On Time Synchronization Issues in Time-Sensitive Networks with Regulators and Nonideal Clocks

LUDOVIC THOMAS, I&C, EPFL, Switzerland
JEAN-YVES LE BOUDEC, I&C, EPFL, Switzerland

Flow reshaping is used in time-sensitive networks (as in the context of IEEE TSN and IETF Detnet) in order to reduce burstiness inside the network and to support the computation of guaranteed latency bounds. This is performed using per-flow regulators (such as the Token Bucket Filter) or interleaved regulators (as with IEEE TSN Asynchronous Traffic Shaping, ATS). The former use one FIFO queue per flow, whereas the latter use one FIFO queue per input port. Both types of regulators are beneficial as they cancel the increase of burstiness due to multiplexing inside the network. It was demonstrated, by using network calculus, that they do not increase the worst-case latency. However, the properties of regulators were established assuming that time is perfect in all network nodes. In reality, nodes use local, imperfect clocks. Time-sensitive networks exist in two flavours: (1) in non-synchronized networks, local clocks run independently at every node and their deviations are not controlled and (2) in synchronized networks, the deviations of local clocks are kept within very small bounds using for example a synchronization protocol (such as PTP) or a satellite based geo-positioning system (such as GPS). We revisit the properties of regulators in both cases. In non-synchronized networks, we show that ignoring the timing inaccuracies can lead to network instability due to unbounded delay in per-flow or interleaved regulators. We propose and analyze two methods (rate and burst cascade, and asynchronous dual arrival-curve method) for avoiding this problem. In synchronized networks, we show that there is no instability with per-flow regulators but, surprisingly, interleaved regulators can lead to instability. To establish these results, we develop a new framework that captures industrial requirements on clocks in both non-synchronized and synchronized networks, and we develop a toolbox that extends network calculus to account for clock imperfections.

CCS Concepts: • **Networks** → **Network performance analysis**; *Time synchronization protocols*; • **Computer systems organization** → *Real-time systems*;

Keywords: Time Sensitive Networks; Time Synchronization; Per-Flow Regulator; Per-Flow Shaper; Interleaved Regulator; Network Calculus

ACM Reference Format:
Ludovic Thomas and Jean-Yves Le Boudec. 2020. On Time Synchronization Issues in Time-Sensitive Networks with Regulators and Nonideal Clocks. In *Proc. ACM Meas. Anal. Comput. Syst.*, Vol. 4, 2, Article 27 (June 2020). ACM, New York, NY. 41 pages. https://doi.org/10.1145/3392145

1 INTRODUCTION

Time-sensitive networks support real-time applications in many industries such as automation [24], avionics [10, 37], space [18], and automobile [31]. Recent work in time-sensitive networks include the time-sensitive networking (TSN) task group of the Institute of Electrical and Electronics Engineers (IEEE) and the Detnet working group of the Internet Engineering Task Force (IETF). Both

Authors' addresses: Ludovic Thomas, ludovic.thomas@epfl.ch, I&C, EPFL, Route Cantonale, Lausanne, 1015, Switzerland; Jean-Yves Le Boudec, jean-yves.leboudec@epfl.ch, I&C, EPFL, Route Cantonale, Lausanne, 1015, Switzerland.

2476-1249/2020/6-ART27 $15.00
https://doi.org/10.1145/3392145

aim to provide deterministic worst-case delay and jitter bounds with seamless reconfiguration [30] and redundancy [29].

Reshaping flows inside the network by means of traffic regulators helps achieve these objectives. Traffic regulators are hardware elements that, placed before a multiplexing stage, remove the increased burstiness due to interference with other flows in previous hops. They support higher scalability and efficiency of time-sensitive networks and enable the computation of guaranteed latency-bounds in networks with cyclic dependencies [46, 57, 61]. They come in two types: the per-flow regulator (PFR) (also called "per-flow shaper") [40, Section 1.7.4] and the interleaved regulator (IR) [39]; the IR processes flow aggregates and the PFR processes each flow individually and requires one queue per flow.

In both types, each flow has its own regulation parameter, usually in terms of burst and rate. Regulators then delay any packet whose release would violate the regulation parameter. A well-known example of a PFR is the Linux's Token-Bucket Filter [60]. Configured with a rate r and a burst b, it makes sure that over any window of duration t, no more than $rt + b$ bits are released by the regulator. Hence, the evaluation of elapsed time is at the heart of the operation of any regulator. When a regulator can base its computations on an ideal clock, previous studies have established that it enjoys the "shaping-for-free" property, i.e., a regulator that removes the burstiness increase caused by a first in, first out (FIFO) system does not increase the worst-case delay of flows [40, Thms 1.5.2 and 1.7.3],[39, Thm 5]. This property is essential to the analysis of time-sensitive networks with regulators.

In reality, the clock used by a regulator is nonideal, and the clocks used by different devices in a network deviate slightly from true time. Time-sensitive networks are either synchronized or non-synchronized. In non-synchronized networks, local clocks run independently at every node and their deviations are not controlled. In synchronized networks, the deviations are kept within bounds, using a time-synchronization protocol or a global navigation satellite system. With time-synchronization methods such as the Precision Time Protocol (PTP) [25], WhiteRabbit [47] or the Global Positioning System (GPS) [54], the clock deviation bound is ~1μs or less; we call such cases "tightly synchronized". Here, the clock deviation is smaller than the latency requirements of network flows (~1ms for avionics systems), and tightly-synchronized networks are typically analyzed as if clocks would be ideal. Some other networks require time synchronization only for network management purposes and use a method such as the Network Time Protocol (NTP) [48], which provides a clock deviation bound of ~100ms; we call such cases "loosely synchronized".

Consider a flow of data shaped at some point in the network by a token-bucket filter with rate r and burst b (Figure 1a). After traversing some network element, say S, the flow typically becomes more bursty and no longer satisfies the burst tolerance b. Assume a per-flow regulator R is applied to the flow at the output of S, in order to re-enforce the burst tolerance b, using a token-bucket filter with same rate r and burst b. The token-bucket filter delays data of the flow that comes out of S in too large bursts. However, if clocks are ideal, the shaping-for-free property means that the worst-case delay of the flow through S and R is the same as the worst-case delay through S alone (i.e., late packets are not delayed by the regulator). Now assume the clocks are nonideal and the network is not synchronized. If the clock at the token-bucket filter R is too slow, the true value of r implemented by R is slightly less than that of the source; this will lead to a slow, but steady, buildup of backlog at the input buffer of R, which might lead over time to an arbitrarily large delay or unexpected loss.

This simple example suggests that clock nonidealities might significantly affect the delay analysis of time-sensitive networks with regulators. It has raised concerns and discussions in the ongoing standardization process of IEEE ATS [32]. In this paper, we provide theoretical foundations to the problem and we determine to what extend delay analyses are affected in non-synchronized and synchronized networks. Our main contributions are:

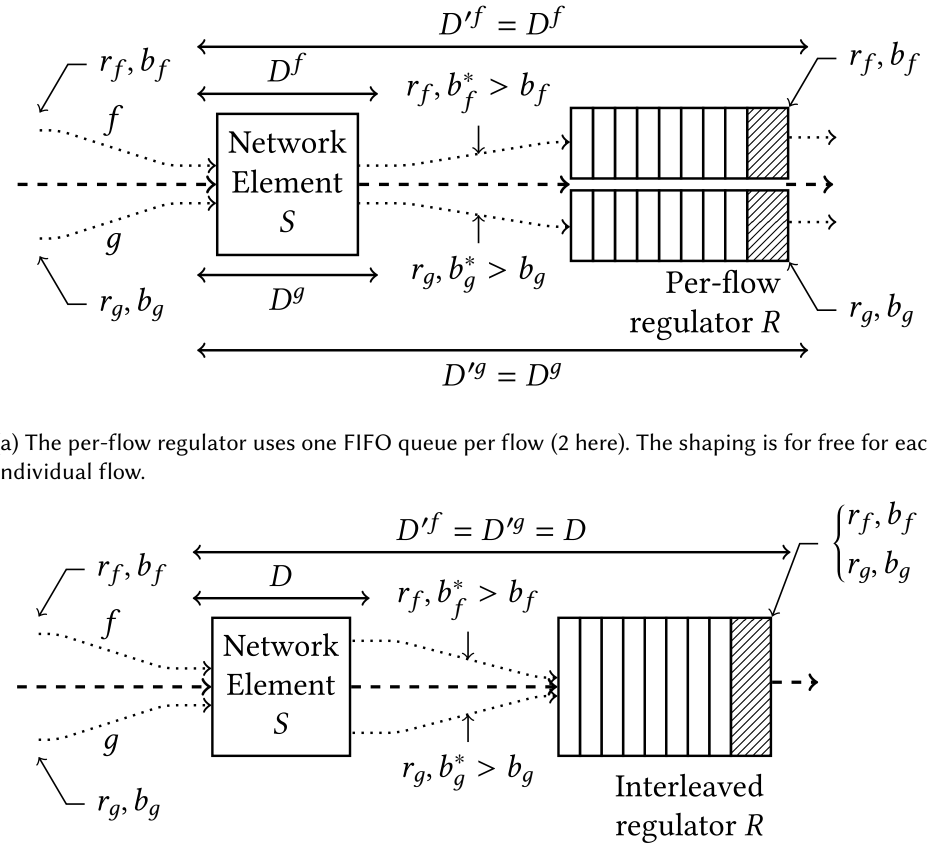
- We propose a time model for non-synchronized and synchronized networks; it can be used for computation of latency bounds. Using the example of TSN, we show that the model parameters can easily be obtained from industrial requirements.
- To compute latency bounds when clocks are nonideal, we propose a toolbox to be used with other network calculus results.
- In non-synchronized networks, we show that the configuration of regulators must be adapted to take into account the clock imperfections. If not adapted, we prove that regulators can yield unbounded latencies or unexpected packet losses.
- For non-synchronized networks, we refine and provide a formal justification for the method proposed in [32, Annex V.8] for configuring the regulators and for avoiding the problem mentioned above. This rate- and burst-cascade method increases the rate and burst tolerance at every regulator along the path of a flow; it requires that the parameters of a regulator depend on the position of the regulator along the flow path, thus it adds complexity to the control plane. It applies to both PFR and IR. We propose an alternative method, asynchronous dual arrival-curve method (ADAM), that uses the same regulator parameters at all re-shaping points on the flow path thus makes the control plane simpler; it applies to PFR only. We also compare the delay bounds obtained with each method, but we leave to future work the practical evaluation of the two methods.
- In synchronized networks, we compute a bound on the delay penalty imposed by PFRs. In tightly-synchronized networks, this penalty is small compared to latency bounds, and the current practice of ignoring it is adequate. In contrast, in loosely-synchronized networks, we show an example where the delay penalty can be significant thus should be taken into account.
- The conclusions are very different in synchronized networks with IRs. We show that, even in tightly-synchronized networks, IRs can yield unbounded delay or unexpected loss if the residual clock inaccuracies are not accounted for. The method of rate and burst cascade can be used to avoid this problem.

In Sections 2 and 3, we present the related work and provide the necessary background on time-sensitive networks, regulators and network calculus. We introduce our assumptions and our time and network models in Section 4. In Section 5, we detail our toolbox of network calculus results. We then analyze regulators in non-synchronized and synchronized networks in Sections 6 and 7, respectively. We make our conclusive remarks in Section 8. Proofs of propositions are available in Appendix A.

2 RELATED WORK

The modeling of clock nonidealities benefits from a solid background in time metrology. In [35], the International Telecommunication Union (ITU) defines fundamental notions and models for clocks used in synchronization networks. These models are further detailed in reference documents such as [36] for the ITU and [26] for the IEEE. Correspondingly, industrial requirements have been defined for clocks to be used in networks. They put constraints and bounds on the clock characteristics defined in the above documents. For instance, for a clock to be used in a synchronized TSN network, it must meet the requirements of [28, Annex B.1].

Many technologies have been developed to perform the time-synchronization of a network. The most common are the use of an external time source such as a global navigation satellite system (GNSS) [54], and the use of time-synchronization protocols such as NTP [42], PTP [25], generalized PTP [28] and WhiteRabbit [47]. Other technologies are tailored for wireless sensor networks [41, 63]. Each comes with various performance analyses: we can cite [48] for NTP, [17] for WhiteRabbit. However, the design of a "good" time-synchronization protocol remains an open



(a) The per-flow regulator uses one FIFO queue per flow (2 here). The shaping is for free for each individual flow.

(b) The interleaved regulator uses only one FIFO queue. If the network element S is FIFO for the aggregate $\{f, g\}$, then the shaping is for free for the aggregate and both flows have the same delay bound $D'^f = D'^g = D$ through S and R.

Fig. 1. Shaping-for free property of two type of regulators in a network with ideal clocks.

issue [21, 56], and each protocol proposition adds to the time-metrology domain by identifying limits of previous protocols [22, 59].

The analyses mention that the precision of time-synchronization protocols depends on the latency and jitter of synchronization messages and of control data. The latency and jitter bounds of time-sensitive networks were studied in numerous occasions using network calculus. For TSN, we can cite [5, 57, 64]. TSN provides many building blocks to provide guaranteed delay bounds. The use of regulators with asynchronous traffic shaping (ATS) is one of them. Regulators have been studied in [46, 57, 62]. Other building blocks include cyclic queuing and forwarding (CQF) [34], credit-based scheduler (CBS) [27] or time-aware shaper (TAS) [33]. Choosing the best set of building blocks for a specific network is an open question [50], and several studies have compared their performance [49]. In this paper, we focus on regulators, thus on ATS.

Interestingly, the reciprocal aspect, the effect of the clock and synchronization nonidealities on the network performances, appears to be much less studied. For example, the above-mentioned studies always assume that clocks are perfect in the network or that time distribution is perfect [49]. Even network simulator 3 (ns-3) [52] has a unique time-base for simulating network events, and the simulation of clock behavior and of time-synchronization protocols requires work-arounds such as the one in [43, Section III]. In this paper, we assess the effect of clock nonidealities on the performance bounds of regulators (ATS) and we leave to future work the evaluation of their effect on other TSN building blocks.

In [3, Section 7.1], the authors consider the nonidealities of clocks to show that the adversarial traffic generation described in [1, 4] does not induce unbounded latencies under realistic network assumptions. However, their traffic model is limited to non-bursty flows [3, Section 3] and their results are obtained through simulations. In this paper, we focus on obtaining upper-bounds on worst-case latencies, and finding the worst-case using simulations is known to be an intractable issue [5, 53].

A seminal work on applying network calculus on networks with nonideal clocks for obtaining worst-case upper-bounds lies in [13–15]. The authors are interested in a bandwidth management method that spreads the time at which frames are scheduled on the Controller Area Network (CAN) bus. Such bandwidth management uses offsets between the time instants at which frames are scheduled. They note that such scheduling of the frames across different nodes would require a time-synchronization mechanism between the network nodes. However, they show that a "weak synchronization" of the nodes (with a 1ms precision bound) already provides significant performance improvements [13, Section 5.3]. Their time-model [13, Section 3.1] is hence limited to synchronized networks (including loosely-synchronized networks). It doesn't consider bounds on the clock frequency offset, but only on the clock time-error function (called the "phase bound" in their paper). In the present paper, we are interested in both synchronized and non-synchronized networks, and we show that taking into consideration the bounds on the frequency offsets helps having tight delay bounds. Last, the authors of [13–15] compute tight arrival curves, service curves and latency bounds of periodic flows on a CAN bus that employs the above-mentioned offset management method [13, Section 4.3.1]. Then they adapt the results to take into account the phase bounds (the time-error bounds) [13, Section 4.3.2]. As a consequence, their analysis of clock nonidealities is limited to a specific network model, with specific service and arrival curves. In this paper, we are interested in computing the effect of clock nonidealities given any service curve, arrival curve or latency bound.

The present paper is also motivated by discussions with industrial partners, specifically in the context of TSN. The ongoing draft for ATS mentions the possible consequences of clock nonidealities when deploying regulators [32, Annex V.8] and proposes some solutions that would benefit from theoretical foundations, as proposed in this paper.

3 TIME-SENSITIVE NETWORKS WITH REGULATORS

Here, we provide some background that is required by the rest of the paper. In time-sensitive networks, delays at network elements have to be bounded in worst case, not in average. To this end, network calculus is often used [7, 11, 12, 40]. This framework uses cumulative functions such as $A(t)$, where $A(t)$ is the total number of bits observed at some observation point between an arbitrary time reference 0 and time t. Traffic flows are assumed to be bounded by arrival curve constraints, namely, constraints of the form: $\forall t \geq s \geq 0, A(t) - A(s) \leq \alpha(t-s)$ (the function α is called "arrival curve"). A frequently used function is $\gamma_{r,b}$ defined by $\gamma_{r,b}(t) = rt + b$ for $t > 0$ and $\gamma_{r,b}(t) = 0$ for $t \leq 0$. It corresponds to a flow that is limited to a rate r and a burst b ("leaky-bucket" arrival curve).

The service offered by a network element is also assumed to be lower bounded by a condition of the form $\forall t \geq 0 : D(t) \geq (A \otimes \beta)(t)$ where A [resp. D] is the input [resp. output] cumulative function, the function β is called "service curve" and the symbol $\otimes$ is the min plus convolution, such that $(A\otimes\beta)(t) = \inf_{s\geq 0}(A(s) + \beta(t-s))$. By Reich's formula [51], a network element reduced to a single server queue with output rate R offers the service curve $\beta(t) = Rt$. If in addition the server can take vacations for durations upper bounded by T per busy period, the system offers a service curve $\lambda_{R,T}(t) = |R(t-T)|^{+}(= \max(0, R(t-T)))$, called "rate-latency" service curve. A FIFO network element that guarantees a delay upper bounded by D offers the service curve δ_D defined by $\delta_D(t) = 0$ for $t \leq D$ and $\delta_D(t) = +\infty$ for $t > D$. The concatenation of network elements that each offers a service curve also offers a service curve equal to the min-plus convolution of service curves. Many schedulers, such as Deficit Round Robin or the Credit Based Shaper of IEEE TSN, are characterized by rate-latency service curves [6, 16].

Classic network calculus results give delay and backlog bounds at a network element, given some arrival-curve and service-curve constraints. They also give bounds on the burst of the output, i.e., arrival curves for the output flows [40, 45].

Time-sensitive networks can be per-flow networks or per-class networks. In the former case, schedulers are per-flow, e.g., there is one Deficit Round Robin queue per flow. It follows that service-curve properties apply to individual flows. In contrast, in class-based networks, schedulers offer a service guarantee (captured by a service curve) to the aggregate of all flows that belong to one specific class; inside a class, the scheduler is FIFO. Providing delay bounds in class-based network is more complicated than in per-flow networks. In particular, we need to compute good bounds on the burst increase that can occur at every hop [9]. Practical solutions for complex class-based networks almost all require that flows are re-shaped individually inside the network.

Flow shaping (or re-shaping) is performed by regulators, that are either per-flow (PFR) or interleaved (IR). A PFR, configured with arrival curve σ for flow f, makes sure that its output satisfies the arrival curve constraint σ (also called "shaping curve"). If the input data of flow f arrives too fast, the packets are stored in the PFR buffer (with one FIFO queue per flow), until the earliest time when it is possible to release the packet without violating the arrival curve constraint.

Note that a regulator, as defined above, controls the arrival curve of any individual flow at its output. As a consequence, asynchronous traffic shaping (ATS) [32] (the standardization of PFR and IR) differs from other types of shapers that act on a per-class basis. For instance, time-aware shaper (TAS) [33] is a scheduler that controls, using gates, when classes may access the link, depending on a global distributed schedule. It cannot control the arrival curve of a single flow that continues to suffer from the contention with other flows of the same class. It is also worth noting that, as regulators only need to measure elapsed time, they are insensitive to constant time offsets. On the contrary, some schedulers such as TAS require the nodes to be time-synchronized.

The input-output characterization of a PFR with concave shaping curve σ is well understood: it can be modelled as the sequence of two virtual systems, a fluid greedy shaper followed by a packetizer [38]. The fluid greedy shaper is similar to the PFR but operates on individual bits: it releases fractions of a packet as soon as possible. The packetizer receives the bit-by-bit output of the former and stores it until a full packet can be released. The former is a min-plus linear system, characterized by the relation $D(t) = (A \otimes \sigma)(t)$ where A [resp. D] is the input [resp. output] cumulative function of the fluid greedy shaper. The latter does not increase the per-packet delay bound. It can be ignored in latency calculations [8]. In particular, a PFR with concave shaping curve σ is a network element that offers σ as service curve.

An IR is similar to a PFR with one large difference. All packets of all flows are stored in a single FIFO queue, the packet at the head of the queue is released at the earliest time when it is possible without violating the arrival curve constraint for this flow, and packets of other flows wait until they appear at the head of the queue [57]. PFRs and IRs have state information per flow but IRs have a single FIFO queue thus are preferred in the context of IEEE TSN, which is typically per-class [32]. Unlike PFRs, no service curve characterization appears to be known for IRs.

When a flow is served in a network element, its burst typically increases, and the increase is large when the flow shares a network element with many bursty flows in the same class. Worse, the more bursty the competing flows are, the larger the burst increase is. The increased burst leads to larger delays and backlogs in downstream nodes, which is the ingredient for a cascade effect and can even lead to instability when there are cyclic dependencies [1, 9]. This does not occur if flows are reshaped by regulators at some or every node [58]. In contrast, if regulators are used, the burst of the output is known and can be imposed to be the same as at the source, which enables us to find good delay bounds.

