A Theory of Traffic Regulators for Deterministic Networks with Application to Interleaved Regulators

Jean-Yves Le Boudec

***Abstract*—We introduce Pi-regulation, a new definition of traffic regulation which extends both the arrival curves of network calculus and Chang's max-plus g-regulation, and also includes new types of regulation such as packet rate limitations. We provide a new exact equivalence between min-plus and max-plus formulations of traffic regulation. We show the existence and a max-plus representation of per-flow minimal regulators, which extends the concepts of packetized greedy shapers and minimal g-regulators. We show that any minimal regulator, placed after any arbitrary system that is FIFO for the flow of interest, does not increase the worst-case delay of the flow. We extend the theory to interleaved regulation and introduce the concept of minimal interleaved regulator. It generalizes the Urgency Based Shaper that was recently proposed by Specht and Samii as a simpler alternative to per-flow regulators in deterministic networks with aggregate scheduling. With this regulator, packets of multiple flows are processed in one FIFO queue and only the packet at the head of the queue is examined against the regulation constraints of its flow. We show that any minimal interleaved regulator, placed after any arbitrary FIFO system does not increase the worst-case delay of the combination.**

***Index Terms*—Network Calculus, FIFO Systems, Regulators, Shapers**

I. Introduction

We are interested in First-In-First-Out (FIFO) or FIFO-per-class systems, as they are found e.g. in industrial networks or in Ethernet or routed networks with delay guarantees, see for example the current time sensitive networking (TSN) group of IEEE 802.1 [1] or the detnet group of IETF [2]. In FIFO networks, the burstiness of a flow increases at every hop where it shares a queuing point with other flows of its class, in direct relation to the burstiness of these flows [3]. The increased burstiness of this flow may, in turn, increase the burstiness of other flows in the same class. This creates a positive feedback loop, which causes large worst-case delays in FIFO per-class networks [4]. Another consequence is that computing worst-case delays in FIFO networks is a very difficult exercise [5]. Even in feed-forward topologies, finding proven bounds that are close to the worst case is difficult: the determination of the worst case is NP-hard [6] and can be addressed by linear programming [7]. Alternative approaches combine the network calculus building blocks of min-plus network calculus in various ways, leading to better bounds by using the Pay Multiplexing Only Once property [8] or the Aggregate Arrival Bounding technique [9]. However, none of these combinations seem to dominate the others and every case requires some tailored analysis [10]. These techniques are available in several software tools [11], [12].

A radical alternative is to avoid cascades of increased burstiness by re-shaping every flow at every hop [13]. This was used in tools that perform industry-floor automatisation by using network calculus [14]. However, this typically requires per-flow queuing at every hop, which defeats the purpose of FIFO networks.

Specht and Samii introduced in [15] a simpler alternative, under the name "Urgency Based Scheduler" (UBS). With a UBS, the packet at the head of the queue is examined against the regulation constraints of its flow; it is released at the earliest time at which this is possible without violating the constraints. Packets that are not at the head of the queue are not examined until they reach the head of the queue. The regulation constraints are either the "Length Rate Quotient" (LRQ) rule, or a leaky bucket constraint (we explain the details in Section III). The motivation is to avoid per-flow queuing, which is perceived as expensive, while keeping per-flow state, which is inexpensive. Similar ideas are used in the context of best-effort traffic in [16].

In [15], Specht and Samii compute a bound on the packet delay for a network of priority queues with constant rate servers and UBS. They use a trajectory analysis. More precisely, end-to-end latency bounds are obtained by examining, for each flow, a sequence of past dequeuing events, and assuming that there are periods of times in the past where queues were empty. The analysis is ad-hoc, and does not separate between the service process and the effect of the UBS. It is very complex and is extremely difficult to validate. Nonetheless, it is quite remarkable, and somewhat unexpected, that Specht and Samii find explicit latency bounds, in spite of the interleaving of flows in the UBS. Also they find, by inspection of formulas, that UBS does not increase the delay bound that they obtain for the priority scheduler. The goal of this paper is to provide a theory that (1) can be used to explain and formally prove these results, (2) isolates the effect of the UBS and of the specific assumptions on the rest of the network, (3) applies to general FIFO systems instead of priority queues, and (4) separates what is specific to leaky bucket and LRQ and what is a consequence of minimal regulation in general.

