal{P}}^{GR,o}_{p,q}(t)$ and $\widetilde{\mathcal{P}}^{GR,e}_{p,q}(t)$ that characterize the difference of the number of clock ticks stored in the remote and local pipes by time t. Different signals are provided for odd and even clock ticks. For example, $\widetilde{\mathcal{P}}^{GEQ,o}_{p,q}(t)$ signals when the number of remote clock ticks is greater than or equal the number of local clock ticks, provided that the last clock tick that entered the local pipe was odd. All these signals are fed, via dedicated channels that add some delay, to the threshold modules of the TG-Alg p.

Behavioral Description:

$$\widetilde{\mathcal{P}}^{GEQ,o}_{p,q}(t) := [r^{rem,GEQ}_{p,q}(t) \geq r^{loc}_{p,q}(t)] \wedge [r^{loc}_{p,q}(t) \in \mathbb{N}_{odd}] \wedge [s^{loc,geq}_{p,q}(t) = 1]$$
$$\widetilde{\mathcal{P}}^{GR,o}_{p,q}(t) := [r^{rem,GR}_{p,q}(t) > r^{loc}_{p,q}(t)] \wedge [r^{loc}_{p,q}(t) \in \mathbb{N}_{odd}] \wedge [s^{loc,gr}_{p,q}(t) = 1]$$

Note that $\widetilde{\mathcal{P}}^{GEQ,o}_{p,q}(t)$ and $\widetilde{\mathcal{P}}^{GR,o}_{p,q}(t)$ need to be valid only if the local pipes contain exactly one tick. These signals are fed into dedicated channels, all of which are initialized to 0: $\langle \mathcal{P}^{GEQ} toGEQ, o, p, q\rangle$ with input $\widetilde{\mathcal{P}}^{GEQ,o}_{p,q}(t)$,

[11] Actually, the pipe delays are accounted for in the delays of the other channels in the signal path, namely, $\langle REM, q, p\rangle$ and $\langle LOC, p, q\rangle$.

output $\widetilde{GEQ}^{o}_{p,q}(t)$ and delay $[\tau^-_{GEQ}, \tau^+_{GEQ}]$, and $\langle \mathcal{P}^{GR}toGR, o, p, q\rangle$ with input $\widetilde{\mathcal{P}}^{GR,o}_{p,q}(t)$, output $\widetilde{GR}^{o}_{p,q}(t)$ and delay $[\tau^-_{GR}, \tau^+_{GR}]$. [Analogously, $\widetilde{\mathcal{P}}^{GEQ,e}_{p,q}(t)$ and $\widetilde{\mathcal{P}}^{GR,e}_{p,q}(t)$ and their channels are defined by substituting $\mathbb{N}_{odd}$ with $\mathbb{N}_{even}$.]
Threshold modules, tick generation module and interconnect: The GR (resp. GEQ) signals from the PCSGs are fed into the threshold modules, which signal within delay $[\tau^-_{TH}, \tau^+_{TH}]$ whether the $f+1$ (resp. $2f+1$) threshold has been reached. The thresholds' outputs are finally combined by the tick generation module, which actually produces every tick exactly once, and broadcasts it via the channels $\langle REM, p, q\rangle$ (within delay $[\tau^-_{rem}, \tau^+_{rem}]$) to all other TG-Algs, and to its own $+/-$ counters via the channels $\langle LOC, p, q\rangle$ (within delay $[\tau^-_{loc}, \tau^+_{loc}]$).

3.4 System-level properties

Correct TG-Algs are required to guarantee the following properties:

(P) **Precision** (see Theorem 2): There is a constant π, such that for every pair of correct nodes $p, q \in C$:

$$\forall t : |b_q(t) - b_p(t)| \leq \pi. \tag{2}$$

(A) **Accuracy** (see Theorem 3): There are constants R^-, O^-, R^+, $O^+ > 0$, such that for every correct node $p \in C$:

$$O^-(t_2 - t_1) - R^- \leq b_p(t_2) - b_p(t_1) \leq O^+(t_2 - t_1) + R^+. \tag{3}$$

(S) **Size**: There are constants $S_{rem,gr}$, $S_{loc,gr}$ and $S_{rem,geq}$, $S_{loc,geq}$, such that for every pair of correct nodes $p, q \in C$

$$s^{loc,gr}_{p,q}(t) \leq S_{loc,gr}, s^{rem,gr}_{p,q}(t) \leq S_{rem,gr}, s^{loc,geq}_{p,q}(t) \leq S_{loc,geq}, \text{ and } s^{rem,geq}_{p,q}(t) \leq S_{rem,geq}.$$

In the following Section 4, we will sketch the cornerstones of our proofs, which show that the TG-Algs at correct nodes indeed satisfy the above properties in all executions complying to the system and failure model, provided that (a) the implementations of correct basic modules specified in Section 3.3 indeed fulfill their specifications, and (b) the additional "global" Constraints 1–3 hold. Our Theorems 2 and 3 will also establish numerical values for precision and accuracy, which only depend on X and the delay parameters introduced in the specifications of the TG-Alg sub-modules in Section 3.3.

4 Correctness Proofs

The first cornerstone of our proofs[12] is the "Interlocking Lemma", which states that an "old" tick $k-2, k-4, \ldots$ is never mixed up with a "recent" tick k when generating tick $k+1$. This property does not come for free, however, but can be guaranteed to hold only if the following timing constraint is satisfied (which is easy to enforce via a suitably defined constraint during place-and-route of a VLSI circuit):

Constraint 1 (Interlocking Constraint). *With the abbreviations* $T_{max} := \tau^+_{TH} + \max(\tau^+_{GR}, \tau^+_{GEQ}) + \tau^+_{loc}$, $T_{min} := \tau^-_{TH} + \min(\tau^-_{GR}, \tau^-_{GEQ}) + \tau^-_{loc} + \tau^-_{Diff}$ *and* $T_{min,dis} := \tau^-_{TH} + \min(\tau^-_{GR}, \tau^-_{GEQ}) + \tau^-_{loc}$, *it must hold that* $T_{max} \leq T_{min} + T_{min,dis}$.

Lemma 1 (Interlocking). *If, for some correct node* p *and* $k' = k+1 \geq 2$, $b_p(t) = k+1$, *then*

(i) either there exists a set Q *of size* $|Q| \geq 2f+1$ *s.t. for* $t' := t - \tau^-_{TH} - \tau^-_{GEQ}$:

$$k \in \mathbb{N}_{even} \Rightarrow \forall q \in Q : \exists t_q \leq t' : \widetilde{\mathcal{P}}^{GEQ,e}_{p,q}(t_q) \wedge r^{loc}_{p,q}(t_q) \geq k$$
$$k \in \mathbb{N}_{odd} \Rightarrow \forall q \in Q : \exists t_q \leq t' : \widetilde{\mathcal{P}}^{GEQ,o}_{p,q}(t_q) \wedge r^{loc}_{p,q}(t_q) \geq k,$$

[12] Lacking space does not allow us to present the complete correctness proof and performance analysis here, which can be found in Section D in the appendix and in the technical report [14].

(ii) or there exists a set Q of size $|Q| \geq f+1$ s.t. for $t' := t - \tau^-_{TH} - \tau^-_{GR}$:

$$k \in \mathbb{N}_{even} \Rightarrow \forall q \in Q : \exists t_q \leq t' : \widetilde{\mathcal{P}}^{GR,e}_{p,q}(t_q) \wedge r^{loc}_{p,q}(t_q) \geq k$$
$$k \in \mathbb{N}_{odd} \Rightarrow \forall q \in Q : \exists t_q \leq t' : \widetilde{\mathcal{P}}^{GR,o}_{p,q}(t_q) \wedge r^{loc}_{p,q}(t_q) \geq k.$$

This result enables us to prove a minimum duration between two successive ticks generated by a correct node:

Lemma 2. *If correct node p sends tick $k \geq 1$ at time $t_{p,k}$, it cannot send tick $k+1$ before $t_{p,k} + T_{min}$.*

The following Lemma 3 in conjunction with an additional Constraint 2 allows us to exclude the possibility of queuing effects in a pipepair $(p,q)_{GR}$ corresponding to correct node q in a correct TG-Alg p. [An analogous Lemma 10 exists for $(p,q)_{GEQ}$.]

Constraint 2 $\tau^+_{Diff} \leq T_{min}$.

Lemma 3. *For any pair of distinct correct nodes p, q and $k \geq 1$: If correct node p sent tick k at $t_{p,k}$ and q sent tick k at $t_{q,k}$, then tick $k-1$ is removed from the local and remote pipe of pipepair $(p,q)_{GR}$ by time $\max\{t_{k,p} + \tau^+_{loc}, t_{k,q} + \tau^+_{rem}\} + \tau^+_{Diff}$, if Constraint 2 holds.*

Proof. The proof is by induction on the number of ticks $k \geq 1$ that are sent by p and q.
Begin ($k=1$): Assume that node p (resp. q) sends tick 1 at $t_{p,1}$ (resp. $t_{q,1}$). Tick 1 will hence arrive at p's local pipe by $t_{p,1} + \tau^+_{loc}$ (resp. at p's remote pipe for q by $t_{q,1} + \tau^+_{rem}$). Since there is no tick that could block the removing of tick 0, the latter is removed by $\max\{t_{p,k} + \tau^+_{loc}, t_{q,k} + \tau^+_{rem}\} + \tau^+_{Diff}$ from both the local and remote pipe because of the Diff-Gate properties.

Step ($k > 1$): For the induction hypothesis, assume that tick $k-2$ is removed from both pipes by $\max\{t_{p,k-1} + \tau^+_{loc}, t_{q,k-1} + \tau^+_{rem}\} + \tau^+_{Diff}$. By analogous arguments as above, tick k arrives in both pipes by t_{both}, with $t_{both} := \max\{t_{p,k} + \tau^+_{loc}, t_{q,k} + \tau^+_{rem}\}$. Because of Lemma 2, consecutive ticks cannot be generated with less than T_{min} distance in-between, i.e., $t_{p,k} - t_{p,k-1} \geq T_{min}$ and $t_{q,k} - t_{q,k-1} \geq T_{min}$. Thus $t_{both} \geq \max\{t_{p,k-1} + \tau^+_{loc}, t_{q,k-1} + \tau^+_{rem}\} + T_{min} \geq \max\{t_{p,k-1} + \tau^+_{loc}, t_{q,k-1} + \tau^+_{rem}\} + \tau^+_{Diff}$ by Constraint 2. By the induction hypothesis, tick $k-2$ is removed by t_{both}. Thus, tick $k-1$ is removed by $t_{both} + \tau^+_{Diff}$. □

The following pivotal Theorem 1 and its proof are a generalization of well-known properties of consistent broadcasting, cp. [10, 11, 30]. To hold true, the following additional timing constraint must be satisfied:

Constraint 3 $T^-_{first} \geq (X+1)(\tau^+_{loc} + \max\{\tau^+_{Diff} + \tau^+_{GR}, \tau^+_{GEQ}\} + \tau^+_{TH})$.

Theorem 1. (Synchronization Properties). *pDARTS satisfies the synchronization properties Unforgeability (U), Progress (P), Quasi-Simultaneity (QS) and Booting-Simultaneity (BS), if Constraints 1, 2, 3 and $n \geq 3f+2$ hold.*

(U) *Unforgeability. If no correct node sends tick $k \geq 1$ by time t, then no correct node sends tick $k+X+1$ by time $t + T^-_{first}$ or earlier, with $T^-_{first} := \tau^-_{rem} + \tau^-_{Diff} + \tau^-_{GEQ} + \tau^-_{TH}$.*

(P) *Progress. If all correct nodes send tick $k \geq X+1$ by time t, then every correct node sends at least tick $k+1$ by time $t + T_P$, with*

$$T_P := \max \left\{ \begin{array}{l} \tau^+_{loc} + \tau^+_{Diff} + \max\{\tau^+_{GEQ}, \tau^+_{GR}\}, \\ \tau^+_{loc} + \tau^+_{GR}, \\ \tau^+_{rem} + \tau^+_{Diff} + \tau^+_{GEQ} - XT_{min} \end{array} \right\} + \tau^+_{TH} \tag{4}$$

(QS) *Quasi-Simultaneity. If some correct node p sends tick $k \geq X+2$ by time t, then every correct node (including p) sends at least tick $k-X-1$ by time $t + T_{QS}$, with*

$$T_{QS} := \max \left\{ \begin{array}{l} \tau^+_{rem} + \tau^+_{GR} + \tau^+_{TH} + X(\tau^+_{loc} + \max\{\tau^+_{Diff} + \tau^+_{GR}, \tau^+_{GEQ}\} + \tau^+_{TH} - T_{min}) - T^-_{first}, \\ B + (X+1)(\tau^+_{loc} + \max\{\tau^+_{GEQ}, \tau^+_{GR}\} + \tau^+_{TH} - T_{min}) - (\tau^+_{loc} - \tau^-_{loc}) - T^-_{first} \end{array} \right\}.$$

(BS) *Booting-Simultaneity. If some correct node sends tick $k \geq X+1$ by time t, then every correct node sends at least tick k by time $t + T_{BS}(k)$, with $T_{BS}(k) := B + \max\{\tau^+_{GEQ}, \tau^+_{GR}\} - \min\{\tau^-_{GEQ}, \tau^-_{GR}\} + \tau^+_{TH} - \tau^-_{TH} + (k-1)(T_P - T_{min})$.*

Theorem 1 eventually leads to our major results:

Theorem 2. (Precision). *The pDARTS algorithm ensures $\forall t : |b_p(t) - b_q(t)| \leq \pi$ for all correct p, q, with precision*

$$\pi := \max\left\{ \left\lfloor \frac{T_{QS}}{T^-_{first}} \right\rfloor (X+1) + \left\lceil \frac{T_{QS} - \left\lfloor \frac{T_{QS}}{T^-_{first}} \right\rfloor T^-_{first}}{T_{min}} \right\rceil + X + 1, \left\lceil \frac{T_{QS}}{T^-_{first}} \right\rceil (X+1) + \left\lceil \frac{T_{QS} - \left(\left\lceil \frac{T_{QS}}{T^-_{first}} \right\rceil - 1 \right) T^-_{first}}{T_{min}} \right\rceil \right\}$$

Note that π depends on the *ratio* of certain timing parameters only, which typically does not change much when, e.g., one migrates the algorithm to a faster VLSI technology.

The following Theorem 3 allows to relate clock time intervals to real-time intervals, and to make statements about the local clock frequency. For example, it reveals that the long-term frequency is within $\left[1/T_P, (X+1)/T^-_{first}\right]$.

Theorem 3. (Accuracy). *Given t_1 and t_2 with $t_2 > t_1 \geq t_{p,X+1}$, the accuracy $b_p(t_2) - b_p(t_1)$ of any correct node p is bounded by*

$$\max\left\{0, \left\lfloor \frac{t_2 - t_1 - \max\left\{ \begin{array}{l} T_{BS}(2X+1), \\ \min\{T_{QS} + (X+1)T_P, T_{BS}(k) \mid k \geq 2X+2\} \end{array} \right\}}{T_P} \right\rfloor + 1 \right\} \tag{5}$$

$$\leq b_p(t_2) - b_p(t_1) \leq \left\lceil \frac{t_2 - t_1}{T^-_{first}} \right\rceil (X+1) - \left\lceil \frac{t_2 - t_1 - \left\lceil \frac{t_2 - t_1}{T^-_{first}} \right\rceil T^-_{first}}{T_{min}} \right\rceil + 1 + \pi$$

Note the term $\min\{T_{QS} + (X+1)T_P, T_{BS}(k)\}$ for $k \geq 2X+2$ in (5), which accounts for the fact that correct nodes may be synchronized very tightly (within $T_{BS}(k)$) after booting, such that $T_{QS} + (X+1)T_P$ would be too conservative. However, when $T_{min} < T_P$, which is typically the case in real systems, the initial synchrony from booting cannot be maintained, since $\forall k \geq 1 : T_{BS}(k+1) > T_{BS}(k)$ as well as $\lim_{k\to\infty} T_{BS}(k) = \infty$ according to (BS) in Theorem 1. Thus, for some k_0, $\forall k \geq k_0 : T_{QS} + (X+1)T_P < T_{BS}(k)$, i.e., the constant bound from (QS) will be tighter.

Finally, we proved that the pipeline sizes $S_{rem,geq}, S_{rem,gr}, S_{loc,geq}$ and $S_{loc,gr}$ at correct TG-Algs are indeed bounded by some constants, which again depend on the ratio of certain timing parameters only.[13]

5 Prototype Implementation and Measurement Results

Since pDARTS uses the same basic blocks as the original DARTS VHDL implementation [5, 12], it was reasonably easy to build an FPGA prototype implementation of pDARTS: Recall that the only substantial change was the doubling of the $+/-$ counters, and the need to initialize all GEQ-$+/-$ counters to X; the latter was accomplished by putting ticks $-X, \ldots, 0$ into their remote pipes upon reset.

Similar to the first prototype of DARTS, we have synthesized a complete system of $n = 5$ pDARTS TG-Algs on an Altera APEX EP20K1000 FPGA. Although FPGAs are not particularly suitable for asynchronous designs due to the fixed structure of the lookup tables and registers, this prototype nevertheless provides a proof of concept and clearly demonstrates the feasibility and efficiency of the pipelined approach.

5.1 Parameter choices

Recall from Section 1 and 2 that the pipelining parameter X is related to the inherent pipeline depth of the end-to-end delay paths in the system. Formally, this is expressed in Constraint 3, which requires $T^-_{first} \geq (X+1)(\tau^+_{loc} +$

[13] Please consult the technical report [14] for the proofs.

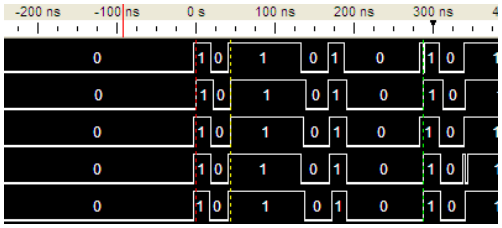

Fig. 5. pDARTS clocks for $X = 2$ (immediately after reset)

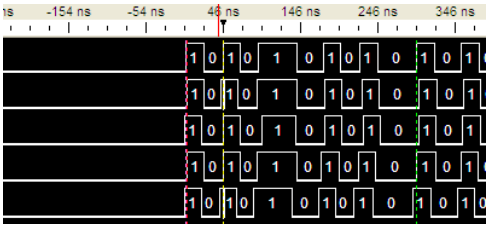

Fig. 6. pDARTS clocks for $X = 4$ (immediately after reset)

$\max\{\tau^+_{Diff} + \tau^+_{GR}, \tau^+_{GEQ}\} + \tau^+_{TH})$. It effectively limits X to a value that allows it to "store" at least X ticks within the fastest end-to-end delay path, even if the X ticks are generated as slowly as possible.

On the other hand, from Theorem 3, it follows that the clock frequency lower bound is $\Omega(1/T_P)$, with T_P given by (4) in Theorem 1. From the term $\tau^+_{rem} + \tau^+_{Diff} + \tau^+_{GEQ} - XT_{min}$ appearing in T_P, it is obvious that, by choosing X sufficiently large, the dependency on a large remote delay bound τ^+_{rem} can be dropped entirely.

From Theorem 2, it is apparent that the worst case precision π increases when increasing X. It is important to note, however, that this is a matter of a "scaling transformation" and *not* a sign of reduced real-time synchronization quality: Since the precision gives the maximum number of *ticks* two clocks can be off at the same real-time, the decrease of the number π is outweight by the increase of the clock frequency (and hence the tick duration). What indeed increases when increasing X is the maximum size of the elastic pipelines, however: In order to be able to experiment with different values of $X \in \{0, 2, 4\}$, we had to choose the conservative size of 8 for every elastic pipeline.

5.2 Measurement results

The above choice of parameters in our FPGA implementation resulted in a local-loop delay (T_{min}) of about 25 ns, which amounts to a maximum local clock frequency of about 20 MHz.[14] In order to be able to demonstrate the benefits of pipelining in this setting, we enforced a remote delay of $\tau^-_{rem} = \tau^+_{rem} \approx 100\ ns$ in the TG-Net. The inherent pipeline depth is hence not smaller than $100/25 = 4$. As revealed by the measurement results for the case $X = 0$, the original DARTS would achieve a clock frequency of less than 5 MHz in this case.

Fig. 5 shows the case $X = 2$. The repetitive pattern consisting of a burst of 2 ticks followed by some idle time in every clock signal is an artefact of our simple initialization approach, in conjunction with an inherent pipeline depth larger than 2: On reset, the TG-Alg generates $X + 1$ transitions and waits until it receives the first transition from sufficiently many remote TG-Alg to generate the next transition.

Fig. 6 finally shows the case $X = 4$. One can observe that the clocks run at a frequency of about 20 MHz, which is the maximum frequency of the local loop. This speed-up by a factor of 4 confirms that pipelining is indeed able to (completely) hide the large remote delay in the system. Interestingly the initially somewhat bursty clock cycles tend to spread out evenly after some time, cp. our comment on (5) after Theorem 3.

6 Related Work

Pipelining is a well-known approach for speeding up *synchronous* distributed algorithms, see e.g. [31, Ch. 6.2.2]. For example, it is possible to speed-up a k-round rotating coordinator-based algorithm (for $k > 1$) like the reliable broadcasting algorithm of [32] by starting coordinator p_i in round $i \mod n$, irrespectively of the fact that coordinator p_i completes by round $i+k$ only. Pipelining is also heavily used in distributed algorithms for "global network problems" in arbitrary network topologies, like the construction of a minimum-weight spanning tree and broadcasting [33, 34]. These algorithms exploit the fact that the processes along a path in the network naturally form a pipeline. On the other hand, in [35], it is shown that not all problems can be sped-up by means of pipelining.

Pipelining is also a well-known technique for reliable data transmission over FIFO channels in asynchronous systems with high bandwidth$\times$delay products: Sliding window protocols allow multiple messages to be sent over the link before the acknowledgment for the first of these messages has been received, see e.g. [36]. Since such protocols are point-to-point and hence involve two processors only, they are not representative for general distributed algorithms.

[14] Our FPGA DARTS prototype achieved about 24 MHz, which is due to the fact that the sizes of the elastic pipelines were 4 instead of 8.