The regulator, however, is itself a queuing system and its impact on delay should be accounted for. Here, an essential property of regulators is used, called "shaping-for-free". For a PFR, it can be stated as follows [39, Theorem 3]. Consider flows, such as f on Figure 1a, which satisfy arrival-curve constraints σ_f (one per flow) and are served in a network element S, which is FIFO for packets inside every flow; after S, the flows are processed by a PFR with the same arrival curves (i.e., σ_f for flow f). Then, for every flow f, the worst-case delay, D^f, for any packet of the flow through S is equal to the worst-case delay, D'^f through S and the PFR. In other words, re-shaping does not increase the worst-case delay of the previous hop (but reduces the worst-case delay at the next hop). If the PFR is replaced by an IR, there is a similar result for the worst-case delay over all flows that are processed by the IR (Figure 1b), assuming S is FIFO for all packets of all flows inside a given class [39, Theorem 5]. The shaping-for-free property is established assuming all clocks are ideal.

In reality, the clocks at the source and at different PFRs or IRs along the path of a flow are likely to be different. We say that the regulators in a time-sensitive network are "non-adapted" if we ignore the clock deviations and apply the shaping-for-free property. As we show in Sections 6 and 7, this can lead to severe problems, in both non-synchronized and synchronized networks.

4 SYSTEM MODEL

We first propose a framework for modeling a clock within a device. We derive this framework for non-synchronized networks and synchronized networks. We show how the clocks requirements within TSN are easily captured under our time model. Finally, we detail the network model under consideration. Notations for the whole paper are available in Table 1.

4.1 General Time-Model

We denote with $\mathcal{H}_{\text{TAI}}$ the true time, i.e. the international atomic time (*temps atomique international*) (TAI). We assume that it represents a continuous quantity. When reading the time indicated by a clock $\mathcal{H}_i$ in the network, only a subset of values are readable, due to the precision of the clock logic. We assume that this clock logic enforces the accessible values to increase when the clock is read over the course of the true time. With this assumption, and since the clock output is observable only at discrete time instants, it is possible to find a continuous, strictly increasing function $h_i(t)$ of the true time, which returns the value that the clock would display at true time t if it had infinite precision (Figure 2). Accessing the values of clock $\mathcal{H}_i$ corresponds to sampling the function h_i at the discrete time instants where the clock logic does a transition. This allows us to introduce the relative time function, which will be useful in Section 5, as follows.

Definition 4.1 (Relative time function $d_{g\to i}$). For any two clocks $\mathcal{H}_g, \mathcal{H}_i$ we define the relative time function from $\mathcal{H}_g$ to $\mathcal{H}_i$ as $d_{g\to i} = h_i \circ h_g^{-1}$, where $\circ$ represents the composition of functions.

Table 1. Notation

t	The measure of a time instant
$\mathcal{H}_i$	The clock of a device or the true time (TAI)
$d_{g\to i}$	The relative time-function between $\mathcal{H}_g$ and $\mathcal{H}_i$
T_{start}	When any of the clocks shows T_{start}, all other clocks have positive values and no device has sent any bit yet.
TAI	International Atomic Time (true time)
η	The timing-jitter bound
ρ	The clock-stability bound
Δ	The time-error bound in a synchronized network
R [resp R^*]	The cumulative arrival function of a flow at the input [resp output] of a device
α [resp α^*]	An arrival curve for a flow at the input [resp output] of a device
β	The service provided by a system (series of devices) to a flow
D'_k [resp D_k]	An upper-bound on the delay of a flow through its k-th hop [resp excluding the regulator of the k-th hop]
$R^{\mathcal{H}_i}, \alpha^{\mathcal{H}_i},$ $\beta^{\mathcal{H}_i}, D^{\mathcal{H}_i}$	We use the super-script to denote the clock used to observe one of the previous notions. Example: $\alpha^{\mathcal{H}_i}$ is the arrival curve as observed with clock $\mathcal{H}_i$.
$\gamma_{r,b}(t) = rt + b, t > 0$ $= 0, t \leq 0$	Leaky-bucket arrival curve of rate r, burst b
$\delta_D(t) = +\infty, t > D$ $= 0, t \leq D$	Service curve of a variable D-bounded delay
$\lambda_{R,T}(t) = \lvert R(t-T)\rvert^+$	Rate-latency service curve of rate R and latency T
$\mathcal{R}_k(r)$ [resp $\mathcal{Q}_k(b)$]	The smallest rate $\geq r$ [resp burst $\geq b$] that can be implemented by regulator Reg$_k$.
$\lvert x\rvert^+$	$\max(0, x)$
$a \wedge b$ [resp $a \vee b$]	$\min(a, b)$ [resp $\max(a, b)$]

$d_{g\to i}(t)$ is the value that clock $\mathcal{H}_i$ would have when clock $\mathcal{H}_g$ shows time t if they both had infinite precision. Note that h_i and h_g are both continuous, strictly increasing. Hence, $d_{g\to i}$ is also continuous, strictly increasing. We note $d^{-1}_{g\to i}$ its inverse. Obviously, $d^{-1}_{g\to i}$ is also equal to $d_{i\to g}$.

The time-metrology literature uses the time-error function [35, Section 4.5.13], which is equal to the time difference between two clocks $\mathcal{H}_i$ and $\mathcal{H}_g$. With our notation, the time-error function between $\mathcal{H}_g$ and $\mathcal{H}_i$ is $d_{g\to i}(t) - t$ for any t measured with $\mathcal{H}_g$.

For two clocks $\mathcal{H}_g, \mathcal{H}_i$, $d_{g\to i}(0)$ is the value of the clock $\mathcal{H}_i$ when the clock $\mathcal{H}_g$ shows zero. We denote by T_{start} the maximum of this value for any pair $(\mathcal{H}_g, \mathcal{H}_i)$, where each of $\mathcal{H}_g, \mathcal{H}_i$ represents a clock in the network or the TAI. We assume that, for any pair of clocks, no device transmits any bit in the network until $d_{g\to i}$ reaches T_{start}. We believe this is not a limiting assumption because we can assume that all devices start with a rough estimation of the true time, hence of each other. Consequently, T_{start} could be in the magnitude of hours or days, whereas the origin of time on network devices usually refers to several years in the past.

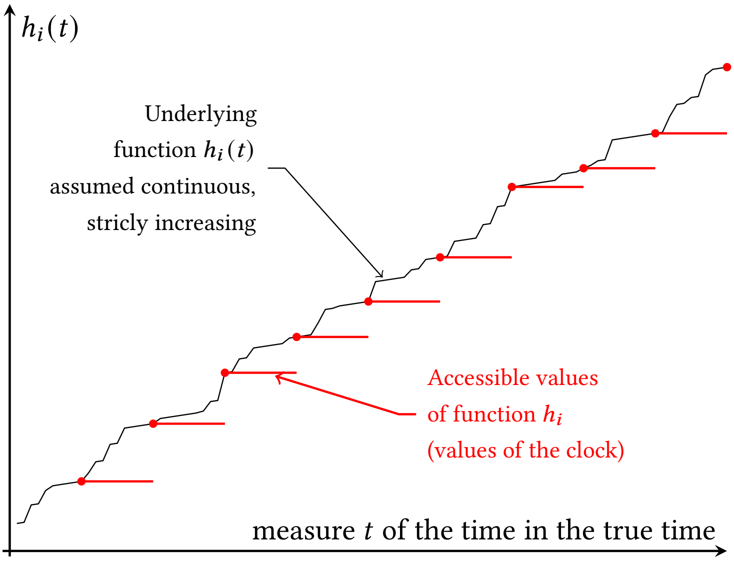


Fig. 2. Function h_i (Section 4.1). Only a subset of values are readable. We assume that the underlying function h_i is continuous, strictly increasing.

In this paper, we consider two time-models:

- The non-synchronized time-model: Clocks are free-running and do not interact with each other, but constraints on their stability can be formulated.
- The synchronized time-model: In addition to the stability requirements on the clocks, a time-synchronization algorithm (such as NTP or PTP), or an external time source (such as a GNSS) is employed. It distributes a time reference to all devices so that their local time matches with each other within a specified bound.

4.2 The Non-Synchronized Time-Model

We consider a clock $\mathcal{H}_i$ in the network. We first assume that this clock does not interact with any other. This corresponds to the *free-running mode* defined in [35, Section 4.4.1]. We assume that, when compared to the true time $\mathcal{H}_{\text{TAI}}$, the clock behaves as per the time-error model provided in [35, Annex I.3], reported below:

$$h_i(t) - t = x_{0,i} + ty_{0,i}(T) + \frac{D_i}{2}t^2 + \psi_i(t) \tag{1}$$

With $x_{0,i}$ the initial time offset of $\mathcal{H}_i$ (relative to the true time), $y_{0,i}(T)$ the frequency offset at constant temperature T, defined relative to the true frequency of 1 second per second, D_i the average frequency drift of clock $\mathcal{H}_i$ caused by its aging, and ψ_i is a random noise component.

As the clock is free-running, $x_{0,i}$ can take any value. Hence, it is impossible to put constraints on the time-error function itself but we can constrain its evolution. Take $s < t$, then

$$h_i(t) - h_i(s) - (t-s) = (t-s)y_{0,i}(T) + \frac{D_i}{2}(t^2 - s^2) + \psi_i(t) - \psi_i(s) \tag{2}$$

The first term is linear with $(t-s)$ and depends on $y_{0,i}(T)$.

Remark: Some time-metrology studies, including [22], denote with “clock drift” the linear evolution of the time-error function, *i.e.* $y_{0,i}(T)$. However, we decide here to remain consistent with the definitions of [35] that defines the *frequency offset* as the linear evolution of the time-error function and the *frequency drift* as the second-order evolution of the time-error function.

In industrial requirements, the frequency offset is usually bounded by a value that depends on the temperature conditions. We note $y_{\max,i}(T)$ the bound on the frequency offset of clock $\mathcal{H}_i$ at constant temperature T and $\rho_{1,i} = \max_{\{T\in\mathcal{T}\}} y_{\max,i}(T)$ its highest value over the whole range of temperatures $T \in \mathcal{T}$ that the network is expected to encounter. It corresponds to term $a_1 + a_2$ in [36, Section 11.2.1].

The second term is in second order with $t - s$ and depends on the relative aging of the clocks.

Remark: [35] defines two main contributors to the frequency drift, *i.e.* to the second-order evolution of the time-error function [35, Section 4.5.4]: the aging of the clock and the external effects, the latter being dominated by the effect of the evolution of the temperature on the clock frequency. Hence, when the temperature depends on the time, which is the case for any real-life single observation such as the one presented in Figure 7 of [22], then the term $t \cdot y_{0,i}(T(t))$ has indeed a second-order (or even higher-order) term that depends on $\frac{dT}{dt}$. However, for a network model, it is generally impossible to predict the evolution of the temperature, *i.e.* the function $T(t)$. Upper-bounding the frequency offset $y_{0,i}(T(t))$ by $\rho_{1,i} = \max_{\{T\in\mathcal{T}\}} y_{\max,i}(T)$ ensures that the model remains conservative for any possible evolution of the temperature. This is also consistent both with *a)* the ITU specifications [36, Section 11.2.1] which increase the linear frequency offset by a factor, a_2 in [36, Section 11.2.1], that accounts for the temperature changes, and keep the aging as the only remaining second-order contributor, and *b)* with the IEEE specifications, which define a frequency offset bound independent from the temperature [28, Annex B.1.1]. As a consequence, the second-order evolution of the time-error function in our model only depends on the aging of the clock, the term $D_i t^2/2$. It depends on the constant aging coefficient D_i of clock $\mathcal{H}_i$.

In industrial requirements, such as TSN [28], this term is often neglected. To verify the assumption that the aging is negligible, we compare the linear coefficient due to the frequency offset, $\rho_{1,i}$, with the linear coefficient due to the aging, $D_i\frac{t+s}{2}$. The following numerical application verify that we can neglect D_i.

Numerical Application to TSN: Call L the order of magnitude of the lifetime of a network. Then the aging coefficient over L is bounded by $D_i \cdot L$. If we take for $\rho_{1,i}$ the value specified in [28, Annex B.1.1] and for D_i (not specified in TSN) the largest value specified in [36, Table 24], we obtain $L = D_i/\rho_{1,i} \approx 10^{-4}/10^{-14} = 10^{10}$s. For the aging to be noticeable, compared to the acceptable frequency offset, the network shall be in operation for more than 300years.

The last term of Equation (2) is made of noise and is further detailed in [36, Section 8]. It has two components. The former is the timing jitter [35, Section 4.1.12], a high-frequency signal. It is usually constrained by a peak-to-peak jitter bound η_i [28, Annex B.1.3.1].

Remark: Due to its stochastic nature, the probability for a clock to present a jitter higher than the specified jitter bound, or to have a frequency offset higher than the frequency offset bound cannot be 0 [36, Note of Section 8.3]. However, providing bounds with a zero probability of excursion would not be possible, neither would it lead to deterministic delay bounds that are at the core of working groups such as TSN or Detnet. We here assume that we can find bounds on the jitter and on the frequency offset with a fulfillment probability high enough so as to keep the probability of an excursion over the lifetime of a network negligible. Then, any clock that does not behave as per the specified bounds is considered as faulty, and the probability of a faulty clock, together with its probable consequences, can be studied using the best practices of the safety-analysis domain [19, 20]. This assumption is also consistent with industrial requirements such as TSN [28, Annex B.1] that typically provides bounds on the jitter and on the frequency offset without specifying the fulfillment probability - hence considering any excursion as a failure.

Numerical Application to TSN: In the TSN requirements [28, Annex B.1.3.1], the jitter of any clock $\mathcal{H}_i$ shall not exceed 2ns peak-to-peak, that is $\eta_i = 2$ns for any $\mathcal{H}_i$ in the network.

The last noise component, the wander [35, Section 4.1.15], is a low-frequency noise signal. As opposed to the jitter or the frequency offset, it is usually constrained using the time deviation (TDEV), a statistical metric. The TDEV of clock $\mathcal{H}_i$, $\mathrm{dev}_i(t-s)$ is an upper-bound on the deviation of the time-error function over an observation period $t-s$. With the same argumentation as before, we assume that we can find $m_i \in \mathbb{R}$ such that the probability of the time-error function to present a wander over $t-s$ higher than $m_i \cdot \mathrm{dev}_i(t-s)$ is negligible over the lifetime of the network, and that such situation can be considered as a failure. Note that the order of magnitude of m_i would typically be around 10 for a normal distribution, as such multiples of the deviation already achieve a very high fulfillment probability.

In the majority of technical requirements, such as for TSN [28, Annex B.1], the TDEV is in the form $\mathrm{dev}_i(t-s) = (t-s)c_i$, with c_i a constant. In some cases, it can even be sub-linear [36, Section 8.1] or negligible [36, Section 8.]. To remain conservative, we consider the linear form and we define $\rho_{2,i} = m_i \cdot c_i$. Hence, the wander of Equation (2) is upper-bounded by $\rho_{2,i} \cdot (t-s)$.

We define the stability of clock $\mathcal{H}_i$ as $\rho_i = 1 + \rho_{1,i} + \rho_{2,i}$. As a consequence, the linear coefficient of Equation (2) is bounded by $\rho_i - 1$. It is worth noting that in general, one of $\rho_{1,i}$,$\rho_{2,i}$ is negligible with respect to the other. For example, when the non-synchronized clock $\mathcal{H}_i$ uses the phase-locking mechanism [35, Section 4.4.4] (also called syntonization or frequency synchronization [55]) with a near-perfect clock representing $\mathcal{H}_{\mathrm{TAI}}$, then $\rho_{1,i}$ is null [35, Appendix I.3]. When phase-locking mechanisms are not used, for instance in TSN, then $\rho_{1,i}$ is usually much higher than c_i, and with our previous remark on the value of m_i, $\rho_{1,i}$ would typically remain much higher than $\rho_{2,i}$.

Numerical Application to TSN: In the TSN requirements, for any clock $\mathcal{H}_i$ in the network, we shall have: $\rho_{1,i} = 100$ppm [28, Annex B.1.1] and $c_i = 5 \cdot 10^{-9}$ [28, Table B.1]. Hence, even with a margin m_i of hundred times the deviation, $\rho_{2,i}$ remains much smaller than $\rho_{1,i}$ and we obtain that any clock $\mathcal{H}_i$ has the stability $\rho_i = 1{+}1 \cdot 10^{-4}$.

From the above considerations, Equation (2) has (1) a jitter, high-frequency term, constrained by η_i, (2) a linear term, bounded by $\rho_i - 1$, and (3) no higher-level terms. We can upper-bound:

$$\forall t \geq s, h_i(t) - h_i(s) \leq (t-s)\rho_i + \ \eta_i \tag{3}$$

We now lower-bound the evolution by changing h_i for h_i^{-1} and we obtain:

$$\forall t \geq s, (t-s-\eta_i)\frac{1}{\rho_i} \leq h_i(t) - h_i(s) \leq (t-s)\rho_i + \ \eta_i \tag{4}$$

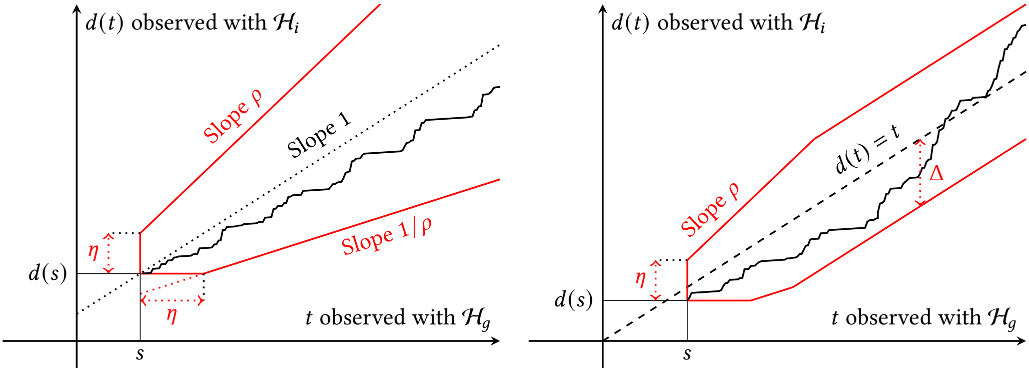
Let us now consider a pair of clocks $(\mathcal{H}_i, \mathcal{H}_g)$. We are interested in bounding the evolution of the relative time between $\mathcal{H}_g$ and $\mathcal{H}_i$, that is $d_{g\to i}(t)$ for t, the measure of a time instant with $\mathcal{H}_g$. We obtain, for $t \geq s$:

$$\begin{aligned} d_{g\to i}(t) - d_{g\to i}(s) &= h_i(h_g^{-1}(t)) - h_i(h_g^{-1}(s)) \\ &\leq (h_g^{-1}(t) - h_g^{-1}(s))\rho_i + \eta_i \\ &\leq (t-s)\rho_i\rho_g + \eta_g\rho_i + \eta_i \end{aligned}$$

For a given network, we define the *clock stability bound* of the network as $\rho = \max_{\{\mathcal{H}_i,\mathcal{H}_g\}}(\rho_i\rho_g)$ and the *timing-jitter bound* of the network as $\eta = \max_{\{\mathcal{H}_i,\mathcal{H}_g\}}(\eta_g\rho_i + \eta_i)$. Then any pair of clocks $(\mathcal{H}_i, \mathcal{H}_g)$ in the network verify $d_{g\to i}(t) - d_{g\to i}(s) \leq \rho(t-s) + \eta$. The lower bound is obtained by symmetry, by flipping the $\mathcal{H}_g$ and $\mathcal{H}_i$ roles.

We have finally obtained the following model: for any pair $(\mathcal{H}_g, \mathcal{H}_i)$, $\forall t \geq s$,

$$\frac{1}{\rho}(t-s-\eta) \leq d(t) - d(s) \leq \rho(t-s) + \eta \tag{5}$$



(a) In the non-synchronized time-model.

(b) In the synchronized time-model.

Fig. 3. Envelope of $d(t)$ (red) and example of a possible evolution of $d(t)$ (black).

where $d = d_{g \to i}$. Note that ρ and η do not depend on $\mathcal{H}_g, \mathcal{H}_i$.

Figure 3a presents, for a given known starting point $(s, d(s))$, the possible evolution space of $d(t)$ in the non-synchronized model as well as a possible trajectory. We note that the time-error function $d(t) - t$ can be unbounded in this model. We also note the symmetry of the non-synchronized envelope: if d meets the stability conditions (5), then d^{-1} also does.

Numerical Application to TSN: For a TSN network, any clock $\mathcal{H}_i$ satisfies $\rho_i = 1{+}1{\cdot}10^{-4}$ and $\eta_i = 2$ns. Hence, for a TSN network, the clock-independent parameters are $\rho = 1{+}2{\cdot}10^{-4}$ and $\eta =$ 4ns. Note that the above values are minimum requirements for clocks to be used as local clocks in TSN. Of course, if better bounds are known from the manufacturer specifications, the values of ρ and η can be updated accordingly.

4.3 The Synchronized Time-Model

In a synchronized network, clocks meeting the above stability requirements are synchronized with each other. The synchronization can be performed using for example PTP [25], generalized PTP defined in TSN [28], NTP [44], WhiteRabbit [47] or a GNSS [54]. When synchronization is used, the time-error function for any pair such as $(\mathcal{H}_g, \mathcal{H}_i)$ is bounded by the precision of the protocol.

For a given network, we define $\Delta \geq 0$ the *time-error bound* of the network and we assume that for any pair $(\mathcal{H}_g, \mathcal{H}_i)$, d meets the constraints of Equation (5), plus

$$\forall t, |d(t) - t| \leq \Delta \tag{6}$$

where $d = d_{g \to i}$. Note that Δ does not depend on $\mathcal{H}_g, \mathcal{H}_i$.