To this end, we introduce the concept of "Minimal Interleaved Regulator", which extends the concept of UBS to a very large class of regulation rules. Like the UBS, a minimal interleaved regulator examines only the packet at the head of the queue; it possibly delays this packet but also, due to FIFO, all following packets, which typically belong to other flows. However, we show that, when a minimal interleaved regulator is placed after an arbitrary FIFO system, the worst-case delay of the combination is the same as without the interleaved regulator. This shaping-for-free property is well-known with per-flow shapers and per-flow service curve elements [17]; as

we show later, it also holds with general per-flow minimal regulators and arbitrary FIFO systems settings, and surprisingly, it continues to hold with minimal *interleaved* regulators.

UBS assumes that flows are regulated either using LRQ or a leaky bucket constraint. Now the former is an instance of Chang's g-regulator [18], which uses max-plus algebra (see Section III-B), whereas the latter is an instance of an arrival curve constraint, which uses min-plus algebra (see Section III-C). It is known that a g-regulation constraint is not equivalent to an arrival curve constraint [19]. In contrast, in [20] Liebeherr shows that there is an isomorphism between the min-plus and max-plus representations of regulation constraints, and of the associated regulators (called "shapers"). Therefore the non-equivalence between Chang's g-regulation and arrival curve constraints is not explained by the choice a max-plus or min-plus algebra. Indeed, the difference is whether traffic constraints are expressed at an arbitrary packet arrival time (as with Chang's g-regulation constraint) or at an arbitrary point in time or space (as with arrival curves or the equivalent max-plus envelope of Liebeherr). In Theorem 1 we give a new result which formulates arrival curve constraints by means of packet arrival times, and thus clarifies the relation between these viewpoints.

This motivates the definition of a new formalism for regulators that encompasses both g-regulation constraints and arrival curves. More precisely, we introduce in Section III the concept of "Pi-regularity" and in Section IV of "minimal $\Pi-$regulator", adapted to the context of FIFO systems. We show that these concepts contain as special cases the classical greedy shaper of network calculus [21] as well as Chang's "minimal regulator" [18]. We show that it also contains other regulators, which cannot be expressed in these classical frameworks, such as the TSN packet rate regulator. We show that any per-flow minimal regulator does not increase the worst-case latency of any system that is FIFO per flow. We also provide a simple, intuitive proof for this result.

In Section V we introduce and analyze interleaved regulators, show their existence and derive in an intuitive and simple way our main result on the maximum latency induced by minimal interleaved regulators. Section II describes the notation and provides some background results. Proofs of lemmas and theorems are in appendix, except for the proofs of the theorems on worst-case latency, which have an independent interest, and are given in the main text.

II. Notation and General Pre-Requisites

A. Packet Sequences and FIFO Systems

We use a notation similar to Chang's marked point process notation in [18]. We consider an arbitrary FIFO system that has packet sequences as input and output. In some cases the input and output packet sequences belong to one single flow, but in general we are interested in packet sequences where packets may belong to different flows.