Still, for general asynchronous distributed algorithms, pipelining is often used implicitly: In case of round-based algorithms, e.g., the round number of a slow processor may arbitrarily lag behind the round number of a fast processor due to messages still in transit. However, we are not aware of any work that explicitly exploits the inherently pipelined architecture of modern distributed systems for speeding-up asynchronous fault-tolerant distributed algorithms.

Modeling approaches: Asynchronous distributed systems theory—in the absence of failures—has been used in the VLSI community for decades [37]: Research on transition signaling [38, 39], delay-insensitivity [40, 41], micropipelines [29], etc. has established a sound basis for dealing with self-timed systems [42]. Since then, much research has been conducted on benefits and limitations of asynchronous circuits. There is a wealth of literature on the arbiter problem [26, 43], which is—like the latch, the inertial delay and the mutex—impossible to solve in a delay-insensitive way [40, 44]. Both arbiter-free problems [45] and a few ways to circumventing the impossibility of implementing arbiters by adding (some) timing properties [46] or order properties [47] have been thoroughly investigated. Existing modeling approaches for asynchronous circuits are based on algebraic trace theory [41, 48] or Petri-nets [45, 49]. Note that such specifications are time-free. Time-augmented asynchronous circuits can be handled by using timed Petri-nets, which assign time intervals to each transition [50–53]. However, to the best of our knowledge, none of these modeling frameworks can deal with failures. On the other hand, timed system models from distributed algorithms research, e.g., [54], can deal with failures, but are not tailored to the specific requirements of VLSI circuits, thus leading to complex specifications.

VLSI clock generation: Since one of the advantages of DARTS is to avoid external clock souces, we do not consider the sizeable body of work on hardware-assisted fault-tolerant clock synchronization (see [55] for an overview) here. The few approaches for distributed clock generation without external clock sources we are aware of are essentially based on a (distributed) ring oscillator, which is formed by gates arranged in a feedback loop. Instead of being dictated by a quartz, the frequency of the generated clock signal is determined by the end-to-end delay of the feedback loop. In [56], a regular structure of closed loops of an odd number of inverters is used for distributed clock generation. Similarly, [57, 58] employs local tick generation cells, arranged in a two-dimensional grid, with each cell inverting its output signal when its four inputs (from the up, down, left and right neighbor) match the current clock output value. Since clock synchronization theory [28] reveals that high connectivity is required for bounded synchronization tightness in presence of failures, however, the sparsely connected designs proposed in [56–58] are not fault-tolerant.

7 Conclusions and Future Work

We demonstrated that pipelining is effective for increasing the clock frequency of the DARTS fault-tolerant distributed clocking approach in VLSI circuits with large bandwidth$\times$delay products. We provided a pipelined version of the original DARTS algorithm, along with some cornerstones of our correctness proof and performance analysis, which is based on a new modeling framework for low-level distributed algorithms. The validity of our results was confirmed by measurement results obtained in an FPGA prototype system.

There are several directions of future work connected with this paper. Besides the further refinement of our modeling approach, we are interested in exploring generic ways for the specification and analysis of pipelined versions of other asynchronous fault-tolerant distributed algorithms.

Appendix

A Introduction to Pipelining

In general, pipelining is applicable in the case of streamed data processing, where a sequence of individual data items is to be processed by a sequence of $x > 1$ actions $a_1, \ldots, a_x$, applied to every data item. Consider a chain of processors $p_1, \ldots, p_x$ connected via storage elements (buffers), which provide the output of p_{i-1} as an input to p_i. Instead of executing all actions by a single processor sequentially, every action a_i is performed by processor p_i here. Consequently, assuming a stream of data items to be processed, every single data item flows through the chain of processors similar as gas or water flows through a pipeline. Flow control can be implemented synchronously or asynchronously here; for example, p_i might just wait until (i) there is a new data item in its input buffer, and (ii) depositing its previous result in its output buffer has completed.

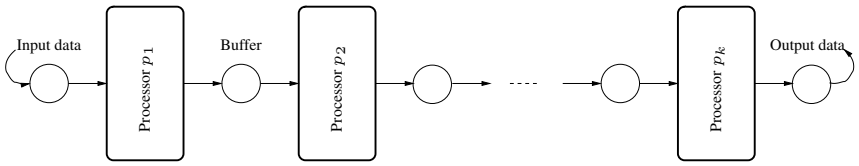

Fig. 7. Pipeline of processors

Assuming that every action takes α seconds of processing time on its processor, the pipeline can process $1/\alpha$ data items per second (ignoring possible buffering overheads).[15] This is a speed-up of a factor x over the at most $1/(x\alpha)$ data items per second that could be digested by a single processor executing all x actions sequentially.

B Hardware Implementation Challenges

In [5], it turned out that several challenging issues must be solved to make a simple algorithm like pDARTS in Fig. 1 suitable for direct implementation in asynchronous digital logic:

How to adapt the original algorithm to circumvent unbounded tick(ℓ), $\ell \in \mathbb{N}$? Clearly, a VLSI distributed algorithm that broadcasts unbounded integers is infeasible. We hence modified the algorithm to only broadcast event-messages (i.e., up/down signal transitions), and maintain the number ℓ_q of event-messages seen from TG-Alg q so far at the receiver. This strategy evidently avoids unbounded message size, albeit not unbounded size of the variable ℓ_q. Fortunately, the inspection of the catchup rule (line 4) and the progress rule (line 7) in Fig. 1 reveals that they only check, for sufficiently many ($f + 1$ and $2f + 1$) q, whether $\ell_q > k$ and $\ell_q \geq k - X$, respectively. We could hence replace the first inequality with $\ell_q - k > 0$, and the second with $\ell_q - k \geq -X$. Consequently, the only state information that actually needs to be maintained by TG-Alg p is the difference of the number of ticks received from q and the number of ticks generated by p locally. This is good news, since pDARTS shall provide synchronized clocks, which means that $\ell_q - k$ should remain bounded (at least for correct p, q). Note that we actually maintain individual copies k_q of k for every receiver q at TG-Alg p, which are all incremented whenever p generates a new tick locally. Since $\ell_q - k_q = \ell_q - c_q - (k_q - c_q)$ for any $0 < c_q \leq \min\{\ell_q, k_q\}$, this trick allows the algorithm to simultaneously decrease both ℓ_q and k_q by c_q, at any time, without changing the difference.

How to ensure atomicity of actions in an asynchronous VLSI implementation? Fault-tolerant distributed computing models assume atomic computing steps at the level of a single node. This abstraction does not apply when the algorithm is implemented via asynchronous digital logic gates, which concurrently and continuously compute their outputs based on their inputs. For example, it is not possible to increment all the copies k_q of k mentioned above simultaneously when a new tick is generated. Thus, explicit synchronization, i.e., serialization of actions via handshaking, or

[15] In fact, this example oversimplifies several aspects, which have to be handled in the detailed analysis: In particular, it does not consider varying stage delays and queueing effects resulting from these variations (where queues may not only overrun, but also influence the stage delays since they depend on the queue's fill level). Note that these effects mandate buffers that can hold more than a single data item between the processors. Moreover, every distributed algorithm involves multiple end-to-end paths, which are usually all different.

implicit synchronization via timing constraints must be employed if two local computations must not interfere with each other (namely, line 4 and line 7 for the same tick(k), cf. "[once]" in Fig. 1).

Fortunately, since the same issues had to be solved for the original DARTS algorithm, the solutions developed there could be reused for pDARTS. Fig. 2 depicts the resulting basic hardware architecture of a pDARTS TG-Alg. Again, this architecture is deceptively simple. The major problem encountered when trying to implement Fig. 2 is the lack of atomicity in any asynchronous digital circuit, which raises another intricate problem:

How to disambiguate GR, GEQ for different ticks? For example, consider the $+/-$ counter for the signal GR at TG-Alg p, corresponding to some remote TG-Alg q, and suppose that p is about to generate tick $k+1$, i.e., $k_q = k$. Assume further that $\ell_q - k_q = 1$, i.e., $\ell_q - k_q > 0$, such that $GR = 1$. This status information is fed into the threshold circuit, which tests whether there are at least f other $+/-$ counters that have set their GR signals as well. If the case, p will eventually generate tick $k+1$ and, hence, increment every k_q. Due to the lack of atomicity, however, the latter cannot be done simultaneously. Consequently, the threshold circuit cannot distinguish the case (i) where still $GR = 1$ since k_q has not been incremented yet, from the case (ii) where $GR = 1$ although k_q has already been incremented, but only *after* a new remote tick has arrived (such that ℓ_q had also been incremented). To circumvent such ambiguities, we split the signal GR into two signals GR^e and GR^o, tied to even and odd ticks, respectively: GR^e is true iff the inequality $\ell_q - k_q > 0$ holds and $k_q \in \mathbb{N}_{even} := 2\mathbb{N}$, whereas GR^o is true iff the inequality $\ell_q - k_q > 0$ holds and $k_q \in \mathbb{N}_{odd} := 2\mathbb{N} + 1$. The associated threshold circuit is also duplicated, providing one for the GR^e signals and one for the GR^o signals. The same splitting is done for the signals GEQ and their threshold circuits. Finally, all their outputs are combined in some straightforward way to finally generate p's odd and even ticks.

Whereas this duplication allows the algorithm to disambiguate between successive ticks, the problem of potentially mixing up GR^e signals based on, say, $k_q = 2$ and $k_{q'} = 4$ still remains. This problem will be solved implicitly, by means of timing constraints: We will formally prove a pivotal *interlocking property* in Lemma 1, which ensures that GR^e, as well as the other status signals, are based on identical k_qs, by proving that a correct TG-Alg p has set all k_q's to, say, 3 before it sets the first k_q to 4.

How to implement the $+/-$ counters? This task turned out to be the most delicate part of the hardware design work. Actually, implementing an asynchronous up/down counter is inherently difficult due to the fact that the up-clock and the down-clock transitions are totally unrelated. They can hence occur arbitrarily close to each other, which causes metastability problems. Another problem is how to correctly generate the status signals GR^o and GEQ^o (resp. GR^e and GEQ^e). They should truly reflect the current counter value, at least during times when they are used. We will specify the detailed properties that must be maintained by those signals in Section 3. Note that our correctness proof and performance analysis will only rely on these properties, i.e., will hold for every implementation of the local $+/-$ counters that fulfills these properties.

Our metastability-free implementation of the $+/-$ counters consists of two elastic pipelines [29], which can be seen as shift registers/FIFO buffer for signal transitions. One is attached to the remote clock signal, the other one is fed by the local clock signal. They are fitted together at their ends via a special *Diff-Gate*, which removes "matching" transitions as soon as they traveled through the pipelines. The status signals GR^o, GEQ^o, GR^e and GEQ^e are derived from monitoring the last few stages of both pipes.

C Complete formal specification of a TG-Alg

In this section, we will describe the complete internal architecture of a TG-Alg, i.e., its sub-modules and interconnect, and formally specify the behavior of the modules, which have been omitted in Section 3.3. We recall that the behavioral properties defined here are *assumed* properties, i.e., properties that must *a priori* be guaranteed by the implementation of the basic modules. [Modules in FCRs hit by a failure may deviate (arbitrarily) from their correct behavior, however.] Based on these basic properties, the correctness proofs sketched in Section 4 and provided in full detail in Section D will show that every TG-Alg belonging to a correct FCR will maintain the system-level properties (precision and accuracy) as specified in Section 3.4.

Fig. 8 shows the general architecture of the TG-Alg executed at node p, cp. also Fig. 2. It consists of two $+/-$ counter modules per remote node, four threshold modules implementing the $f+1$ and $2f+1$ rules in Fig. 2, and a tick broadcast module that finally generates p's clock ticks $b_p(t)$.

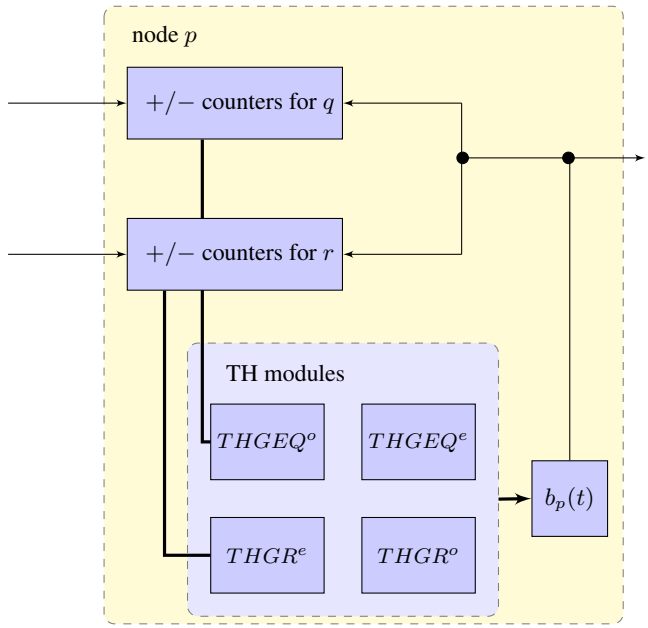

Fig. 8. Architecture of node p

The GEQ $+/-$ counter with its basic modules, namely, the $(p,q)_{GEQ}$ pair of elastic pipelines, the Diff-Gate and the PCSG, has already been specified in Section 3.3. We will now specify the behaviors of the remaining modules.

GR $+/-$ counter: The behavior of $(p,q)_{GR}$ and its constituent basic modules is the same as of the pipepair $(p,q)_{GEQ}$, except that ticks are removed as follows:

- For $k=-1$ and $k=0$, the arrival time $t^{GR}_{rem,k}$ for remote virtual tick k in $(p,q)_{GR}$ is defined as $t^{GR}_{rem,k} := t_{p,boot}$.
- For $k=-1$, the virtual tick -1 is removed at booting time, i.e., at $t_{p,boot}$.
- For all $k \geq 0$, if
  - tick $k+1$ shows up at the output $r^{rem,GR}_{p,q}(t^{GR}_{rem,k+1})$ of the remote pipe of $(p,q)_{GR}$ at time $t^{GR}_{rem,k+1}$, and
  - tick $k+1$ shows up at the output $r^{loc,GR}_{p,q}(t^{GR}_{loc,k+1})$ of the local pipe of $(p,q)_{GR}$ at time $t^{GR}_{loc,k+1}$, and
  - tick $k-1$ is removed at time $t^{GR}_{rmv,k-1}$,

  then tick k is removed at $t^{GR}_{rmv,k}$, with

$$t_{rmv,k} \in \max\{t^{GR}_{rem,k+1}, t^{GR}_{loc,k+1}, t^{GR}_{rmv,k-1}\} + \left[\tau^-_{Diff}, \tau^+_{Diff}\right]. \tag{6}$$

On top of the above defined signals $r^{loc}_{p,q}(t)$ and $d^{GR}_{p,q}(t)$, we define the size of the local pipe of the pipepair $(p,q)_{GR}$ at time t as

$$s^{loc,gr}_{p,q}(t) := r^{loc}_{p,q}(t) - d^{GR}_{p,q}(t).$$

Connection from PCSGs to Threshold modules: These four signals are fed into four dedicated channels, all of which are initialized to 0.

- Channel $\langle \mathcal{P}^{GEQ} toGEQ, o, p, q\rangle$ with input $\widetilde{\mathcal{P}}^{GEQ,o}_{p,q}(t)$, output $\widetilde{GEQ}^o_{p,q}(t)$ and delay $\left[\tau^-_{GEQ}, \tau^+_{GEQ}\right]$
- Channel $\langle \mathcal{P}^{GEQ} toGEQ, e, p, q\rangle$ with input $\widetilde{\mathcal{P}}^{GEQ,e}_{p,q}(t)$, output $\widetilde{GEQ}^e_{p,q}(t)$ and delay $\left[\tau^-_{GEQ}, \tau^+_{GEQ}\right]$
- Channel $\langle \mathcal{P}^{GR} toGR, o, p, q\rangle$ with input $\widetilde{\mathcal{P}}^{GR,o}_{p,q}(t)$, output $\widetilde{GR}^o_{p,q}(t)$ and delay $\left[\tau^-_{GR}, \tau^+_{GR}\right]$
- Channel $\langle \mathcal{P}^{GR} toGR, e, p, q\rangle$ with input $\widetilde{\mathcal{P}}^{GR,e}_{p,q}(t)$, output $\widetilde{GR}^e_{p,q}(t)$ and delay $\left[\tau^-_{GR}, \tau^+_{GR}\right]$

Threshold modules: Let $Q := P \setminus p$ be the set of remote TG-Algs for p. The signals provided via the PCSG-channels associated with $(p,q)_{GEQ}$ and $(p,q)_{GR}$, for all $q \in Q$, are fed into four dedicated threshold modules in the TG-Alg p. If the number of active $\widetilde{GEQ}^{o/e}_{p,q}(t)$ resp. $\widetilde{GR}^{o/e}_{p,q}(t)$ signals exceeds the $2f+1$ resp. $f+1$ threshold, the corresponding threshold signal $\widetilde{THGEQ}^{o/e}_{p}(t)$ resp. $\widetilde{THGR}^{o/e}_{p}(t)$ becomes active within $\left[\tau^-_{TH}, \tau^+_{TH}\right]$. This property will be formalized as a boolean predicate involving the sum of the counting functions of a threshold module's input ports that is fed into a channel.

Behavioral Description: Let $Q := P \setminus p$ be the set of remote nodes. The Thresholds modules are defined as four channels, all initialized to 0:

- Channel $\langle \sum GEQtoTHGEQ, o, p\rangle$ with input $\sum_{q\in Q} \widetilde{GEQ}^{o}_{p,q}(t) \geq 2f+1$, output $\widetilde{THGEQ}^{o}_{p}(t)$ and delay in $\left[\tau^-_{TH}, \tau^+_{TH}\right]$
- Channel $\langle \sum GEQtoTHGEQ, e, p\rangle$ with input $\sum_{q\in Q} \widetilde{GEQ}^{e}_{p,q}(t) \geq 2f+1$, output $\widetilde{THGEQ}^{e}_{p}(t)$ and delay in $\left[\tau^-_{TH}, \tau^+_{TH}\right]$
- Channel $\langle \sum GRtoTHGR, o, p\rangle$ with input $\sum_{q\in Q} \widetilde{GR}^{o}_{p,q}(t) \geq f+1$, output $\widetilde{THGR}^{o}_{p}(t)$ and delay in $\left[\tau^-_{TH}, \tau^+_{TH}\right]$
- Channel $\langle \sum GRtoTHGR, o, p\rangle$ with input $\sum_{q\in Q} \widetilde{GR}^{e}_{p,q}(t) \geq f+1$, output $\widetilde{THGR}^{e}_{p}(t)$ and delay in $\left[\tau^-_{TH}, \tau^+_{TH}\right]$

Note that a predicate can be a function too, so $\sum_{q\in Q} \widetilde{GEQ}^{o}_{p,q}(t) \geq 2f+1$ is a valid predicate.

Tick generation module: The TG-Alg p generates the next clock tick, at some time t, when (i) both threshold outputs for the previously generated tick, say, $\widetilde{THGEQ}^{o}_{p}(t)$ and $\widetilde{THGR}^{o}_{p}(t)$, are inactive again, and (ii) at least one threshold output $\widetilde{THGEQ}^{e}_{p}(t)$ or $\widetilde{THGR}^{e}_{p}(t)$ for the current tick becomes active. We will refer to (i) as the *disabling path* and to (ii) as the *enabling path* in the sequel. The Tick generation module hence has four input ports connected to the threshold outputs, and a single output port represented by the counting function $b_p(t)$, which is the number of ticks generated by p by time t. Finally, $b_p(t)$ is distributed to the local pipes (in $(p,q)_{GEQ}$ and $(p,q)_{GR}$) at TG-Alg p and to the remote pipes corresponding to p at all TG-Algs $q \in P \setminus \{p\}$ via dedicated channels $\langle LOC, p, q\rangle$ and $\langle REM, p, q\rangle$.

Behavioral Description: Let the signal B_p be defined by its event trace $\widehat{B}_p$ as the set for which

$$
\begin{aligned}
&(0,0) \in \widehat{B}_p,\\
&(\widetilde{THGEQ}^{o}_{p}(t) \vee \widetilde{THGR}^{o}_{p}(t)) \wedge \neg(\widetilde{THGEQ}^{e}_{p}(t) \vee \widetilde{THGR}^{e}_{p}(t)) \Leftrightarrow (t,0) \in \widehat{B}_p,\\
&(\widetilde{THGEQ}^{e}_{p}(t) \vee \widetilde{THGR}^{e}_{p}(t)) \wedge \neg(\widetilde{THGEQ}^{o}_{p}(t) \vee \widetilde{THGR}^{o}_{p}(t)) \Leftrightarrow (t,1) \in \widehat{B}_p.
\end{aligned}
$$

Then $b_p(t)$ is defined as the counting function of B_p, with initial value $b_p(0) = 0$.

Interconnect: The channels $\langle LOC, p, q\rangle$ and $\langle REM, p, q\rangle$ for distributing $b_p(t)$ are all initialized to 0 and adhere to the following specifications:

- $n-1$ local channels, one for each $q \in P \setminus \{p\}$: Channel $\langle LOC, p, q\rangle$ with input $b_p(t)$, output $r^{loc}_{p,q}(t)$ and delay in $\left[\tau^-_{loc}, \tau^+_{loc}\right]$.
- $n-1$ remote channels, one for each $q \in P \setminus \{p\}$: Channel $\langle REM, p, q\rangle$ with input $b_p(t)$, output $r^{rem}_{q,p}(t)$ and delay in $[\tau^-_{rem}, \tau^+_{rem}]$.

D Complete correctness proofs

For ease of notation we will sometimes need to express the left limit of a point in time. For this purpose, we define $t^{\rightarrow} := \lim_{\varepsilon>0,\varepsilon\rightarrow 0} t - \varepsilon$. For example, this notation allows to easily formalize $t_{p,k}$ as the time t for which $b_p(t^{\rightarrow})+1 = b_p(t) = k$ holds. We will start our formal treatment with a simple observation of channel properties.