Remark: Clocks of a synchronized network have synchronization servos that control their frequency. This control can be used to perform some form of syntonization based on the synchronization messages as described in [25, Section 12.1]. Another example, is NTP's second leap: for example, Google's public NTP services smear the second leap by increasing the clock frequency by 11.6ppm over a duration of 24h [23]. In both cases, the frequency offset used by the servo to control the clock might exceed the non-synchronized stability requirements of Equation (5). This is handled in our model by defining, for any clock $\mathcal{H}_i$, another linear term in the form $(t-s)\rho_{3,i}$ in Equation (2), where $\rho_{3,i}$ upper-bounds the properties of the servo of clock $\mathcal{H}_i$. Then, ρ_i is redefined as $\rho_i = 1 + \rho_{1,i} + \rho_{2,i} + \rho_{3,i}$ and the network-wide parameter ρ is redefined accordingly.

Numerical Application to TSN: For tightly-synchronized networks, we take $\Delta = 1\mu s$ from [28, Normative Annex B.3]. For loosely-synchronized networks, we select the precision of NTP. In [42,

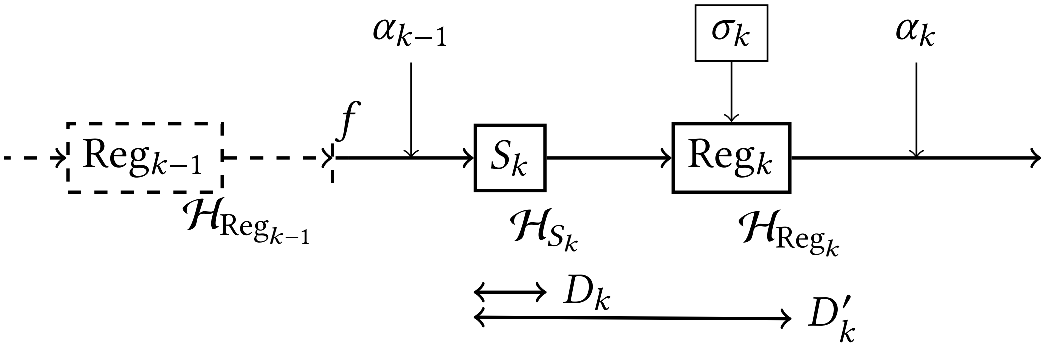


Fig. 4. Network calculus model for a regulated flow. At each hop, the flow goes through a network element S_k and is then regulated by the regulator Reg_k. Each device of the model has its own internal clock, noted at the bottom right.

Figure 27], NTP defines a "step threshold" of 125ms, and a survey of the NTP network performed in [48] notes that this value is hardly exceeded if the clock is synchronized [48, Section IV.B.1]. Hence, we take Δ =125ms.

Figure 3b presents, for a given known starting point $(s, d(s))$, the possible evolution space of $d(t)$ in the synchronized model as well as a possible trajectory. Note that the Δ envelope is not centered on the starting point but on the $d(t) = t$ function. Here again, we keep the symmetry noted previously on the non-synchronized constraints.

4.4 Network Model

We model the network as a set of "network elements" and "regulators". Each flow f has a non-cyclic path made of a series of connected network elements $(S_k)_{k=1\ldots n+1}$, with k the index of the network element in the path of the considered flow f. Each flow f is assumed to be processed by a regulator after each network element in its path, except the last one (Figure 4). We call "kth hop" of this flow the sequence $S_k - \mathrm{Reg}_k$. Each regulator can either be a PFR or an IR. For each flow f and each index $k = 1 \ldots n$, we note Reg_k the regulator that processes flow f after the network element S_k. On Reg_k, we assume that we can configure a shaping curve σ_k with a burst b_{Reg_k} and a rate r_{Reg_k} for this flow. Practical implementations support only limited accuracy, and we note $Q_k(b)$ [resp. $\mathcal{R}_k(r)$] the lowest value that is configurable by this regulator and that is higher than b [resp r]. By convention, Reg_0 denotes the source of the flow. When a set of flows share the same network element S, they do not need to share the same regulator just after S. However, when a set of flows share a same IR, we assume that they also share the same upstream network element and that this network element is FIFO for the considered set of flows.

For each network element S, we assume that if we know the arrival curve of each flow g going through S, then for any flow f going through S we can obtain a delay bound D_S^f for the flow through the network element.

Each network element and each regulator (including, for each flow f, its source Reg_0) is called a device. We assume that each device uses a local clock noted at the bottom right of the device, as in Figure 4. In particular, any regulator releases packets according to the regulation rules that depend on its type [2, 39] (see also Section 3), but the computations for its operation are performed using its local clock. When we need to detail the clock used to observe a delay, an arrival curve or a service curve, we put it in superscript.

Each flow f exits it source Reg_0 with a leaky-bucket arrival curve of rate $r_0 = r_{\text{Reg}_0}$ and burst $b_0 = b_{\text{Reg}_0}$, when observed with $\mathcal{H}_{\text{Reg}_0}$. For each flow f, we note α_k an arrival curve of the flow at the output of its k-th hop and D_k [resp D'_k] an upper bound on the delay for the flow through S_k [resp through the combination made of S_k and Reg_k]. As the flow f is not processed by any regulator after the last network element in its path S_{n+1}, for each flow f we note by convention $D'_{n+1} = D_{n+1}$.

Remark: This general model also applies in the specific case where several devices share the same clock, for instance if they are located within a same network node. We can indeed write that the clocks of the considered devices compose a sub-network, with parameters $\rho' = \rho$, $\eta' = \eta$, and that they are synchronized together as per the constraints of Equation (6) with a sub-network synchronization bound Δ' very small or null. However, the editors of IEEE P802.1Qcr (asynchronous traffic shaping) mention in the ongoing draft that even two devices sharing the same oscillator of a same node can have different notions of time [32, Section 8.6.11.2]. They also mention that this difference needs to be accounted for in their ad-hoc delay analysis in [32, Informative Annex V.6]. In such case, we take for the sub-network parameters $\rho' = \rho$, $\eta' = \eta$ and $\Delta' = \max(|\text{ClockOffsetMax}|, |\text{ClockOffsetMin}|)$, where ClockOffsetMax and ClockOffsetMin are defined in [32, Section 8.6.11.2].

5 NETWORK CALCULUS TOOLBOX FOR NETWORKS WITH NONIDEAL CLOCKS

The three-bound theorem of [40] is valid whenever all the notions used in the theorem are expressed using the same clock: the arrival curves, the service curves, and the delays. Therefore, we propose a toolbox that can be used to change the clock used to observe one of the above notions; this results in an extension of network calculus.

In the entire section, we consider a device j and a flow f entering this device (see Figure 5). Whenever f and j are unambiguous, the dependency of the defined functions and notions on j, f will be omitted. We also consider two clocks $\mathcal{H}_i$ and $\mathcal{H}_g$ and we denote with d the function $d_{g\to i}$.

5.1 Results on Delays

We are interested in a bound on the measure, with the TAI, of the delay that flow f suffers through device j. If a delay bound is known with a different clock $\mathcal{H}_i$, the following proposition enables us to retrieve a bound as seen with any clock $\mathcal{H}_g$ (especially with the TAI).

Proposition 5.1 (Changing the clock for a delay bound). *If $D^{\mathcal{H}_i}$ is an upper-bound on a delay measured with clock $\mathcal{H}_i$, then $(d \oslash d)(D^{\mathcal{H}_i})$ is an upper-bound on the delay measured with $\mathcal{H}_g$, where $\oslash$ denotes the min-plus de-convolution[1] and d is the function $d_{g\to i}$.*

The proof is in Appendix A.1. When a delay is observed with the TAI, we call it a TAI delay.

Remark: A measure of an end-to-end (ETE) delay is often a subtraction of two other measures: the time value given by a clock on the recipient's side when the packet arrives and the time value given by a clock located on the source side when the packet departs. If the clocks used to timestamps these two events are not perfectly synchronized with the TAI, the measured delay may violate a delay bound computed in the TAI using the work presented in this paper. Nevertheless, we consider the TAI delay as being the "true delay" experienced by the flow, and any real-world measurement should take into account the uncertainties of the measurement clocks as per the best-practices of the time metrology domain [26, 35].

[1] $f \oslash g : t \mapsto \sup_\tau [f(t+\tau) - g(\tau)]$, see [40]

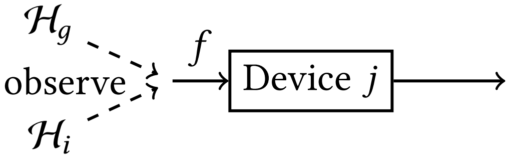


Fig. 5. Clocks $\mathcal{H}_g$ and $\mathcal{H}_i$ observe flow f entering device j

In the *Non-Synchronized Time Model*, we have, for any t, τ, $(d \oslash d)(\tau) \leq \rho\tau + \eta$ and thus $D^{\mathcal{H}_g} \leq \rho D^{\mathcal{H}_i} + \eta$. In the *Synchronized Time Model*, we obtain $(d \oslash d)(\tau) \leq \min(\rho\tau + \eta, \tau + 2\Delta)$ and thus $D^{\mathcal{H}_g} \leq \min\left(\rho D^{\mathcal{H}_i} + \eta, D^{\mathcal{H}_i} + 2\Delta\right)$.

Numerical Application to TSN: For both time models, for values of the delay bound $D^{\mathcal{H}_i}$ ranging from 1μsec to 200msec, the relative increase on the bound ranges from 0.4% to 0.02%. Practically, the effect of clock nonidealities on the *definition* of delay bounds in time-sensitive networks can thus be ignored.

5.2 Results on Arrival Curves

From [40], the cumulative function $R(t)$ of the flow entering the device j is defined, when measured with the TAI, as the number of bits entering the device between the instant measured as 0 in the TAI and the instant measured as t in the TAI. When we use clock $\mathcal{H}_i$ instead of TAI, we obtain a different cumulative function. We call $R^{\mathcal{H}_i}(t)$ the cumulative function obtained when counting the number of bits entering a device between the instant measured as 0 using $\mathcal{H}_i$ and the instant measured as t using $\mathcal{H}_i$ (Figure 5). The following proposition gives the relation between cumulative functions obtained with different clocks:

Proposition 5.2 (Changing the clock of a cumulative function). *For any clock pair $(\mathcal{H}_g, \mathcal{H}_i)$:*

$$\forall t, R^{\mathcal{H}_g}(t) = R^{\mathcal{H}_i}(d_{g\to i}(t))$$

The proof of the proposition is given in Appendix A.2. We now define the concept of an arrival curve observed with a clock.

Definition 5.3. We say that a wide-sense increasing function α is an arrival curve for the flow entering the device j when observed with clock $\mathcal{H}_i$ if $\forall t, \tau, R^{\mathcal{H}_i}(t+\tau) - R^{\mathcal{H}_i}(t) \leq \alpha(\tau)$. We note such a function $\alpha^{\mathcal{H}_i}$.

Remark: In particular, when $\mathcal{H}_i$ is the TAI, we retrieve the definition of an arrival curve, as defined in [40, Chapter 1].

Proposition 5.4 (Changing the clock of an arrival curve). *If $\alpha^{\mathcal{H}_i}$ is an arrival curve for the flow entering the device and being observed with $\mathcal{H}_i$, then an arrival curve for the flow observed with $\mathcal{H}_g$ is $\alpha^{\mathcal{H}_g} : t \mapsto \alpha^{\mathcal{H}_i}((d \oslash d)(t))$, where $\oslash$ is the min-plus de-convolution and $d = d_{g\to i}$.*

The proof of the proposition is available in Appendix A.3. We can now find an arrival curve for a flow as observed from any clock as long as we know one that is observed with one clock. We now apply the proposition to our two clock models.

Non-Synchronized Time Model. Here, for any t, τ, $(d \oslash d)(\tau) \leq (\rho\tau + \eta)$, thus, if $\alpha^{\mathcal{H}_i}$ is an arrival curve observed with $\mathcal{H}_i$, then $\alpha^{\mathcal{H}_g} : t \mapsto \alpha^{\mathcal{H}_i}(\rho t + \eta)$ is an arrival curve observed with $\mathcal{H}_g$.

Example: *Application to leaky-bucket arrival curves.* Assume $\alpha^{\mathcal{H}_i}$ is a leaky-bucket arrival curve with burst b_i and rate r_i, then $\alpha^{\mathcal{H}_g}$ is also a leaky-bucket arrival curve with burst $b_g = b_i + r_i\eta \geq b_i$ and a rate $r_g = \rho r_i \geq r_i$. We can also write $\alpha^{\mathcal{H}_g} = \gamma_{\rho r_i, b_i + r_i\eta}$.

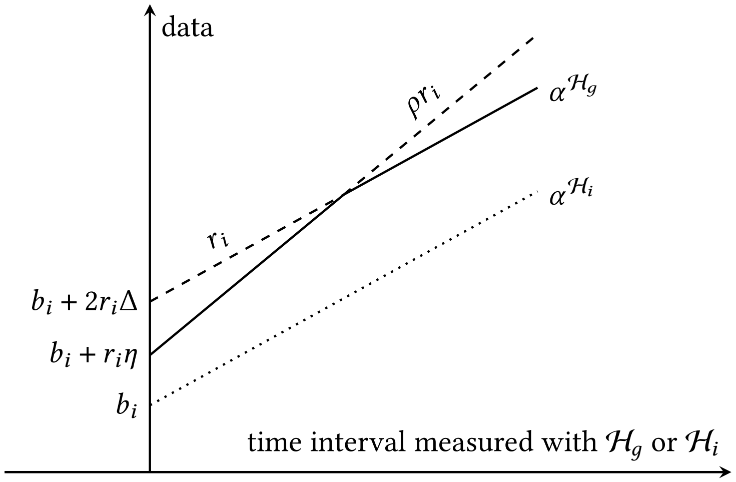


Fig. 6. $\mathcal{H}_i$ is synchronized with $\mathcal{H}_g$. A flow f has the leaky-bucket arrival curve $\alpha^{\mathcal{H}_i}$ when it is viewed from $\mathcal{H}_i$. Then, f has the arrival curve $\alpha^{\mathcal{H}_g}$ when it is viewed from $\mathcal{H}_g$.

Table 2. Relations between an arrival curve as observed with $\mathcal{H}_i$ and an arrival curve as observed with $\mathcal{H}_g$.

	Arrival curve	
	General	Leaky-Bucket
in $\mathcal{H}_i$	$\alpha^{\mathcal{H}_i}(t)$	$\gamma_{r,b}$
in $\mathcal{H}_g$, general	$\alpha^{\mathcal{H}_i}((d_{g\to i} \oslash d_{g\to i})(t))$	–
in $\mathcal{H}_g$, non-sync	$\alpha^{\mathcal{H}_i}(\rho t + \eta)$	$\gamma_{r\rho,b+r\eta}$
in $\mathcal{H}_g$, sync	$\alpha^{\mathcal{H}_i}(\min[\rho t + \eta, t + 2\Delta])$	$\gamma_{r\rho,b+r\eta} \wedge \gamma_{r,b+2r\Delta}$

Numerical Application to TSN: This represents 0.02% of a rate increase and a burst increase that is below 1 bit for most flows (below 250Mbits/s) as η =4ns.

Synchronized Time Model. Here, we additionally obtain $\forall\tau, (d \oslash d)(\tau) \leq \tau + 2\Delta$. Consequently, if $\alpha^{\mathcal{H}_i}$ is an arrival curve observed with $\mathcal{H}_i$, then $\alpha^{\mathcal{H}_g} : t \mapsto \alpha^{\mathcal{H}_i}(\min(t + 2\Delta, \rho t + \eta))$ is an arrival curve observed with $\mathcal{H}_g$. As arrival curves are wide-sense increasing, we also have $\alpha^{\mathcal{H}_g} : t \mapsto \min\left[\alpha^{\mathcal{H}_i}(t + 2\Delta), \alpha^{\mathcal{H}_i}(\rho t + \eta)\right]$.

Example: *Application to leaky-bucket arrival curves.* Assume $\alpha^{\mathcal{H}_i}$ is a leaky-bucket arrival curve of burst b_i and rate r_i. Then the function $t \mapsto \alpha^{\mathcal{H}_i}(t + 2\Delta)$ is a leaky-bucket arrival curve of rate r_i and burst $b_i + 2r_i\Delta$. And $t \mapsto \alpha^{\mathcal{H}_i}(\rho t + \eta)$ is a leaky-bucket arrival curve of rate $r = \rho r_i$ and burst $b_i + r_i\eta$. Hence, $\alpha^{\mathcal{H}_g} = \gamma_{r_i,b_i+2r_i\Delta} \wedge \gamma_{\rho r_i,b_i+r_i\eta}$, the minimum of these two arrival curves (Figure 6). According to the numerical application, we expect to have $2r_i\Delta$ much higher than $r_i\eta$. In Figure 6, we respect this order but not the scale.

Numerical Application to TSN: In a synchronized network, we have a second additional arrival curve with an unchanged rate but with an increased burst. For a flow of 500kbits/s, this represents a burst increase of 125kbits for a loosely-synchronized network and 1bit for a tightly-synchronized network. For loosely-synchronized networks, due to the high burst increase, the other part of the arrival curve, $\gamma_{\rho r_i,b_i+r_i\eta}$ needs to be used in order to obtain tight bounds.

Table 2 regroups the results of this subsection. Note that the table can be used for any pair of clocks $(\mathcal{H}_g, \mathcal{H}_i)$, each being either a clock in the network or the TAI.

5.3 Results on Service Curves

As for the arrival curve concept, we define the service curve concept relative to a clock:

Definition 5.5. We say that a wide-sense increasing function β is a service curve of the device j for the flow when observed with $\mathcal{H}_i$ if $\beta(0) = 0$ and $\forall t \geq 0, R^{*\mathcal{H}_i}(t) \geq \inf_{0\leq s\leq t} R^{\mathcal{H}_i}(s) + \beta(t-s)$. We note such a function $\beta^{\mathcal{H}_i}$.

Remark: In particular, when $\mathcal{H}_i$ is the TAI, we retrieve the definition of a service curve, as defined in [40, Chapter 1].

Proposition 5.6 (Changing the clock for a service curve). *If $\beta^{\mathcal{H}_i}$ is a service curve observed with $\mathcal{H}_i$, then a service curve observed with $\mathcal{H}_g$ is $\beta^{\mathcal{H}_g} : t \mapsto \beta^{\mathcal{H}_i}((d\overline{\oslash}d)(t))$, where $d = d_{g\to i}$ and $d\overline{\oslash}d(t) = \inf_{u\geq 0}[d(t+u) - d(u)]$ (max-plus de-convolution [40, Section 3.2.1]).*

The proof is in Appendix A.4. Again, we now apply the proposition to the two time models:

Non-Synchronized Time Model. Here, $\forall t, (d\overline{\oslash}d)(t) \geq \frac{1}{\rho}|t-\eta|^+$, thus if $\beta^{\mathcal{H}_i}$ is a service curve viewed from $\mathcal{H}_i$, then $\beta^{\mathcal{H}_g} : t \mapsto \beta^{\mathcal{H}_i}(\frac{1}{\rho}|t-\eta|^+)$ is a service curve viewed from $\mathcal{H}_g$.

Example: *Application to rate-latency service curves.* Assume $\beta^{\mathcal{H}_i}$ is a rate-latency service curve with rate R_i and latency T_i, then $\beta^{\mathcal{H}_g}$ is a rate-latency service curve with rate $R_g = R_i/\rho \leq R_i$ and latency $T_g = \eta + \rho T_i \geq T_i$.

Numerical Application to TSN: The service observed with another clock has a guaranteed rate reduced by 0.02% and a slightly increased latency. For a latency of 10μs, this represents 6ns of added latency.

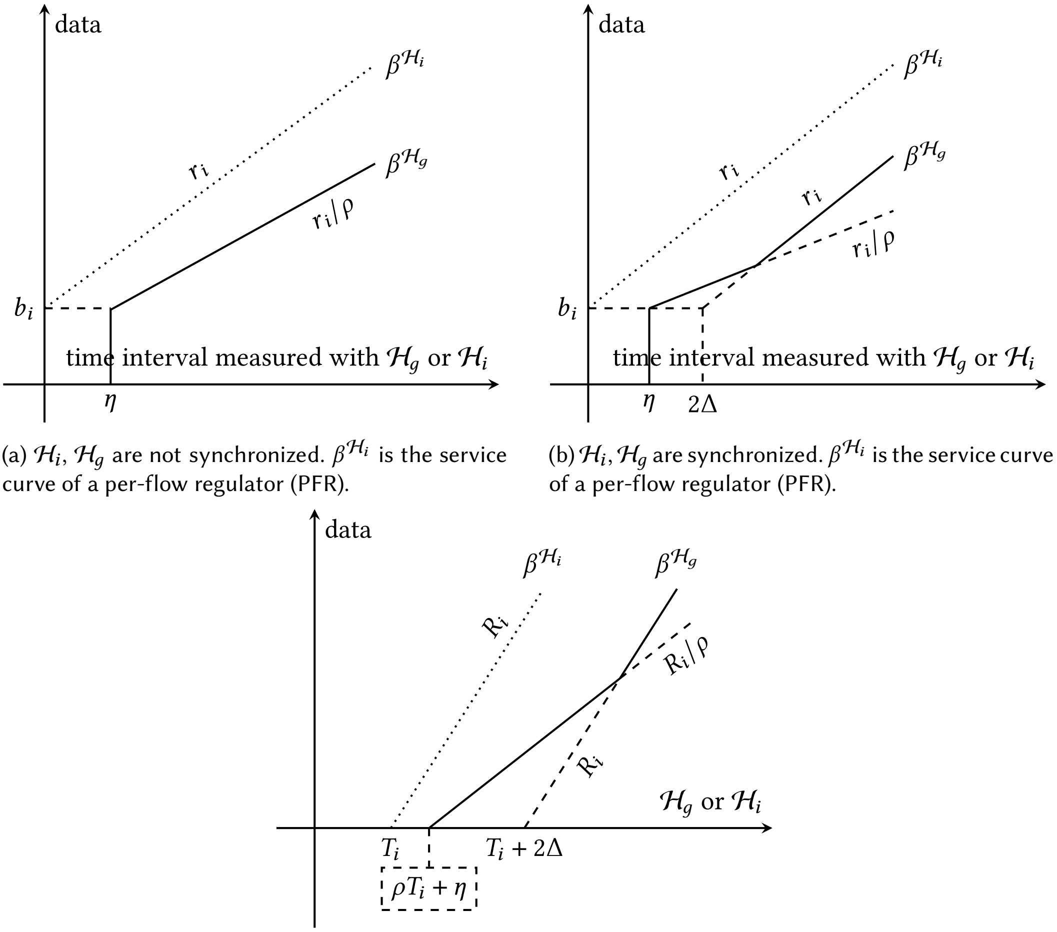
Example: *Application to PFR service curves.* Recall from Section 3 that, when observed with its internal clock, a PFR offers to the regulated flow a service curve equal to its shaping curve. If we observe a token-bucket filter with rate r_i and burst b_i with a different clock, $\mathcal{H}_g$, we obtain the service curve $\beta^{\mathcal{H}_g} = \delta_\eta \otimes \gamma_{r_i/\rho, b_i}$. This corresponds to the leaky-bucket service curve with a burst b_i and a service rate $r_g = r_i/\rho \leq r_i$, delayed by delay η (Figure 7a).