- $\mathbb{N} = \{0, 1, 2, ...\}$ and $\mathbb{N}^+ = \{1, 2, 3, ...\}$.
- $\mathbb{R}^+ = [0, \infty)$ is the set of non-negative real numbers.
- $\mathcal{F}$ is the set of sequences such that $A \in \mathcal{F}$ whenever $A = (A_1, A_2, ...)$ with $A_n \in [-\infty, +\infty)$ for $n \in \mathbb{N}^+$.
- $\mathcal{F}_{inc}$ is the subset of $\mathcal{F}$ of wide-sense increasing sequences, i.e. $A \in \mathcal{F}_{inc}$ if and only if $A \in \mathcal{F}$ and $A_n \leq A_{n+1}$ for all $n \in \mathbb{N}^+$.
- $\mathcal{G}$ is the set of positive integer-valued sequences. i.e. $L \in \mathcal{G}$ whenever $L = (L_1, L_2, ...)$ with $L_n \in \mathbb{N}^+$ for $n \in \mathbb{N}^+$.
- $x \vee y$ denotes the maximum of x and y, for $x, y \in [-\infty, +\infty]$.
- Whenever $h \in \mathbb{R}$ and $A \in \mathcal{F}$, $A + h$ and $h + A$ denote the sequence A' such that $A'_n = A_n + h$ for all $n \in \mathbb{N}^+$.
- For A, A' in $\mathcal{F}$ the notation $A \leq A'$ means that $A_n \leq A'_n$ for all $n \in \mathbb{N}^+$.
- The supremum of an empty set is $-\infty$.
- The infimum of an empty set is ∞.
- The summation of an empty set is 0.
- $\mathbf{1}_{\{C\}}$ is equal to 1 when the condition C is true and is equal to 0 otherwise.
- $\mathbb{I}_{\{C\}}$ is equal to 1 when the condition C is true and is equal to $+\infty$ otherwise.
- For $x \in \mathbb{R}$, $\lceil x \rceil$ is the ceiling of x, namely the smallest integer $\geq x$; $\lfloor x \rfloor$ is the floor of x, namely the largest integer $\leq x$.

A general packet sequence is a triple such as (A, L, F) where:

1) The first element A is the sequence of packet dates, i.e. the time instants $A = (A_1, A_2, ...)$ at which the packets are observed. We assume that packet numbering follows chronological order; simultaneous packet observation times in the same sequence are possible. Thus we require that $A \in \mathcal{F}_{inc}$. Depending on the context, we may denote the sequence of packet dates with A (for arrivals) or D (for departures) etc.
2) The second element is a sequence of packet lengths $L = (L_1, L_2, ...)$ with $L \in \mathcal{G}$. Packet lengths are counted in some arbitrary data unit, typically in bytes or words of a fixed number of bytes.
3) The third element is a sequence of flow numbers $F = (F_1, F_2, ...)$ with $F_n \in \mathbb{N}^+$, and $n \in \mathbb{N}^+$. In other words, $F_n = f$ means that packet n belongs to flow f. To avoid cumbersome notation, we assume without loss of generality that the set of flow numbers is finite and that the subsequence of packets of flow f is infinite, for every f.

With this notation, we can express a FIFO system as a system that maps a given input packet sequence (A, L, F) to an output packet sequence (D, L, F) such that $A \leq D$. Observe that our definition of packet sequences allows simultaneous packet arrivals; inside a packet sequence, different packets with identical dates must be numbered differently, i.e., we assume that there is a tie-breaking rule for ordering packets. Our definition of FIFO system requires that such a numbering is preserved.

When a packet sequence is for a single flow, we simply describe it as a couple such as (A, L). A system S is FIFO for a flow with input sequence (A, L) if the output packet sequence for this flow is (D, L) for some $D \in \mathcal{F}_{inc}$ such that $A \leq D$.

B. Notation for Flows Inside a Packet Sequence

In Section V we need some specific notation for flows inside a packet sequence. Given a sequence F of flow numbers, we define $I()$ as the function that returns the index of packet n in its flow. In other words:

$$I(n) = \text{card}\left\{m \in \mathbb{N}^+ : m \leq n \text{ and } F_m = F_n\right\} \quad (1)$$

We also define the function ind() such that $\text{ind}(f,i)$ is the index in the packet sequence of the i^{th} packet of flow f; in other words:

$$\text{ind}(f,i) = n \Leftrightarrow (F_n = f \text{ and } I(n) = i) \quad (2)$$

Note that the functions $I()$ and ind() depend on the packet sequence F but we leave out the dependency on F for the sake of simplicity in notation.