Lemma 4. *If X is a channel with sending port X^s, receiving port X^r and latencies in $\left[\tau^-_X, \tau^+_X\right]$, then the following properties hold:*

(Ps) $t_1 \leq t_2 \Rightarrow X^s(t_1) \leq X^s(t_2)$

(Pr) $t_1 \leq t_2 \Rightarrow X^r(t_1) \leq X^r(t_2)$
(Pmax) $X^r(t + \tau_X^+) \geq X^s(t)$
(Pmin) $X^r(t + \tau_X^-) \leq X^s(t)$

Proof. (Ps) and (Pr) follow immediately from the definition of a counting function and the fact that both $\hat{X}^s$ and $\hat{X}^r$ are event traces. (Pmax) and (Pmin) follow from the fact that f bijectively maps the (non-idempotent) events occurring in $\hat{X}^s$ by time t to the (non-idempotent) events occurring in $\hat{X}^r$ by time $f(t)$, in conjunction with $f(t) \in t + \left[\tau_X^-, \tau_X^+\right]$ according to (1).

The following Lemmas 5 and 6 allow to infer bounds on the number of ticks received from a remote node when the size of the local pipe is 1. We start with $r_{p,q}^{rem,GEQ}(t)$, which is the number of remote ticks that traversed the remote pipe towards the Diff-Gate.

Lemma 5. *If for correct node $p \in C$ and a different correct node $q \in C \setminus \{p\}$, at time t it holds that*

$$(r_{p,q}^{loc}(t) = k) \wedge (s_{p,q}^{loc,geq}(t) = 1) \textit{ resp. } (r_{p,q}^{loc}(t) = k) \wedge (s_{p,q}^{loc,gr}(t) = 1)$$

for some $k \geq 1$, then it must hold that

$$r_{p,q}^{loc}(t - \tau_{Diff}^-) \geq k \textit{ and } r_{p,q}^{rem,GEQ}(t - \tau_{Diff}^-) \geq k \textit{ resp.}$$
$$r_{p,q}^{loc}(t - \tau_{Diff}^-) \geq k \textit{ and } r_{p,q}^{rem,GR}(t - \tau_{Diff}^-) \geq k.$$

Proof.

$$\begin{aligned}(r_{p,q}^{loc}(t) = k) \wedge (s_{p,q}^{loc,geq}(t) = 1) &\equiv (r_{p,q}^{loc}(t) = k) \wedge (r_{p,q}^{loc}(t) - d_{p,q}^{GEQ}(t) = 1) \\ &\Rightarrow d_{p,q}^{GEQ}(t) = k - 1 \end{aligned} \tag{7}$$

Let $t_{rmv,k-1}$ be the time tick $k-1$ is removed from the pipepair $(p,q)_{GEQ}$. From (7) it follows that

$$t_{rmv,k-1} \leq t. \tag{8}$$

Now assume by contradiction that

$$\begin{aligned} r_{p,q}^{loc}(t - \tau_{Diff}^-) < k \text{ or} \\ r_{p,q}^{rem,GEQ}(t - \tau_{Diff}^-) < k. \end{aligned} \tag{9}$$

Denoting with $t_{loc,k}$, resp., $t_{rem,k}$ the time tick k arrives at the local port, resp., remote port of pipepair $(p,q)_{GEQ}$, it follows that

$$t_{loc,k} > t - \tau_{Diff}^- \text{ resp.} \tag{10}$$
$$t_{rem,k} > t - \tau_{Diff}^-. \tag{11}$$

Combination of (8) with (10), resp., (11) yields

$$t_{rmv,k-1} < t_{loc,k} + \tau_{Diff}^- \text{ resp.} \tag{12}$$
$$t_{rmv,k-1} < t_{rem,k} + \tau_{Diff}^-. \tag{13}$$

From the behavioral specification of the diff-gate we know, however, that

$$\begin{aligned} t_{rmv,k-1} \geq t_{loc,k} + \tau_{Diff}^- \text{ and} \\ t_{rmv,k-1} \geq t_{rem,k} + \tau_{Diff}^-, \end{aligned}$$

contradicting (12) and (13). The proof for GR is analogous.

Recalling $r^{rem}_{p,q}(t) = r^{rem,GEQ}_{p,q}(t) - X$ and $r^{rem}_{p,q}(t) = r^{rem,GR}_{p,q}(t)$, this finally yields

Lemma 6. *If for correct node $p \in C$ and a different correct node $q \in C \setminus \{p\}$, at time t it holds that*

$$(r^{loc}_{p,q}(t) = k) \wedge (s^{loc,geq}_{p,q}(t) = 1) \textit{ resp. } (r^{loc}_{p,q}(t) = k) \wedge (s^{loc,gr}_{p,q}(t) = 1)$$

for some $k \geq 1$, then it must hold that

$$\begin{aligned} &r^{loc}_{p,q}(t - \tau^-_{Diff}) \geq k \textit{ and } r^{rem}_{p,q}(t - \tau^-_{Diff}) \geq k - X, \textit{ resp.} \\ &r^{loc}_{p,q}(t - \tau^-_{Diff}) \geq k \textit{ and } r^{rem}_{p,q}(t - \tau^-_{Diff}) \geq k. \end{aligned}$$

Note that $r^{rem}_{p,q}(t - \tau^-_{Diff}) \geq k - X$ is actually void for $k \leq X$, since $\forall t \geq 0 : r^{rem}_{p,q}(t) \geq 0$.

We next establish a main result, the Interlocking Lemma, which states that an "old" tick $k-2, k-4, \ldots$ is never mixed up with a "new" tick k when generating tick $k+1$. This does not come for free but requires

Constraint 4 (Interlocking Constraint). *With the abbreviations*

$$\begin{aligned} T_{max} &:= \tau^+_{TH} + \max(\tau^+_{GR}, \tau^+_{GEQ}) + \tau^+_{loc} \\ T_{min} &:= \tau^-_{TH} + \min(\tau^-_{GR}, \tau^-_{GEQ}) + \tau^-_{loc} + \tau^-_{Diff} \\ T_{min,dis} &:= \tau^-_{TH} + \min(\tau^-_{GR}, \tau^-_{GEQ}) + \tau^-_{loc} \end{aligned} \tag{14}$$

$T_{max} \leq T_{min} + T_{min,dis}$ *must hold.*

Lemma 7 (Interlocking). *If, for some correct node p and $k' = k+1 \geq 2$, $b_p(t) = k+1$, then*

(i) either there exists a set Q of size $|Q| \geq 2f+1$ s.t. for $t' := t - \tau^-_{TH} - \tau^-_{GEQ}$:

$$\begin{aligned} k \in \mathbb{N}_{even} &\Rightarrow \forall q \in Q : \exists t_q \leq t' : \widetilde{\mathcal{P}}^{GEQ,e}_{p,q}(t_q) \wedge r^{loc}_{p,q}(t_q) \geq k \\ k \in \mathbb{N}_{odd} &\Rightarrow \forall q \in Q : \exists t_q \leq t' : \widetilde{\mathcal{P}}^{GEQ,o}_{p,q}(t_q) \wedge r^{loc}_{p,q}(t_q) \geq k \end{aligned}$$

(ii) or there exists a set Q of size $|Q| \geq f+1$ s.t. for $t' := t - \tau^-_{TH} - \tau^-_{GR}$:

$$\begin{aligned} k \in \mathbb{N}_{even} &\Rightarrow \forall q \in Q : \exists t_q \leq t' : \widetilde{\mathcal{P}}^{GR,e}_{p,q}(t_q) \wedge r^{loc}_{p,q}(t_q) \geq k \\ k \in \mathbb{N}_{odd} &\Rightarrow \forall q \in Q : \exists t_q \leq t' : \widetilde{\mathcal{P}}^{GR,o}_{p,q}(t_q) \wedge r^{loc}_{p,q}(t_q) \geq k \end{aligned}$$

Proof. The proof is by induction on $k' \geq 2$, the number of ticks sent by node p. The lemma is first shown for tick $k' = 2$. Then we assume that some tick $k' > 2$ is the first tick for which the lemma does not hold. By investigating the causes that lead to this tick, we observe contradictions to Constraint 4.

Begin ($k' = 2$): Assume p sends tick 2 at time $t_{p,k'}$. Then by the algorithm either (a) $\widetilde{THGEQ}^o_p(t_{p,k'})$ or (b) $\widetilde{THGR}^o_p(t_{p,k'})$ must have held. We consider both cases:

case (a): If $\widetilde{THGEQ}^o_p(t_{p,k'})$, then by the algorithm, there must be a set $Q \subseteq P \setminus \{p\}$, of size $|Q| \geq 2f+1$, s.t., for $t' := f^{-1}\left(\langle \sum GEQtoTHGEQ, o, p\rangle; t_{p,k'}\right)$

$$\forall q \in Q : \widetilde{GEQ}^o_{p,q}(t').$$

Again by the algorithm and with $t_q := f^{-1}\left(\langle \mathcal{P}^{GEQ} toGEQ, o, p, q\rangle; t'\right)$ defined for every $q \in Q$ we obtain

$$\forall q \in Q : \widetilde{\mathcal{P}}^{GEQ,o}_{p,q}(t_q)$$

and by this

$$\widetilde{\mathcal{P}}^{GEQ,o}_{p,q}(t_q) \equiv [r^{rem,GEQ}_{p,q}(t_q) \geq r^{loc}_{p,q}(t_q)] \wedge [r^{loc}_{p,q}(t_q) \in \mathbb{N}_{odd}] \wedge [s^{loc,geq}_{p,q}(t_q) = 1]. \tag{15}$$

Since $r^{loc}_{p,q}(t_q) \geq 0$ from reset on and $r^{loc}_{p,q}(t_q) \in \mathbb{N}_{odd}$,

$$r^{loc}_{p,q}(t_q) \geq 1 = k. \tag{16}$$

Finally, from the channel properties, we know that

$$\begin{aligned} t_{p,k'} - t_q &= t_{p,k'} - f^{-1}\left(\langle \mathcal{P}^{GEQ} toGEQ, o, p, q\rangle; f^{-1}\left(\left\langle \sum GEQtoTHGEQ, o, p\right\rangle; t_{p,k'}\right)\right) \\ &\geq \tau^-_{TH} + \tau^-_{GEQ}. \end{aligned}$$

case (b): If $\widetilde{THGR}^o_p(t_{p,k'})$, then by the algorithm, there must be a set $Q \subseteq P \setminus \{p\}$, of size $|Q| \geq f+1$, s.t., for $t' := f^{-1}\left(\langle \sum GRtoTHGR, o, p\rangle; t_{p,k'}\right)$

$$\forall q \in Q : \widetilde{GR}^o_{p,q}(t').$$

By the algorithm and with $t_q := f^{-1}\left(\langle \mathcal{P}^{GR} toGR, o, p, q\rangle; t'\right)$ defined for every $q \in Q$ this implies

$$\forall q \in Q : \widetilde{\mathcal{P}}^{GR,o}_{p,q}(t_q) \tag{17}$$

and by this

$$\widetilde{\mathcal{P}}^{GR,o}_{p,q}(t_q) \equiv [r^{rem,GR}_{p,q}(t_q) > r^{loc}_{p,q}(t_q)] \wedge [r^{loc}_{p,q}(t_q) \in \mathbb{N}_{odd}] \wedge [s^{loc,gr}_{p,q}(t_q) = 1]. \tag{18}$$

Since $r^{loc}_{p,q}(t_q) \geq 0$ from reset on and $r^{loc}_{p,q}(t_q) \in \mathbb{N}_{odd}$,

$$r^{loc}_{p,q}(t_q) \geq 1 = k. \tag{19}$$

From the channel properties, we know that

$$\begin{aligned} t_{p,k'} - t_q &= t_{p,k'} - f^{-1}\left(\langle \mathcal{P}^{GR} toGR, o, p, q\rangle; f^{-1}\left(\left\langle \sum GRtoTHGR, o, p\right\rangle; t_{p,k'}\right)\right) \\ &\geq \tau^-_{TH} + \tau^-_{GR}. \end{aligned}$$

Step ($k' \geq 3$): Assume by contradiction that k' is the first tick, for which the lemma does not hold. Let $t_{p,k'}$ be the time p sends tick k'. Assume, wlog., that $k \in \mathbb{N}_{odd}$[16]. We will establish two delay bounds, one on the enabling path and the other on the disabling path.

Enabling path: To send tick k' at time $t_{p,k'}$, by the algorithm, at least one of the two threshold signals must have been enabled, i.e., either (a) $\widetilde{THGEQ}^o_p(t_{p,k'})$ or (b) $\widetilde{THGR}^o_p(t_{p,k'})$ must have held. We consider both cases:

case (a): If $\widetilde{THGEQ}^o_p(t_{p,k'})$, then by the algorithm, there must be a set $Q \subseteq P \setminus \{p\}$, of size $|Q| \geq 2f+1$, s.t., for $t' := f^{-1}\left(\langle \sum GEQtoTHGEQ, o, p\rangle; t_{p,k'}\right)$

$$\forall q \in Q : \widetilde{GEQ}^o_{p,q}(t'). \tag{20}$$

Again by the algorithm and with $t_q := f^{-1}\left(\langle \mathcal{P}^{GEQ} toGEQ, o, p, q\rangle; t'\right)$ defined for every $q \in Q$ this implies

$$\forall q \in Q : \widetilde{\mathcal{P}}^{GEQ,o}_{p,q}(t_q) \tag{21}$$

and by this

$$\widetilde{\mathcal{P}}^{GEQ,o}_{p,q}(t_q) \equiv [r^{rem,GEQ}_{p,q}(t_q) \geq r^{loc}_{p,q}(t_q)] \wedge [r^{loc}_{p,q}(t_q) \in \mathbb{N}_{odd}] \wedge [s^{loc,geq}_{p,q}(t_q) = 1]. \tag{22}$$

[16] The proof for $k \in \mathbb{N}_{even}$ is analogous.

By the channel properties

$$\tau_{TH}^{-} + \tau_{GEQ}^{-} \leq t_{p,k'} - t_q \leq \tau_{TH}^{+} + \tau_{GEQ}^{+}. \tag{23}$$

Assuming $\forall q \in Q : r_{p,q}^{loc}(t_q) \geq k$ yields the desired result of the Lemma. Thus, we only have to investigate the negation in order to establish a contradiction:

$$\exists q \in Q : r_{p,q}^{loc}(t_q) < k.$$

Since, by (22), $r_{p,q}^{loc}(t_q) \in \mathbb{N}_{odd}$, we must have

$$\exists q \in Q : r_{p,q}^{loc}(t_q) \leq k - 2.$$

Thus tick $k-1$ must have been received at the local pipe of pipepair $(p,q)_{GEQ}$ at time $t_{rcv,k-1}$, with

$$t_{rcv,k-1} > t_q.$$

Combination with (23) yields

$$t_{p,k'} - t_{rcv,k-1} < \tau_{TH}^{+} + \tau_{GEQ}^{+}. \tag{24}$$

Let $t_{p,k-1}$ be the sending time of tick $k-1$. Clearly, by the local channel properties

$$\begin{aligned} &t_{p,k-1} \geq t_{rcv,k-1} - \tau_{loc}^{+} \quad \Rightarrow \\ &t_{p,k'} - t_{p,k-1} < \tau_{TH}^{+} + \tau_{GEQ}^{+} + \tau_{loc}^{+}. \end{aligned} \tag{25}$$

Disabling path: Let $t_{p,k}$ be the sending time of tick k. By the induction hypothesis, we know that the lemma holds for tick k, with $k-1 \in \mathbb{N}_{even}$ and hence $k-1 \geq 2$. According to our lemma, we have to distinguish two cases (i) and (ii):

case (a.i): Assume that implication (i) is valid, i.e., there exists a set Q' of size $|Q'| \geq 2f+1$, s.t., for

$$t_{q'} := f^{-1}\left(\langle \mathcal{P}^{GEQ} toGEQ, e, p, q' \rangle; f^{-1}\left(\left\langle \sum GEQtoTHGEQ, e, p \right\rangle; t_{p,k}\right)\right)$$

it holds that

$$\forall q' \in Q' : \widetilde{\mathcal{P}}_{p,q'}^{GEQ,e}(t_{q'}) \wedge r_{p,q'}^{loc}(t_{q'}) \geq k-1. \tag{26}$$

Thus tick $k-1$ clearly must have arrived locally at every pipepair $(p,q')_{GEQ}$ at time $t_{q',rcv,k-1}$, with

$$t_{q',rcv,k-1} \leq t_{q'} - \tau_{Diff}^{-}$$

by Lemma 5. Further by the properties of the local channels

$$t_{q',rcv,k-1} - t_{p,k-1} \geq \tau_{loc}^{-}.$$

Thus

$$t_{p,k} - t_{p,k-1} \geq \tau_{TH}^{-} + \tau_{GEQ}^{-} + \tau_{Diff}^{-} + \tau_{loc}^{-}. \tag{27}$$

From the algorithm, we further deduce that at time $t_{p,k'}$

$$\neg \widetilde{THGEQ}_p^e(t_{p,k'}) \tag{28}$$

must hold, i.e., the threshold signals which generated tick k must be inactive again. Thus by the algorithm it must hold that

$$\nexists Q'', |Q''| \geq 2f+1 : \forall q'' \in Q'' : \widetilde{\mathcal{P}}^{GEQ,e}_{p,q''}(t_{q''}). \tag{29}$$

with

$$t_{q''} := f^{-1}\left(\langle \mathcal{P}^{GEQ} toGEQ, e, p, q' \rangle; f^{-1}\left(\left\langle \sum GEQtoTHGEQ, e, p \right\rangle; t_{p,k'}\right)\right) \tag{30}$$

Let us choose $Q'' := Q'$. Because of the FIFO property of the channels and because of $t_{p,k} < t_{p,k'}$, we obtain

$$t_{q'} < t_{q''}.$$

Clearly there has to be at least one $q' \in Q'$ for which

$$\widetilde{\mathcal{P}}^{GEQ,e}_{p,q'}(t_{q'}) \text{ but } \neg\widetilde{\mathcal{P}}^{GEQ,e}_{p,q'}(t_{q''}), \tag{31}$$

since otherwise $Q'' := Q'$ would have been a choice for Q'' contradicting (29). (31), however, can only hold, if tick k has been received locally at pipepair $(p, q')_{GEQ}$ at time $t_{q',rcv,k}$, with

$$t_{q',rcv,k} \leq t_{q''}.$$

According to (30) and the channel properties, we hence obtain

$$t_{p,k'} - t_{q',rcv,k} \geq \tau^-_{TH} + \tau^-_{GEQ}. \tag{32}$$

and by the local channel properties,

$$t_{q',rcv,k} - t_{p,k} \geq \tau^-_{loc}. \tag{33}$$

Finally, (27) together with (32) and (33) yields

$$t_{p,k'} - t_{p,k-1} \geq (\tau^-_{TH} + \tau^-_{GEQ} + \tau^-_{Diff} + \tau^-_{loc}) + (\tau^-_{TH} + \tau^-_{GEQ} + \tau^-_{loc}). \tag{34}$$

case (a.ii): Assuming that implication (ii) is valid, i.e., there exists a set Q' of size $|Q'| \geq f+1$, s.t.

$$\forall q' \in Q' : \widetilde{\mathcal{P}}^{GR,e}_{p,q'}(t_{q'}) \wedge r^{loc}_{p,q'}(t_{q'}) \geq k-1$$

with

$$t_{q'} := f^{-1}\left(\langle \mathcal{P}^{GR} toGR, e, p, q' \rangle; f^{-1}\left(\left\langle \sum GRtoTHGR, e, p \right\rangle; t_{p,k}\right)\right).$$

Thus tick $k-1$ has arrived at local pipepair $(p, q')_{GR}$ at time $t_{q',rcv,k-1}$, with

$$t_{q',rcv,k-1} \leq t_{q'} - \tau^-_{Diff} \tag{35}$$

by application of Lemma 5. By (35) together with the local channel's delay

$$t_{p,k} - t_{p,k-1} \geq \tau^-_{TH} + \tau^-_{GR} + \tau^-_{Diff} + \tau^-_{loc}. \tag{36}$$

From the algorithm we further know that tick k' could not have been sent without

$$\neg\widetilde{THGR}^{e}_{p}(t_{p,k'}).$$

Thus it must hold, that

$$\nexists Q'', |Q''| \geq f+1 : \forall q'' \in Q'' : \widetilde{\mathcal{P}}^{GR,e}_{p,q''}(t_{q''}). \tag{37}$$

with

$$t_{q''} := f^{-1}\left(\langle \mathcal{P}^{GR} toGR, e, p, q' \rangle; f^{-1}\left(\left\langle \sum GRtoTHGR, e, p \right\rangle; t_{p,k'}\right)\right). \tag{38}$$

Let us choose $Q'' := Q'$. Because of the FIFO property of the channels and because of $t_{p,k} < t_{p,k'}$, we obtain

$$t_{q'} < t_{q''}.$$

Clearly there has to be at least one $q' \in Q'$ for which

$$\widetilde{\mathcal{P}}^{GR,e}_{p,q'}(t_{q'}) \text{ but } \neg\widetilde{\mathcal{P}}^{GR,e}_{p,q'}(t_{q''}), \tag{39}$$

since otherwise $Q'' := Q'$ would have been a choice for Q'' contradicting (37). (39), however, can only hold, if tick k has been received locally at pipepair $(p, q')_{GR}$ at time $t_{q',rcv,k}$, with

$$t_{q',rcv,k} \leq t_{q''}.$$

According to (38) and the channel properties, we hence obtain

$$t_{p,k'} - t_{q',rcv,k} \geq \tau^-_{TH} + \tau^-_{GR}.$$

and by the local channel delay,

$$t_{p,k'} - t_{p,k} \geq \tau^-_{TH} + \tau^-_{GR} + \tau^-_{loc}.$$

Finally together with (36) this yields

$$t_{p,k'} - t_{p,k-1} \geq (\tau^-_{TH} + \tau^-_{GR} + \tau^-_{Diff} + \tau^-_{loc}) + (\tau^-_{TH} + \tau^-_{GR} + \tau^-_{loc}). \tag{40}$$

Combination of (a.i) and (a.ii): Combining (25), (34) and (40) leads to

$$\begin{aligned}(\tau^-_{TH} + \min\{\tau^-_{GEQ}, \tau^-_{GR}\} + \tau^-_{Diff} + \tau^-_{loc}) + (\tau^-_{TH} + \min\{\tau^-_{GEQ}, \tau^-_{GR}\} + \tau^-_{loc}) \\ \leq t_{p,k'} - t_{p,k-1} < \tau^+_{TH} + \tau^+_{GEQ} + \tau^+_{loc}\end{aligned} \tag{41}$$

which is a contradiction to Constraint 4.

case (b): If $\widetilde{THGR}^{o}_{p}(t_{p,k'})$, then by the algorithm, there must be a set $Q \subseteq P \setminus \{p\}$, of size $|Q| \geq f+1$, s.t., for $t' := f^{-1}\left(\langle \sum GRtoTHGR, o, p \rangle; t_{p,k'}\right)$

$$\forall q \in Q : \widetilde{GR}^{o}_{p,q}(t').$$

By analogous arguments as in case (a), we obtain

$$t_{p,k'} - t_{p,k-1} < \tau^+_{TH} + \tau^+_{GR} + \tau^+_{loc}. \tag{42}$$

Disabling path: By analogous case distinction as in case (a), we obtain:

case (b.i):

$$t_{p,k'} - t_{p,k-1} \geq (\tau^-_{TH} + \tau^-_{GEQ} + \tau^-_{Diff} + \tau^-_{loc}) + (\tau^-_{TH} + \tau^-_{GEQ} + \tau^-_{loc}). \tag{43}$$

case (b.ii):

$$t_{p,k'} - t_{p,k-1} \geq (\tau^-_{TH} + \tau^-_{GR} + \tau^-_{Diff} + \tau^-_{loc}) + (\tau^-_{TH} + \tau^-_{GR} + \tau^-_{loc}). \tag{44}$$

Combination of (b.i) and (b.ii): Combining (42), (43) and (44) leads to

$$\begin{aligned}(\tau^-_{TH} + \min\{\tau^-_{GEQ}, \tau^-_{GR}\} + \tau^-_{Diff} + \tau^-_{loc}) + (\tau^-_{TH} + \min\{\tau^-_{GEQ}, \tau^-_{GR}\} + \tau^-_{loc}) \\ \leq t_{p,k'} - t_{p,k-1} < \tau^+_{TH} + \tau^+_{GR} + \tau^+_{loc}\end{aligned} \tag{45}$$

which is a contradiction to Constraint 4, too.