Synchronized Time Model. If $\beta^{\mathcal{H}_i}$ is a service curve viewed from $\mathcal{H}_i$, then $\beta^{\mathcal{H}_g} : t \mapsto \beta^{\mathcal{H}_i}(\max[\frac{1}{\rho}(t-\eta), t-2\Delta, 0])$ is a service curve viewed from $\mathcal{H}_g$.

Example: *Application to rate-latency service curves.* Assume $\beta^{\mathcal{H}_i}$ is a rate-latency service curve with rate R_i and latency T_i, then a service curve as observed with $\mathcal{H}_g$ is $\forall t, \beta^{\mathcal{H}_g}(t) = \lambda_{R_i/\rho, \eta+\rho T_i} \vee \lambda_{R_i, T_i+2\Delta}$ where $\vee$ denotes the maximum. I.e., $\beta^{\mathcal{H}_g}$ is the maximum between the rate-latency service curve of rate R_i/ρ, latency $\eta + \rho T_i$ and the rate-latency service curve of rate R_i and latency $T_i + 2\Delta$. Note that if $\rho - 1$ is small and with our previous remark in Section 5.2 about η and Δ, we expect the latency $\eta + \rho T_i$ to be less than the latency $T_i + 2\Delta$. Hence, the shape of $\beta^{\mathcal{H}_g}$ is given in Figure 7c.

Numerical Application to TSN: In a synchronized network, compared to the non-synchronized time model, we also obtain a service with an unchanged rate but an increased latency: 2μs for a tightly-synchronized network and 300ms for a loosely-synchronized network.

Example: *Application to PFR service curves.* Assume $\beta^{\mathcal{H}_i}$ is a service curve offered by a token-bucket filter observed with its internal clock, with burst b_i and rate r_i, then a service curve of this PFR as observed with $\mathcal{H}_g$ is $\forall t, \beta^{\mathcal{H}_g}(t) = b_i + r_i \max[\frac{1}{\rho}(t-\eta), t-2\Delta, 0]$. Ie, $\beta^{\mathcal{H}_g} = (\gamma_{r_i/\rho, b_i} \otimes \delta_\eta) \vee (\gamma_{r_i, b_i} \otimes \delta_{2\Delta})$ (Figure 7b). Table 3 regroups the results of this subsection. Note that the table can be used for any pair of clocks $(\mathcal{H}_g, \mathcal{H}_i)$, each being either a clock in the network or the TAI.



(a) $\mathcal{H}_i$, $\mathcal{H}_g$ are not synchronized. $\beta^{\mathcal{H}_i}$ is the service curve of a per-flow regulator (PFR).

(b) $\mathcal{H}_i$, $\mathcal{H}_g$ are synchronized. $\beta^{\mathcal{H}_i}$ is the service curve of a per-flow regulator (PFR).

(c) $\mathcal{H}_i$, $\mathcal{H}_g$ are synchronized. $\beta^{\mathcal{H}_i}$ is a rate-latency service curve.

Fig. 7. If the system offers service curve $\beta_{\mathcal{H}_i}$ when observed with $\mathcal{H}_i$, it offers service curve $\beta_{\mathcal{H}_g}$ when observed with $\mathcal{H}_g$.

Table 3. Relations between a service curve of a system observed with $\mathcal{H}_i$ and a service curve of the same system observed with $\mathcal{H}_g$.

	Service curve		
	General	Rate-Latency	Leaky-Bucket
in $\mathcal{H}_i$	$\beta^{\mathcal{H}_i}(t)$	$\lambda_{R,T}$	$\gamma_{r,b}$
in $\mathcal{H}_g$, general	$\beta^{\mathcal{H}_i}((d_{g\to i}\overline{\oslash}d_{g\to i})(t))$	–	–
in $\mathcal{H}_g$, non-sync	$\beta^{\mathcal{H}_i}(1/\rho \cdot \lvert t-\eta\rvert^{+})$	$\lambda_{R/\rho,\rho T+\eta}$	$\delta_\eta \otimes \gamma_{r/\rho,b}$
in $\mathcal{H}_g$, sync	$\beta^{\mathcal{H}_i}(\max[1/\rho \cdot (t-\eta), t-2\Delta, 0])$	$\lambda_{R/\rho,\rho T+\eta}$ $\vee\lambda_{R,T+2\Delta}$	$(\delta_\eta \otimes \gamma_{r/\rho,b})$ $\vee(\delta_{2\Delta} \otimes \gamma_{r,b})$

5.4 Results on Regulators

Recall that the operation of a PFR requires measuring times, therefore, a system that is a PFR when observed with its own clock might no longer be a PFR when observed with a different clock. Indeed, assume, for instance, that the PFR's clock runs faster than expected when observed with an external clock. Then the external observer will see that the device violates the configured shaping curve and cannot be a PFR. Conversely, if the clock of the device is two slow from an external point of view, then the external observer will conclude that the device is not a PFR because it does not release the packets as soon as possible. Worse, the device clock may oscillate between being too fast or being too slow.

Similar observations can be done with the IR. Hence, the property of being a PFR or an IR is only valid when the regulator is observed with its internal clock. Consequently, none of the properties of the IR [39] or the PFR [40, Sections 1.5 and 1.7] are expected to hold when observed with an external clock, in general.

One of them states that, at the ouput of the regulator, a flow has the shaping curve as arrival curve. Observed with an external clock, this property does not hold but an arrival curve as observed with this external clock can be retrieved using Table 2.

In the following sections, we focus on the shaping-for-free property (Section 3), as it is underlies all delay computations in deterministic networks with regulators. Proposition 6.1 proves that in non-synchronized networks, the shaping-for-free property holds neither for a PFR nor for an IR. For synchronized networks, Section 7.1 computes a lower and an upper bound on the worst-case penalty incurred by a PFR. Finally, Proposition 7.3 proves that the IR cannot provide any delay bound.

6 NON-SYNCHRONIZED NETWORKS WITH REGULATORS

We now combine the set of results of the previous section with other network calculus results to analyze non-synchronized networks containing regulators. We focus on a network with flows that are leaky-bucket constrained at their sources.

6.1 Instability of Non-Adapted Regulators in Non-Synchronized Networks

Proposition 6.1 (Instability of non-adapted regulators in non-synchronized networks). *Consider a non-synchronized network with $\rho > 1$. Consider a network element S that is FIFO per flow and guarantees to a flow f a delay $\leq D$ if the flow satisfies an arrival curve $\gamma_{r,b}$ (both measured in TAI). After processing by S, the flow is submitted to a non-adapted regulator (PFR or IR) (Figure 1) with shaping curve $\gamma_{r,b}$. There exists adversarial source clocks within our time model and adversarial traffic generation satisfying the $\gamma_{r,b}$ arrival curve such that the delay of the flow through the regulator, measured using any clock, is unbounded.*

The proof is in Appendix A.5. Recall that if the clocks would be ideal, the worst-case delay of the flow would not be increased by the regulator and the total delay would thus be $\leq D$. In contrast, with nonideal clocks, the total delay is unbounded, thus "shaping-for-free" does not hold.

As illustration, consider for simplicity a PFR or an IR regulating only one flow: in this situation, both devices present the same behavior. We consider the source of a leaky-bucket constrained flow with all packets of same size, burst $b = 1$ packet and rate $r = 1$ packet per time unit. In the first line of Figure 8, we show the traffic trace at the output of the source, observed with its clock. The source is greedy, its sends at its maximum rate of one packet per time unit.

Now assume that the FIFO system S has zero delay, whatever the clock used to observe it. Then the first timeline in Figure 8 is also the traffic trace at the input of the regulator, but observed with the source clock. Assume now that the clock of the regulator is running slower than the one of the

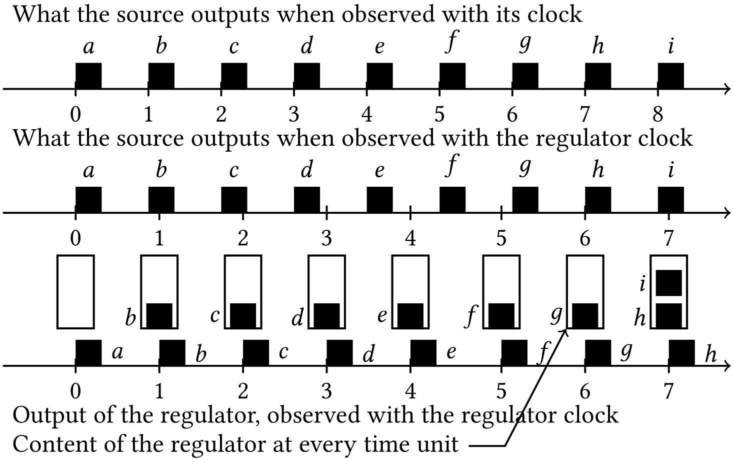

What the source outputs when observed with its clock
a
b
c
d
e
f
g
h
i
0
1
2
3
4
5
6
7
8
What the source outputs when observed with the regulator clock
a
b
c
d
e
f
g
h
i
0
1
2
3
4
5
6
7
b
c
d
e
f
g
i
h
a
b
c
d
e
f
g
h
0
1
2
3
4
5
6
7
Output of the regulator, observed with the regulator clock
Content of the regulator at every time unit

Fig. 8. Adversarial case triggering the instability for non-synchronized non-adapted regulators.

source. Here we exaggerate the phenomenon and we take $d_{\text{source}\to\text{regulator}}(t) = s_0\, t$ with $s_0 = 6/7$. The regulator is non-adapted. It enforces the arrival curve of the source (one packet of burst, one packet per unit of time) by using its internal clock.

Seen with the regulator clock, the incoming traffic trace is given with the second timeline in Figure 8. Packet b appears to arrive too soon (at time 0) and needs to be stored in the regulator. It is released when the regulator clock reaches time 1. When measured with the regulator clock, packet b suffers a delay of $1/7$ time unit, and $1/6$ time unit when observed with the source clock.

We note that each packet has an increasing delay. At local time 7, the regulator has now two packets in its buffer, and it will increase to three packets at time 14, and so forth. We observe that the delays and the buffer occupation is linearly increasing. Hence, we have an unstable system with no delay bound. In practice, buffer sizes are finite and instead of unbounded delay, we will see unexpected packet losses (this contradicts the purpose of time-sensitive networks, which assume zero loss due to buffer overflow).

Numerical Application to TSN: Looking at the example, we observe that the delays observed with the source clock increase linearly with time and with $1/s_0 - 1$, which can be as large as $\rho - 1$. In a TSN context, this represents approximately 200μs of increased worst-case delay per second of operation.

The example presented above is inspired by a remark made in the TSN ATS draft [32, Annex V.8]: "*If the upstream device [...] runs faster than nominal and [the] downstream Bridge [...] runs slower than nominal, the backlog as well as the per hop delay in the downstream Bridge could grow under peak conditions*". In our model, the "upstream device" represents the upstream regulator, or the source, the "downstream Bridge" represents the next regulator, and "peak conditions" refer to a greedy source.

6.2 The Rate and Burst Cascade for Non-Synchronized Networks with Regulators

This last quoted remark highlights how a first solution for the instability problem can be formulated: one can make sure that whatever the clock conditions (but within the constraints of Equation (5)), the downstream device will always have an output rate higher than the input. This requires increasing slightly the nominal rate of the regulator; and because this increase is performed at every hop, it generates a rate cascade that was first described in [32, Annex V.8]. In this section, we refine the method into a rate an burst cascade, with the following differences:

- We consider the jitter tolerance η of the clocks, unlike [32].
- The discussion in [32, Annex V.7], that is related to the difference between the theoretical IR behavior and its concrete implementation in ATS is not considered here.
- We take into account the finite resolution of the configurable rates and bursts on the regulators, whereas [32] assumes that any rate or burst is configurable.

Last, we use the network calculus toolbox in Section 5 to prove that the rate burst cascade ensures the stability of the network, and we also compute per-hop delay bounds.

The rate and burst cascade works as follows. For each flow f, we use the notation and the reference configuration in Section 4.4 and Figure 4.

Step 1: For each flow f, and each hop $k = 1 \dots n$ in its path, configure Reg_k with $r_{\mathrm{Reg}_k} = \mathcal{R}_k(\rho r_{\mathrm{Reg}_{k-1}})$ and $b_{\mathrm{Reg}_k} = \mathcal{Q}_k(b_{\mathrm{Reg}_{k-1}} + \eta r_{\mathrm{Reg}_{k-1}})$. Recall that $\mathcal{R}_k(r)$ [resp. $\mathcal{Q}_k(b)$] denote the configurable rate [resp. burst] higher than r [resp. b] for this regulator. Recall also that ρ and η are network-wide parameters that depend neither on the considered clock nor on k.

Step 2: For each flow f, and each hop $k = 1 \dots n$ in its path, the configured shaping curve σ_{k-1} is an arrival curve at the output of the regulator Reg_{k-1}, when observed with the clock of the regulator (Section 5.4). Using Table 2 with $\mathcal{H}_g = \mathcal{H}_{\mathrm{TAI}}$ and $\mathcal{H}_i = \mathcal{H}_{\mathrm{Reg}_{k-1}}$, it follows that flow f has a leaky-bucket arrival curve $\alpha_k^{\mathcal{H}_{\mathrm{TAI}}}$ of rate $\rho r_{\mathrm{Reg}_{k-1}}$ and burst $b_{\mathrm{Reg}_{k-1}} + \eta r_{\mathrm{Reg}_{k-1}}$ at the input of S_k, when observed with $\mathcal{H}_{\mathrm{TAI}}$.

For each network element S in the network and each flow f' crossing S, the arrival curve of f' at the input of S, observed with $\mathcal{H}_{\mathrm{TAI}}$ is given by the above $\alpha_k^{\mathcal{H}_{\mathrm{TAI}}}$, with k the index of S in the path of flow f'. According to the assumptions of Section 4.4, we can compute a TAI delay bound $D_S^{f,\mathcal{H}_{\mathrm{TAI}}}$ of any flow f that goes through S.

Step 3: For each flow f, and each hop $k = 1 \dots n$ in its path, compute the TAI delay bound, $D_k'^{\mathcal{H}_{\mathrm{TAI}}}$, of the flow through the sequence $S_k - \mathrm{Reg}_k$, using the next proposition:

Proposition 6.2. *If regulators are configured as in Step 1, for each flow f that goes through network element S, a bound on the TAI delay of the flow through the concatenation of S and the next regulator is $D'^{f,\mathcal{H}_{\mathrm{TAI}}} = \rho^2 D^{f,\mathcal{H}_{\mathrm{TAI}}} + \eta(1+\rho)$ where $D^{f,\mathcal{H}_{\mathrm{TAI}}}$ is a bound on the TAI delay of flow f through S, computed in Step 2.*

The proof is in Appendix A.6. Intuitively, with the clock of the regulator, the arrival curve of the flow at the input of S is $\leq$ the shaping curve of the regulator, due to the inflation of rate and burst; hence shaping-for-free holds with this clock.

Numerical Application to TSN: Assume that the regulators' configurations have infinite precision ($\mathcal{R}, \mathcal{Q}$ are the identity function). With these settings, the rate and burst cascade method increases the rate [resp the burst] of the flow by approx. 0.02% [resp less than one bit for most flows] at each hop.

The rate and burst cascade has the drawback that the configuration of a regulator depends on its position on the flow path. This puts complexity on the control plane, specifically, for computing, distributing and managing the configuration. This motivates us to propose next an alternative method which, however, works only with PFRs.

6.3 The ADAM method for Non-Synchronized Networks with Per-Flow Regulators

The goal of the asynchronous dual arrival-curve method (ADAM) is, for any given flow, to have the same parameters at all regulators along the flow path. Thus,when applying ADAM, we require

Algorithm 1 Computing the TAI delay for a flow through its hop m using the ADAM method

```
Require: $\{D_k^{\mathcal{H}_{\mathrm{TAI}}}\}_k$, the set of all the TAI delay bounds for the flow through the systems $S_k$ in its
    path (from Step 2).
Require: $m$, the index at which to compute the TAI hop delay.
Require: $r_0$, $b_0$, the rate and burst of the flow at the source, observed with the source's clock. $W$,
    the rate margin.
Require: $\eta$, $\rho$ the network-wide parameters of the time-model.
1: function ComputeDelayHopM($\{D_k^{\mathcal{H}_{\mathrm{TAI}}}\}_k$,$m$,$(r_0, b_0)$,$W$,$\rho$,$\eta$)
2:     $r_2 \leftarrow \rho r_0$
3:     $b_{2,0} \leftarrow b_0 + \eta r_0$
4:     for $k = 1 \dots m$ do
5:         $D_k'^{\mathcal{H}_{\mathrm{TAI}}} \leftarrow D_k^{\mathcal{H}_{\mathrm{TAI}}} + \eta(1+\rho) + \frac{b_{2,k-1} - b_0 - \eta W r_0}{\rho r_0} \frac{\rho^2 - 1}{W - 1}$
6:         $b_{2,k} \leftarrow b_{2,k-1} + \rho r_0 \cdot D_k'^{\mathcal{H}_{\mathrm{TAI}}}$        ▷ See Proposition 6.3
7:     end for
8:     return $D_m'^{\mathcal{H}_{\mathrm{TAI}}}$
9: end function
```

that the rounding functions $\mathcal{R}_k, \mathcal{Q}_k$ (which capture the accuracy with which regulator parameters are actually implemented) are the same at all network nodes (and we consequently drop the index k for these functions). We also require that all the regulators in the network be PFRs.

The main idea of ADAM is to establish that each flow f has an arrival curve, expressed in TAI, of the form $\alpha_k = \alpha_1 \wedge \alpha_{2,k}$ at the output of its k-th hop where α_1, $\alpha_{2,k}$ are leaky-bucket arrival curves and the former is independent of the hop index.

Step 1: For each flow f, find a rate margin W such that $W \geq \rho^2$ and Wr_0 can be exactly implemented, i.e $\mathcal{R}(Wr_0) = Wr_0$. Configure the shaping curves at all regulators along the path of the flow with rate $r_{\mathrm{Reg}_k} = Wr_0$ and burst $b_{\mathrm{Reg}_k} = b_0$. Since ρ is a network-wide parameter that does not depend on a clock, all regulators except the source have the same configuration, independent of the hop index $k = 1 ... n$. Here r_0, b_0 are the rate and burst at the source (which depend on f, though the dependency on f is not shown for simplicity of notation).

Step 2: Any flow f that goes through a network element S is output by the regulator at its previous hop. Thus, using the same justification as step 2 of the rate and burst cascade, it has the arrival curve $\alpha_1 = \gamma_{\rho W r_0, b_0 + \eta W r_0}$ (i.e. leaky bucket with rate $\rho W r_0$ and burst $b_0 + \eta W r_0$), when observed with $\mathcal{H}_{\mathrm{TAI}}$.

Then, again using the same justification as step 2 of the rate and burst cascade, compute a TAI delay bound $D_S^{f,\mathcal{H}_{\mathrm{TAI}}}$ through any network element S and for any flow f that goes through it.

Step 3: For each flow f and each hop $k = 1 \dots n$ in its path, compute a TAI delay bound, $D_k'^{\mathcal{H}_{\mathrm{TAI}}}$, of the flow through the sequence $S_k - \mathrm{Reg}_k$, using Algorithm 1.