When a flow f is present in a packet sequence (A, L, F) we define A^f [resp. L^f] as the subsequence extracted from A [resp. L] by keeping only the packet dates [resp. lengths] corresponding to a packet of flow f, namely

$$A^f_i = A_{\text{ind}(f,i)},\ L^f_i = L_{\text{ind}(f,i)} \text{ for all } i \in \mathbb{N} \quad (3)$$

For example, assume the sequence of flow numbers is $F = (3,4,1,2,1,3...)$, i.e. the first packet belongs to flow 3 ($F_1 = 3$), the second to flow 4 ($F_2 = 4$), etc. Packets 3 and 5 belong to flow 1, packet 5 is the second packet of flow 1 so $\text{ind}(1,2) = 5$, $I(5) = 2$, $A^1 = (A_3, A_5, ...)$ and $A^3 = (A_1, A_6, ...)$.

C. Pseudo-Inverses

Let $f()$ be a wide-sense increasing function $\mathbb{R}^+ \to \mathbb{R}^+$. Let $f^{\downarrow}() : \mathbb{R}^+ \to [0,\infty]$ be its lower pseudo-inverse, defined by [18], [20]

$$f^{\downarrow}(x) = \inf\{s \geq 0 \text{ such that } f(s) \geq x\} \quad (4)$$
$$= \sup\{s \geq 0 \text{ such that } f(s) < x\} \quad (5)$$

The lower pseudo-inverse is the same as the pseudo-inverse in [21]. Similarly, let $f^{\uparrow}() : \mathbb{R}^+ \to [0,\infty]$ be its upper pseudo-inverse, defined by [18], [20]

$$f^{\uparrow}(x) = \sup\{s \geq 0 \text{ such that } f(s) \leq x\} \quad (6)$$
$$= \inf\{s \geq 0 \text{ such that } f(s) > x\} \quad (7)$$

Note that $f^{\downarrow}(0) = 0$. Furthermore [21, Theorem 3.1.2]:

$$f(t) \geq x \Rightarrow t \geq f^{\downarrow}(x) \quad (8)$$

and

$$t > f^{\downarrow}(x) \Rightarrow f(t) \geq x \quad (9)$$

but the converse may not hold. However, we have:

Lemma 1: If f is right-continuous then for all $t, x \in \mathbb{R}^+$:

$$t \geq f^{\downarrow}(x) \Leftrightarrow f(t) \geq x \quad (10)$$

It is known [22, Theorem (4.2.1)] that $f()$ is not necessarily continuous but (1) its set of discontinuities is countable and (2) $f()$ has a limit to the right and to the left at every point. We denote with $f^+()$ the right-limit of $f()$, defined for $t \in \mathbb{R}^+$ by $f^+(t) = \lim_{s \to t, s > t} f(s) = \inf_{s > t} f(s)$. Note that $f^+()$ is right-continuous and whenever $0 \leq s < t$:

$$f(s) \leq f^+(s) \leq f(t) \quad (11)$$

Similarly, we denote with $f^-()$ the left-limit of $f()$, defined for $t \in \mathbb{R}^+$ by $f^-(t) = \lim_{s \to t, s < t} f(s) = \sup_{s < t} f(s)$.

Last, we will use the following results, which are true for any wide-sense increasing function $f() : \mathbb{R}^+ \to \mathbb{R}^+$

Lemma 2: $f^{\downarrow} = (f^+)^{\downarrow}$ and $f^{\uparrow} = (f^-)^{\uparrow}$.

Lemma 3: $(f^{\uparrow})^- = f^{\downarrow}$ and $(f^{\downarrow})^+ = f^{\uparrow}$.

Proofs of the lemmas are in appendix.

III. Pi-Regularity

In this section and the next section we are interested in regulation of a single flow. We start by introducing a new concept, "Pi-regularity", which extends both arrival curve constraints and Chang's g-regularity. This concept will prove to be essential in analyzing interleaved regulators in Section V.

A. Definition of Pi-Regularity

Our new definition of regularity uses an operator, say Π, which must satisfy the following conditions.