The next lemma enables us to prove a minimum duration between two successive tick generated by a correct node.

Lemma 8. *Assume that correct node p sends tick $k \geq 1$ at time $t_{p,k}$. Then, it cannot send tick $k+1$ before $t_{p,k}+T_{min}$.*

Proof. Assume that a correct node p sends tick k, wlog. $k \in \mathbb{N}_{even}$, at time $t_{p,k}$. Now assume by contradiction that p sends tick $k+1$ at time $t_{p,k+1}$ with

$$t_{p,k+1} - t_{p,k} < T_{min}. \tag{46}$$

We apply Lemma 7 to the sending of tick $k+1$. Thus either implication (i) or (ii) has to be true:

case (i): There exists at least one correct node $q \in Q$ among the $2f+1$ nodes Q, s.t., for some $t_q \leq t_{p,k+1} - \tau^-_{TH} - \tau^-_{GEQ}$

$$\begin{aligned}&\widetilde{\mathcal{P}}^{GEQ,e}_{p,q}(t_q) \wedge (r^{loc}_{p,q}(t_q) \geq k) \quad \Rightarrow \\ &(s^{loc,geq}_{p,q}(t_q) = 1) \wedge (r^{loc}_{p,q}(t_q) \geq k).\end{aligned}$$

By applying Lemma 5, we obtain

$$r^{loc}_{p,q}(t_q - \tau^-_{Diff}) \geq k. \tag{47}$$

Further, by the local channel property (Pmin) established in Lemma 4

$$r^{loc}_{p,q}(t_q - \tau^-_{Diff}) \leq b_p(t_q - \tau^-_{Diff} - \tau^-_{loc}). \tag{48}$$

Combination of (47) and (48) yields

$$b_p(t_q - \tau^-_{Diff} - \tau^-_{loc}) \geq k,$$

i.e., tick k must have been sent at time $t_{p,k}$ with

$$\begin{aligned}t_{p,k} &\leq t_q - \tau^-_{Diff} - \tau^-_{loc} \\ &\leq t_{p,k+1} - \tau^-_{Diff} - \tau^-_{TH} - \tau^-_{GEQ} - \tau^-_{loc} \\ &\leq t_{p,k+1} - \tau^-_{Diff} - \tau^-_{TH} - \min\{\tau^-_{GEQ}, \tau^-_{GR}\} - \tau^-_{loc} \\ &= t_{p,k+1} - T_{min},\end{aligned}$$

contradicting (46).

case (ii): The contradiction is derived by analogous means as in case (i).

Lemma 9, 10 and 11 together with Constraint 5 below allow us to exclude the possibility of queuing effects inside a pipepair $(p,q)_{GEQ}$ of GEQ local/remote pipes $r^{loc,GEQ}_{p,q}(t)$ and $r^{rem,GEQ}_{p,q}(t)$ or $(p,q)_{GR}$ of GR local/remote pipes $r^{loc,GR}_{p,q}(t)$ and $r^{rem,GR}_{p,q}(t)$ at correct node p and corresponding to correct node q.

Constraint 5 $\tau^+_{Diff} \leq T_{min}$

Lemma 9. *For any pair of distinct correct nodes p, q and $k \geq 1$: If node p sent tick k at $t_{p,k}$ and q sent tick k at $t_{q,k}$, then tick $k-1$ is removed from the local and remote pipe of pipepair $(p,q)_{GR}$ by time $\max\{t_{p,k}+\tau^+_{loc}, t_{q,k}+\tau^+_{rem}\} + \tau^+_{Diff}$, if Constraint 5 holds.*

Proof. The proof is by induction on the number of ticks $k \geq 1$ that are sent by p and q.
Begin ($k = 1$): Assume that p sends tick 1 at $t_{p,k}$, which hence reaches the end of the local pipe of $(p,q)_{GR}$ at p by $t_{p,k} + \tau^+_{loc}$. Further assume that q sends tick 1 at $t_{q,k}$, which hence reaches the end of the corresponding remote pipe of $(p,q)_{GR}$ at p by $t_{q,k} + \tau^+_{rem}$. Since there is no tick that could block the removing of tick 0, it is removed by $\max\{t_{p,k} + \tau^+_{loc}, t_{q,k} + \tau^+_{rem}\} + \tau^+_{Diff}$ from both the local and remote pipe according to the Diff-gate properties.

Step ($k > 1$): As the induction hypothesis, assume that tick $k-2$ is removed from both pipes by $\max\{t_{p,k-1} + \tau^+_{loc}, t_{q,k-1} + \tau^+_{rem}\} + \tau^+_{Diff}$. By analogous arguments as above, tick k arrives in both pipes by t_{both}, with

$$t_{both} := \max\{t_{p,k} + \tau^+_{loc}, t_{q,k} + \tau^+_{rem}\}.$$

Because of Lemma 8, consecutive ticks cannot be generated with less than T_{min} distance in-between, i.e.,

$$\begin{aligned} t_{p,k} - t_{p,k-1} &\geq T_{min} \text{ and} \\ t_{q,k} - t_{q,k-1} &\geq T_{min}. \end{aligned}$$

Thus

$$\begin{aligned} t_{both} &\geq \max\{t_{p,k-1} + \tau^+_{loc}, t_{q,k-1} + \tau^+_{rem}\} + T_{min} \\ &\geq \max\{t_{p,k-1} + \tau^+_{loc}, t_{q,k-1} + \tau^+_{rem}\} + \tau^+_{Diff} \end{aligned}$$

by Constraint 5. According to the induction hypothesis, tick $k-2$ is removed by t_{both}. Thus tick $k-1$ is removed by $t_{both} + \tau^+_{Diff}$ from both the local and remote pipe according to the Diff-Gate properties.

Lemma 10. *For any pair of distinct correct nodes p, q and $1 \leq k \leq X$: If node p sent tick k at $t_{p,k}$, then tick $k-1$ is removed from the outputs of local and remote pipe of pipepair $(p,q)_{GEQ}$ by time $t_{p,k} + \tau^+_{loc} + \tau^+_{Diff}$, if Constraint 5 holds.*

Proof. For the GEQ pipe, we need to consider the initial fill-up of X additional virtual ticks in the remote pipeline $(p,q)_{GEQ}$ at booting time. The proof is by induction on k.
Begin ($k = 1$): Assume that p sends tick 1 at $t_{p,1}$. Tick 1 will arrive at the output $r^{loc,GEQ}_{p,q}(t)$ of the local pipe of $(p,q)_{GEQ}$ at p by $t_{p,1} + \tau^+_{loc}$. The corresponding remote pipe has already been initialized with tick $1-X$, so tick 1 shows up at $r^{rem,GEQ}_{p,q}(t)$ right from the beginning. Thus, by the behavior of the Diff-gate, tick 0 is removed from the output of the remote and local pipes at $t_{rmv,0} \leq t_{p,1} + \tau^+_{loc} + \tau^+_{Diff}$.
Tick $k \geq 2$: As the induction hypothesis, assume that the lemma is true for k. Further, assume that p sends tick $k+1 \leq X$ at time $t_{p,k+1}$. Since tick $k+1-X$ has been filled into the remote pipe at startup, tick $k+1$ shows up at $r^{rem,GEQ}_{p,q}(t)$ right from the beginning. Hence, tick k is removed at

$$t_{rmv,k} \leq \max\{t_{p,k+1} + \tau^+_{loc}, t_{rmv,k-1}\} + \tau^+_{Diff}.$$

It remains to be shown that $t_{p,k+1} + \tau^+_{loc} \geq t_{rmv,k-1}$. Because of Lemma 8, consecutive ticks cannot be generated with less than T_{min} distance in-between. Thus

$$\begin{aligned} t_{p,k} + \tau^+_{loc} &\geq t_{p,k-1} + T_{min} + \tau^+_{loc} \\ &\geq t_{p,k-1} + \tau^+_{Diff} + \tau^+_{loc} \end{aligned}$$

by Constraint 5 and

$$\geq t_{rmv,k-1} \tag{49}$$

by the induction hypothesis. Thus, tick $k-1$ is removed by $t_{rmv,k} \leq t_{p,k+1} + \tau^+_{loc} + \tau^+_{Diff}$.

Lemma 11. *For any pair of distinct correct nodes p, q and $k \geq X+1$: If node p sent tick k at $t_{p,k}$ and node q sent tick $k-X$ at $t_{q,k-X}$, then tick $k-1$ is removed from the outputs of local and remote pipe of pipepair $(p,q)_{GEQ}$ by time $\max\{t_{p,k}+\tau^+_{loc}, t_{q,k-X}+\tau^+_{rem}\}+\tau^+_{Diff}$, if Constraint 5 holds.*

Proof. The proof is by induction on k.
Begin ($k = X+1$): Assume that p sends tick $X+1$ at $t_{p,X+1}$ and q sends tick 1 at $t_{q,1}$. Clearly, both ticks will have reached the end of both local and remote pipeline $(p,q)_{GEQ}$ by $t_{both} := \max\{t_{p,X+1}+\tau^+_{loc}, t_{q,1}+\tau^+_{rem}\}$, hence

$$t_{both} \geq t_{p,X+1}+\tau^+_{loc}.$$

Thus tick X is removed from both pipes at

$$t_{rmv,X} \leq \max\{t_{both}, t_{rmv,X-1}\}+\tau^+_{Diff}.$$

It thus remains to be shown that $t_{both} \geq t_{rmv,X-1}$. By Lemma 8, consecutive ticks cannot be generated with less than T_{min} distance in-between. Thus

$$\begin{aligned} t_{p,X+1}+\tau^+_{loc} &\geq t_{p,X}+T_{min}+\tau^+_{loc} \\ &\geq t_{p,X}+\tau^+_{Diff}+\tau^+_{loc} \end{aligned}$$

by Constraint 5 and

$$\geq t_{rmv,X-1} \tag{50}$$

by Lemma 10 for $k = X$. Thus, tick X is removed by $t_{rmv,X} \leq t_{both}+\tau^+_{Diff}$ as asserted.

Step ($k \to k+1$): As induction hypothesis assume that the lemma holds for $k \geq X+1$. Assume that p sends tick $k+1$ at $t_{p,k+1}$ and q sends tick $k+1-X$ at $t_{q,k+1-X}$. Clearly both ticks will have arrived in the local and remote pipeline $(p,q)_{GEQ}$ by

$$t_{both} := \max\{t_{p,k+1}+\tau^+_{loc}, t_{q,k+1-X}+\tau^+_{rem}\}.$$

Thus tick k is removed from both pipes at

$$t_{rmv,k} \leq \max\{t_{both}, t_{rmv,k-1}\}+\tau^+_{Diff}.$$

It thus remains to be shown that $t_{both} \geq t_{rmv,k-1}$. By Lemma 8, consecutive ticks cannot be generated with less than T_{min} distance in-between. Thus

$$\begin{aligned} &\max\{t_{p,k+1}+\tau^+_{loc}, t_{q,k+1-X}+\tau^+_{rem}\} \geq \\ &\max\{t_{p,k}+T_{min}+\tau^+_{loc}, t_{q,k-X}+T_{min}+\tau^+_{rem}\} \geq \\ &\max\{t_{p,k}+\tau^+_{loc}, t_{q,k-X}+\tau^+_{rem}\}+\tau^+_{Diff} \geq \end{aligned} \tag{51}$$

by Constraint 5 and

$$\geq t_{rmv,k-1} \tag{52}$$

by the induction hypothesis. Thus tick k is removed by $t_{rmv,k} \leq \max\{t_{p,k}+\tau^+_{loc}, t_{q,k-X}+\tau^+_{rem}\}+\tau^+_{Diff}$.

The next lemma establishes that the GR-rule really does its duty: If sufficiently many $(f+1)$ correct nodes send tick k messages, these are received at any correct node p, which then sends tick k itself if it has not already done so.

Lemma 12. *For all correct nodes p and tick numbers $k \geq 1$: If there exists a set Q of correct nodes with $|Q| \geq f+1$, s.t., for all $q \in Q$:*

(i) all nodes q send tick $k+1$ by $t_{Q,k+1}$,
(ii) p sends tick k by $t_{p,k}$,

then p sends tick $k+1$ by $t' := \max\{t_{p,k} + \tau^+_{loc} + \tau^+_{Diff} + \tau^+_{GR}, t_{p,k} + \tau^+_{loc} + \tau^+_{GEQ}, t_{Q,k+1} + \tau^+_{rem} + \tau^+_{GR}\} + \tau^+_{TH}$.

Proof. Wlog. assume that $k \in \mathbb{N}_{even}$. We distinguish between two cases:

(a) Assume, that $\exists q \in Q : r^{loc}_{p,q}(t') > k$. Since $r^{loc}_{p,q}(t') \leq b_p(t')$, i.e., no tick can be locally received if it has not been sent before, $b_p(t') > k$ must hold. Thus, p has already sent tick $k+1$ by t'.
(b) Otherwise,

$$\forall q \in Q : r^{loc}_{p,q}(t') \leq k. \tag{53}$$

By Lemma 8 it follows that all nodes q send tick $k \geq 1$ by $t_{Q,k}$ with $t_{Q,k} \leq t_{Q,k+1} - T_{min}$. By applying Lemma 9, we conclude, that tick $k-1$ is removed from all of p's pipes corresponding to all $q \in Q$ by

$$\begin{aligned} t_{del,k-1} &:= \max\{t_{p,k} + \tau^+_{loc}, t_{Q,k} + \tau^+_{rem}\} + \tau^+_{Diff} \\ &\leq \max\{t_{p,k} + \tau^+_{loc}, t_{Q,k+1} - T_{min} + \tau^+_{rem}\} + \tau^+_{Diff} \\ &\leq \max\{t_{p,k} + \tau^+_{loc} + \tau^+_{Diff}, t_{Q,k+1} + \tau^+_{rem}\}. \end{aligned} \tag{54}$$

Clearly, since all nodes $q \in Q$ are correct and have sent tick $k+1$ by $t_{Q,k+1}$, tick $k+1$ must arrive at p's remote pipe corresponding to q by $t_{rcv,Q} := t_{Q,k+1} + \tau^+_{rem}$. Furthermore, p must receive tick k in all its local pipes by $t_{rcv,p} := t_{p,k} + \tau^+_{loc}$, i.e., at $t_{rcv} := \max\{t_{rcv,p}, t_{rcv,Q}, t_{del,k-1}\} \leq \max\{t_{p,k} + \tau^+_{loc} + \tau^+_{Diff}, t_{Q,k+1} + \tau^+_{rem}\}$ it holds that:

$$r^{loc}_{p,q}(t_{rcv}) \geq k \qquad r^{rem}_{p,q}(t_{rcv}) \geq k+1 \qquad d^{GR}_{p,q}(t) \geq k-1 \tag{55}$$

By combination of (55) with (53), it holds that for all $\xi \in [t_{rcv}, t']$:

$$r^{loc}_{p,q}(\xi) = k \quad \text{and} \quad r^{rem}_{p,q}(\xi) > k. \tag{56}$$

Furthermore,

$$\begin{aligned} \widetilde{\mathcal{P}}^{GR,e}_{p,q}(\xi) &\equiv (r^{rem}_{p,q}(\xi) > r^{loc}_{p,q}(\xi)) \wedge (r^{loc}_{p,q}(\xi) \in \mathbb{N}_{even}) \wedge (s^{loc,gr}_{p,q}(\xi) = 1) \\ &\equiv (r^{rem}_{p,q}(\xi) > k) \wedge (r^{loc}_{p,q}(\xi) - d^{GR}_{p,q}(\xi) = 1) \\ &\equiv (d^{GR}_{p,q}(\xi) = k-1). \end{aligned} \tag{57}$$

Assuming $\exists q \in Q : d^{GR}_{p,q}(\xi) > k-1$ implies $r^{loc}_{p,q}(\xi) > k$, contradicting (56). Thus, $\forall q \in Q : d^{GR}_{p,q}(t) = k-1$ must hold, which implies that $\forall q \in Q : \widetilde{\mathcal{P}}^{GR,e}_{p,q}(\xi)$ is true for $\xi \in [t_{rcv}, t']$.
By the algorithm, $\forall q \in Q : \widetilde{GR}^e_{p,q}(\xi)$ is true for $\xi \in \left[t_{rcv} + \tau^+_{GR}, t'\right]$, such that

$$\widetilde{THGR}^e_p(\xi) \tag{58}$$

is true for

$$\begin{aligned} &\xi \in \left[t_{rcv} + \tau^+_{GR} + \tau^+_{TH}, t'\right] = \\ &\quad \left[\max\{t_{p,k} + \tau^+_{loc} + \tau^+_{Diff}, t_{Q,k+1} + \tau^+_{rem}\} + \tau^+_{GR} + \tau^+_{TH}, t'\right]. \end{aligned}$$

It remains to be shown that the disabling path cannot inhibit the generation of tick $k+1$ at p. For the sake of contradiction, assume that the disabling path can enforce $t_{p,k+1} > t'$.

Because of the lemma's assumption (ii), tick k must eventually arrive in all p's local pipes, i.e., $\forall r \in P \setminus \{p\}$: $r^{loc}_{p,r}(\xi) \geq k$, for $\xi \in \left[t_{p,k} + \tau^+_{loc}, t'\right]$. Note that, since p did not send tick $k+1$ yet, this actually implies $r^{loc}_{p,r}(\xi) = k$. As $k \in \mathbb{N}_{even}$,

$$\forall r \in P \setminus \{p\} : \neg \left(\widetilde{\mathcal{P}}^{GR,o}_{p,r}(\xi) \vee \widetilde{\mathcal{P}}^{GEQ,o}_{p,r}(\xi) \right).$$

By the algorithm and analogous arguments as before,

$$\neg \left(\widetilde{THGR}^{o}_{p}(\xi) \vee \widetilde{THGEQ}^{o}_{p}(\xi) \right) \tag{59}$$

for $\xi \in \left[t_{p,k} + \tau^+_{loc} + \max\{\tau^+_{GR}, \tau^+_{GEQ}\} + \tau^+_{TH}, t'\right]$.
Combining (58) and (59) and noting that $\max\{\tau^+_{GR} + \tau^+_{Diff}, \max\{\tau^+_{GR}, \tau^+_{GEQ}\}\} = \max\{\tau^+_{GR} + \tau^+_{Diff}, \tau^+_{GEQ}\}$, it is apparent that p must send tick $k+1$ by t', providing the required contradiction.

Analogous to Lemma 12, the next lemma states that the GEQ-rule does its duty: If a set of at least $2f+1$ correct nodes send tick k, then any correct node p broadcasts a tick $k+1$ message.

Lemma 13. *For all correct nodes p and tick numbers $k \geq X+1$: If there exists a set Q of correct nodes with $|Q| \geq 2f+1$, s.t., for all $q \in Q$:*

(i) all nodes q send tick $k-X$ by $t_{Q,k-X}$,
(ii) p sends tick k by $t_{p,k}$,

then p sends tick $k+1$ by
$t' := \max\{t_{p,k} + \tau^+_{loc} + \tau^+_{Diff} + \tau^+_{GEQ}, t_{p,k} + \tau^+_{loc} + \tau^+_{GR}, t_{Q,k-X} + \tau^+_{rem} + \tau^+_{Diff} + \tau^+_{GEQ}\} + \tau^+_{TH}$.