Proposition 6.3 (Correctness of Algorithm 1). *For a flow f that has n hops and for $m = 1 \dots n$:*

(1) *Let $\alpha_{2,0}$ be the leaky-bucket function with rate r_2 (Line 2) and burst $b_{2,0}$ (Line 3); it is an arrival curve for the flow at its source when observed with $\mathcal{H}_{TAI}$.*
(2) *Let $\alpha_{2,k}$ be the leaky-bucket function with rate r_2 (Line 2) and burst $b_{2,k}$ (Line 6). It is an arrival curve of the flow observed with $\mathcal{H}_{TAI}$ at the output of Reg_k.*
(3) *$D_k'^{\mathcal{H}_{TAI}}$ is a TAI delay bound for the flow through the concatenation $S_k - Reg_k$, for $k = 1 \dots m$.*

The proof is in Appendix A.7. Observe that, unlike with the rate and burst cascade method, here the regulators do increase (slightly) the delay bound, specifically, shaping-for-free does not hold. The proof captures this increase by using a service-curve characterization of PFRs, together with the arrival curve $\alpha_1 \wedge \alpha_{2,k-1}$ for the flow at the input of S_k observed with $\mathcal{H}_{\text{TAI}}$. Then it applies the Network Calculus results using these curves observed with $\mathcal{H}_{\text{TAI}}$. As we do not know any service-curve characterization for IRs, we are not able to extend this method to IRs.

Also note that the delay bound in Step 2 is computed using the arrival curve α_1 and not the full arrival curve known by the method. This is because Step 2 is performed at every node in the network, before knowing the results of Step 3 that is performed per-flow. Using the result of Step 3 in Step 2 is possible in feed-forward networks and might in some cases lead to slightly smaller delay bounds. However, one of the major applications of regulators is in non-feedforward networks. Therefore, we do not explore such possible optimizations in this paper.

Numerical Application to TSN: We consider a network with one unique flow going through n nodes, each with a FIFO system and a regulator. Figure 9 compares, for one example and for both ADAM and the rate and burst cascade, the increase of the end-to-end TAI delay bound with respect to the theoretical situation with ideal clocks. The delay-bound increase is larger with ADAM but in both cases it is very small (less than 4% for paths smaller than 10 hops).

7 SYNCHRONIZED NETWORKS WITH REGULATORS

In synchronized networks, we expect that unbounded delays due to non-adapted regulators cannot occur, as clock rates cannot diverge for long periods of time. We now examine to which extent this holds. We study separately PFRs and IRs.

7.1 Delay Penalty of Non-Adapted PFRs in Synchronized Networks

Even when synchronized, a non-adapted PFR increases the worst-case delay, i.e. does not "shape for free":

Proposition 7.1 (Lower bound on the worst-case delay penalty of synchronized non-adapted per-flow regulators). *For any leaky-bucket arrival curve $\gamma_{r,b}$, there exists a network configuration such that (1) one flow has the arrival curve $\gamma_{r,b}$ at the source, (2) the flow goes through one network element followed by one non-adapted PFR (hence with shaping curve $\gamma_{r,b}$), (3) the clocks of the source and the PFR are synchronized with time-error bound Δ, and (4) the PFR increases the worst-case delay of the flow by at least Δ.*

The proof is in Appendix A.8. We give here a less-formal example. We take the same example as for Section 6.1 and Figure 8, with the source clock being the TAI, the source being greedy, and the system S_k having no delay. But this time (Figure 10), the regulator clock is too slow by $s_1 = 6/7$ between Time Units 0 and 7 (measured with the TAI), and then it regains the same speed, with a time function $d_{\text{Source}\rightarrow\text{ PFR}}(t) = t - \Delta$. We exaggerate Δ to equal one unit of time. Then we can note that $d_{\text{Source}\rightarrow\text{ PFR}}(t)$ meets Equations (5) and (6) with $\rho \geq s_1$. We observe in Figure 10 that the TAI delay of each packet is increasing for packets from a to h, and packets h to j have a delay of one unit, that is Δ.

Thanks to the synchronization, though, the delay penalty of the PFR can be controlled:

Proposition 7.2 (Upper bound on the delay penalty of synchronized non-adapted per-flow regulators). *Assume a flow is regulated at the source with rate r_0 and burst b_0 and goes through a sequence of n concatenations of network elements and regulators. The regulators are nonadapted (i.e. have shaping curve γ_{r_0,b_0}), and the network is synchronized with time-error bound Δ. Let $D_k^{\mathcal{H}_{TAI}}$ be an*

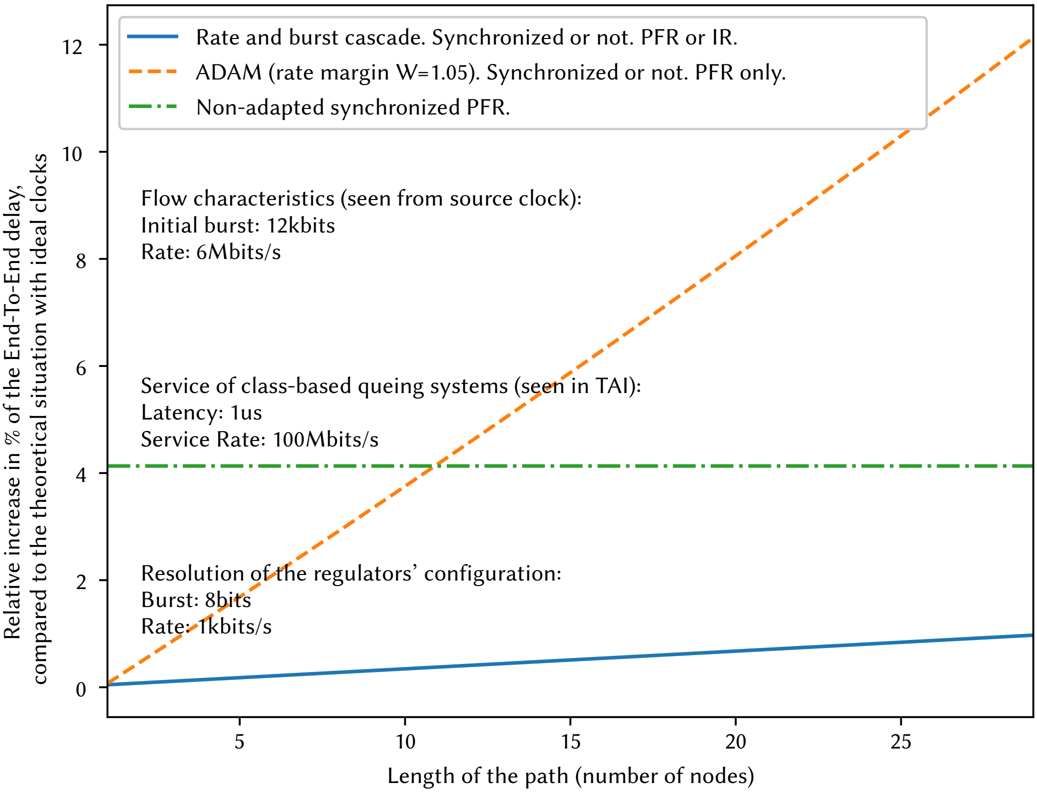


Fig. 9. Comparison of the relative increase of the TAI end-to-end delay bound of a flow obtained with several methods, with respect to the theoretical situation with ideal clocks, and as a function of the path length.

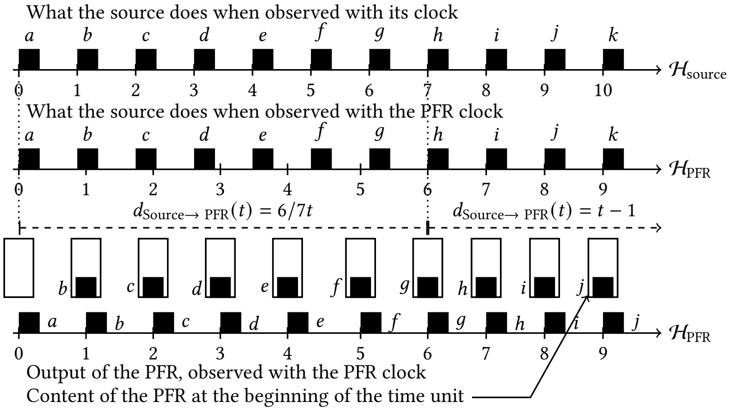

What the source does when observed with its clock
a b c d e f g h i j k
0 1 2 3 4 5 6 7 8 9 10
$\mathcal{H}_{\text{source}}$
What the source does when observed with the PFR clock
a b c d e f g h i j k
0 1 2 3 4 5 6 7 8 9
$\mathcal{H}_{\text{PFR}}$
$d_{\text{Source}\to\text{ PFR}}(t) = 6/7t$
$d_{\text{Source}\to\text{ PFR}}(t) = t-1$
b c d e f g h i j
a b c d e f g h i j
0 1 2 3 4 5 6 7 8 9
$\mathcal{H}_{\text{PFR}}$
Output of the PFR, observed with the PFR clock
Content of the PFR at the beginning of the time unit

Fig. 10. Adversarial case achieving the lower penalty bound for synchronized non-adapted PFRs.

upper bound on the TAI delay at S_k, the kth network element on the path of the flow. A bound on the TAI delay of the flow through the concatenation of S_k and the next regulator is $D_k'^{\mathcal{H}_{TAI}} = D_k^{\mathcal{H}_{TAI}} + 4\Delta$.

The proof is in Appendix A.9. Note that a TAI delay bound $D_k^{\mathcal{H}_{\text{TAI}}}$ can be obtained by using the fact that the shaping curve is a valid arrival curve at the output of the previous regulator. Hence we can use the last line of Table 2; when observed with $\mathcal{H}_{\text{TAI}}$, the flow enters S_k with a double arrival-curve constraint: a leaky-bucket arrival curve of rate ρr_0 and burst $b_0 + r_0\eta$, and a leaky-bucket arrival curve of rate r_0 and burst $b_0 + 2r_0\Delta$.

It follows from Propositions 7.1 and 7.2 that the worst-case penalty on the TAI per-hop delay of non-adapted PFRs in synchronized networks is between Δ and 4Δ, i.e., is of the order of magnitude of the synchronization precision. For tightly-synchronized networks and PFRs, the current practice of ignoring clock nonidealities is thus perfectly acceptable. However, in loosely-synchronized networks, the value of Δ ($\sim$ 125ms) is larger than the required delay bound for flows with stringent delay requirements. The two solutions (rate and burst cascade, and ADAM) that apply to non-synchronized networks also apply here and can be used.

Numerical Application to TSN: In Figure 9, we compare, with the same conditions as in Section 6.3, the delay bound with non-adapted tightly-synchronized PFRs, obtained with Proposition 7.2 with the methods of Section 6. We observe that they perform similarly, each being within a few percents of the ideal-clocks case.

7.2 Instability of Non-Adapted IRs in Synchronized Networks

For the interleaved regulator, however, the conclusions are very different. Indeed, the IR may yield unbounded latencies, even with tightly-synchronized networks, as shown in the following proposition.

Proposition 7.3 (Instability of non-adapted synchronized interleaved regulators). *Consider an interleaved regulator as in Figure 11, with n upstream systems. Each upstream system j outputs P_j flows $\{f_{j,p}\}_{p=1}^{P_j}$, each with a known arrival curve $\alpha_{f_{j,p}}^{\mathcal{H}_j}$ when observed with clock $\mathcal{H}_j$. Assume the interleaved regulator is non-adapted: $\forall(j,p), \sigma_{f_{j,p}}^{\mathcal{H}_{IR}} = \alpha_{f_{j,p}}^{\mathcal{H}_j}$. Finally, assume that the clocks $\mathcal{H}_{IR}$,$\mathcal{H}_{FIFO}$ and each of the $\mathcal{H}_j$ are synchronized (as per Equations (5) and (6)). Then, for any parameters n, ρ, η, Δ with $n \geq 3$, $\rho > 1$, $\eta \geq 0$ and $\Delta > 0$, there exists a FIFO system, adversarial clocks for $\mathcal{H}_{IR}$,$\mathcal{H}_{FIFO}$ and $\{\mathcal{H}_j\}_j$, and adversarial traffic generation of the upstream systems, such that the flows crossing the IR have unbounded latency within the IR, when observed with any clock of the network.*

The proof is in Appendix A.10. We give now an informal example of a missed deadline. Consider Figure 12 that gives the traffic profile that is expected to enter the previous FIFO system. It corresponds to two flows, each one with a leaky-bucket arrival curve with a burst of one packet and a rate of 1/2 packet per unit of time. Assume that the IR is configured with this contract. Assume that System 1 generates Flow 1 and System 2 generates Flow 2. Each System believes, according to its clock, that it generates in Figure 13a, its flow's packets as per the expected behaviour of Figure 12. But the clock of System 2 suddenly increases its speed relative to the IR clock between Units 1 and 3 (Figure 13a). When seen from the clock of the IR (here assumed to share the same clock as system 1), this sudden speeding of Clock 2 increases the burst of Flow 2. Even worse, it allows Packet $2b$ to squeeze in before Packet $1b$, when $2b$ was supposed to come after, as per Figure 12.

Assume a simulation of the FIFO system that provides the output given in Figure 13b. The worst-case delay through the FIFO is five units of TAI time, reached by Packet $1a$. For Packet $1b$ to meet this deadline for the entire hop (FIFO+IR), it must be released no later than Time Unit 7. But recall that the IR looks only at the head-of-line packet and all the following packets, even from

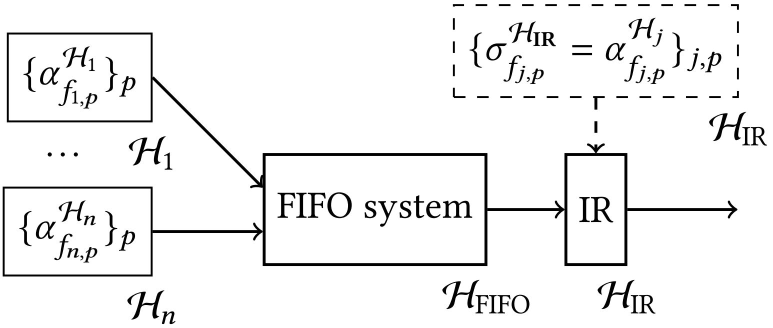

$\{\alpha^{\mathcal{H}_1}_{f_{1,p}}\}_p$
...
$\mathcal{H}_1$
$\{\alpha^{\mathcal{H}_n}_{f_{n,p}}\}_p$
$\mathcal{H}_n$
FIFO system
$\mathcal{H}_{\text{FIFO}}$
$\{\sigma^{\mathcal{H}_{\text{IR}}}_{f_{j,p}} = \alpha^{\mathcal{H}_j}_{f_{j,p}}\}_{j,p}$
IR
$\mathcal{H}_{\text{IR}}$
$\mathcal{H}_{\text{IR}}$

Fig. 11. Adversarial situation with several sources.

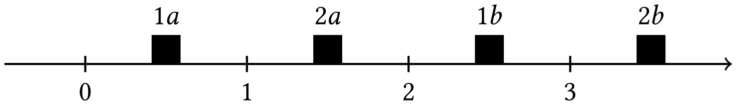
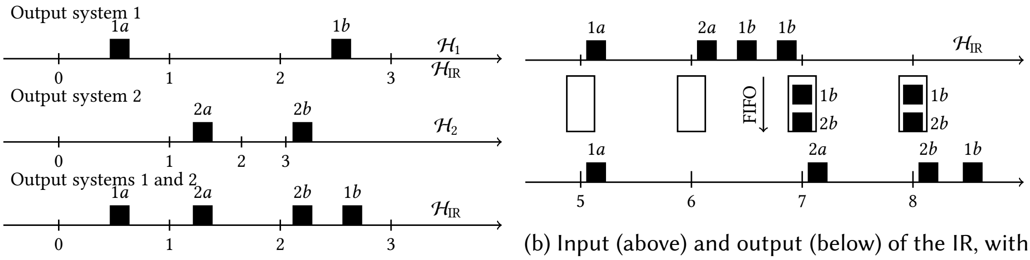


Fig. 12. Expected input of the previous FIFO system.



(a) Output of each source.

(b) Input (above) and output (below) of the IR, with $\mathcal{H}_{\text{IR}}$.

Fig. 13. Example of a missed deadline caused by nonideal clocks. Packet $1b$ misses its deadline (5 units of time).

other flows, need to wait for this head packet to be released before being processed by the IR. $1b$, blocked by $2b$, is released no earlier than Time Unit 8. Thus, $1b$ has missed its deadline.

Numerical Application to TSN: In the proof, the delay divergence increases at a rate $\sqrt{\rho}-1$ for any number of previous sources $n \geq 3$, and any synchronization precision $\Delta > 0$. This corresponds to 100μs of increased worst-case delay per second of network operation for both tightly- or loosely-synchronized networks.

8 CONCLUSION

We have developed a theory for adding clock nonidealities to network calculus. We have applied the theory to time-sensitive networks with regulators and have obtained the following conclusions.

In loosely-synchronized and in non-synchronized networks, regulators are affected by clock issues; this leads to large or unbounded delays if not properly addressed. We have proposed two solutions for PFRs: the rate and burst cascade, and ADAM. The former imposes that the regulator parameters depend on the position of the regulator on the flow path, which complexifies the control plane. For interleaved regulators, only the rate and burst cascade applies.

In tightly-synchronized networks, per-flow regulators are affected, but the delay penalty is of the order of the time-error bound and can be neglected. In contrast, interleaved regulators are affected well beyond the time-error bound, and this can lead to unbounded delays if the issue is not properly addressed. Therefore, a solution such as the method of rate and burst cascade should be applied with interleaved regulators, even in tightly-synchronized networks.

In this paper, we proposed a network calculus toolbox for networks with nonideal clocks, and we then focused on applying the toolbox on asynchronous traffic shaping (ATS). Computing the effect of nonideal clocks on other mechanisms and non-work conservatives scheduler such as TAS, CBS, the damper, constitute a first interesting axis of future work.

We also proposed methods to manage the regulator parameters of ATS within a network, focusing on delay bounds obtained with Network Calculus. The evaluation of these methods under realistic network and clock conditions, using simulations, constitute another axis of future work.

ACKNOWLEDGEMENTS

This work was supported by Huawei Technologies Co., Ltd. in the framework of the project *Large Scale Deterministic Network* (project number EPFL-LCA2-001). The authors thank Bryan Liubingyang for his contribution to the idea of this paper and Ahlem Mifdaoui for many fruitful discussions on an early version.

REFERENCES

[1] Matthew Andrews. 2009. Instability of FIFO in the Permanent Sessions Model at Arbitrarily Small Network Loads. *ACM Trans. Algorithms* 5, 3 (July 2009), 33:1–33:29. https://doi.org/10.1145/1541885.1541894 http://doi.acm.org/10.1145/1541885.1541894.

[2] H. Ayed, A. Mifdaoui, and C. Fraboul. 2014. Hierarchical Traffic Shaping and Frame Packing to Reduce Bandwidth Utilization in the AFDX. In *Proceedings of the 9th IEEE International Symposium on Industrial Embedded Systems (SIES 2014)*. 77–86. https://doi.org/10.1109/SIES.2014.6871190

[3] Daniel S. Berger, Martin Karsten, and Jens Schmitt. 2014. On the Relevance of Adversarial Queueing Theory in Practice. In *The 2014 ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS '14)*. Association for Computing Machinery, Austin, Texas, USA, 343–354. https://doi.org/10.1145/2591971.2592006 https://doi.org/10.1145/2591971.2592006.

[4] Rajat. Bhattacharjee, Ashish. Goel, and Zvi. Lotker. 2005. Instability of FIFO at Arbitrarily Low Rates in the Adversarial Queueing Model. *SIAM J. Comput.* 34, 2 (Jan. 2005), 318–332. https://doi.org/10.1137/S0097539703426805 https://epubs.siam.org/doi/10.1137/S0097539703426805.

[5] A. Bouillard, M. Boyer, and E.L. Corronc. 2018. *Deterministic Network Calculus: From Theory to Practical Implementation*. Wiley. http://doi.org/10.1002/9781119440284.

[6] Marc Boyer, Giovanni Stea, and William Mangoua Sofack. 2012. Deficit Round Robin with network calculus. In *6th International ICST Conference on Performance Evaluation Methodologies and Tools*. IEEE, 138–147.

[7] Cheng-Shang Chang. 2000. *Performance Guarantees in Communication Networks*. Springer-Verlag, London. https://www.springer.com/gp/book/9781852332266.

[8] Cheng-Shang Chang and Yih Haur Lin. 1998. A general framework for deterministic service guarantees in telecommunication networks with variable length packets. In *1998 Sixth International Workshop on Quality of Service (IWQoS'98)(Cat. No. 98EX136)*. IEEE, 49–58.

[9] Anna Charny and Jean-Yves Le Boudec. 2000. Delay Bounds in a Network with Aggregate Scheduling. In *Quality of Future Internet Services (Lecture Notes in Computer Science)*, Jon Crowcroft, James Roberts, and Mikhail I. Smirnov (Eds.). Springer Berlin Heidelberg, 1–13. https://link.springer.com/chapter/10.1007/3-540-39939-9_1.

[10] AEE Committee et al. 2002. Aircraft Data Network Part 7, Avionics Full Duplex Switched Ethernet (AFDX) Network, ARINC Specification 664. *Annapolis, Maryland: Aeronautical Radio* (2002).

[11] R. L. Cruz. 1991. A Calculus for Network Delay. I. Network Elements in Isolation. *IEEE Transactions on Information Theory* 37, 1 (Jan. 1991), 114–131. https://doi.org/10.1109/18.61109 http://doi.org/10.1109/18.61109.

[12] R. L. Cruz. 1991. A Calculus for Network Delay. II. Network Analysis. *IEEE Transactions on Information Theory* 37, 1 (Jan. 1991), 132–141. https://doi.org/10.1109/18.61110 http://doi.org/10.1109/18.61110.

[13] Hugo Daigmorte and Marc Boyer. 2016. Traversal Time for Weakly Synchronized CAN Bus. In *Proceedings of the 24th International Conference on Real-Time Networks and Systems (RTNS '16)*. Association for Computing Machinery, Brest, France, 35–44. https://doi.org/10.1145/2997465.2997477 https://doi.org/10.1145/2997465.2997477.