- C1 Π is a mapping $\mathcal{F}_{inc} \times \mathcal{G} \to \mathcal{F}$, i.e. Π takes as argument a single-flow packet sequence (A, L) and transforms it into a sequence of time instants. The output sequence is in $\mathcal{F}$, i.e. is not necessarily monotonic.
- C2 Π **is causal**, in the following sense: if $\Pi(A, L) = A'$ then the value of A'_n may depend on $A_1, ...A_{n-1}$ and $L_1, ..., L_n$ but not on A_m for $m \geq n$ nor on L_m for $m \geq n+1$.
- C3 Π **is homogeneous with respect to** A**:** $\Pi(A + h, L) = \Pi(A, L) + h$ for any constant $h \in \mathbb{R}$ and any sequences $A \in \mathcal{F}_{inc}, L \in \mathcal{G}$.
- C4 Π **is isotone with respect to** A**:** whenever $A, A' \in \mathcal{F}_{inc}$ are such that $A \leq A'$ then also $\Pi(A, L) \leq \Pi(A', L)$ for any sequence $L \in \mathcal{G}$.

Observe that the causality condition C2 is a little unusual, as it does not allow A'_n to depend on A_n. This is required for the theory to work; in particular, if we would allow A'_n to depend on A_n, the max-plus representations of the minimal regulators in Eq.(44) and Eq.(49) would be circular and would not be useful.

Note that if an operator Π is causal and homogeneous with respect to A then necessarily $\Pi(A, L)_1 = -\infty$ for any input (A, L). This can be derived by observing first that $\Pi(A, L)_1$ is independent of A by causality and therefore has the form $c(L) \in [-\infty, \infty)$. Second, by homogeneity $\Pi(A + h, L) = \Pi(A, L) + h$ for any real number h so $c(L) = c(L) + h$ for any h, which is possible only if $c(L) = -\infty$.

In the next three subsections we give several examples of such operators. They all have the form

$$\Pi(A, L)_n = \max_{1 \leq m \leq n-1} \{A_m + H_{m,n}(L)\} \quad (12)$$

for some appropriate choice of the array $H_{m,n}(L)$ (with $H_{m,n}(L) \in [-\infty, \infty)$ for $m < n \in \mathbb{N}^+$, $L \in \mathcal{G}$ and such

that $H_{m,n}$ does not depend on L_j for $j \geq n+1$). An operator defined by an equation of the form Eq.(12) is a max-plus-linear operator and clearly satisfies C1-C4. Note that here the identity $\Pi(A)_1 = -\infty$ follows from the fact that the max in Eq.(12) is $-\infty$ when the set of indices is empty.

Definition 1 (Pi-Regularity.): Given some operator Π that satisfies C1-C4, we say that a single-flow packet sequence (A, L) is $\Pi-$regular if $A \geq \Pi(A, L)$.

We next give some examples and show how this definition extends existing frameworks.

B. Chang's g-Regularity

Given some wide-sense increasing function (or sequence) $g() : \mathbb{N} \to \mathbb{R}^+$ such that $g(0) = 0$, Chang [18] defines a single-flow packet sequence (A, L) as g-regular if and only if for all $m, n \in \mathbb{N}^+$ such that $m < n$ we have

$$A_n - A_m \geq g\left(L_m + ... + L_{n-1}\right) \tag{13}$$

It is immediate to see that g-regularity is a special case of Pi-regularity with the operator Π given by

$$\Pi(A, L)_n = \max_{1 \leq m \leq n-1} \left\{ A_m + g\left(\sum_{j=m}^{n-1} L_j \right) \right\} \tag{14}$$

In the case where $g(x) = x/r$ for some r, g-regularity is called the "Length Rate Quotient" (LRQ) constraint with rate r in [15]. Any flow that is observed on a physical communication link of rate r satisfies the LRQ(r) constraint. Because in this case g is linear, it can easily be seen that, for the LRQ(r) regulation constraint, Eq.(13) is equivalent to the simpler condition

$$A_n - A_{n-1} \geq \frac{L_{n-1}}{r} \tag{15}$$

for all $n \in \mathbb{N}^+$, $n \geq 2$. Therefore, LRQ(r) regularity is an instance of Pi-regularity with the operator $\Pi^{LRQ(r)}$ given by

$$\begin{array}{l} \Pi^{LRQ(r)}(A, L)_n = A_{n-1} + \frac{L_{n-1}}{r} \text{ for } n \geq 2 \\ \Pi^{LRQ(r)}(A, L)_1 = -\infty \end{array} \tag{16}$$