Proof. Wlog. assume, that $k \in \mathbb{N}_{even}$ and distinguish two cases:

(a) Assume that $\exists q \in Q : r^{loc}_{p,q}(t') > k$. Since $r^{loc}_{p,q}(t') \leq b_p(t')$, as no tick can be locally received if it has not been sent before, $b_p(t') > k$ must hold. Thus, p has already sent tick $k+1$ by t'.
(b) Otherwise,

$$\forall q \in Q : r^{loc}_{p,q}(t') \leq k. \tag{60}$$

Since $k \geq X+2$, we may apply Lemma 11, concluding, that tick $k-1$ is removed from the local and remote geq pipes corresponding to $q \in Q$ by

$$t_{del,k-1} \leq \max\{t_{p,k} + \tau^+_{loc}, t_{Q,k-X} + \tau^+_{rem}\} + \tau^+_{Diff}.$$

Clearly, since all nodes $q \in Q$ are correct and have sent tick $k-X$ by $t_{Q,k-X}$, tick $k-X$ must arrive at the remote pipe $(p,q)_{GEQ}$ by $t_{rcv,Q} := t_{Q,k-X} + \tau^+_{rem}$. Further, p must receive tick k in all its local pipes by $t_{rcv,p} := t_{p,k} + \tau^+_{loc}$. Thus, at

$$\begin{aligned} t_{rcv} := &\max\{t_{rcv,p}, t_{rcv,Q}, t_{del,k-1}\} \leq \\ &\max\{t_{p,k} + \tau^+_{loc}, t_{Q,k-X} + \tau^+_{rem}\} + \tau^+_{Diff} \end{aligned}$$

it holds that:

$$r^{loc}_{p,q}(t_{rcv}) \geq k, \qquad r^{rem}_{p,q}(t_{rcv}) \geq k-X \Rightarrow r^{rem,GEQ}_{p,q}(t_{rcv}) \geq k, \qquad d^{GEQ}_{p,q}(t_{rcv}) \geq k-1 \tag{61}$$

By combination of (61) with (60), it holds that for all $\xi \in [t_{rcv}, t']$:

$$r^{loc}_{p,q}(\xi) = k \quad \text{and} \quad r^{rem,GEQ}_{p,q}(\xi) \geq k. \tag{62}$$

Furthermore,

$$\begin{aligned}\widetilde{\mathcal{P}}^{GEQ,e}_{p,q}(\xi) &\equiv (r^{rem,GEQ}_{p,q}(\xi) \geq r^{loc}_{p,q}(\xi)) \wedge (r^{loc}_{p,q}(\xi) \in \mathbb{N}_{even}) \wedge (s^{loc,geq}_{p,q}(\xi) = 1) \\ &\equiv (r^{rem,GEQ}_{p,q}(\xi) \geq k) \wedge (r^{loc}_{p,q}(\xi) - d^{GEQ}_{p,q}(\xi) = 1) \\ &\equiv (d^{GEQ}_{p,q}(\xi) = k-1). \end{aligned} \tag{63}$$

Assuming $\exists q \in Q : d^{GEQ}_{p,q}(\xi) > k-1$ implies $r^{loc}_{p,q}(\xi) > k$, contradicting (62). Thus $\forall q \in Q : d^{GEQ}_{p,q}(t) = k-1$ must hold, which implies that $\forall q \in Q : \widetilde{\mathcal{P}}^{GEQ,e}_{p,q}(\xi)$ is true for $\xi \in [t_{rcv}, t']$.
By the algorithm, $\forall q \in Q : \widetilde{GEQ}^{e}_{p,q}(\xi)$ is true for $\xi \in \left[t_{rcv} + \tau^{+}_{GEQ}, t'\right]$, such that

$$\widetilde{THGEQ}^{e}_{p}(\xi) \tag{64}$$

is true for

$$\begin{aligned}\xi &\in \left[t_{rcv} + \tau^{+}_{GEQ} + \tau^{+}_{TH}, t'\right] \\ &= \left[\max\{t_{p,k} + \tau^{+}_{loc}, t_{Q,k-X} + \tau^{+}_{rem}\} + \tau^{+}_{Diff} + \tau^{+}_{GEQ} + \tau^{+}_{TH}, t'\right].\end{aligned}$$

It remains to be shown that the disabling path cannot inhibit the generation of tick $k+1$ at p. For the sake of contradiction, assume that the disabling path can enforce $t_{p,k+1} > t'$.
Because of the lemma's assumption (ii), tick k must eventually arrive in all p's local pipes, i.e., $\forall r \in P \setminus \{p\} : r^{loc}_{p,r}(\xi) \geq k$ and hence $r^{loc}_{p,r}(\xi) = k$ (as tick $k+1$ has not been sent yet) for $\xi \in \left[t_{p,k} + \tau^{+}_{loc}, t'\right]$. Since $k \in \mathbb{N}_{even}$,

$$\forall r \in P \setminus \{p\} : \neg\left(\widetilde{\mathcal{P}}^{GR,o}_{p,r}(\xi) \vee \widetilde{\mathcal{P}}^{GEQ,o}_{p,r}(\xi)\right).$$

By the algorithm and analogous arguments as before,

$$\neg\left(\widetilde{THGR}^{o}_{p}(\xi) \vee \widetilde{THGEQ}^{o}_{p}(\xi)\right) \tag{65}$$

for $\xi \in \left[t_{p,k} + \tau^{+}_{loc} + \max\{\tau^{+}_{GR}, \tau^{+}_{GEQ}\} + \tau^{+}_{TH}, t'\right]$.
Combining (64) and (65) and noting that $\max\{\tau^{+}_{GEQ} + \tau^{+}_{Diff}, \max\{\tau^{+}_{GR}, \tau^{+}_{GEQ}\}\} = \max\{\tau^{+}_{GEQ} + \tau^{+}_{Diff}, \tau^{+}_{GR}\}$, it is apparent that p must send tick $k+1$ by t', providing the required contradiction.

Theorem 4 and its proof are a generalization of the well-known consistent broadcasting results of [10, 11, 30]. For correctness, we additionally require

Constraint 6 $T^{-}_{first} \geq (X+1)(\tau^{+}_{loc} + \max\{\tau^{+}_{Diff} + \tau^{+}_{GR}, \tau^{+}_{GEQ}\} + \tau^{+}_{TH})$,

where T^{-}_{first} is defined in (66) below.

Theorem 4. (Synchronization Properties). *The algorithm satisfies the synchronization properties Unforgeability (U), Progress (P), Quasi-Simultaneity (QS) and Booting-Simultaneity (BS), if Constraints 4, 5, 6 and $n \geq 3f+2$ hold.*

(U) *Unforgeability. If no correct node sends tick $k \geq 1$ by time t, then no correct node sends tick $k+X+1$ by time $t + T^{-}_{first}$ or earlier, with*

$$T^{-}_{first} := \tau^{-}_{rem} + \tau^{-}_{Diff} + \tau^{-}_{GEQ} + \tau^{-}_{TH}. \tag{66}$$

(P) *Progress. If all correct nodes send tick $k \geq X+1$ by time t, then every correct node sends at least tick $k+1$ by time $t + T_P$, with*

$$T_P := \max\left\{\begin{array}{l}\tau^{+}_{loc} + \tau^{+}_{Diff} + \max\{\tau^{+}_{GEQ}, \tau^{+}_{GR}\}, \\ \tau^{+}_{loc} + \tau^{+}_{GR}, \\ \tau^{+}_{rem} + \tau^{+}_{Diff} + \tau^{+}_{GEQ} - XT_{min}\end{array}\right\} + \tau^{+}_{TH}.$$

(QS) *Quasi-Simultaneity. If some correct node p sends tick $k \geq X+2$ by time t, then every correct node (p included) sends at least tick $k-X-1$ by time $t+T_{QS}$, with*

$$T_{QS} := \max \left\{ \begin{array}{l} \tau^+_{rem} + \tau^+_{GR} + \tau^+_{TH} + X(\tau^+_{loc} + \max\{\tau^+_{Diff} + \tau^+_{GR}, \tau^+_{GEQ}\} + \tau^+_{TH} - T_{min}) - T^-_{first}, \\ B + (X+1)(\tau^+_{loc} + \max\{\tau^+_{Diff} + \tau^+_{GEQ}, \tau^+_{GR}\} + \tau^+_{TH} - T_{min}) - (\tau^+_{loc} - \tau^-_{loc}) - T^-_{first} \end{array} \right\}.$$

(BS) *Booting-Simultaneity. If some correct node sends tick $k \geq X+1$ by time t, then every correct node sends at least tick k by time $t+T_{BS}(k)$, with*

$$\begin{aligned} T_{BS}(k) := B + \max\{\tau^+_{GEQ}, \tau^+_{GR}\} - \min\{\tau^-_{GEQ}, \tau^-_{GR}\} + \tau^+_{TH} - \tau^-_{TH} + \\ + (k-1)(T_P - T_{min}). \end{aligned} \tag{67}$$

We will show the properties Unforgeability (U), Progress (P), Quasi-Simultaneity (QS) and Booting-Simultaneity (BS) one after the other.

Unforgeability (U)

Proof. Let p be the first correct node that sends tick $k+X+1 \geq X+2$, at time $t_{p,k+X+1}$. Wlog. assume that $k+X \in \mathbb{N}_{even}$. We apply Lemma 7 and consider the two possible implications:

(i) Since $|Q| \geq 2f+1$ here, there must be a subset $C' \subseteq Q$ of correct nodes of size $|C'| \geq f+1$. Clearly, it must hold that

$$\forall r \in C' : \exists t_r \leq t' : \widetilde{\mathcal{P}}^{GEQ,e}_{p,r}(t_r) \wedge r^{loc}_{p,r}(t_r) \geq k+X$$

with $t' = t_{p,k+X+1} - \tau^-_{TH} - \tau^-_{GEQ}$. By applying Lemma 5, we obtain for each of the $r \in C'$

$$r^{rem,GEQ}_{p,r}(t' - \tau^-_{Diff}) \geq k+X \quad \text{and hence} \quad r^{rem}_{p,r}(t' - \tau^-_{Diff}) \geq k.$$

By the remote channel properties this implies $b_r(t'') \geq k$ with

$$\begin{aligned} t'' &:= t' - \tau^-_{Diff} - \tau^-_{rem} \\ &= t_{p,k+X+1} - \tau^-_{TH} - \tau^-_{GEQ} - \tau^-_{Diff} - \tau^-_{rem} \\ &= t_{p,k+X+1} - T^-_{first} \end{aligned}$$

i.e., r — and hence also the first correct node that sent tick k — has sent tick k by $t_{p,k+X+1} - T^-_{first}$.

(ii) Since $|Q| \geq f+1$ here, there must be at least one correct nodes $r \neq p$ among Q for which $r^{rem}_{p,r}(t') \geq k+X+1$ with $t' = t_{p,k+X+1} - \tau^-_{TH} - \tau^-_{GR}$. For r, this implies $b_r(t') \geq k+X+1$ — a contradiction to the assumption that p was the first correct node to send tick $k+X+1$.

Progress (P)

Proof. Assume that all correct nodes — we denote them by C, $|C| \geq 2f+2$ — sent tick $k \geq X+1$ by time t. Let $p \in C$ be arbitrary. By applying Lemma 8 repeatedly to all correct nodes $Q = C \setminus \{p\}$ without p, it follows that $\forall q \in Q$, $t_{q,k-X} \leq t - XT_{min}$. We may now apply Lemma 13 with $Q = C \setminus \{p\}$ and the bounds $t_{p,k} = t$, $t_{Q,k-X} = t - XT_{min}$. Thus, p must send tick $k+1$ by

$$\begin{aligned} t' &= \max \left\{ \begin{array}{l} t_{p,k} + \tau^+_{loc} + \tau^+_{Diff} + \tau^+_{GEQ}, \\ t_{p,k} + \tau^+_{loc} + \tau^+_{GR}, \\ t_{Q,k-X} + \tau^+_{rem} + \tau^+_{Diff} + \tau^+_{GEQ} \end{array} \right\} + \tau^+_{TH} \\ &\leq t + \max \left\{ \begin{array}{l} \tau^+_{loc} + \tau^+_{Diff} + \tau^+_{GEQ}, \\ \tau^+_{loc} + \tau^+_{GR}, \\ \tau^+_{rem} + \tau^+_{Diff} + \tau^+_{GEQ} - XT_{min} \end{array} \right\} + \tau^+_{TH} \\ &\leq t + \max \left\{ \begin{array}{l} \tau^+_{loc} + \tau^+_{Diff} + \max\{\tau^+_{GEQ}, \tau^+_{GR}\}, \\ \tau^+_{loc} + \tau^+_{GR}, \\ \tau^+_{rem} + \tau^+_{Diff} + \tau^+_{GEQ} - XT_{min} \end{array} \right\} + \tau^+_{TH} \\ &= t + T_P \end{aligned}$$

We proceed with some technical lemmas.

Lemma 14. *If the first correct node p sends tick $k' = k+1 \geq X+2$ at time $t_{p,k'}$, then at $t' := t_{p,k'} - \tau^-_{TH} - \tau^-_{GEQ} - \tau^-_{Diff}$ it must hold that: There exists a set Q of size $|Q| \geq 2f+1$ s.t.*

$$\forall q \in Q : r^{rem,GEQ}_{p,q}(t') \geq k.$$

Proof. Analogous to the proof of (U), we apply Lemma 7 and consider the two possible implications:

(i) This case exactly matches the implication of our lemma.
(ii) Since $|Q| \geq f+1$ here, there must be at least one correct node $r \neq p$ among Q which has already sent tick k' before p did[17] — contradicting the assumption that p is the first correct one to send tick k'.

Let $t_{boot} = \min_{p \in C}\{t_{p,boot}\}$ be the time when the first correct node completes booting; recall that we actually assumed $t_{boot} = 0$.

Lemma 15. *Every correct node must send tick 1 within the interval $[t_{first,1}, t_{last,1}]$ with*

$$t_{first,1} \geq t_{boot} + \min\{\tau^-_{GEQ}, \tau^-_{GR}\} + \tau^-_{TH} \text{ and}$$
$$t_{last,1} \leq t_{boot} + B + \max\{\tau^+_{GEQ}, \tau^+_{GR}\} + \tau^+_{TH}.$$

Proof. Since the first correct node boots at t_{boot}, the first correct node that sends tick 1 does so at $t_{first,1}$ with

$$t_{first,1} \geq t_{boot} + \min\{\tau^-_{GEQ}, \tau^-_{GR}\} + \tau^-_{TH}.$$

All other correct nodes boot by $t_{boot} + B$. Thus all correct nodes must send tick 1 by $t_{last,1}$ with

$$t_{last,1} \leq t_{boot} + B + \max\{\tau^+_{GEQ}, \tau^+_{GR}\} + \tau^+_{TH}.$$

Lemma 16. *Every correct node p must send tick $k \geq 1$ at $t_{p,k}$ with*

$$t_{p,k} \geq t_{boot} + \min\{\tau^-_{GEQ}, \tau^-_{GR}\} + \tau^-_{TH} + (k-1)T_{min}. \tag{68}$$

Proof. The proof is by induction on $k \geq 1$.
Begin ($k = 1$): By application of Lemma 15

$$t_{p,1} \geq t_{boot} + \min\{\tau^-_{GEQ}, \tau^-_{GR}\} + \tau^-_{TH}. \tag{69}$$

Step ($k \to k+1$): Assume the lemma is true for $k \geq 1$. From Lemma 8,

$$t_{p,k+1} - t_{p,k} \geq T_{min}.$$

Applying the induction hypothesis yields

$$\begin{aligned} t_{p,k+1} &= t_{p,k+1} - t_{p,k} + t_{p,k} \\ &\geq T_{min} + t_{boot} + \min\{\tau^-_{GEQ}, \tau^-_{GR}\} + \tau^-_{TH} + (k-1)T_{min} \\ &= t_{boot} + \min\{\tau^-_{GEQ}, \tau^-_{GR}\} + \tau^-_{TH} + kT_{min}. \end{aligned}$$

Lemma 17. *Every correct node p must send tick $1 \leq k \leq X+1$ at $t_{p,k}$ with*

$$t_{p,k} \leq t_{boot} + B + \max\{\tau^+_{GEQ}, \tau^+_{GR}\} + \tau^+_{TH} + (k-1)(\tau^+_{loc} + \max\{\tau^+_{Diff} + \tau^+_{GEQ}, \tau^+_{GR}\} + \tau^+_{TH}).$$

[17] By analogous arguments as in the proof of (U)

Proof. The proof is by induction on k.
Begin ($k = 1$): This follows from applying Lemma 15, yielding

$$t_{p,1} \leq t_{boot} + B + \max\{\tau^+_{GEQ}, \tau^+_{GR}\} + \tau^+_{TH}.$$

Step ($k \to k+1$): Assume as induction hypothesis that the lemma holds for $1 \leq k \leq X$. Further assume, wlog., that $k \in \mathbb{N}_{even}$. By the behavioral specification of the remote pipes, we know that for all remote nodes q, and $t \geq t_{p,boot}$,

$$r^{rem,GEQ}_{p,q}(t) \geq X.$$

By the induction hypothesis, tick k is sent by p at time

$$t_{p,k} \leq t_{boot} + B + \max\{\tau^+_{GEQ}, \tau^+_{GR}\} + \tau^+_{TH} + (k-1)(\tau^+_{loc} + \max\{\tau^+_{Diff} + \tau^+_{GEQ}, \tau^+_{GR}\} + \tau^+_{TH})$$

and by the local channel properties arrives at all local GEQ pipes $(p,q)_{GEQ}$ by

$$t_{p,rcv,k} = t_{p,k} + \tau^+_{loc}$$

yielding

$$r^{loc,GEQ}_{p,q}(t_{p,rcv,k}) \geq k.$$

Let

$$I = \left[t_{p,rcv,k}, t_{boot} + B + \max\{\tau^+_{GEQ}, \tau^+_{GR}\} + \tau^+_{TH} + k(\tau^+_{loc} + \max\{\tau^+_{GEQ}, \tau^+_{GR}\} + \tau^+_{TH})\right].$$

We distinguish between two cases:
(i) Assume, that there exists a remote $q \in P \setminus \{p\}$ and a $\xi \in I$, s.t.,

$$r^{loc,GEQ}_{p,q}(\xi) > k.$$

In this case, p must already have generated tick $k+1$ by ξ, yielding the desired result.
(ii) Otherwise assume, that

$$\forall q \in P \setminus \{p\}, \xi \in I : r^{loc,GEQ}_{p,q}(\xi) = k \tag{70}$$

Since $k \in \mathbb{N}_{even}$, all odd GEQ/GR signals are disabled, i.e.,

$$\neg(\widetilde{\mathcal{P}}^{GEQ,o}_{p,q}(\xi) \vee \widetilde{\mathcal{P}}^{GR,o}_{p,q}(\xi)). \tag{71}$$

Further, since $k \leq X$,

$$\forall q \in P \setminus \{p\}, \xi \in I : r^{rem,GEQ}_{p,q}(\xi) \geq r^{loc,GEQ}_{p,q}(\xi) = k \tag{72}$$

and by Lemma 10, tick $k-1$ is removed from both the local and remote pipe of pipepair $(p,q)_{GEQ}$ by

$$t_{p,rmv,k-1} = t_{p,k} + \tau^+_{loc} + \tau^+_{Diff}.$$

Thus for all $q \in P \setminus \{p\}$ and $\xi \in I'$, with

$$I' = \left[t_{p,k} + \tau^+_{loc} + \tau^+_{Diff}, t_{boot} + B + \max\{\tau^+_{GEQ}, \tau^+_{GR}\} + \tau^+_{TH} + k(\tau^+_{loc} + \max\{\tau^+_{Diff} + \tau^+_{GEQ}, \tau^+_{GR}\} + \tau^+_{TH})\right],$$

it holds that

$$\widetilde{\mathcal{P}}^{GEQ,e}_{p,q}(\xi) \tag{73}$$

Because of the channel properties and the algorithm, we deduce from (71) and (73) that p must generate tick $k+1$ at

$$t_{p,k+1} \leq t_{p,k} + \tau^+_{loc} \max\{\tau^+_{Diff} + \tau^+_{GEQ}, \tau^+_{GR}\} + \tau^+_{TH}$$

and by the induction hypothesis

$$\leq t_{boot} + B + \max\{\tau^+_{GEQ}, \tau^+_{GR}\} + \tau^+_{TH} + k(\tau^+_{loc} + \max\{\tau^+_{Diff} + \tau^+_{GEQ}, \tau^+_{GR}\} + \tau^+_{TH}).$$

Quasi Simultaneity (QS)

Proof. The proof is by induction on the number of ticks $k \geq X+2$ sent by the first correct node.
Begin ($X+2 \leq k \leq 2X+2$): Assume the first node that sends tick $k \geq X+2$ does so at $t_{first,k}$. By Unforgeability,

$$t_{first,k} \geq t_{first,k-X-1} + T^-_{first}. \tag{74}$$

By Lemma 16, it must further hold that

$$t_{first,k-X-1} \geq t_{boot} + \min\{\tau^-_{GEQ}, \tau^-_{GR}\} + \tau^-_{TH} + (k-X-2)T_{min}.$$

Thus from (74) we obtain

$$t_{first,k} \geq t_{boot} + \min\{\tau^-_{GEQ}, \tau^-_{GR}\} + \tau^-_{TH} + (k-X-2)T_{min} + T^-_{first}.$$

By Lemma 17, all other correct nodes must send tick $k-X-1$ with $1 \leq k-X-1 \leq X+1$ by $t_{last,k-X-1}$ with

$$t_{last,k-X-1} \leq t_{boot} + B + \max\{\tau^+_{GEQ}, \tau^+_{GR}\} + \tau^+_{TH} + (k-X-2)(\tau^+_{loc} + \max\{\tau^+_{Diff} + \tau^+_{GEQ}, \tau^+_{GR}\} + \tau^+_{TH}).$$

Thus

$$\begin{aligned} t_{last,k-X-1} - t_{first,k} \leq & B + \max\{\tau^+_{GEQ}, \tau^+_{GR}\} + \tau^+_{TH} - (\min\{\tau^-_{GEQ}, \tau^-_{GR}\} + \tau^-_{TH}) + \\ & (k-X-2)(\tau^+_{loc} + \max\{\tau^+_{Diff} + \tau^+_{GEQ}, \tau^+_{GR}\} + \tau^+_{TH} - T_{min}) - T^-_{first} \\ \leq & B + \max\{\tau^+_{GEQ}, \tau^+_{GR}\} + \tau^+_{TH} - (\min\{\tau^-_{GEQ}, \tau^-_{GR}\} + \tau^-_{TH}) + \\ & X(\tau^+_{loc} + \max\{\tau^+_{Diff} + \tau^+_{GEQ}, \tau^+_{GR}\} + \tau^+_{TH} - T_{min}) - T^-_{first} \\ \leq & T_{QS}. \end{aligned}$$

Step ($k-X-1 \to k$)**:** As induction hypothesis, assume, that the lemma holds for $k-X-1 \geq X+2$, i.e., all correct nodes send tick $k-2X-2$ by $t_{last,k-2X-2} \leq t_{first,k-X-1} + T_{QS}$.