[14] Hugo Daigmorte and Marc Boyer. 2017. Evaluation of Admissible CAN Bus Load with Weak Synchronization Mechanism. In *Proceedings of the 25th International Conference on Real-Time Networks and Systems (RTNS '17)*. Association for Computing Machinery, Grenoble, France, 277–286. https://doi.org/10.1145/3139258.3139261 https://doi.org/10.1145/3139258.3139261.

[15] Hugo Daigmorte, Marc Boyer, and Jörn Migge. 2017. Reducing CAN Latencies by Use of Weak Synchronization between Stations.

[16] Hugo Daigmorte, Marc Boyer, and Luxi Zhao. 2018. Modelling in network calculus a TSN architecture mixing Time-Triggered, Credit Based Shaper and Best-Effort queues. (2018).

[17] E. F. Dierikx, A. E. Wallin, T. Fordell, J. Myyry, P. Koponen, M. Merimaa, T. J. Pinkert, J. C. J. Koelemeij, H. Z. Peek, and R. Smets. 2016. White Rabbit Precision Time Protocol on Long-Distance Fiber Links. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control* 63, 7 (July 2016), 945–952. https://doi.org/10.1109/TUFFC.2016.2518122

[18] ECSS. 2008. SpaceWire – Links, Nodes, Routers and Networks (31 July 2008) | European Cooperation for Space Standardization. *ECSS-E-ST-50-12C* (July 2008). https://ecss.nl/standard/ecss-e-st-50-12c-spacewire-links-nodes-routers-and-networks/.

[19] ECSS. 2009. *ECSS-Q-ST-30-02C – Failure Modes, Effects (and Criticality) Analysis (FMEA/FMECA) – (6 March2009) | European Cooperation for Space Standardization.* Technical Report. https://ecss.nl/standard/ecss-q-st-30-02c-failure-modes-effects-and-criticality-analysis-fmeafmeca/.

[20] ECSSS. 2008. *ECSS-Q-ST-40-02C – Hazard Analysis (15 November 2008) | European Cooperation for Space Standardization.* Technical Report. https://ecss.nl/standard/ecss-q-st-40-02c-hazard-analysis/.

[21] Nikolaos M. Freris, Scott R. Graham, and P. R. Kumar. 2011. Fundamental Limits on Synchronizing Clocks Over Networks. *IEEE Trans. Automat. Control* 56, 6 (June 2011), 1352–1364. https://doi.org/10.1109/TAC.2010.2089210 Conference Name: IEEE Transactions on Automatic Control.

[22] Yilong Geng, Shiyu Liu, Zi Yin, Ashish Naik, Balaji Prabhakar, Mendel Rosenblum, and Amin Vahdat. 2018. Exploiting a Natural Network Effect for Scalable, Fine-Grained Clock Synchronization. In *15th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 18)*. 81–94. https://www.usenix.org/node/211256.

[23] Google. [n.d.]. Leap Smear | Public NTP. https://developers.google.com/time/smear. Library Catalog: developers.google.com.

[24] IEC and IEEE. 2019. IEC/IEEE 60802 - Time-Sensitive Networking Profile for Industrial Automation. IEC/IEEE 60802 (D1.1) (2019). http://www.ieee802.org/1/files/private/60802-drafts/d1/60802-d1-1.pdf.

[25] IEEE. 2008. IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems. *IEEE Std 1588-2008 (Revision of IEEE Std 1588-2002)* (July 2008), 1–300. https://doi.org/10.1109/IEEESTD.2008.4579760

[26] IEEE. 2009. IEEE Standard Definitions of Physical Quantities for Fundamental Frequency and Time Metrology—Random Instabilities. *IEEE Std Std 1139-2008* (Feb. 2009), c1–35. https://doi.org/10.1109/IEEESTD.2008.4797525

[27] IEEE. 2010. IEEE Standard for Local and Metropolitan Area Networks– Virtual Bridged Local Area Networks Amendment 12: Forwarding and Queuing Enhancements for Time-Sensitive Streams. *IEEE Std 802.1Qav-2009 (Amendment to IEEE Std 802.1Q-2005)* (Jan. 2010), 1–72. https://doi.org/10.1109/IEEESTD.2010.8684664 Conference Name: IEEE Std 802.1Qav-2009 (Amendment to IEEE Std 802.1Q-2005).

[28] IEEE. 2011. IEEE Standard for Local and Metropolitan Area Networks - Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks. *IEEE Std 802.1AS-2011* (March 2011), 1–292. https://doi.org/10.1109/IEEESTD.2011.5741898

[29] IEEE. 2017. IEEE Standard for Local and Metropolitan Area Networks–Frame Replication and Elimination for Reliability. *IEEE Std 802.1CB-2017* (Oct. 2017), 1–102. https://doi.org/10.1109/IEEESTD.2017.8091139

[30] IEEE. 2018. IEEE Standard for Local and Metropolitan Area Networks–Bridges and Bridged Networks – Amendment 31: Stream Reservation Protocol (SRP) Enhancements and Performance Improvements. *IEEE Std 802.1Qcc-2018 (Amendment to IEEE Std 802.1Q-2018 as amended by IEEE Std 802.1Qcp-2018)* (Oct. 2018), 1–208. https://doi.org/10.1109/IEEESTD.2018.8514112

[31] IEEE. 2019. Draft Standard for Local and Metropolitan Area Networks — Time-Sensitive Networking Profile for Automotive In-Vehicle Ethernet Communications. *IEEE P802.1DG™/D1.1* In IEEE802.1 private repository. To obtain the access credentials, visit https://www.ietf.org/proceedings/52/slides/bridge-0/tsld003.htm or contact the IEEE802.1 chair. (Oct. 2019). http://www.ieee802.org/1/files/private/dg-drafts/d1/802-1DG-d1-1.pdf.

[32] IEEE. 2019. Draft Standard for Local and Metropolitan Area Networks—Bridges and Bridged Networks—Amendment: Asynchronous Traffic Shaping. *IEEE P802.1Qcr/D2.0* In IEEE802.1 private repository. To obtain the access credentials, visit https://www.ietf.org/proceedings/52/slides/bridge-0/tsld003.htm or contact the IEEE802.1 chair. (Dec. 2019). http://www.ieee802.org/1/files/private/cr-drafts/d2/802-1Qcr-d2-0.pdf.

[33] ISO-IEC-IEEE. 2018. ISO/IEC/IEEE International Standard – Information Technology – Telecommunications and Information Exchange between Systems – Local and Metropolitan Area Networks – Specific Requirements – Part 1Q: Bridges and Bridged Networks AMENDMENT 3: Enhancements for Scheduled Traffic. *ISO/IEC/IEEE 8802-1Q:2016/Amd.3:2017(E)* (Feb. 2018), 1–62. https://doi.org/10.1109/IEEESTD.2018.8290654 Conference Name: ISO/IEC/IEEE 8802-1Q:2016/Amd.3:2017(E).

[34] ISO-IEC-IEEE. 2019. IEEE/ISO/IEC International Standard - Information Technology - Telecommunications and Information Exchange between Systems - Local and Metropolitan Area Networks - Specific Requirements - Part 1Q: Bridges and Bridged Networks - AMENDMENT 7: Cyclic Queuing and Forwarding. *ISO/IEC/IEEE 8802-1Q:2016/Amd.7:2019(E)* (March 2019), 1–34. https://doi.org/10.1109/IEEESTD.2019.8664711 Conference Name: ISO/IEC/IEEE 8802-1Q:2016/Amd.7:2019(E).

[35] ITU. 1996. Definitions and Terminology for Synchronization Networks. *ITU G.810* (1996). https://www.itu.int/rec/T-REC-G.810-199608-I/en.

[36] ITU. 2004. Timing Requirements of Slave Clocks Suitable for Use as Node Clocks in Synchronization Networks. *ITU G.812* (2004). https://www.itu.int/rec/T-REC-G.812-200406-I/en.

[37] Hermann Kopetz, Astrit Ademaj, Petr Grillinger, and Klaus Steinhammer. 2005. The Time-Triggered Ethernet (TTE) Design. In *Object-Oriented Real-Time Distributed Computing, 2005. ISORC 2005. Eighth IEEE International Symposium on*. IEEE, 22–33.

[38] J-Y Le Boudec. 2002. Some properties of variable length packet shapers. *IEEE/ACM Transactions on Networking* 10, 3 (2002), 329–337.

[39] Jean-Yves Le Boudec. 2018. A Theory of Traffic Regulators for Deterministic Networks With Application to Interleaved Regulators. *IEEE/ACM Transactions on Networking* 26, 6 (Dec. 2018), 2721–2733. https://doi.org/10.1109/TNET.2018.2875191 http://doi.org/10.1109/TNET.2018.2875191.

[40] Jean-Yves Le Boudec and Patrick Thiran. 2001. *Network Calculus: A Theory of Deterministic Queuing Systems for the Internet*. Springer-Verlag, Berlin Heidelberg. https://www.springer.com/us/book/9783540421849.

[41] Jun Liu, Zhaohui Wang, Jun-Hong Cui, Shengli Zhou, and Bo Yang. 2016. A Joint Time Synchronization and Localization Design for Mobile Underwater Sensor Networks. *IEEE Transactions on Mobile Computing* 15, 3 (March 2016), 530–543. https://doi.org/10.1109/TMC.2015.2410777 Conference Name: IEEE Transactions on Mobile Computing.

[42] Jim Martin, Jack Burbank, William Kasch, and Professor David L. Mills. 2010. Network Time Protocol Version 4: Protocol and Algorithms Specification. RFC 5905. https://doi.org/10.17487/RFC5905

[43] Tatsuya Maruyama, Tsutomu Yamada, Shouji Yoshida, Mitsuyasu Kido, and Chikashi Komatsu. 2015. NS-3 Based IEEE 1588 Synchronization Simulator for Multi-Hop Network. In *2015 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control, and Communication (ISPCS)*. 99–104. https://doi.org/10.1109/ISPCS.2015.7324691

[44] D. Mills, J. Martin, J. Burbank, and W. Kasch. 2010. *Network Time Protocol Version 4: Protocol and Algorithms Specification*. Technical Report RFC5905. RFC Editor. RFC5905 pages. https://doi.org/10.17487/rfc5905 https://www.rfc-editor.org/info/rfc5905.

[45] Ehsan Mohammadpour, Eleni Stai, and Jean-Yves Le Boudec. 2019. Improved Delay Bound for a Service Curve Element with Known Transmission Rate. *IEEE Networking Letters* 1, 4 (2019), 156–159.

[46] E. Mohammadpour, E. Stai, M. Mohiuddin, and J. Le Boudec. 2018. Latency and Backlog Bounds in Time-Sensitive Networking with Credit Based Shapers and Asynchronous Traffic Shaping. In *2018 30th International Teletraffic Congress (ITC 30)*, Vol. 02. 1–6. https://doi.org/10.1109/ITC30.2018.10053 http://doi.org/10.1109/ITC30.2018.10053.

[47] P. Moreira, J. Serrano, T. Wlostowski, P. Loschmidt, and G. Gaderer. 2009. White Rabbit: Sub-Nanosecond Timing Distribution over Ethernet. In *Control and Communication 2009 International Symposium on Precision Clock Synchronization for Measurement*. 1–5. https://doi.org/10.1109/ISPCS.2009.5340196

[48] Cristina D. Murta, Pedro R. Torres Jr., and Prasant Mohapatra. 2006. QRPp1-4: Characterizing Quality of Time and Topology in a Time Synchronization Network. In *IEEE Globecom 2006*. 1–5. https://doi.org/10.1109/GLOCOM.2006.467

[49] Ahmed Nasrallah, Akhilesh S. Thyagaturu, Ziyad Alharbi, Cuixiang Wang, Xing Shao, Martin Reisslein, and Hesham Elbakoury. 2019. Performance Comparison of IEEE 802.1 TSN Time Aware Shaper (TAS) and Asynchronous Traffic Shaper (ATS). *IEEE Access* 7 (2019), 44165–44181. https://doi.org/10.1109/ACCESS.2019.2908613 Conference Name: IEEE Access.

[50] Nicolas NAVET, Tieu Long MAI, and Jörn MIGGE. 2019. *Using Machine Learningto SpeedUp the Design Space Exploration of Ethernet TSN Networks*. Technical Report.

https://orbilu.uni.lu/bitstream/10993/38604/1/feasibility-with-ml.pdf.

[51] Ilkka Norros. 1994. A storage model with self-similar input. *Queueing systems* 16, 3-4 (1994), 387–396.

[52] nsnam. 2011. Ns-3 Network Simulator. Project Homepage. https://www.nsnam.org/.

[53] Linh T. X. Phan, Samarjit Chakraborty, P. S. Thiagarajan, and Lothar Thiele. 2007. Composing Functional and State-Based Performance Models for Analyzing Heterogeneous Real-Time Systems. In *28th IEEE International Real-Time Systems Symposium (RTSS 2007)*. 343–352. https://doi.org/10.1109/RTSS.2007.46

[54] Edward Powers and Jorg Hahn. 2004. GPS and Galileo UTC Time Distribution. In *2004 18th European Frequency and Time Forum (EFTF 2004)*. 484–488. https://doi.org/10.1049/cp:20040914

[55] ITUTG Recommendation. 2006. 8261/Y. 1361 timing and synchronization aspects in packet networks. *International Telecommun. Union* (2006).

[56] Julien Ridoux and Darryl Veitch. 2010. Principles of Robust Timing over the Internet. *Commun. ACM* (May 2010). https://dl.acm.org/doi/abs/10.1145/1735223.1735241.

[57] Johannes Specht and Soheil Samii. 2016. Urgency-based scheduler for time-sensitive switched ethernet networks. In *Real-Time Systems (ECRTS), 2016 28th Euromicro Conference on*. IEEE, 75–85.

[58] Ludovic Thomas, Jean-Yves Le Boudec, and Ahlem Mifdaoui. 2019. On Cyclic Dependencies and Regulators in Time-Sensitive Networks. In *2019 IEEE Real-Time Systems Symposium (RTSS)*. 299–311. https://doi.org/10.1109/RTSS46320.2019.00035

[59] Darryl Veitch, Satish Babu, and Attila Pàsztor. 2004. Robust Synchronization of Software Clocks across the Internet | Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement. In *IMC '04: Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement*. Taormina Sicily, Italy. https://dl.acm.org/doi/abs/10.1145/1028788.1028817.

[60] Kurt Wagner. 2001. Short evaluation of linux's token-bucket-filter (TBF) queuing discipline. *http://www.docum.org/stef.coene/qos/docs/other/tbf02_kw.ps* (2001).

[61] Ernesto Wandeler, Alexander Maxiaguine, and Lothar Thiele. 2006. Performance analysis of greedy shapers in real-time systems. In *Proceedings of the Design Automation & Test in Europe Conference*, Vol. 1. IEEE, 6–pp.

[62] E. Wandeler, A. Maxiaguine, and L. Thiele. 2006. Performance Analysis of Greedy Shapers in Real-Time Systems. In *Proceedings of the Design Automation Test in Europe Conference*, Vol. 1. 6 pp.–. https://doi.org/10.1109/DATE.2006.243801

[63] Jie Wu, Liyi Zhang, Yu Bai, and Yunshan Sun. 2015. Cluster-Based Consensus Time Synchronization for Wireless Sensor Networks. *IEEE Sensors Journal* 15, 3 (March 2015), 1404–1413. https://doi.org/10.1109/JSEN.2014.2363471 Conference Name: IEEE Sensors Journal.

[64] Lin Zhao, Feng He, Ershuai Li, and Jun Lu. 2018. Comparison of Time Sensitive Networking (TSN) and TTEthernet. In *2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*. 1–7. https://doi.org/10.1109/DASC.2018.8569454

A APPENDIX

A.1 Proof of Proposition 5.1

Proof of Proposition 5.1. Call $A(t)$ the time, measured with $\mathcal{H}_i$, at which a packet that has entered device j at t leaves the device. By definition, for any time t observed with $\mathcal{H}_i$, $A(t) - t \leq D$. As d is increasing, for any time t measured in $\mathcal{H}_i$, $d(A(t)) - d(t) \leq d(t + D) - d(t) \leq \sup_{t'}(d(t' + D) - d(t')) = (d \oslash d)(D)$.

Note that $d(A(t))$ also equals $A(d(t))$ because the packet that enters device j at t observed with $\mathcal{H}_i$ [resp $d(t)$ observed with $\mathcal{H}_g$] leaves device j at $A(t)$ [resp $A(d(t))$]. Hence for any t observed using $\mathcal{H}_i$, $A(d(t)) - d(t) \leq (d \oslash d)(D)$, which gives the result. □

A.2 Proof of Proposition 5.2

Proof of Proposition 5.2. Note that the same number of bits of the flow enter the device j between the time instants measured as t_1 and t_2 using $\mathcal{H}_i$ and between the time instants measured as $d_{g\to i}(t_1)$ and $d_{g\to i}(t_2)$ using $\mathcal{H}_g$. We now split the proof into two situations.

Case $d(0) \geq 0$. If the origin of $\mathcal{H}_i$ is before the origin of $\mathcal{H}_g$, then the situation is as in Figure 14, where we represent face-to-face the two clocks. For any t, the number of bits observed using $\mathcal{H}_i$ between 0 and $d(t)$ equals the number of bits observed between 0 and $d(0)$, plus the number of bits observed between $d(0)$ and $d(t)$. That second term also equals the number of bits observed

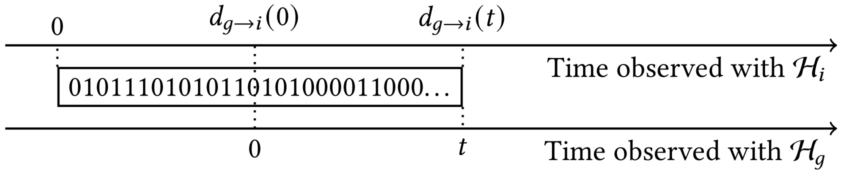

0
$d_{g\to i}(0)$
$d_{g\to i}(t)$
0101111010101110101000011000...
Time observed with $\mathcal{H}_i$
0
t
Time observed with $\mathcal{H}_g$

Fig. 14. Case where the initial offset of the local clock $\mathcal{H}_i$ is positive compared to the reference clock $\mathcal{H}_g$.

between 0 and t using $\mathcal{H}_g$ (Figure 14), i.e., $R^{\mathcal{H}_g}(t)$. Hence,

$$\forall t, R^{\mathcal{H}_i}(d(t)) = R^{\mathcal{H}_i}(d(0)) + R^{\mathcal{H}_g}(t)$$

Due to the definition of T_{start}, no bit could have been sent before the time measured as 0 using $\mathcal{H}_g$. Consequently, $R^{\mathcal{H}_i}(d(0)) = 0$, and we have the result.

Case $d(0) < 0$. If the origin of clock $\mathcal{H}_i$ is after the origin of clock $\mathcal{H}_g$, then by symmetry, we simply flip the pair and obtain the result from case *a).* □

A.3 Proof of Proposition 5.4

Proof of Proposition 5.4. For any t, τ, we have

$$\begin{aligned}
&R^{\mathcal{H}_g}(t+\tau) - R^{\mathcal{H}_g}(t)\\
&= R^{\mathcal{H}_i}(d(t+\tau)) - R^{\mathcal{H}_i}(d(t))\\
&\qquad \text{per Proposition 5.2}\\
&\leq \alpha^{\mathcal{H}_i}(d(t+\tau) - d(t))\\
&\qquad \text{because } \alpha^{\mathcal{H}_i} \text{ is an arrival curve when observed with } \mathcal{H}_i\\
&\leq \sup_{t'\geq 0}\left[\alpha^{\mathcal{H}_i}(d(t'+\tau) - d(t'))\right]\\
&\leq \alpha^{\mathcal{H}_i}(d \oslash d(\tau))\\
&\qquad \text{as } \alpha^{\mathcal{H}_i} \text{ is a wide-sense increasing function}
\end{aligned}$$

Also, per Equation (5), $(d \oslash d)(\tau) \leq \rho\tau + \eta$, and $d \oslash d$ is always finite in our model. □

A.4 Proof of Proposition 5.6

Proof of Proposition 5.6. Fix any arbitrary $t \geq 0$. Then

$$\begin{aligned}
R^{*\mathcal{H}_g}(t) &= R^{*\mathcal{H}_i}(d(t))\\
&\geq \inf_{0\leq\sigma\leq d(t)}\left[R^{\mathcal{H}_i}(\sigma) + \beta^{\mathcal{H}_i}(d(t)-\sigma)\right]
\end{aligned}$$

As per Proposition 5.2 and because $\beta^{\mathcal{H}_i}$ is a service curve observed with $\mathcal{H}_i$. As d is strictly increasing, for any time instant σ measured with $\mathcal{H}_i$, we note $\sigma = d(s)$ with s the measure using $\mathcal{H}_g$. Hence,

$$\begin{aligned}
&R^{*\mathcal{H}_g}(t)\\
&\geq \inf_{d^{-1}(0)\leq s\leq t}\left[R^{\mathcal{H}_i}(d(s)) + \beta^{\mathcal{H}_i}(d(t)-d(s))\right]
\end{aligned}$$

If $t < T_{\text{start}}$ (defined in Section 4.1), the result holds because no bit has been transmitted: $R^{*\mathcal{H}_g}(t) = 0$ and $\forall s \leq t, R^{\mathcal{H}_i}(d(s)) = 0$ and $\inf_{d^{-1}(0)\leq s \leq t} \beta^{\mathcal{H}_i}(d(t) - d(s)) = 0$, obtained for $s = t$.