C. Arrival Curve Constraint

This is a classical network calculus constraint, originally expressed with min-plus algebra. It uses some wide-sense increasing function $\sigma() : \mathbb{R}^+ \to \mathbb{R}^+$, called "arrival curve" or "min-plus traffic envelope"; it can always be assumed without loss of generality that σ is sub-additive and $\sigma(0) = 0$ (but we don't need such an assumption in the following theorems). The celebrated leaky bucket constraint LB(r, b) [23] corresponds to $\sigma(t) = rt + b$ for $t > 0$ and $\sigma(0) = 0$, where r is the leaky bucket rate and b the burstiness. The arrival curve constraint is expressed in terms of the cumulative arrival function $R(t)$, defined for $t \geq 0$, and which can be derived from the single-flow packet sequence (A, L) by

$$R(t) = \sum_{n \in \mathbb{N}^+} L_n \mathbf{1}_{\{A_n < t\}} \tag{17}$$

where the indicator function $\mathbf{1}_{\{A_n < t\}}$ has the value 1 when the condition $\{A_n < t\}$ is true and 0 otherwise. In this context, we assume that time is nonnegative and in particular $A_n \geq 0$. The arrival curve constraint requires that

$$R(t) - R(s) \leq \sigma(t - s) \text{ for all } 0 \leq s \leq t \tag{18}$$

Liebeherr shows in [20] that the arrival curve constraint can be expressed in max-plus algebra. To this end, he introduces the arrival time function $T() = R^{\uparrow}()$, which is equal to the upper pseudo-inverse of cumulative arrival function $R()$, and is also given by

$$T(x) = \inf_{n \in \mathbb{N}^+} \left(A_n \mathbb{I}_{\{L_1 + ... + L_n > x\}} \right) \text{ for } x \geq 0 \tag{19}$$

where the indicator function $\mathbb{I}_{\{L_1+...+A_n>x\}}$ has the value 1 when the condition $\{A_n < t\}$ is true and $+\infty$ otherwise. Note that $R()$ can be recovered from $T()$ since $R() = T^{\downarrow}()$. Liebeherr shows that the arrival curve constraint Eq.(18) is equivalent to the condition

$$T(y) - T(x) \geq \lambda(y - x) \text{ for all } 0 \leq x \leq y \tag{20}$$

with $\lambda() = \sigma^{\uparrow}()$. In other words, the upper pseudo-inverse $\sigma^{\uparrow}()$ is a max-plus traffic envelope of $T()$ if and only if $\sigma()$ is an arrival curve, or min-plus traffic envelope, of $R()$. If we enforce that $\sigma()$ be left continuous, then $\sigma()$ is the lower pseudo-inverse of $\sigma^{\uparrow}()$ and thus there is exact equivalence between min-plus and max-plus traffic envelopes.

To make the link between arrival curve constraints and Pi-regularity, we need to go one step further and understand the relation between traffic constraints expressed at an arbitrary point in time (or space, as in Eq.(18) and Eq.(20)) and constraints that are expressed at packet arrival times. This is provided by the following theorem.

Theorem 1: [Formulation of Arrival Curve Constraint by Means of Packet Arrival Times] Consider a single-flow packet sequence (A, L) and the associated cumulative arrival function $R()$ given by Eq.(17). Let $\sigma()$ be some wide-sense increasing function, $\sigma^{+}()$ its right-limit and $\sigma^{\downarrow}()$ the lower pseudo-inverse of $\sigma()$ (and hence, by Lemma 2 also of $\sigma^{+}()$).