Let $t_{first,k}$ be the time the first correct node, say p, sends tick $k > 2X+2$. By Lemma 14, there exists a set Q of size $|Q| \geq 2f+1$, s.t., at time $t' = t_{first,k} - \tau^-_{TH} - \tau^-_{GEQ} - \tau^-_{Diff}$:

$$\forall q \in Q : r^{rem}_{p,q}(t') = r^{rem,GEQ}_{p,q}(t') - X \geq k-X-1. \tag{75}$$

Clearly there is a subset $\bar{Q} \subseteq Q$ of correct nodes among Q of size at least $|\bar{Q}| \geq f+1$. Let $\bar{Q}' := C \setminus (\bar{Q} \cup \{p\})$. Now consider the partitioning of correct nodes $C = \bar{Q} \cup \{p\} \cup \bar{Q}'$. We will prove the lemma separately for each of the three partitions: In case $q \in \bar{Q}$, by (75) and the remote channel properties, q has sent tick $k-X-1$ by $t' - \tau^-_{rem} < t_{first,k}$. In case $q = p$, it has sent tick k by $t_{first,k}$. The interesting case $q \in \bar{Q}'$ is shown in the following: Any $\bar{q} \in \bar{Q}$, since it is correct, must have sent tick $k-X-1$ by

$$t_{\bar{Q},k-X-1} := t_{first,k} - T^-_{first} \tag{76}$$

according to (U). Clearly, $t_{first,k-X-1} \leq t_{\bar{Q},k-X-1}$. Thus we may apply the induction hypothesis, yielding, for every $r \in \bar{Q}'$,

$$t_{r,k-2X-2} \leq t_{last,k-2X-2} \leq t_{first,k-X-1} + T_{QS}. \tag{77}$$

Further, by repeatedly applying Lemma 8, we obtain for the $f+1$ nodes $q \in \bar{Q}$

$$t_{q,k-2X-1} \leq t_{first,k-X-1} - XT_{min} \tag{78}$$

and more generally for $X+1 \geq m \geq 1$

$$t_{q,k-2X-2+m} \leq t_{first,k-X-1} - (X+1-m)T_{min}. \tag{79}$$

We may now apply Lemma 12 to node r and the set of correct nodes $\bar{Q}$ with $t_{r,k-2X-2}$ and $t_{q,k-2X-1}$ from (77) and (78). This yields: r sends tick $k-2X-1$ by

$$\begin{aligned}
t_{r,k-2X-1} &:= \max\left\{\begin{array}{l} t_{r,k-2X-2}+\tau^+_{loc}+\tau^+_{Diff}+\tau^+_{GR}, \\ t_{r,k-2X-2}+\tau^+_{loc}+\tau^+_{GEQ}, \\ t_{q,k-2X-1}+\tau^+_{rem}+\tau^+_{GR} \end{array}\right\}+\tau^+_{TH} \\
&\leq \max\left\{\begin{array}{l} t_{first,k-X-1}+T_{QS}+\tau^+_{loc}+\tau^+_{Diff}+\tau^+_{GR}, \\ t_{first,k-X-1}+T_{QS}+\tau^+_{loc}+\tau^+_{GEQ}, \\ t_{first,k-X-1}-XT_{min}+\tau^+_{rem}+\tau^+_{GR} \end{array}\right\}+\tau^+_{TH} \\
&\leq t_{first,k-X-1}+\max\left\{\begin{array}{l} T_{QS}+\tau^+_{loc}+\max\{\tau^+_{Diff}+\tau^+_{GR},\tau^+_{GEQ}\}, \\ -XT_{min}+\tau^+_{rem}+\tau^+_{GR} \end{array}\right\}+\tau^+_{TH}. \qquad (80)
\end{aligned}$$

Proceeding with simplifying the last $\max$-term, we first assume

$$\begin{aligned}
&T_{QS}+\tau^+_{loc}+\max\{\tau^+_{Diff}+\tau^+_{GR},\tau^+_{GEQ}\} < -XT_{min}+\tau^+_{rem}+\tau^+_{GR} \Leftrightarrow \\
&T_{QS} < \tau^+_{rem}+\tau^+_{GR}-(XT_{min}+\tau^+_{loc}+\max\{\tau^+_{Diff}+\tau^+_{GR},\tau^+_{GEQ}\}). \qquad (81)
\end{aligned}$$

The definition of T_{QS} implies that

$$T_{QS} \geq \tau^+_{rem}+\tau^+_{GR}+\tau^+_{TH}+X(\tau^+_{loc}+\max\{\tau^+_{Diff}+\tau^+_{GR},\tau^+_{GEQ}\}+\tau^+_{TH}-T_{min})-T^-_{first}. \qquad (82)$$

Combination of (81) and (82) yields

$$T^-_{first} > (X+1)(\tau^+_{loc}+\max\{\tau^+_{Diff}+\tau^+_{GR},\tau^+_{GEQ}\}+\tau^+_{TH}),$$

which is a contradiction to Constraint 6.

So let us assume

$$\begin{aligned}
&T_{QS}+\tau^+_{loc}+\max\{\tau^+_{Diff}+\tau^+_{GR},\tau^+_{GEQ}\} \geq -XT_{min}+\tau^+_{rem}+\tau^+_{GR} \Leftrightarrow \\
&T_{QS} \geq \tau^+_{rem}+\tau^+_{GR}-(XT_{min}+\tau^+_{loc}+\max\{\tau^+_{Diff}+\tau^+_{GR},\tau^+_{GEQ}\}). \qquad (83)
\end{aligned}$$

For the purpose of readability, we define variables $\ell_m, 0 \leq m \leq X+1$, as follows:

$$\begin{aligned}
&t_{r,k-2X-2} \leq \ell_0 := t_{first,k-X-1}+T_{QS} \\
&t_{r,k-2X-1} \leq \ell_1 := t_{first,k-X-1}+T_{QS}+\tau^+_{loc}+\max\{\tau^+_{Diff}+\tau^+_{GR},\tau^+_{GEQ}\}+\tau^+_{TH}
\end{aligned}$$

and further, by inductively applying Lemma 12 and (79) as in (80), for $2 \leq m \leq X+1$,

$$\begin{aligned}
t_{r,k-2X-2+m} &\leq \ell_m \\
&:= \max\left\{\begin{array}{l} \ell_{m-1}+\tau^+_{loc}+\max\{\tau^+_{Diff}+\tau^+_{GR},\tau^+_{GEQ}\}, \\ t_{first,k-X-1}-(X+1-m)T_{min}+\tau^+_{rem}+\tau^+_{GR} \end{array}\right\}+\tau^+_{TH}. \qquad (84)
\end{aligned}$$

We will show by induction on m, that

(a) The first term is always larger than the second, i.e., for all $2 \leq m \leq X+1$,

$$\ell_{m-1}+\tau^+_{loc}+\max\{\tau^+_{Diff}+\tau^+_{GR},\tau^+_{GEQ}\}+\tau^+_{TH} \geq t_{first,k-X-1}-(X+1-m)T_{min}+\tau^+_{rem}+\tau^+_{GR}+\tau^+_{TH}$$

(b) $t_{r,k-2X-2+m} \leq \ell_m = t_{first,k-X-1}+T_{QS}+m(\tau^+_{loc}+\max\{\tau^+_{Diff}+\tau^+_{GR},\tau^+_{GEQ}\}+\tau^+_{TH})$.

Begin ($m = 2$): By definition

$$\begin{aligned} t_{r,k-2X} &\leq \ell_2 \\ &= t_{first,k-X-1} + \max \left\{ \begin{array}{l} T_{QS} + 2(\tau^+_{loc} + \max\{\tau^+_{Diff} + \tau^+_{GR}, \tau^+_{GEQ}\} + \tau^+_{TH}), \\ -(X-1)T_{min} + \tau^+_{rem} + \tau^+_{GR} + \tau^+_{TH} \end{array} \right\}. \end{aligned} \tag{85}$$

We will show that (a) holds, i.e.,

$$T_{QS} + 2(\tau^+_{loc} + \max\{\tau^+_{Diff} + \tau^+_{GR}, \tau^+_{GEQ}\} + \tau^+_{TH}) \geq -(X-1)T_{min} + \tau^+_{rem} + \tau^+_{GR} + \tau^+_{TH}. \tag{86}$$

By assumption (83), we obtain

$$\begin{aligned} &T_{QS} + 2(\tau^+_{loc} + \max\{\tau^+_{Diff} + \tau^+_{GR}, \tau^+_{GEQ}\} + \tau^+_{TH}) \geq \\ &\tau^+_{rem} + \tau^+_{GR} - XT_{min} + \tau^+_{loc} + \max\{\tau^+_{Diff} + \tau^+_{GR}, \tau^+_{GEQ}\} + 2\tau^+_{TH}. \end{aligned}$$

By transitivity it remains to be proven that

$$\begin{aligned} &\tau^+_{rem} + \tau^+_{GR} - XT_{min} + \tau^+_{loc} + \max\{\tau^+_{Diff} + \tau^+_{GR}, \tau^+_{GEQ}\} + 2\tau^+_{TH} \geq \\ &-(X-1)T_{min} + \tau^+_{rem} + \tau^+_{GR} + \tau^+_{TH} \end{aligned} \tag{87}$$

By algebraic manipulations, (87) turns out to be equivalent to

$$T_{min} \leq \tau^+_{loc} + \max\{\tau^+_{Diff} + \tau^+_{GR}, \tau^+_{GEQ}\} + \tau^+_{TH}$$

which is fulfilled by the definition of T_{min}, proving (a) correct. We may thus rewrite (85) as

$$\begin{aligned} t_{r,k-2X} &\leq \ell_2 \\ &= t_{first,k-X-1} + T_{QS} + 2(\tau^+_{loc} + \max\{\tau^+_{Diff} + \tau^+_{GR}, \tau^+_{GEQ}\} + \tau^+_{TH}). \end{aligned}$$

proving (b) correct.
Step ($m \to m+1$): As induction hypothesis assume that (a) and (b) hold for m. By definition, for $3 \leq m+1 \leq X+1$

$$\begin{aligned} t_{r,k-2X-2+m+1} &\leq \ell_{m+1} \\ &= \max \left\{ \begin{array}{l} \ell_m + \tau^+_{loc} + \max\{\tau^+_{Diff} + \tau^+_{GR}, \tau^+_{GEQ}\}, \\ t_{first,k-X-1} - (X+1-(m+1))T_{min} + \tau^+_{rem} + \tau^+_{GR} \end{array} \right\} + \tau^+_{TH}. \end{aligned} \tag{88}$$

We will first show that (a) holds, where $A := \tau^+_{loc} + \max\{\tau^+_{Diff} + \tau^+_{GR}, \tau^+_{GEQ}\} + \tau^+_{TH}$ for the purpose of readability. Because of the induction hypothesis

$$\ell_{m-1} + A = t_{first,k-X-1} - (X+1-m)T_{min} + \tau^+_{rem} + \tau^+_{GR} + \tau^+_{TH}. \tag{89}$$

Since by definition $A \geq T_{min}$ we may add A to the left side and T_{min} to the right side of (89), obtaining

$$(\ell_{m-1} + A) + A \geq t_{first,k-X-1} - (X+1-(m+1))T_{min} + \tau^+_{rem} + \tau^+_{GR} + \tau^+_{TH}$$

Because of the $\max$ operator in (84),

$$\ell_m \geq \ell_{m-1} + A$$

and thus by transitivity

$$\ell_m + A \geq t_{first,k-X-1} - (X+1-(m+1))T_{min} + \tau^+_{rem} + \tau^+_{GR} + \tau^+_{TH}$$

proving (a) correct. We may thus rewrite (88) as

$$\begin{aligned} t_{r,k-2X-2+m+1} &\leq \ell_{m+1} \\ &= \ell_m + \tau^+_{loc} + \max\{\tau^+_{Diff} + \tau^+_{GR}, \tau^+_{GEQ}\} + \tau^+_{TH} \end{aligned}$$

and because of the induction hypothesis

$$= t_{first,k-X-1} + T_{QS} + (m+1)(\tau^+_{loc} + \max\{\tau^+_{Diff} + \tau^+_{GR}, \tau^+_{GEQ}\} + \tau^+_{TH})$$

proving (b) correct.

Thus for $m = X + 1$, it follows that

$$\begin{aligned} t_{r,k-X-1} &\leq \ell_{X+1} \\ &= t_{first,k-X-1} + T_{QS} + (X+1)(\tau^+_{loc} + \max\{\tau^+_{Diff} + \tau^+_{GR}, \tau^+_{GEQ}\} + \tau^+_{TH}). \end{aligned} \tag{90}$$

To complete the proof, we proceed from (74) and will show

$$t_{r,k-X-1} \leq t_{first,k-X-1} + T^-_{first} + T_{QS} \leq t_{first,k} + T_{QS}.$$

Due to (90), this holds if

$$\begin{aligned} &t_{first,k-X-1} + T_{QS} + (X+1)(\tau^+_{loc} + \max\{\tau^+_{Diff} + \tau^+_{GR}, \tau^+_{GEQ}\} + \tau^+_{TH}) \leq \\ &t_{first,k-X-1} + T^-_{first} + T_{QS}. \end{aligned} \tag{91}$$

Since the latter is equivalent to

$$T^-_{first} \geq (X+1)(\tau^+_{loc} + \max\{\tau^+_{Diff} + \tau^+_{GR}, \tau^+_{GEQ}\} + \tau^+_{TH}),$$

this holds due to Constraint 6.

Booting Simultaneity (BS)

Proof. The proof is by induction on $k \geq 1$.

Begin ($1 \leq k \leq X + 1$)**:** By Lemma 17 and Lemma 16,

$$t_{first,k} \geq t_{boot} + \min\{\tau^-_{GEQ}, \tau^-_{GR}\} + \tau^-_{TH} + (k-1)T_{min}$$

and

$$\begin{aligned} t_{last,k} &\leq t_{boot} + B + \max\{\tau^+_{GEQ}, \tau^+_{GR}\} + \tau^+_{TH} + (k-1)(\tau^+_{loc} + \max\{\tau^+_{Diff} + \tau^+_{GEQ}, \tau^+_{GR}\} + \tau^+_{TH}) \\ &\leq t_{boot} + B + \max\{\tau^+_{GEQ}, \tau^+_{GR}\} + \tau^+_{TH} + (k-1)T_P. \end{aligned}$$

Thus

$$\begin{aligned} t_{last,k} - t_{first,k} \leq & B + \max\{\tau^+_{GEQ}, \tau^+_{GR}\} + \tau^+_{TH} - \min\{\tau^-_{GEQ}, \tau^-_{GR}\} - \tau^-_{TH} + \\ & (k-1)(T_P - T_{min}) \\ = & T_{BS}(k). \end{aligned}$$

Step ($k \geq X + 1 \rightarrow k + 1$)**:** Assume the Lemma is true for k. From Lemma 8 and (P), it follows that

$$\begin{aligned} t_{first,k+1} - t_{first,k-X} &\geq (X+1)T_{min} \\ t_{last,k+1} - t_{last,k} &\leq T_P. \end{aligned}$$

In combination with the induction hypothesis this yields:

$$\begin{aligned} t_{last,k+1} - t_{first,k+1} &= (t_{last,k+1} - t_{last,k}) + (t_{last,k} - t_{first,k}) - (t_{first,k+1} - t_{first,k}) \\ &\leq T_P + T_{BS}(k) - T_{min} \\ &= T_{BS}(k+1). \end{aligned}$$

Having completed the proof of our major Theorem 4, we proceed with a sequence of technical lemmas devoted to tick sequences.

Lemma 18. *Given a sequence of consecutive events $t_i, i > 0$, where $t_i < t_{i+1}$. Assume that the minimum inter-event times are restricted by*

$$\forall i > 0 : t_{i+1} - t_i \geq T_{min} \tag{92}$$

$$\forall i > 0 : t_{i+X+1} - t_i \geq T^-_{first} \geq (X+1)T_{min}. \tag{93}$$

Then, for the number of events N that can occur in the interval $I = (t_j, \tau)$, for some $j > 0$ and $\tau > t_j$, it holds that

$$N \leq \left(\left\lceil \frac{\tau - t_j}{T^-_{first}} \right\rceil - 1 \right)(X+1) + \left\lceil \frac{\tau - t_j - \left(\left\lceil \frac{\tau - t_j}{T^-_{first}} \right\rceil - 1 \right) T^-_{first}}{T_{min}} \right\rceil - 1.$$

Proof. Abbreviate $k = \left\lceil \frac{\tau - t_j}{T^-_{first}} \right\rceil - 1$ and $d = \left\lceil \frac{\tau - t_j - \left(\left\lceil \frac{\tau - t_j}{T^-_{first}} \right\rceil - 1 \right) T^-_{first}}{T_{min}} \right\rceil = \left\lceil \frac{\tau - t_j - kT^-_{first}}{T_{min}} \right\rceil$, then we can write $N \leq k(X+1) + d - 1$ Assume by contradiction that

$$N \geq k(X+1) + d. \tag{94}$$

By (94) and the definition of N,

$$t_{j+k(X+1)+d} \leq t_{j+N} < \tau. \tag{95}$$

Thus $t_{j+N} - t_j \geq t_{j+k(X+1)+d} - t_{j+k(X+1)} + t_{j+k(X+1)} - t_j$. By a generalized form of (92) and (93), $t_{j+k(X+1)+d} - t_{j+k(X+1)} \geq dT_{min}$ and $t_{j+k(X+1)} - t_j \geq kT^-_{first}$. Combining these results yields

$$\begin{aligned} t_{j+N} - t_j &\geq dT_{min} + kT^-_{first} \\ &= \left\lceil \frac{\tau - t_j - kT^-_{first}}{T_{min}} \right\rceil T_{min} + kT^-_{first} \\ &\geq \tau - t_j - kT^-_{first} + kT^-_{first} \\ &= \tau - t_j \end{aligned}$$

i.e., $t_{j+N} \geq \tau$, a contradiction to (95).

Lemma 19. *Given a sequence of consecutive events $t_i, i \geq 0$, where $t_i < t_{i+1}$. Assume that the minimum inter-event times are restricted by*

$$\forall i > 0 : t_{i+1} - t_i \geq T_{min} \tag{96}$$

$$\forall i > 0 : t_{i+X+1} - t_i \geq T^-_{first} \geq (X+1)T_{min}. \tag{97}$$

Then, for the number of events N that can occur in the interval $I = (t_j, \tau]$, with $j > 0$ and $\tau > t_j$, it holds that

$$N \leq \left\lfloor \frac{\tau - t_j}{T^-_{first}} \right\rfloor (X+1) + \left\lceil \frac{\tau - t_j - \left\lfloor \frac{\tau - t_j}{T^-_{first}} \right\rfloor T^-_{first}}{T_{min}} \right\rceil.$$

Proof. Abbreviating $k = \left\lfloor \frac{\tau - t_j}{T^-_{first}} \right\rfloor$ and $d = \left\lceil \frac{\tau - t_j - \left\lfloor \frac{\tau - t_j}{T^-_{first}} \right\rfloor T^-_{first}}{T_{min}} \right\rceil = \frac{\tau - t_j - kT^-_{first}}{T_{min}}$, we can write $N \leq k(X+1) + d$. Assume by contradiction that

$$N > k(X+1) + d. \tag{98}$$

Then,

$$t_{j+N} \leq \tau. \tag{99}$$

By a generalized form of (96) and (97), $t_{j+N} - t_j > t_{j+k(X+1)+d} - t_{j+k(X+1)} + t_{j+k(X+1)} - t_j \geq dT_{min} + kT^-_{first}$. This yields

$$\begin{aligned} t_{j+N} - t_j &> dT_{min} + kT^-_{first} && (100)\\ &= \frac{\tau - t_j - kT^-_{first}}{T_{min}} T_{min} + kT^-_{first} && (101)\\ &\geq \tau - t_j - kT^-_{first} + kT^-_{first} && (102) \end{aligned}$$

i.e., $t_{j+N} > \tau$, a contradiction to (99).

The following Lemma 20 is a weaker form of Lemma 19, where the beginning of the interval I not necessarily coincides with some event occurrence time t_j, i.e., I is not "aligned".

Lemma 20. *Given a sequence of consecutive events $t_i, i \geq 0$, where $t_i < t_{i+1}$. Assume that the minimum inter-event times are restricted by*

$$\begin{aligned} \forall i > 0 : t_{i+1} - t_i &\geq T_{min} && (103)\\ \forall i > 0 : t_{i+X+1} - t_i &\geq T^-_{first} \geq (X+1)T_{min}. && (104) \end{aligned}$$

Then, for the number of events N that can occur in the interval $I = (t, \tau]$, for some $t \geq 0$ and $\tau > t$, it holds that

$$N \leq \left\lceil \frac{\tau - t}{T^-_{first}} \right\rceil (X+1) + \left\lceil \frac{\tau - t - \left\lceil \frac{\tau - t}{T^-_{first}} \right\rceil T^-_{first}}{T_{min}} \right\rceil + 1.$$

Proof. Abbreviating $k = \left\lceil \frac{\tau - t}{T^-_{first}} \right\rceil$ and $d = \left\lceil \frac{\tau - t - \left\lceil \frac{\tau - t}{T^-_{first}} \right\rceil T^-_{first}}{T_{min}} \right\rceil = \left\lceil \frac{\tau - t - kT^-_{first}}{T_{min}} \right\rceil$, we can write $N \leq k(X+1) + d + 1$. We distinguish between two cases: (i) $N \geq 1$, i.e., there $\exists k : t_{k+1} \in I$ and (ii) $N = 0$.
ad (i): Assume by contradiction that

$$N > k(X+1) + d + 1. \tag{105}$$

By the definition of N and assumption of $N \geq 1$, there must be a j, s.t.

$$\begin{aligned} t_{j+1} &> t && (106)\\ t_{j+N} &\leq \tau. && (107) \end{aligned}$$

By (107), (106), (103) and (104),

$$\begin{aligned} t_{j+N} - t &> t_{j+N} - t_{j+1} \\ &\geq t_{j+k(X+1)+d+1} - t_{j+k(X+1)+1} + t_{j+k(X+1)+1} - t_{j+1} \\ &\geq dT_{min} + kT^{-}_{first} \\ &= \left\lceil \frac{\tau - t - kT^{-}_{first}}{T_{min}} \right\rceil T_{min} + kT^{-}_{first} \\ &= \tau - t \end{aligned} \tag{108}$$

Clearly, (108) contradicts (107).

ad (ii): Obviously $N = 0 \leq \left\lceil \frac{\tau - t}{T^{-}_{first}} \right\rceil (X+1) + \left\lceil \frac{\tau - t - \left\lceil \frac{\tau - t}{T^{-}_{first}} \right\rceil T^{-}_{first}}{T_{min}} \right\rceil + 1$ holds for $\tau > t$.