Let's now assume $t \geq T_{\text{start}}$ and call $A : s \mapsto R^{\mathcal{H}_i}(d(s)) + \beta^{\mathcal{H}_i}(d(t) - d(s))$. As $T_{\text{start}} \geq d^{-1}(0)$, we can split the domain of the $\inf_s A(s)$ into the two cases $s \leq T_{\text{start}}$ and $s \geq T_{\text{start}}$, the result being the minimum of the two obtained inf.

For $s < T_{\text{start}}$, $\beta^{\mathcal{H}_i}(d(t) - d(s)) \geq \beta^{\mathcal{H}_i}(d(t) - d(T_{\text{start}}))$ because d and $\beta^{\mathcal{H}_i}$ are both increasing functions. On the other hand, based on the assumption on T_{start}, $R^{\mathcal{H}_i}(d(s)) = R^{\mathcal{H}_i}(d(T_{\text{start}}))$ as both quantities equal 0 bit.

Consequently,

$$\inf_{d^{-1}(0)\leq s \leq T_{\text{start}}} A(s) \geq A(T_{\text{start}})$$

and $\inf_s A(s)$ is obtained for $s \in [T_{\text{start}}, t]$, i.e.,

$$R^{*\mathcal{H}_g}(t) \geq \inf_{T_{\text{start}} \leq s \leq t} A(s)$$

By definition $T_{\text{start}} \geq 0$, so $[T_{\text{start}}, t] \subset [0, t]$, and hence,

$$\begin{aligned} R^{*\mathcal{H}_g}(t) &\geq \inf_{0\leq s\leq t} A(s) \\ &\geq \inf_{0\leq s\leq t} \left[R^{\mathcal{H}_i}(d(s)) + \beta^{\mathcal{H}_i}(d(t) - d(s)) \right] \\ \geq \inf_{0\leq s\leq t} &\begin{bmatrix} R^{\mathcal{H}_i}(d(s)) + \\ \inf_{0\leq u\leq s} \beta^{\mathcal{H}_i}(d(t - s + u) - d(u)) \end{bmatrix} \\ \geq \inf_{0\leq s\leq t} &\left[R^{\mathcal{H}_i}(d(s)) + \beta^{\mathcal{H}_i}((d\overline{\oslash}d)(t-s)) \right] \end{aligned}$$

Because $\beta^{\mathcal{H}_i}$ is wide-sense increasing. Now, as $R^{\mathcal{H}_i}(d(s)) = R^{\mathcal{H}_g}(s)$ per Proposition 5.2, we obtain

$$R^{*\mathcal{H}_g}(t) \geq \inf_{0\leq s\leq t} \left[R^{\mathcal{H}_g}(s) + \beta^{\mathcal{H}_i}((d\overline{\oslash}d)(t-s)) \right]$$

which proves that $t \mapsto \beta^{\mathcal{H}_i}((d\overline{\oslash}d)(t-s))$ is a service curve observed with $\mathcal{H}_g$. □

A.5 Proof of Proposition 6.1

Proof of Proposition 6.1. Consider a flow such as f in Figure 1, assume that the source of the flow is at the input to system S and call $\mathcal{H}_1$ the clock used to define the arrival curve $\alpha^{\mathcal{H}_1} = \gamma_{r,b}$ at the source. Assume the regulator is a PFR and let $\mathcal{H}_{\text{Reg}} \neq \mathcal{H}_1$ be its clock. Since the PFR is non-adapted, it is configured with shaping curve $\sigma^{\mathcal{H}_{\text{Reg}}} = \alpha^{\mathcal{H}_1}$.

Let the flow be greedy after T_{start} and generate packets of size ℓ bits, with $\ell \leq b$. Its cumulative arrival function at the source, measured in $\mathcal{H}_1$, is

$$R^{\mathcal{H}_1}(t) = \left\lfloor \frac{\alpha^{\mathcal{H}_1}(|t - T_{\text{start}}|^+)}{\ell} \right\rfloor \ell \tag{7}$$

where $\lfloor\cdot\rfloor$ is the floor function. The fact that this flow satisfies the arrival-curve constraint $\alpha^{\mathcal{H}_1}$ in $\mathcal{H}_1$ follows from [38, Thm III.2]. Also, system S provides a delay bound D in TAI thus, by Section 5.1, a bound $D_1 = \rho D + \eta$ in $\mathcal{H}_1$. Let $R'^{\mathcal{H}_1}$ be the cumulative arrival function of flow f at the input of the PFR, observed with $\mathcal{H}_1$. Thus, for every t,

$$R^{\mathcal{H}_1}(t - D_1) \leq R'^{\mathcal{H}_1}(t) \tag{8}$$

Let $d_{1\to\text{Reg}}(t) = t/\rho$ be the relative time function between $\mathcal{H}_1$ and $\mathcal{H}_{\text{Reg}}$. The function meets the conditions of Equation (5). Using Proposition 5.2, the cumulative arrival function of flow f at the input of the PFR, when observed with the PFR clock $\mathcal{H}_{\text{Reg}}$, is given by

$$R'^{\mathcal{H}_{\text{Reg}}}(\tau) = R'^{\mathcal{H}_1}\left(d^{-1}_{1\to\text{Reg}}(\tau)\right) = R'^{\mathcal{H}_1}(\rho\tau) \tag{9}$$

Network calculus results are valid as long as all the notions are in the same time reference. Recall from Section 3 that the PFR can be modelled as a fluid greedy shaper followed by a packetizer, and the latter can be ignored for delay computations. The output of the fluid greedy shaper, observed with $\mathcal{H}_{\text{Reg}}$, is thus $R^{*\mathcal{H}_{\text{Reg}}} = R'^{\mathcal{H}_{\text{Reg}}} \otimes \gamma_{r,b}$. It follows that, for all τ, $R^{*\mathcal{H}_{\text{Reg}}}(\tau) \leq R^{\mathcal{H}_{\text{Reg}}}(T_{\text{start}}) + \alpha^{\mathcal{H}^1}(\tau - T_{\text{start}}) = r|\tau - T_{\text{start}}|^+ + b$ by definition of T_{start}.

We now show that for any $e > 0$, the worst-case delay through the PFR, measured in $\mathcal{H}_{\text{Reg}}$, is $\geq e$. From the previous equation, it follows that, for $\tau \geq T_{\text{start}}$,

$$R^{*\mathcal{H}_{\text{Reg}}}(\tau + e) \leq r(\tau + e - T_{\text{start}}) + b \tag{10}$$

Combine Eqs (7)–(10) and obtain

$$\begin{aligned} R^{*\mathcal{H}_{\text{Reg}}}(\tau + e) - R'^{\mathcal{H}_{\text{Reg}}}(\tau) &\leq r(\tau + e - T_{\text{start}}) + b - R^{\mathcal{H}_1}(\rho\tau - D_1) \\ &= r(\tau + e - T_{\text{start}}) + b - \left\lfloor \frac{r(\rho\tau - D_1 T_{\text{start}}) + b}{\ell} \right\rfloor \ell \\ &\leq r(\tau + e - T_{\text{start}}) + b - (r(\rho\tau - D_1 - T_{\text{start}}) + b) + \ell \\ &= r(1-\rho)\tau + re + rD_1 + \ell \end{aligned} \tag{11}$$

Thus $R^{*\mathcal{H}_{\text{Reg}}}(\tau + e) - R'^{\mathcal{H}_{\text{Reg}}}(\tau) < 0$ whenever $\tau > \left(\frac{re + rD_1 + \ell}{(\rho-1)r} \vee T_{\text{start}}\right)$; it follows that the delay, measured with $\mathcal{H}_{\text{Reg}}$, for packets arrived at the PFR after time $\left(\frac{re + rD_1 + \ell}{(\rho-1)r} \vee T_{\text{start}}\right)$ is larger than e. This holds for any arbitrary $e > 0$, therefore the delay measured with $\mathcal{H}_{\text{Reg}}$ is unbounded.

By Section 5.1, this also proves that the delay is not bounded when viewed from any clock of the network.

For the IR, the same adversarial example applies because an IR processing only one flow has the same behavior as a PFR [39]. □

A.6 Proof of Proposition 6.2

Proof of Proposition 6.2. Assume that S is the kth hop for flow f and note $S = S_k$ as in Figure 4. When observed with $\mathcal{H}_{\text{Reg}_{k-1}}$, the flow has the arrival curve $\sigma_{k-1}^{\mathcal{H}_{\text{Reg}k-1}}$ at the output of Reg_{k-1}. We now apply Table 2 with $\mathcal{H}_g = \mathcal{H}_{\text{Reg}_k}$ and $\mathcal{H}_i = \mathcal{H}_{\text{Reg}_{k-1}}$. We obtain that, when observed with the clock of the next regulator, $\mathcal{H}_{\text{Reg}_k}$, the flow leaves Reg_{k-1} with a leaky-bucket arrival curve $\alpha_{k-1}^{\mathcal{H}_{\text{Reg}k}}$ of rate $\rho r_{\text{Reg}_{k-1}}$ and burst $\text{Reg}_{k-1} + \eta r_{\text{Reg}_{k-1}}$.

From the configuration of Reg_{k-1} and Reg_k, we note that $\alpha_{k-1}^{\mathcal{H}_{\text{Reg}k}} \leq \sigma_k^{\mathcal{H}_{\text{Reg}k}}$. Consequently, all the conditions for the shaping-for-free property are met when observed with clock $\mathcal{H}_{\text{Reg}_k}$. We apply the respective theorems for both the PFR [40, Thm 1.5.2] and the IR [39, Thm 5] using this clock. An upper bound on the delay for the flow through the system S_k as measured with $\mathcal{H}_{\text{Reg}_k}$ is $\rho D_k^{\mathcal{H}_{\text{TAI}}} + \eta$ (Proposition 5.1). Applying the shaping-for-free property, this is also a valid delay bound for the flow trough the whole hop (S_k followed by regulator), when measuring with $\mathcal{H}_{\text{Reg}_k}$. To obtain a delay bound back in the measurement clock $\mathcal{H}_{\text{TAI}}$, we apply again Proposition 5.1, which gives the result. □

A.7 Proof of Proposition 6.3

We first establish the following lemma.

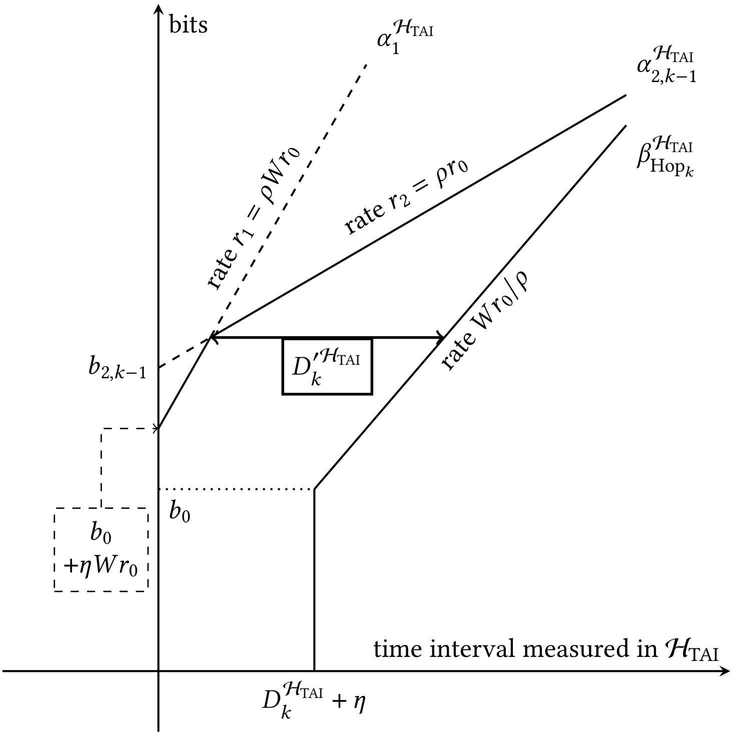


Fig. 15. TAI delay bound computation for the flow through the whole hop (S_k followed by regulator) in the ADAM method. The knowledge of $\alpha_{2,k}$ is required to provide a bounded delay whereas the knowledge of α_1 helps having a tight delay bound.

Lemma A.1. *Assume that $\alpha_{2,k-1}$ is an arrival curve for the flow at the input of the S_k, observed in $\mathcal{H}_{TAI}$. Then*

(1) $D_k'^{\mathcal{H}_{TAI}} = D_k^{\mathcal{H}_{TAI}} + \eta(1+\rho) + \frac{b_{2,k-1} - b_0 - \eta W r_0}{\rho r_0} \frac{\rho^2 - 1}{W - 1}$ *is a TAI delay bound for the flow through the concatenation of S_k and Reg_k.*

(2) $\gamma_{\rho r_0, b_{2,k-1} + \rho r_0 \cdot D_k'^{\mathcal{H}_{TAI}}}$ *is an arrival curve for the flow observed in $\mathcal{H}_{TAI}$ at the output of Reg_k.*

Proof. (1) We use the property that the shaping curve of a PFR is also a service curve, when observed with its own clock. Using the results of Table 3 with $\mathcal{H}_g = \mathcal{H}_{\text{TAI}}$ and $\mathcal{H}_i = \mathcal{H}_{\text{Reg}_k}$, we obtain a service curve of the PFR when observed with $\mathcal{H}_{\text{TAI}}$: $\beta_{\text{Reg}_k}^{\mathcal{H}_{\text{TAI}}} = \delta_\eta \otimes \gamma_{W r_0/\rho, b_0}$. Additionally, as $D_k^{\mathcal{H}_{\text{TAI}}}$ is a delay bound through S_k, observed with $\mathcal{H}_{\text{TAI}}$, $\delta_{D_k^{\mathcal{H}_{\text{TAI}}}}$ is a service curve of S_k when observed with $\mathcal{H}_{\text{TAI}}$. Due to the concatenation property of service curves [40, Thm 1.4.6], applied with $\mathcal{H}_{\text{TAI}}$, the whole k-th hop offers, when observed with $\mathcal{H}_{\text{TAI}}$, the service curve $\beta_{\text{Hop}_k}^{\mathcal{H}_{\text{TAI}}} = \delta_{D_k^{\mathcal{H}_{\text{TAI}}}} \otimes \delta_\eta \otimes \gamma_{W r_0/\rho, b_0}$to the flow. Its shape is given in Figure 15.

Conversely, the flow has, when observed with $\mathcal{H}_{\text{TAI}}$ and at the input of the hop, both α_1 (PFR output arrival-curve property) and $\alpha_{2,k-1}$ (induction assumption) as arrival curves. Their shape are given in Figure 15.

We apply the Network Calculus three-bound theorem [40, Thm 1.4.2] to obtain a delay bound as the maximal horizontal distance between $\alpha_1 \otimes \alpha_2$ and β_{Hop_k}, reached at the location marked on

Figure 15. Note that knowing only α_1 does not prove the existence of a maximal horizontal distance because $Wr_0/\rho < \rho W r_0$ (for $\rho > 1$). However, knowing also $\alpha_{2,k-1}$ proves it because $Wr_0/\rho \geq \rho r_0$ because $W \geq \rho^2$. Geometrical considerations gives the result in the proposition.

(2) Use the output flow part of the three-bound theorem [40, Thm 1.4.3], applied with $\mathcal{H}_{\text{TAI}}$. □

Proof of Proposition 6.3. (1) Applying Table 2 with $\mathcal{H}_g = \mathcal{H}_{\text{TAI}}$ and $\mathcal{H}_i = \mathcal{H}_{\text{Reg}_0}$ proves that $\alpha_{2,0}$ is an arrival curve for the flow at its source when observed with $\mathcal{H}_{\text{TAI}}$.

(2) and (3): The combination of the two statements is shown by induction on k. The base step $k = 0$ follows from the previous item. The induction step follows immediately from items (2) and (3) of Lemma A.1

□

A.8 Proof of Proposition 7.1

Proof of Proposition 7.1. Take the clock of the source to be exactly the TAI and let $\mathcal{H}_{\text{Reg}} \neq \mathcal{H}_{\text{TAI}}$ be the clock of the PFR. Since the PFR is non-adapted, it is configured with shaping curve $\sigma^{\mathcal{H}_{\text{Reg}}} = \alpha^{\mathcal{H}_{\text{TAI}}}$.

Let $x_1 = T_{\text{start}} + \frac{\rho\Delta}{\rho-1}$ and take, for the adversarial relative time function $d_{\text{TAI}\to\text{PFR}}$, the following piecewise linear function

$$d_{\text{TAI}\to\text{PFR}}(t) = \begin{cases} t & \text{if } t \leq T_{\text{start}} \\ \frac{1}{\rho}(t - T_{\text{start}}) + T_{\text{start}} & \text{if } T_{\text{start}} < t \leq x_1 \\ t - \Delta & \text{if } x_1 < t \end{cases}$$

The shape of $d_{\text{TAI}\to\text{PFR}}$ is given in Figure 16. It is continuous, strictly increasing, and meets the constraints of Equations (5) and (6).

As in Appendix A.5, let the source be greedy after T_{start} and generate packets of size ℓ bits, with $b \geq \ell$. Its cumulative arrival function at the source, measured in $\mathcal{H}_1$, is

$$R^{\mathcal{H}_{\text{TAI}}}(t) = \left\lfloor \frac{\alpha^{\mathcal{H}_{\text{TAI}}}(|t - T_{\text{start}}|^+)}{\ell} \right\rfloor \ell \tag{12}$$

where $\lfloor\cdot\rfloor$ is the floor function. The fact that this flow satisfies the arrival-curve constraint $\alpha^{\mathcal{H}_{\text{TAI}}}$ in $\mathcal{H}_{\text{TAI}}$ follows again from [38, Thm III.2].

Consider now the first network element for the flow, S_1 in Figure 4, as having no delay, for any observation clock.

Let R' be the cumulative arrival function in the PFR, that is at the output of the network element. Then for any t, $R'^{\mathcal{H}_{\text{TAI}}}(t) = R^{\mathcal{H}_{\text{TAI}}}(t)$ and for any τ, $R'^{\mathcal{H}_{\text{Reg}}}(\tau) = R^{\mathcal{H}_{\text{Reg}}}(\tau)$.

With the same arguments as in Appendix A.5, we model the PFR as a fluid greedy shaper followed by a packetizer. The latter can be ignored for delay computations. The output of the fluid greedy shaper, observed with $\mathcal{H}_{\text{Reg}}$, is thus $R^{*\mathcal{H}_{\text{Reg}}} = R'^{\mathcal{H}_{\text{Reg}}} \otimes \gamma_{r,b}$. It follows that, for all τ, $R^{*\mathcal{H}_{\text{Reg}}}(\tau) \leq R^{\mathcal{H}_{\text{Reg}}}(T_{\text{start}}) + \alpha^{\mathcal{H}_{\text{TAI}}}(\tau - T_{\text{start}}) = r|\tau - T_{\text{start}}|^+ + b$ by definition of T_{start}.

Now let x_2 be the next TAI time after x_1 at which the source finishes sending a packet. Then by definition of R and R'

$$R'^{\mathcal{H}_{\text{TAI}}}(x_2) = R^{\mathcal{H}_{\text{TAI}}}(x_2) = \alpha^{\mathcal{H}_{\text{TAI}}}(x_2 - T_{\text{start}}) \tag{13}$$

Now consider $x_2+\Delta$ and compute the cumulative output of the regulator at $x_2+\Delta$: $R^{*\mathcal{H}_{\text{TAI}}}(x_2+\Delta) = R^{\mathcal{H}_{\text{Reg}}}(d_{\text{TAI}\to\text{PFR}}(x_2 + \Delta)) = R^{\mathcal{H}_{\text{Reg}}}(x_2)$ because $x_2 \geq x_1$ and by definition of $d_{\text{TAI}\to\text{PFR}}$.

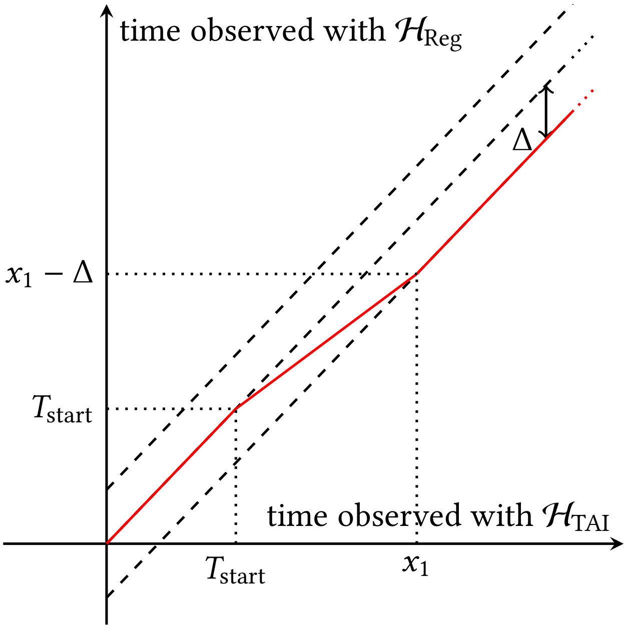


Fig. 16. Shape of the time function $d_{\text{TAI}\to\text{PFR}}$. When the TAI reaches T_{start}, the PFR clock $\mathcal{H}_{\text{Reg}}$ starts counting the time slower (with a relative factor $1/\rho$) until its time is late by Δ compared to the TAI. After that point, it counts the time at the same speed but remain Δ seconds late.