The following three conditions are equivalent:

1) The arrival curve constraint in Eq.(18) is satisfied;
2) For any m, n with $1 \leq m \leq n$:

$$\sum_{j=m}^{n} L_j \leq \sigma^{+}(A_n - A_m) \tag{21}$$

3) For any m, n with $1 \leq m \leq n$:

$$A_n - A_m \geq \sigma^{\downarrow}\left(\sum_{j=m}^{n} L_j \right) \tag{22}$$

The proof is in appendix. We give above a version of the theorem using the arrival curve $\sigma()$. An equivalent formulation can be given, assuming that we are given not the arrival curve $\sigma()$ but the max-plus traffic envelope $\lambda()$. Then define $\sigma = \lambda^{\downarrow}$ so that Eq.(18) is equivalent to Eq.(20). Using Lemma 3 we have $\lambda^{\uparrow} = \sigma^{+}$ and $\lambda^{-} = \sigma^{\downarrow}$, so that Eq.(21) is equivalent to

$$\sum_{j=m}^{n} L_j \leq \lambda^{\uparrow}(A_n - A_m) \tag{23}$$

and Eq.(22) is equivalent to

$$A_n - A_m \geq \lambda^- \left(\sum_{j=m}^{n} L_j \right) \tag{24}$$

which shows the duality between min-plus and max-plus representations.

Theorem 1 shows that an arrival curve constraint, which is originally defined by using cumulative arrival functions, can also be expressed *exactly* by using packet arrival times. Thus, it establishes an exact equivalence between a min-plus oriented representation, and a max-plus oriented representation.

An important outcome is that the condition in Eq.(22) or Eq.(24) has the form in Eq.(12); this shows that an arrival curve constraint is a special form of Pi-regularity. The operator Π that corresponds to an arrival curve $\sigma()$ is given by

$$\Pi(A, L)_n = \max_{1 \leq m \leq n-1} \left\{ A_m + \sigma^{\downarrow} \left(\sum_{j=m}^{n} L_j \right) \right\} \tag{25}$$

The theorem also sheds some light on the difference between arrival curve constraints and Chang's g-regularity. Notice that the arrival curve constraint in Eq.(18) implies the condition

$$\sum_{j=m}^{n-1} L_j \leq \sigma(A_n - A_m), \tag{26}$$

for any $m \leq n \in \mathbb{N}^+$. This can be derived from a direct application of Eq.(18) to $s = A_m, t = A_n$. However, it can easily be seen that the converse is not true, i.e. this last condition is not equivalent to the arrival curve constraint – compare to Eq.(21). Observe now that the definition of g-regularity involves only equations such as Eq.(26). In particular, it does not involve the length L_n of the current packet, which explains why it cannot be equivalent to arrival curve constraints and why published relations between the two involve a bound with a term in $L^{\max}$, the maximum packet size of the flow.

In the rest of this subsection we apply the above theorem to two classic arrival curve constraints.

1) Leaky Bucket Constraints: For the single-leaky-bucket constraint LB(r, b), we can take $\sigma(t) = rt + b$ for some positive r and b, so that $\sigma^+(t) = \sigma(t)$ and $\sigma^{\downarrow}(x) = 0 \vee \frac{x-b}{r}$. The condition Eq.(22) is equivalent to

$$A_n \geq A_m + \frac{\left(\sum_{j=m}^{n} L_j \right) - b}{r} \tag{27}$$

and the Pi-regularity condition can be expressed by

$$\begin{aligned} A_n \quad \geq \quad & \left(A_1 + \frac{\left(\sum_{j=1}^{n} L_j \right) - b}{r} \right) \vee \ldots \\ & \vee \left(A_{n-1} + \frac{\left(\sum_{j=n-1}^{n} L_j \right) - b}{r} \right) \end{aligned} \tag{28}$$

Note that $A_n \geq A_m$ is always true by construction for any $1 \leq m \leq n-1$. Therefore, the constraint in the previous equation is equivalent to

$$A_n \geq \max_{1 \leq m \leq n-1} \left\{ A_m + \frac{\left(\sum_{j=m}^{n} L_j \right) - b}{r} \right\} \tag{29}$$