Lemma 21. (Last Progress) *If the last correct node sends tick $k \geq X+1$ at $t_{last,k}$, then the last correct node sends tick $k+N$, $N \geq 0$ by $t_{last,k+N}$, with*

$$t_{last,k+N} - t_{last,k} \leq NT_P.$$

Proof. The proof is by induction on N:
Begin ($N = 0$): Clearly $t_{last,k} - t_{last,k} \leq 0$ is true.
Step ($N > 0$): As the induction hypothesis, assume that the Lemma is true for $N-1$. Then, by applying (P),

$$\begin{aligned} t_{last,k+N} - t_{last,k} &= (t_{last,k+N} - t_{last,k+N-1}) + (t_{last,k+N-1} - t_{last,k}) \\ &\leq T_P + (N-1)T_P \\ &= NT_P. \end{aligned}$$

Lemma 22. (Progress by (QS)) *If p is a correct node which sends tick $k \geq 2X+2$ at $t_{p,k}$, then p sends tick $k+N$, $N \geq -X-1$, at $t_{p,k+N}$ with*

$$t_{p,k+N} - t_{p,k} \leq (N+X+1)T_P + T_{QS}.$$

Proof. With Lemma 21 and (QS), it follows that

$$\begin{aligned} t_{p,k+N} - t_{p,k} &\leq t_{last,k+N} - t_{first,k} \\ &= (t_{last,k+N} - t_{last,k-X-1}) + (t_{last,k-X-1} - t_{first,k}) \\ &\leq (N+X+1)T_P + T_{QS}. \end{aligned}$$

Lemma 23. (Progress by (BS)) *If p is a correct node which sends tick $k \geq X+1$ at $t_{p,k}$, then p sends tick $k+N$, $N \geq 1$, at $t_{p,k+N}$ with*

$$t_{p,k+N} - t_{p,k} \leq NT_P + T_{BS}(k).$$

Proof. With Lemma 21 and (BS) it follows that

$$\begin{aligned} t_{p,k+N} - t_{p,k} &\leq t_{last,k+N} - t_{first,k} \\ &= (t_{last,k+N} - t_{last,k}) + (t_{last,k} - t_{first,k}) \\ &\leq NT_P + T_{BS}(k). \end{aligned}$$

Theorem 5 states a bound π on the precision of our algorithm, i.e., for every pair of correct nodes $p, q \in C : \forall t : |b_q(t) - b_p(t)| \leq \pi$.

Theorem 5. (Precision).

$$\pi := \max \left\{ \begin{array}{l} \left\lfloor \frac{T_{QS}}{T^-_{first}} \right\rfloor (X+1) + \left\lceil \frac{T_{QS} - \left\lfloor \frac{T_{QS}}{T^-_{first}} \right\rfloor T^-_{first}}{T_{min}} \right\rceil + X + 1, \\ \left\lceil \frac{T_{QS}}{T^-_{first}} \right\rceil (X+1) + \left\lceil \frac{T_{QS} - \left(\left\lceil \frac{T_{QS}}{T^-_{first}} \right\rceil - 1 \right) T^-_{first}}{T_{min}} \right\rceil \end{array} \right\}$$

is a valid precision-bound.

Proof. Let p, q be two distinct correct nodes. Clearly for all t, $|b_q(t) - b_p(t)| \leq b^{max}(t) - b^{min}(t)$, where $b^{max}(t)$ and $b^{min}(t)$ denote the number of clock ticks generated by the first and last correct node in the system a time t. The following proof establishes a bound on the latter term. For this purpose we distinguish between three cases for t: (i) $t \in [0, t_{last,1})$, (ii) $t = t_{last,k}$ for some $k \geq 1$ and (iii) $t \in (t_{last,k}, t_{last,k+1})$ for some $k \geq 1$.
ad (i):

$$\begin{aligned} & b^{max}(t) - b^{min}(t) \leq \\ & b^{max}(\overrightarrow{t_{last,1}}) - b^{min}(0) = \\ & b^{max}(\overrightarrow{t_{last,1}}) = \\ & b^{max}(\overrightarrow{t_{last,1}}) - b^{max}(t_{first,X+2}) + b^{max}(t_{first,X+2}) \leq \end{aligned}$$

by Lemma 18

$$\left\lceil \frac{t_{last,1} - t_{first,X+2}}{T^-_{first}} \right\rceil (X+1) + \left\lceil \frac{t_{last,1} - t_{first,X+2} - \left(\left\lceil \frac{t_{last,1} - t_{first,X+2}}{T^-_{first}} \right\rceil - 1 \right) T^-_{first}}{T_{min}} \right\rceil \leq$$

by (QS)

$$\left\lceil \frac{T_{QS}}{T^-_{first}} \right\rceil (X+1) + \left\lceil \frac{T_{QS} - \left(\left\lceil \frac{T_{QS}}{T^-_{first}} \right\rceil - 1 \right) T^-_{first}}{T_{min}} \right\rceil$$

ad (ii):

$$\begin{aligned} & b^{max}(t_{last,k}) - b^{min}(t_{last,k}) = \\ & b^{max}(t_{last,k}) - k = \\ & b^{max}(t_{last,k}) - b^{max}(t_{first,k+X+1}) + b^{max}(t_{first,k+X+1}) - k = \\ & b^{max}(t_{last,k}) - b^{max}(t_{first,k+X+1}) + X + 1 = \end{aligned}$$

by Lemma 19

$$\left\lfloor \frac{t_{last,k} - t_{first,k+X+1}}{T^-_{first}} \right\rfloor (X+1) - \left\lceil \frac{t_{last,k} - t_{first,k+X+1} - \left\lfloor \frac{t_{last,k} - t_{first,k+X+1}}{T^-_{first}} \right\rfloor T^-_{first}}{T_{min}} \right\rceil + X + 1$$

by (QS)

$$\left\lfloor \frac{T_{QS}}{T^-_{first}} \right\rfloor (X+1) + \left\lceil \frac{T_{QS} - \left\lfloor \frac{T_{QS}}{T^-_{first}} \right\rfloor T^-_{first}}{T_{min}} \right\rceil + X + 1$$

ad (iii):

$$\begin{aligned}
&b^{max}(t) - b^{min}(t) \leq \\
&b^{max}(t^{\rightarrow}_{last,k+1}) - k = \\
&b^{max}(t^{\rightarrow}_{last,k+1}) - b^{max}(t_{first,k+X+2}) + b^{max}(t_{first,k+X+2}) - k \leq
\end{aligned}$$

by Lemma 18

$$\left\lceil \frac{t_{last,k+1} - t_{first,k+X+2}}{T^-_{first}} \right\rceil (X+1) + \left\lceil \frac{t_{last,k+1} - t_{first,k+X+2} - \left(\left\lceil \frac{t_{last,k+1} - t_{first,k+X+2}}{T^-_{first}} \right\rceil - 1 \right) T^-_{first}}{T_{min}} \right\rceil \leq$$

by (QS)

$$\left\lceil \frac{T_{QS}}{T^-_{first}} \right\rceil (X+1) + \left\lceil \frac{T_{QS} - \left(\left\lceil \frac{T_{QS}}{T^-_{first}} \right\rceil - 1 \right) T^-_{first}}{T_{min}} \right\rceil$$

Combination of (i), (ii) and (iii) allows us to state the precision-bound π for arbitrary t:

$$\pi := \max \left\{ \begin{array}{l} \left\lfloor \frac{T_{QS}}{T^-_{first}} \right\rfloor (X+1) + \left\lceil \frac{T_{QS} - \left\lfloor \frac{T_{QS}}{T^-_{first}} \right\rfloor T^-_{first}}{T_{min}} \right\rceil + X + 1, \\ \left\lceil \frac{T_{QS}}{T^-_{first}} \right\rceil (X+1) + \left\lceil \frac{T_{QS} - \left(\left\lceil \frac{T_{QS}}{T^-_{first}} \right\rceil - 1 \right) T^-_{first}}{T_{min}} \right\rceil \end{array} \right\}$$

Theorem 6 (Accuracy) can be used to bound the number of tick messages generated locally at a correct node p during some real time interval $(t_1, t_2]$, i.e., allows to make statements about the local clock frequency. For example, it reveals that the long-term frequency is within $\left[1/T_P, (X+1)/T^-_{first}\right]$.

Theorem 6. (Accuracy). *Given t_1 and t_2 with $t_2 > t_1 \geq t_{p,X+1}$, the accuracy $b_p(t_2) - b_p(t_1)$ of any correct node p is bounded by*

$$\begin{aligned}
&\max \left\{ 0, \left\lfloor \frac{t_2 - t_1 - \max \left\{ \begin{array}{l} T_{BS}(2X+1), \\ \min \{T_{QS} + (X+1)T_P, T_{BS}(k) \mid k \geq 2X+2\} \end{array} \right\}}{T_P} \right\rfloor + 1 \right\} \\
&\leq b_p(t_2) - b_p(t_1) \leq \\
&\leq \left\lceil \frac{t_2 - t_1}{T^-_{first}} \right\rceil (X+1) - \left\lceil \frac{t_2 - t_1 - \left\lceil \frac{t_2 - t_1}{T^-_{first}} \right\rceil T^-_{first}}{T_{min}} \right\rceil + 1 + \pi
\end{aligned}$$

Proof of the upper bound

Proof. Since

$$\begin{aligned} b_p(t_2) - b_p(t_1) &\leq b^{max}(t_2) - b^{min}(t_1) \\ &\leq (b^{max}(t_2) - b^{max}(t_1)) + \left(b^{max}(t_1) - b^{min}(t_1)\right), \end{aligned} \tag{109}$$

and both terms of (109) can be bounded by applying Lemma 20 and Theorem 5, namely,

$$b^{max}(t_2) - b^{max}(t_1) \leq \left\lceil \frac{t_2 - t_1}{T^-_{first}} \right\rceil (X+1) - \left\lceil \frac{t_2 - t_1 - \left\lceil \frac{t_2 - t_1}{T^-_{first}} \right\rceil T^-_{first}}{T_{min}} \right\rceil + 1 \tag{110}$$

$$b^{max}(t_1) - b^{min}(t_1) \leq \pi, \tag{111}$$

we obtain

$$b_p(t_2) - b_p(t_1) \leq \left\lceil \frac{t_2 - t_1}{T^-_{first}} \right\rceil (X+1) - \left\lceil \frac{t_2 - t_1 - \left\lceil \frac{t_2 - t_1}{T^-_{first}} \right\rceil T^-_{first}}{T_{min}} \right\rceil + 1 + \pi. \tag{112}$$

Proof of the lower bound

Proof. Let $k = b_p(t_1) \geq X + 1$ and $N \geq 0$, s.t., $k + N = b_p(t_2)$. Clearly, such k and N exist, since p must have sent tick $X + 1$ by t_1. By the definition of k and N,

$$t_{p,k} \leq t_1 \tag{113}$$

$$t_{p,k+N+1} > t_2. \tag{114}$$

For $k \geq 2X + 2$, we apply Lemma 22 together with (113) and (114), yielding

$$\begin{aligned} t_2 - t_1 &< t_{p,k+N+1} - t_{p,k} \\ &\leq (N + X + 2)T_P + T_{QS} \Rightarrow \\ N &> \frac{t_2 - t_1 - T_{QS} - (X+2)T_P}{T_P} \Rightarrow \\ N &\geq \left\lfloor \frac{t_2 - t_1 - T_{QS} - (X+1)T_P}{T_P} \right\rfloor + 1. \end{aligned} \tag{115}$$

For $k \geq X + 1$, we apply Lemma 23, obtaining

$$\begin{aligned} t_2 - t_1 &< (N+1)T_P + T_{BS}(k) \Rightarrow \\ N &> \frac{t_2 - t_1 - T_{BS}(k) - T_P}{T_P} \Rightarrow \\ N &\geq \left\lfloor \frac{t_2 - t_1 - T_{BS}(k)}{T_P} \right\rfloor + 1. \end{aligned} \tag{116}$$

Combining the bounds (115), (116) and the trivial bound 0 yields (with $b_p(t_1) = k$)

$$b_p(t_2) - b_p(t_1) = N \geq \begin{cases} \max\left\{0, \left\lfloor \frac{t_2 - t_1 - T_{BS}(k)}{T_P} \right\rfloor + 1\right\} & \text{if } k \leq 2X+1, \\ \max\left\{0, \left\lfloor \frac{t_2 - t_1 - \min\left\{T_{BS}(k), T_{QS} + (X+1)T_P\right\}}{T_P} \right\rfloor + 1\right\} & \text{if } k \geq 2X+2. \end{cases} \tag{117}$$

In case k is not known, a valid bound is the minimum of all lower bounds, i.e.,

$$b_p(t_2) - b_p(t_1) \geq \max\left\{0, \left\lfloor \frac{t_2 - t_1 - \max\left\{\begin{array}{l} T_{BS}(2X+1), \\ \min\{T_{QS} + (X+1)T_P, T_{BS}(k) \mid k \geq 2X+2\} \end{array}\right\}}{T_P} \right\rfloor + 1 \right\} \tag{118}$$

Note the term $\min\{T_{BS}(k), T_{QS} + (X+1)T_P\}$ for $k \geq 2X+2$ in both (117) and (118). It accounts for the fact that correct nodes may be synchronized very tightly after booting (within $T_{BS}(k)$), such that $T_{QS} + (X+1)T_P$ would be too conservative. Clearly, however, when $T_{min} < T_P$, which is typically the case in real systems, the initial synchrony from booting cannot be maintained: In systems with $T_{min} < T_P$, we have $\forall k \geq 1 : T_{BS}(k+1) > T_{BS}(k)$ as well as $\lim_{k\to\infty} T_{BS}(k) = \infty$. Thus, at some k_0, $\forall k \geq k_0 : T_{QS} + (X+1)T_P < T_{BS}(k)$, i.e., the constant bound from (QS) will be tighter — preventing the nodes from drifting apart without bound.

Our final two theorems establish that the local and remote pipeline size is indeed bounded. We first establish a bound on the maximum time tick $k-2$ can exist in the system before it is eliminated by the Diff-Gate at all correct node's pipes.

Lemma 24. *If the first correct node has sent tick $k \geq X+2$ by time t, then every correct node $p \in C$ has removed tick $k-X-2$ from all its pipepairs $(p,q)_{GR}$ corresponding to correct nodes $q \in C$ by time $t + T_{del,gr}$, with*

$$T_{del,gr} := T_{QS} + \max\{\tau^+_{loc}, \tau^+_{rem}\} + \tau^+_{Diff}.$$

Equivalently,

$$d^{GR}_{p,q}(t + T_{del,gr}) \geq b^{max}(t) - X - 2.$$

Proof. Consider a pair of pipes $(p,q)_{GR}$, located at $p \in C$, corresponding to a different $q \in C$. Furthermore, assume that $b^{max}(t) = k$ holds at time t. We are interested in how much later tick $k-X-2$ is removed from this pipe-pair.

Clearly, $t_{first,k} \leq t$ and, by (QS),

$$\begin{aligned} t_{p,k-X-1} &\leq t_{last,k-X-1} \\ &\leq t_{first,k} + T_{QS} \text{ and} && (119) \\ t_{q,k-X-1} &\leq t_{first,k} + T_{QS}. && (120) \end{aligned}$$

We may now apply Lemma 9 in combination with (119) and (120), which ensures that tick $k-X-2$ is removed from the pipepair $(p,q)_{GR}$ by time $t_{rmv,k-X-2}$, with

$$\begin{aligned} t_{rmv,k-X-2} &\leq \max\{t_{p,k-X-1} + \tau^+_{loc}, t_{q,k-X-1} + \tau^+_{rem}\} + \tau^+_{Diff} \\ &\leq t_{first,k} + T_{QS} + \max\{\tau^+_{loc}, \tau^+_{rem}\} + \tau^+_{Diff} \\ &= t_{first,k} + T_{del,gr} \\ &\leq t + T_{del,gr}. \end{aligned}$$

Lemma 25. *If the first correct node has sent tick $k \geq 2X+2$ by time t, then every correct node $p \in C$ has removed tick $k-2$ from all its pipepairs, $(p,q)_{GEQ}$, corresponding to correct nodes $q \in C$ by time $t + T_{del,geq}$, with*

$$T_{del,geq} := T_{QS} + \max\{XT_P + \tau^+_{loc}, \tau^+_{rem}\} + \tau^+_{Diff}.$$

Equivalently,

$$d^{GEQ}_{p,q}(t + T_{del,geq}) \geq b^{max}(t) - 2.$$

Proof. Consider a pair of pipes $(p,q)_{GEQ}$, located at $p \in C$, corresponding to a different $q \in C$. Furthermore, assume that $b^{max}(t) = k$ holds at time t. We are interested in how much later tick $k-2$ is removed from this pipepair:

Clearly, $t_{first,k} \leq t$ and, by (QS) and Lemma 22,

$$\begin{aligned} t_{p,k-1} &\leq t_{last,k-1} \\ &\leq t_{first,k} + XT_P + T_{QS} \text{ and} && (121) \\ t_{q,k-X-1} &\leq t_{first,k} + T_{QS}. && (122) \end{aligned}$$

Since $k-1 \geq X+1$, we may now apply Lemma 11 in combination with (121) and (122), which ensures that tick $k-2$ is removed from the pipepair $(p,q)_{GEQ}$ by time $t_{rmv,k-2}$, with

$$\begin{aligned} t_{rmv,k-2} &\leq \max\{t_{p,k-1} + \tau^+_{loc}, t_{q,k-X-1} + \tau^+_{rem}\} + \tau^+_{Diff} \\ &\leq t_{first,k} + T_{QS} + \max\{XT_P + \tau^+_{loc}, \tau^+_{rem}\} + \tau^+_{Diff} \\ &= t_{first,k} + T_{del,geq} \\ &\leq t + T_{del,geq}. \end{aligned}$$

Theorem 7. (Local pipeline size bound, GR). *For every pair of distinct correct nodes $p,q \in C$, $s^{loc,gr}_{p,q}(t)$ is bounded by $S_{loc,gr}$, with*

$$S_{loc,gr} := \max \left\{ \begin{array}{l} \left\lceil \frac{T_{del,gr}-\tau^-_{loc}}{T^-_{first}} \right\rceil (X+1) + \left\lceil \frac{T_{del,gr}-\tau^-_{loc}-\left\lceil \frac{T_{del,gr}-\tau^-_{loc}}{T^-_{first}} \right\rceil T^-_{first}}{T_{min}} \right\rceil + X + 3, \\ \left\lfloor \frac{T_{del,gr}-\tau^-_{loc}}{T^-_{first}} \right\rfloor (X+1) + \left\lceil \frac{T_{del,gr}-\tau^-_{loc}-\left\lfloor \frac{T_{del,gr}-\tau^-_{loc}}{T^-_{first}} \right\rfloor T^-_{first}}{T_{min}} \right\rceil + X + 3 \end{array} \right\}.$$

Proof. Choose two arbitrary distinct correct node p,q and consider $s^{loc,gr}_{p,q}(t)$. We distinguish between two cases for t: (i) $t \geq t_{first,X+2} + T_{del,gr}$ and (ii) $t < t_{first,X+2} + T_{del,gr}$.

ad (i):

$$\begin{aligned} s^{loc,gr}_{p,q}(t) &= r^{loc,GR}_{p,q}(t) - d^{GR}_{p,q}(t) \\ &\leq b_p(t-\tau^-_{loc}) - d^{GR}_{p,q}(t) \\ &\leq b^{max}(t-\tau^-_{loc}) - d^{GR}_{p,q}(t) \\ &= \left(b^{max}(t-\tau^-_{loc}) - b^{max}(t-T_{del,gr})\right) + \left(b^{max}(t-T_{del,gr}) - d^{GR}_{p,q}(t)\right) \\ &\leq \left\lceil \frac{T_{del,gr}-\tau^-_{loc}}{T^-_{first}} \right\rceil (X+1) + \left\lceil \frac{T_{del,gr}-\tau^-_{loc}-\left\lceil \frac{T_{del,gr}-\tau^-_{loc}}{T^-_{first}} \right\rceil T^-_{first}}{T_{min}} \right\rceil + X + 3 && (123) \\ &\leq S_{loc,gr}, \end{aligned}$$

where (123) follows from applying Lemma 20 and Lemma 24.

ad (ii): Since $d_{p,q}^{GR}(t) \geq -1$ must always hold, we obtain

$$\begin{aligned}
s_{p,q}^{loc,gr}(t) &= r_{p,q}^{loc,GR}(t) - d_{p,q}^{GR}(t) \\
&\leq b_p(t - \tau_{loc}^-) - d_{p,q}^{GR}(t) \\
&\leq b^{max}(t - \tau_{loc}^-) + 1 \\
&\leq b^{max}(t_{first,X+2} + T_{del,gr} - \tau_{loc}^-) + 1 \\
&\leq \left(b^{max}(t_{first,X+2} + T_{del,gr} - \tau_{loc}^-) - b^{max}(t_{first,X+2})\right) + b^{max}(t_{first,X+2}) + 1 && (124) \\
&\leq \left\lfloor \frac{T_{del,gr} - \tau_{loc}^-}{T_{first}^-} \right\rfloor (X+1) + \left\lceil \frac{T_{del,gr} - \tau_{loc}^- - \left\lfloor \frac{T_{del,gr} - \tau_{loc}^-}{T_{first}^-} \right\rfloor T_{first}^-}{T_{min}} \right\rceil + X + 3 && (125) \\
&\leq S_{loc,gr},
\end{aligned}$$

where (125) follows from Lemma 19.