Yet $R^{\mathcal{H}_{\text{Reg}}}(x_2) \leq \alpha^{\mathcal{H}_{\text{TAI}}}(x_2 - T_{\text{start}})$ so

$$R^{*\mathcal{H}_{\text{TAI}}}(x_2 + \Delta) \leq \alpha^{\mathcal{H}_{\text{TAI}}}(x_2 - T_{\text{start}}) \tag{14}$$

Combining Equations (13) and (14) proves

$$R^{*\mathcal{H}_{\text{TAI}}}(x_2 + \Delta) - R^{\mathcal{H}_{\text{TAI}}}(x_2) \leq 0 \tag{15}$$

Equation (15) proves that the delay of the packet output at x_2 (observed with $\mathcal{H}_{\text{TAI}}$) from the source exits the greedy shaper at $x_2 + \Delta$ (observed with $\mathcal{H}_{\text{TAI}}$). It has hence suffered a delay of Δ measured with $\mathcal{H}_{\text{TAI}}$, which is Δ more than the worst-case delay through the network element. The worst-case delay is hence lower-bounded by this reachable value. □

A.9 Proof of Proposition 7.2

Proof of Proposition 7.2. We use the service-curve property of PFRs. The kth PFR is non-adapted, its configuration is

$$\begin{cases} r_{\text{Reg}_k} = r_0 \\ b_{\text{Reg}_k} = b_0 \end{cases}$$

One one hand, $\gamma_{r_{\text{Reg}_k}, b_{\text{Reg}_k}}$ is a service curve of each PFR when observed with its clock $\mathcal{H}_{\text{Reg}_k}$. We use the synchronized part of Table 3 with $\mathcal{H}_g = \mathcal{H}_{\text{TAI}}$ and $\mathcal{H}_i = \mathcal{H}_{\text{Reg}_k}$ and obtain that $\beta^{\mathcal{H}_{\text{TAI}}}_{\text{PFR}_k} = \left(\delta_\eta \otimes \gamma_{r_0/\rho, b_0}\right) \vee \left(\delta_{2\Delta} \otimes \gamma_{r_0, b_0}\right)$ is a service curve of the PFR when observed with $\mathcal{H}_{\text{TAI}}$.

On the other hand, $\gamma_{r_{\text{Reg}_{k-1}}, b_{\text{Reg}_{k-1}}}$ is an arrival curve of the flow at the input of the k-th hop, when observed with $\mathcal{H}_{\text{Reg}_{k-1}}$. We apply the synchronized part of Table 2 with $\mathcal{H}_g = \mathcal{H}_{\text{TAI}}$ and $\mathcal{H}_i = \mathcal{H}_{\text{Reg}_{k-1}}$ and obtain that $\alpha^{\mathcal{H}_{\text{TAI}}}_{k-1} = \gamma_{\rho r_0, b_0 + r_0\eta} \wedge \gamma_{r_0, b_0 + 2r_0\Delta}$ is an arrival curve of the flow at the input of hop k when observed with $\mathcal{H}_{\text{TAI}}$. Its shape is given in Figure 17.

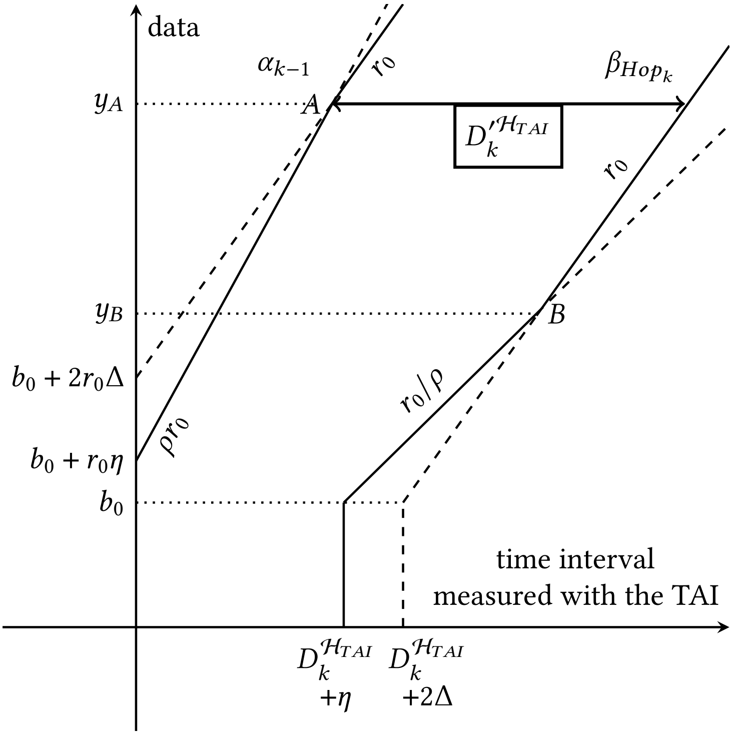


Fig. 17. Delay bound computation for the flow through the whole kth hop (S_k followed by regulator) when the regulator is a PFR and the network is synchronized.

Assume now that the delay through S_k, $D_k^{\mathcal{H}_{\mathrm{TAI}}}$ is computed using the above $\alpha_{k-1}^{\mathcal{H}_{\mathrm{TAI}}}$ as an arrival curve in S_k when observed with $\mathcal{H}_{\mathrm{TAI}}$. The whole hop ($S_k$ followed by regulator) offers, when observed with $\mathcal{H}_{\mathrm{TAI}}$, the service curve $\beta_{\mathrm{Hop}_k}^{\mathcal{H}_{\mathrm{TAI}}} = \delta_{D_k^{\mathcal{H}_{\mathrm{TAI}}}} \otimes \beta_{\mathrm{PFR}_k}^{\mathcal{H}_{\mathrm{TAI}}}$. Its shape is given in Figure 17.

We then compute the maximal horizontal distance in the figure. Geometrical considerations give

$$\begin{cases} y_A = b_0 + \dfrac{r_0}{\rho - 1}(2\Delta\rho - \eta) \\ y_B = b_0 + \dfrac{r_0}{\rho - 1}(2\Delta - \eta) \end{cases}$$

Hence, $y_A \geq y_B$ and the maximum horizontal distance is reached at A. We obtain

$$D_k'^{\mathcal{H}_{\mathrm{TAI}}} = D_k^{\mathcal{H}_{\mathrm{TAI}}} + 4\Delta \tag{16}$$

□

A.10 Proof of Proposition 7.3

Proof of Proposition 7.3. Take $\eta \geq 0, \rho > 1, \Delta > 0$ and $n \geq 3$. The following example respects the constraints of the model and has unbounded flow latencies.

We consider n sources, each source inputs a single flow to the FIFO system. We choose a FIFO system with infinite service and with $\mathcal{H}_{\mathrm{FIFO}} = \mathcal{H}_{\mathrm{IR}}$ such that the delay of the flows trough the FIFO system, when observed in the clock $\mathcal{H}_{\mathrm{IR}}$, is null. We further take $\mathcal{H}_{\mathrm{IR}} = \mathcal{H}_{\mathrm{TAI}}$. If the TAI delay is not bounded, then it is also not bounded in any other clock that meets the stability requirements of Equation (5).

Adversarial Clocks. We consider a starting point $x_1 \geq T_{\text{start}}$. We choose a slope s_1 such that $1 < s_1 \leq \min(1.5, \sqrt{\rho})$ and define

$$I = \frac{\Delta s_1}{s_1 - 1}$$

We also consider any $\epsilon > 0$ such that $\epsilon < I(1 - \frac{1}{s_1})$. ϵ is well defined because $s_1 > 1$. We also define $\tau = nI/s_1 + n\epsilon$ and $x_j = x_1 + (j-1)I/s_1 + (j-1)\epsilon$ for $j = 1 \ldots n$. Now, for every clock j, we choose as relative time function $d_{\text{IR}\to j}$ the following piecewise linear function

$$d_{\text{IR}\to j} = \begin{cases} t - \Delta/2 & \text{if } t \leq x_j \\ s_1(t - x_j) + x_j - \Delta/2 & \text{if } x_j < t \leq x_j + I/s_1 \\ \left.\begin{matrix} 1/s_1(t - x_j + I/s_1) \\ +I + x_j - \Delta/2 \end{matrix}\right\} & \text{if } x_j + \dfrac{I}{s_1} < t \leq x_j + \dfrac{I}{s_1} + I \\ t - \Delta/2 & \text{if } x_j + \dfrac{I}{s_1} + I < t \leq x_j + \tau \\ \tau + d_j(t - \tau) & \text{if } x_j + \tau < t \end{cases}$$

The shape of function $d_{\text{IR}\to j}$ is available in Figure 18. The slope between x_j and x_j+I/s_1 corresponds to the exaggerated compression of the time line in Figure 13a of the example in Section 7.2. We obtain directly the following properties for any j

- $d_{\text{IR}\to j}$ is continuous and strictly increasing
- The time-error function $t \mapsto d_{\text{IR}\to j}(t) - t$ is periodic with period τ for $t \geq x_j$

Also, for any j, j' $d_{j\to j'}(t) = d_{\text{IR}\to j'}(d^{-1}_{\text{IR}\to j}(t))$. As $s_1 \leq \sqrt{\rho}$ and $|d_{\text{IR}\to j}(t) - t| \leq \Delta/2$, any pair of clocks $(\mathcal{H}_j, \mathcal{H}_{j'})$ meets the constraints of Equations (5) and (6), which shows that our adversarial clocks are within the synchronized time model proposed in Section 4.1.

Adversarial Traffic Generation. : Let l be any arbitrary data size. Each source j is configured to send a packet of size l when its local clock reaches $d_j(x_j) + k\tau$ and $d_j(x_j) + k\tau + I$ for all $k \in \mathbb{N}$. Figure 19 presents the traffic generation of source j within one period, observed with its internal clock $\mathcal{H}_j$. When observed with $\mathcal{H}_j$, the traffic generation is periodic of period τ.

As $n \geq 3$ and $s_1 < 1.5$, $\tau \geq 2I + 2\epsilon \geq 2I$ and $\tau - I \geq I$. Hence, the minimum duration between two packets generated by source j, counted using $\mathcal{H}_j$ is I. This proves that each flow exits its respective source j with a leaky-bucket arrival curve $\gamma_{\frac{l}{I},l}$ (rate l/I, burst l) when observed using $\mathcal{H}_j$. We now assume that the interleaved regulator is configured with the same leaky-bucket arrival curve $\gamma_{\frac{l}{I},l}$ for all the flows.

Figure 20 presents the timeline of packets generated by source j but as observed with $\mathcal{H}_{\text{IR}}$. The IR has to regulate the same timeline for all the n inputs, based on its configuration. For $j = 1 \ldots n$ and for $k \in \mathbb{N}$ we note $A^1_{j,k} = x_j + k\tau$ the arrival time in the IR of the first packet of the the kth period of source j measured with $\mathcal{H}_{\text{IR}}$ and $A^2_{j,k} = x_j + k\tau + \frac{I}{s_1}$ the arrival time of the second packet, still measured with $\mathcal{H}_{\text{IR}}$. Also, note $D^1_{j,k}$ and $D^2_{j,k}$ their respective release time out of the IR, again measured using $\mathcal{H}_{\text{IR}}$.

Figure 21 presents the arrival and release times of the packets for two consecutive sources. Assume for instance that the source has been idle for a while, then $D^1_{j,k} = A^1_{j,k}$. The instant $A^2_{j,k}$ is, from the perspective of $\mathcal{H}_{\text{IR}}$, too soon by $I(1 - \frac{1}{s_1})$. Using the IR equations [39], the IR has to delay

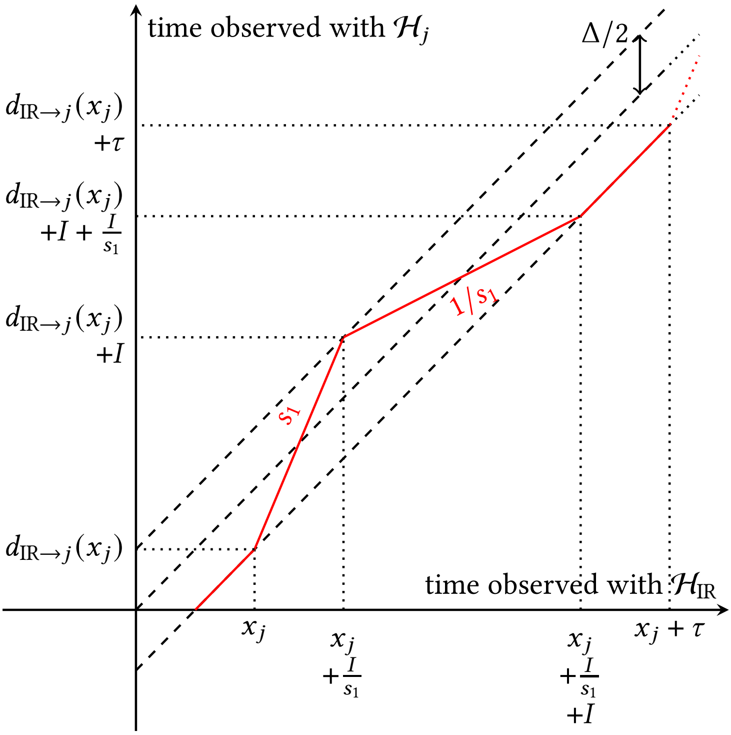

time observed with $\mathcal{H}_j$
$\Delta/2$
$d_{\mathrm{IR}\to j}(x_j)$ $+\tau$
$d_{\mathrm{IR}\to j}(x_j)$ $+I+\frac{I}{s_1}$
$d_{\mathrm{IR}\to j}(x_j)$ $+I$
$d_{\mathrm{IR}\to j}(x_j)$
$1/s_1$
s_1
time observed with $\mathcal{H}_{\mathrm{IR}}$
x_j
x_j $+\frac{I}{s_1}$
x_j $+\frac{I}{s_1}$ $+I$
$x_j+\tau$

Fig. 18. Shape of the time function $d_{\mathrm{IR}\to j}$. The shape is periodic, with period τ. When clock $\mathcal{H}_{\mathrm{IR}}$ reaches x_j, clock $\mathcal{H}_j$ starts counting the time faster (with a relative factor s_1) until $\mathcal{H}_{\mathrm{IR}}$ counts I/s_1 more seconds. Then, $\mathcal{H}_j$ counts the time slower (with a relative factor $1/s_1$). When the time-error function between $\mathcal{H}_{\mathrm{IR}}$ and $\mathcal{H}_j$ reaches $-\Delta/2$, $\mathcal{H}_j$ counts the time at the same speed as $\mathcal{H}_{\mathrm{IR}}$ until the next period starts.

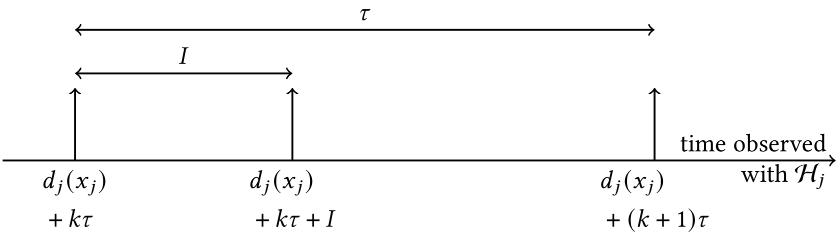


Fig. 19. Generation of packets as observed with $\mathcal{H}_j$. The traffic profile is periodic of period τ. Source j sends a packet when the internal clock reaches $d_j(x_j)+k\tau$ for some $k\in\mathbb{N}$, then it sends another packet after a duration of I counted using $\mathcal{H}_j$, and finally restarts at the next period.

the packet and

$$\forall j=1\dots n, \forall k\in\mathbb{N},\quad D^2_{j,k}\geq D^1_{j,k}+I \tag{17}$$

As $x_{j+1}=x_j+\frac{I}{s_1}+\epsilon$, packet $A^1_{j+1,k}$ arrives ϵ seconds after $A^2_{j,k}$ (measured using $\mathcal{H}_{\mathrm{IR}}$). As $\epsilon<I(1-\frac{1}{s_1})$, the packet at $A^1_{j+1,k}$ arrives before the previous packet of the previous source could be released out of the IR. Because the IR only looks at the head-of-line packet, and using the IR equations, we

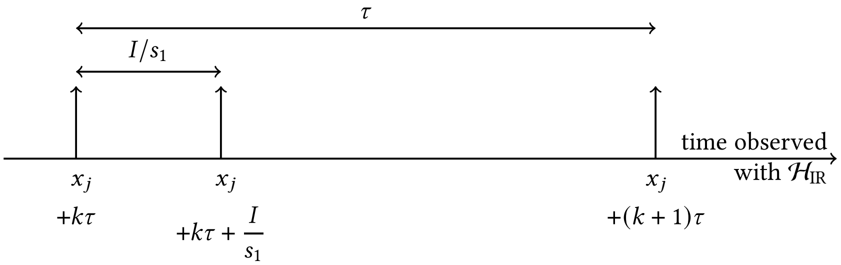


Fig. 20. Generation of packets as observed with $\mathcal{H}_{\mathrm{IR}}$. The traffic profile is periodic of period τ. When observing with $\mathcal{H}_{\mathrm{IR}}$, source j sends packets at $x_j + k\tau$ and $x_j + k\tau + \frac{I}{s_1}$ for all $k \in \mathbb{N}$.

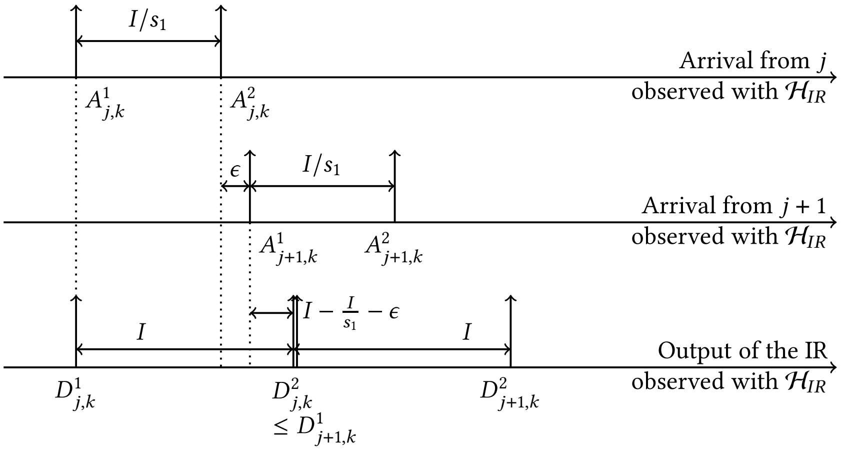


Fig. 21. Traffic arrival from two successive upstream sources, as observed with $\mathcal{H}_{\mathrm{IR}}$ and release time of the packets, again observed with $\mathcal{H}_{\mathrm{IR}}$.

obtain

$$\forall j = 1 \dots n-1, \forall k \in \mathbb{N}, \quad D^1_{j+1,k} \geq D^2_{j,k} \tag{18}$$

Combining Equations 17 and 18 gives, by induction,

$$D^2_{n,k} \geq D^1_{1,k} + nI \tag{19}$$

Now we can note that

$$\begin{aligned}
A^1_{1,k+1} &= x_1 + k\tau + \tau \\
&= x_1 + k\tau + n\frac{I}{s_1} + n\epsilon \\
&= x_1 + (n-1)\frac{I}{s_1} + (n-1)\epsilon + k\tau + \frac{I}{s_1} + \epsilon \\
&= x_n + k\tau + \frac{I}{s_1} + \epsilon \\
&= A^2_{n,k} + \epsilon
\end{aligned}$$

Hence, the first packet of the $(k+1)$th period of the first upstream source arrives ϵ seconds (counted with $\mathcal{H}_{\text{IR}}$) after the second packet of the kth period of the last source, so we also have

$$D^1_{1,k+1} \geq D^2_{n,k} \tag{20}$$

Combining equations 17 and 20 gives

$$D^1_{1,k} \geq D^1_{1,1} + (k-1)nI = x_1 + (k-1)nI \tag{21}$$

Because we have $D^1_{1,1} = x_1$, as the network was empty before. The delay suffered through the IR by the first packet of the kth period of the first source is, when measured with $\mathcal{H}_{\text{IR}}$

$$D^1_{1,k} - A^1_{1,k} \geq x_1 + (k-1)nI - x_1 - (k-1)\tau \tag{22}$$

$$\geq x_1 + (k-1)nI - x_1 - (k-1)n\frac{I}{s_1} - (k-1)n\epsilon \tag{23}$$

$$\geq (k-1)n\left(I\left(1 - \frac{1}{s_1}\right) - \epsilon\right) \tag{24}$$

As we have arbitrary selected ϵ such that $\epsilon < I(1 - \frac{1}{s_1})$, we obtain $I(1 - \frac{1}{s_1}) - \epsilon > 0$ thus the above delay lower-bound diverges as k increases, so the delay through the IR is unbounded when seen from $\mathcal{H}_{\text{IR}}$, which proves the instability.

Remark: Equation (24) proves that at each period of duration τ, the worst-case delay increases by $nI(1 - \frac{1}{s_1}) - n\epsilon$. The divergence of the worst-case delay per second is

$$\text{div} = \frac{nI(1 - \frac{1}{s_1}) - n\epsilon}{\frac{nI}{s_1} + n\epsilon}$$

This divergence is valid for any $\epsilon > 0$, with $\epsilon < I(1 - \frac{1}{s_1})$. Taking $\epsilon \longrightarrow 0$, the divergence can be as large as

$$\lim_{\epsilon \longrightarrow 0} \text{div} = \frac{nI(1 - \frac{1}{s_1})}{\frac{nI}{s_1}} = s_1 - 1$$

s_1 can be as large as $\sqrt{\rho}$, so the divergence of the worst-case delay can be as large as $\sqrt{\rho} - 1$, for any $n \geq 3, \Delta > 0$. □

Received January 2020; revised February 2020; accepted March 2020