Theorem 8. (Local pipeline size bound, GEQ). *For every pair of distinct correct nodes $p, q \in C$, $s_{p,q}^{loc,geq}(t)$ is bounded by $S_{loc,geq}$, with*

$$S_{loc,geq} := \max \left\{ \begin{array}{l} \left\lceil \frac{T_{del,geq} - \tau_{loc}^-}{T_{first}^-} \right\rceil (X+1) + \left\lceil \frac{T_{del,geq} - \tau_{loc}^- - \left\lfloor \frac{T_{del,geq} - \tau_{loc}^-}{T_{first}^-} \right\rfloor T_{first}^-}{T_{min}} \right\rceil + 3, \\ \left\lfloor \frac{T_{del,geq} - \tau_{loc}^-}{T_{first}^-} \right\rfloor (X+1) + \left\lceil \frac{T_{del,geq} - \tau_{loc}^- - \left\lfloor \frac{T_{del,geq} - \tau_{loc}^-}{T_{first}^-} \right\rfloor T_{first}^-}{T_{min}} \right\rceil + 2X + 3 \end{array} \right\}.$$

Proof. Choose two arbitrary distinct correct node p, q and consider $s_{p,q}^{loc,geq}(t)$. We distinguish between two cases for t: (i) $t \geq t_{first,2X+2} + T_{del,geq}$ and (ii) $t < t_{first,2X+2} + T_{del,geq}$.

ad (i):

$$\begin{aligned}
s_{p,q}^{loc,geq}(t) &= r_{p,q}^{loc,GEQ}(t) - d_{p,q}^{GEQ}(t) \\
&\leq b_p(t - \tau_{loc}^-) - d_{p,q}^{GEQ}(t) \\
&\leq b^{max}(t - \tau_{loc}^-) - d_{p,q}^{GEQ}(t) \\
&= \left(b^{max}(t - \tau_{loc}^-) - b^{max}(t - T_{del,geq})\right) + \left(b^{max}(t - T_{del,geq}) - d_{p,q}^{GEQ}(t)\right) \\
&\leq \left\lceil \frac{T_{del,geq} - \tau_{loc}^-}{T_{first}^-} \right\rceil (X+1) + \left\lceil \frac{T_{del,geq} - \tau_{loc}^- - \left\lceil \frac{T_{del,geq} - \tau_{loc}^-}{T_{first}^-} \right\rceil T_{first}^-}{T_{min}} \right\rceil + 3 && (126) \\
&\leq S_{loc,geq},
\end{aligned}$$

where (126) follows from applying Lemma 20 and Lemma 25.

ad (ii): Since $d_{p,q}^{GEQ}(t) \geq -1$ must always hold, we obtain

$$\begin{aligned}
s_{p,q}^{loc,geq}(t) &= r_{p,q}^{loc,GEQ}(t) - d_{p,q}^{GEQ}(t) \\
&\leq b_p(t - \tau_{loc}^-) - d_{p,q}^{GEQ}(t) \\
&\leq b^{max}(t - \tau_{loc}^-) + 1 \\
&\leq b^{max}(t_{first,2X+2} + T_{del,geq} - \tau_{loc}^-) + 1 \\
&\leq \left(b^{max}(t_{first,2X+2} + T_{del,geq} - \tau_{loc}^-) - b^{max}(t_{first,2X+2})\right) + b^{max}(t_{first,2X+2}) + 1 \qquad (127) \\
&\leq \left\lfloor \frac{T_{del,geq} - \tau_{loc}^-}{T_{first}^-} \right\rfloor (X+1) + \left\lceil \frac{T_{del,geq} - \tau_{loc}^- - \left\lfloor \frac{T_{del,geq} - \tau_{loc}^-}{T_{first}^-} \right\rfloor T_{first}^-}{T_{min}} \right\rceil + 2X + 3 \qquad (128) \\
&\leq S_{loc,geq},
\end{aligned}$$

where (128) follows from Lemma 19.

Theorem 9. (Remote pipeline size bound, GR). *For every pair of distinct correct nodes $p, q \in C$, $s_{p,q}^{rem,gr}(t)$ is bounded by $S_{rem,gr}$, with*

$$S_{rem,gr} := \max \left\{ \begin{array}{l} \left\lceil \frac{T_{del,gr} - \tau_{rem}^-}{T_{first}^-} \right\rceil (X+1) + \left\lceil \frac{T_{del,gr} - \tau_{rem}^- - \left\lfloor \frac{T_{del,gr} - \tau_{rem}^-}{T_{first}^-} \right\rfloor T_{first}^-}{T_{min}} \right\rceil + X + 3, \\ \left\lfloor \frac{T_{del,gr} - \tau_{rem}^-}{T_{first}^-} \right\rfloor (X+1) + \left\lceil \frac{T_{del,gr} - \tau_{rem}^- - \left\lfloor \frac{T_{del,gr} - \tau_{rem}^-}{T_{first}^-} \right\rfloor T_{first}^-}{T_{min}} \right\rceil + X + 3 \end{array} \right\}.$$

Proof. Choose two arbitrary distinct correct node p, q and consider $s_{p,q}^{rem,gr}(t)$. We distinguish between two cases for t: (i) $t \geq t_{first,X+2} + T_{del,gr}$ and (ii) $t < t_{first,X+2} + T_{del,gr}$.

ad (i):

$$\begin{aligned}
s_{p,q}^{rem,gr}(t) &= r_{p,q}^{rem,GR}(t) - d_{p,q}^{GR}(t) \\
&\leq b_p(t - \tau_{rem}^-) - d_{p,q}^{GR}(t) \\
&\leq b^{max}(t - \tau_{rem}^-) - d_{p,q}^{GR}(t) \\
&= \left(b^{max}(t - \tau_{rem}^-) - b^{max}(t - T_{del,gr})\right) + \left(b^{max}(t - T_{del,gr}) - d_{p,q}^{GR}(t)\right) \\
&\leq \left\lceil \frac{T_{del,gr} - \tau_{rem}^-}{T_{first}^-} \right\rceil (X+1) + \left\lceil \frac{T_{del,gr} - \tau_{rem}^- - \left\lceil \frac{T_{del,gr} - \tau_{rem}^-}{T_{first}^-} \right\rceil T_{first}^-}{T_{min}} \right\rceil + X + 3 \qquad (129) \\
&\leq S_{rem,gr},
\end{aligned}$$

where (129) follows from applying Lemma 20 and Lemma 24.

ad (ii): Since $d_{p,q}^{GR}(t) \geq -1$ must always hold, we obtain

$$\begin{aligned}
s_{p,q}^{rem,gr}(t) &= r_{p,q}^{rem,GR}(t) - d_{p,q}^{GR}(t) \\
&\leq b_p(t - \tau_{rem}^-) - d_{p,q}^{GR}(t) \\
&\leq b^{max}(t - \tau_{rem}^-) + 1 \\
&\leq b^{max}(t_{first,X+2} + T_{del,gr} - \tau_{rem}^-) + 1 \\
&\leq \left(b^{max}(t_{first,X+2} + T_{del,gr} - \tau_{rem}^-) - b^{max}(t_{first,X+2})\right) + b^{max}(t_{first,X+2}) + 1 && (130) \\
&\leq \left\lfloor \frac{T_{del,gr} - \tau_{rem}^-}{T_{first}^-} \right\rfloor (X+1) + \left\lceil \frac{T_{del,gr} - \tau_{rem}^- - \left\lfloor \frac{T_{del,gr} - \tau_{rem}^-}{T_{first}^-} \right\rfloor T_{first}^-}{T_{min}} \right\rceil + X + 3 && (131) \\
&\leq S_{rem,gr},
\end{aligned}$$

where (131) follows from Lemma 19.

Theorem 10. (Remote pipeline size bound, GEQ). *For every pair of distinct correct nodes $p, q \in C$, $s_{p,q}^{rem,geq}(t)$ is bounded by $S_{rem,geq}$, with*

$$S_{rem,geq} := \max \left\{ \begin{array}{l} \left\lceil \frac{T_{del,geq} - \tau_{rem}^-}{T_{first}^-} \right\rceil (X+1) + \left\lceil \frac{T_{del,geq} - \tau_{rem}^- - \left\lfloor \frac{T_{del,geq} - \tau_{rem}^-}{T_{first}^-} \right\rfloor T_{first}^-}{T_{min}} \right\rceil + X + 3, \\ \left\lfloor \frac{T_{del,geq} - \tau_{rem}^-}{T_{first}^-} \right\rfloor (X+1) + \left\lceil \frac{T_{del,geq} - \tau_{rem}^- - \left\lfloor \frac{T_{del,geq} - \tau_{rem}^-}{T_{first}^-} \right\rfloor T_{first}^-}{T_{min}} \right\rceil + 3X + 3 \end{array} \right\}.$$

Proof. Choose two arbitrary distinct correct node p, q and consider $s_{p,q}^{rem,geq}(t)$. We distinguish between two cases for t: (i) $t \geq t_{first,2X+2} + T_{del,geq}$ and (ii) $t < t_{first,2X+2} + T_{del,geq}$.

ad (i):

$$\begin{aligned}
s_{p,q}^{rem,geq}(t) &= r_{p,q}^{rem,GEQ}(t) - d_{p,q}^{GEQ}(t) \\
&\leq b_p(t - \tau_{rem}^-) + X - d_{p,q}^{GEQ}(t) \\
&\leq b^{max}(t - \tau_{rem}^-) + X - d_{p,q}^{GEQ}(t) \\
&= \left(b^{max}(t - \tau_{rem}^-) - b^{max}(t - T_{del,geq})\right) + \left(b^{max}(t - T_{del,geq}) - d_{p,q}^{GEQ}(t)\right) + X \\
&\leq \left\lceil \frac{T_{del,geq} - \tau_{rem}^-}{T_{first}^-} \right\rceil (X+1) + \left\lceil \frac{T_{del,geq} - \tau_{rem}^- - \left\lfloor \frac{T_{del,geq} - \tau_{rem}^-}{T_{first}^-} \right\rfloor T_{first}^-}{T_{min}} \right\rceil + X + 3 && (132) \\
&\leq S_{rem,geq},
\end{aligned}$$

where (132) follows from applying Lemma 20 and Lemma 25.

ad (ii): Since $d_{p,q}^{GEQ}(t) \geq -1$ must always hold, we obtain

$$
\begin{aligned}
s_{p,q}^{rem,geq}(t) &= r_{p,q}^{rem,GEQ}(t) - d_{p,q}^{GEQ}(t) \\
&\leq b_p(t - \tau_{rem}^{-}) + X - d_{p,q}^{GEQ}(t) \\
&\leq b^{max}(t - \tau_{rem}^{-}) + 1 + X \\
&\leq b^{max}(t_{first,2X+2} + T_{del,geq} - \tau_{rem}^{-}) + 1 + X \\
&\leq \left(b^{max}(t_{first,2X+2} + T_{del,geq} - \tau_{rem}^{-}) - b^{max}(t_{first,2X+2})\right) + b^{max}(t_{first,2X+2}) + 1 + X \qquad (133) \\
&\leq \left\lfloor \frac{T_{del,geq} - \tau_{rem}^{-}}{T_{first}^{-}} \right\rfloor (X+1) + \left\lceil \frac{T_{del,geq} - \tau_{rem}^{-} - \left\lfloor \frac{T_{del,geq} - \tau_{rem}^{-}}{T_{first}^{-}} \right\rfloor T_{first}^{-}}{T_{min}} \right\rceil + 3X + 3 \qquad (134) \\
&\leq S_{rem,geq},
\end{aligned}
$$

where (134) follows from Lemma 19.

References

1. Charron-Bost, B., Dolev, S., Ebergen, J., Schmid, U.: 08371 summary – fault-tolerant distributed algorithms on vlsi chips. In Charron-Bost, B., Dolev, S., Ebergen, J., Schmid, U., eds.: Fault-Tolerant Distributed Algorithms on VLSI Chips. Number 08371 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany (2009) http://drops.dagstuhl.de/opus/frontdoor.php?source_opus=1927.
2. Schmid, U.: Keynote: Distributed algorithms and VLSI. In: Proceedings of the 10th International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS'08). Volume 5340 of Lecture Notes in Computer Science., Detroit, USA, Springer Verlag (2008) 3 (http://www.vmars.tuwien.ac.at/documents/extern/2467/sss08.pdf).
3. Moscibroda, T., Mutlu, O.: Distributed order scheduling and its application to multi-core dram controllers. In: Proceedings of the 27th ACM Symposium on Principles of Distribute d Computing (PODC'08), Toronto, Canada (2008) 365–374
4. Ferri, C., Moreshet, T., Bahar, R.I., Benini, L., Herlihy, M.: A hardware/software framework for supporting transactional memory in a mpsoc environment. SIGARCH Comput. Archit. News **35** (2007) 47–54
5. Fuegger, M., Schmid, U., Fuchs, G., Kempf, G.: Fault-Tolerant Distributed Clock Generation in VLSI Systems-on-Chip. In: Proceedings of the Sixth European Dependable Computing Conference (EDCC-6), IEEE Computer Society Press (2006) 87–96
6. Dolev, S., Haviv, Y.: Self-stabilizing microprocessors, analyzing and overcoming soft-errors. IEEE Transactions on Computers **55** (2006) 385–399
7. Dolev, S., Tzachar, N.: Brief announcment: Corruption resilient fountain codes. In: DISC. (2008) 502–503
8. Friedman, R., Mostefaoui, A., Rajsbaum, S., Raynal, M.: Asynchronous agreement and its relation with error-correcting codes. IEEE Trans. Comput. **56** (2007) 865–875
9. Dielacher, A., Fuegger, M., Schmid, U.: How to speed-up fault-tolerant clock generation in vlsi systems-on-chip via pipelining. In: Proceedings of the 27th ACM Symposium on Principles of Distributed Computing (PODC'08), ACM Press (2008) 423 An extended version is available as RR 15/2009, Institut für Technische Informatik, TU-Wien, http://www.vmars.tuwien.ac.at/documents/extern/2571/techreport.pdf.
10. Srikanth, T.K., Toueg, S.: Optimal clock synchronization. Journal of the ACM **34** (1987) 626–645
11. Widder, J., Schmid, U.: Achieving synchrony without clocks. Research Report 49/2005, Technische Universität Wien, Institut für Technische Informatik (2005) (to appear in Distributed Computing, 2009).
12. Ferringer, M., Fuchs, G., Steininger, A., Kempf, G.: VLSI Implementation of a Fault-Tolerant Distributed Clock Generation. IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems (DFT2006) (2006) 563–571
13. Fischer, M.J., Lynch, N.A., Paterson, M.S.: Impossibility of distributed consensus with one faulty process. Journal of the ACM **32** (1985) 374–382
14. Dielacher, A., Fuegger, M., Schmid, U.: How to speed-up fault-tolerant clock generation in vlsi systems-on-chip via pipelining. Research Report 15/2009, Technische Universität Wien, Institut für Technische Informatik, Treitlstr. 1-3/182-1, 1040 Vienna, Austria (2009) http://www.vmars.tuwien.ac.at/documents/extern/2571/techreport.pdf.
15. Chapiro, D.M.: Globally-Asynchronous Locally-Synchronous Systems. PhD thesis, Stanford University (1984)
16. Polzer, T., Handl, T., Steininger, A.: A metastability-free multi-synchronous communication scheme for fault-tolerant socs. Research Report 10/2009, Technische Universität Wien, Institut für Technische Informatik, Treitlstr. 1-3/182-1, 1040 Vienna, Austria (2009)
17. Schmid, U., Steininger, A.: Dezentrale Fehlertolerante Taktgenerierung in VLSI Chips. Research Report 69/2004, Technische Universität Wien, Institut für Technische Informatik (2004) (Österr. Patentanmeldung A 1223/2004).
18. Le Lann, G., Schmid, U.: How to implement a timer-free perfect failure detector in partially synchronous systems. Technical Report 183/1-127, Department of Automation, Technische Universität Wien (2003) (Replaced by Research Report 28/2005, Institut für Technische Informatik, TU Wien, 2005.).
19. Widder, J., Le Lann, G., Schmid, U.: Failure detection with booting in partially synchronous systems. In: Proceedings of the 5th European Dependable Computing Conference (EDCC-5). Volume 3463 of LNCS., Budapest, Hungary, Springer Verlag (2005) 20–37
20. Lynch, N.: Distributed Algorithms. Morgan Kaufman Publishers, Inc., San Francisco, USA (1996)
21. Grahsl, J., Handl, T., Steininger, A.: Exploring the usefulness of the gate-level stuck-at fault model for Muller C-elements. In: Proceedings 20. Workshop für Testmethoden und Zuverlässigkeit von Schaltungen und Systemen (TuZ'08), Vienna, Austria (2008) 165–169
22. Koren, I., Koren, Z.: Defect tolerance in VLSI circuits: Techniques and yield analysis. Proceedings of the IEEE **86** (1998) 1819–1838
23. Baumann, R.: Radiation-induced soft errors in advanced semiconductor technologies. IEEE Transactions on Device and Materials Reliability **5** (2005) 305–316
24. Normand, E.: Single-event effects in avionics. IEEE Transactions on Nuclear Science **43** (1996) 461–474
25. Maza, M.S., Aranda, M.L.: Analysis of clock distribution networks in the presence of crosstalk and groundbounce. In: Proceedings International IEEE Conference on Electronics, Circuits, and Systems (ICECS). (2001) 773–776
26. Lamport, L.: Buridan's principle. Technical report, SRI Technical Report (1984)
27. Fuchs, G., Fuegger, M., Steininger, A.: On the threat of metastability in an asynchronous fault-tolerant clock generation scheme. Research Report 11/2009, Technische Universität Wien, Institut für Technische Informatik, Treitlstr. 1-3/182-1, 1040 Vienna, Austria (2009)
28. Dolev, D., Halpern, J.Y., Strong, H.R.: On the possibility and impossibility of achieving clock synchronization. Journal of Computer and System Sciences **32** (1986) 230–250
29. Sutherland, I.E.: Micropipelines. Communications of the ACM, Turing Award **32** (1989) 720–738 ISSN:0001-0782.
30. Schmid, U.: How to model link failures: A perception-based fault model. In: Proceedings of the International Conference on Dependable Systems and Networks (DSN'01), Göteborg, Sweden (2001) 57–66
31. Santoro, N.: Design and Analysis of Distributed Algorithms. Wiley Series on Parallel and Distributed Computing. John Wiley & Sons (2007)

32. Chandra, T.D., Toueg, S.: Time and message efficient reliable broadcasts. In: Proceedings 4th International Workshop on Distributed Algorithms (WDAG'90). (1990) 289–303
33. Garay, J.A., Kutten, S., Peleg, D.: A sublinear time distributed algorithm for minimum-weight spanning trees. SIAM J. Comput. **27** (1998) 302–316
34. Awerbuch, B., Cidon, I., Kutten, S.: Optimal maintenance of a spanning tree. J. ACM **55** (2008) 1–45
35. Bar-Noy, A., Naor, J., Naor, M.: One-bit algorithms. Distributed Computing **4** (1990) 3–8
36. Tanenbaum, A.S.: Computer Networks. 4 edn. Prentice Hall (2003)
37. (ed.), E.M.C.: Editorial: Distributed computing issue in hardware design. Distributed Computing **1** (1986) 185–186
38. Seitz, C.L.: System timing. In: Introduction to VLSI Systems. Addison Wesley (1980) 218–262
39. Black, D.L.: On the existince of delay-insensitive fair arbiters: Trace theory and its limitations. Distributed Computing **1** (1986) 205–225
40. Martin, A.J.: The limitations to delay-insensitivity in asynchronous circuits. In: AUSCRYPT '90: Proceedings of the sixth MIT conference on Advanced research in VLSI, Cambridge, MA, USA, MIT Press (1990) 263–278
41. Ebergen, J.C.: A formal approach to designing delay-insensitive circuits. Distributed Computing **5** (1991) 107–119
42. Hauck, S.: Asynchronous design methodologies: An overview. Proceedings of the IEEE **83** (1995) 69–93
43. Marino, L.: General theory of metastable operation. IEEE Transactions on Computers **C-30** (1981) 107–115
44. Barros, J.C., Johnson, B.W.: Equivalence of the arbiter, the synchronizer, the latch, and the inertial delay. IEEE Trans. Comput. **32** (1983) 603–614
45. Lamport, L.: Arbitration-free synchronization. Distributed Computing **16** (2003) 219–237
46. Martin, A.J.: Compiling communicating processes into delay-insensitive VLSI circuits. Distributed Computing **1** (1986) 226–234
47. Stevens, K.S., Ginosar, R., Rotem, S.: Relative timing [asynchronous design]. IEEE Trans. VLSI Syst. **11** (2003) 129–140
48. Marzullo, K.A.: Tolerating failures of continuous-valued sensors. ACM Transactions on Computer Systems **8** (1990) 284–304
49. Yakovlev, A., Lavagno, L., Sangiovanni-Vincentelli, A.: A unified signal transition graph model for asynchronous control circuit synthesis. In: Proceedings of the 1992 IEEE/ACM international conference on Computer-aided design (ICCAD'92), Los Alamitos, CA, USA, IEEE Computer Society Press (1992) 104–111
50. Rokicki, T., Myers, C.J.: Automatic verification of timed circuits. In: CAV. (1994) 468–480
51. Myers, C.J., Meng, T.H.Y.: Synthesis of timed asynchronous circuits. IEEE Trans. VLSI Syst. **1** (1993) 106–119
52. Belluomini, W., Myers, C.J.: Verification of timed systems using posets. In: CAV. (1998) 403–415
53. Yoneda, T., Kitai, T., Myers, C.J.: Automatic derivation of timing constraints by failure analysis. In: Proceedings 14th International Conference on Computer Aided Verification (CAV'02). Volume 2404 of Lecture Notes in Computer Science., Springer (2002) 195–208
54. Kaynar, D.K., Lynch, N., Segala, R., Vaandrager, F.: Timed I/O automata: A mathematical framework for modeling and analyzing real-time systems. Proceedings 24th IEEE International Real-Time Systems Symposium (RTSS'03) **00** (2003) 166–177
55. Schmid, U., Klasek, J., Mandl, T., Nachtnebel, H., Cadek, G.R., Kerö, N.: A Network Time Interface M-Module for distributing GPS-time over LANs. Real-Time Systems **18** (2000) 24–57
56. Maza, M.S., Aranda, M.L.: Interconnected rings and oscillators as gigahertz clock distribution nets. In: GLSVLSI '03: Proceedings of the 13th ACM Great Lakes symposium on VLSI, ACM Press (2003) 41–44
57. Fairbanks, S., Moore, S.: Self-timed circuitry for global clocking. In: Proceedings of the Eleventh International IEEE Symposium on Advanced Research in Asynchronous Circuits and Systems. (2005) 86–96
58. Fairbanks, S.: Method and apparatus for a distributed clock generator (2004) US patent no. US2004108876.