The operator $\Pi^{LB(r,b)}$ that corresponds to the leaky bucket constraint LB(r, b) is therefore given by

$$\Pi^{LB(r,b)}(A, L)_n = \max_{1 \leq m \leq n-1} \left\{ A_m + \frac{\left(\sum_{j=m}^{n} L_j \right) - b}{r} \right\} \tag{30}$$

2) Staircase Arrival Curve: The staircase arrival curve SC(τ, b) is defined for $\tau > 0, b > 0$ by

$$\sigma(t) = b \left\lceil \frac{t}{\tau} \right\rceil, \ t \geq 0 \tag{31}$$

and is used to express the constraint that at most b data units can be observed over any window of fixed duration τ [21]. Here $\sigma()$ is left-continuous and straightforward computations give:

$$\sigma^+(t) = b \left\lfloor \frac{t}{\tau} + 1 \right\rfloor, \ t \geq 0 \tag{32}$$

$$\sigma^{\downarrow}(x) = \tau \left\lceil \frac{x}{b} - 1 \right\rceil, \ x > 0 \tag{33}$$

$$\sigma^{\downarrow}(0) = 0 \tag{34}$$

The operator $\Pi^{SC(\tau,b)}$ that corresponds to the staircase arrival curve SC(τ, b) is therefore given by

$$\Pi^{SC(\tau,b)}(A, L)_n = \max_{1 \leq m \leq n-1} \left\{ A_m + \tau \left\lceil \frac{\left(\sum_{j=m}^{n} L_j \right) - b}{b} \right\rceil \right\} \tag{35}$$

D. Limits on Packet Rate

Packet rate limitations are used to put limits on the processing demand in networking boxes. For example, the IEEE TSN working group specifies a limit, say $K \in \mathbb{N}^+$ on the number of packets sent by a flow over a specified duration, say $\tau \geq 0$. If all packets are of the same size, say ℓ, this is the same as a staircase arrival curve constraint SC(τ, b) with $b = K\ell$. By Theorem 1 and Eq.(33), in this specific cases such a packet rate constraint is equivalent to

$$A_n - A_m \geq \tau \left\lceil \frac{n - m + 1 - K}{K} \right\rceil \tag{36}$$

If packets are not all of the same size, this cannot be specified exactly as an arrival curve constraint nor as a g-regulation constraint. However, it is still expressed by the constraint in Eq.(36). Therefore, this "TSN packet rate regulation" is an instance of Pi-regularity, with the operator $\Pi^{TSN(\tau,K)}$ defined by

$$\Pi^{TSN(\tau,K)}(A, L)_n = \max_{1 \leq m \leq n-1} \left\{ A_m + \tau \left\lceil \frac{n - m + 1 - K}{K} \right\rceil \right\} \tag{37}$$

Other forms of packet rate limitations can be defined. For example, a packet spacing constraint PS(τ) can be defined by

$$A_n - A_{n-1} \geq \tau, \text{ for all } n = 2, 3... \tag{38}$$

where $\tau \geq 0$ is the spacing interval. Note the analogy with an LRQ regulation constraint. This regulation constraint, is obviously an instance of Pi-regularity, with the operator $\Pi^{PS(\tau)}$ given by

$$\begin{array}{l} \Pi^{PS(\tau)}(A,L)_n = A_{n-1} + \tau \text{ for } n \geq 2 \\ \Pi^{PS(\tau)}(A,L)_1 = -\infty \end{array} \tag{39}$$

Last, similar to leaky bucket constraints, we can define a packet rate constraint by allowing a packet rate limit ρ with some packet burstiness constraint K. In other words, the "packet burstiness" constraint PB(ρ, K) specifies that the number of packets observed over an interval of duration t must be upper bounded by $\rho t + K$. A benefit of this formulation is its superposition property, i.e. if every flow f in a set S of flows is PB(ρ_f, K_f) constrained, then the superposition is PB(ρ, K) constrained, with $\rho = \sum_{f \in S} \rho_f$ and $K = \sum_{f \in S} K_f$. This follows immediately from the definition.

Similarly to Eq.(29), the PB(ρ, K) constraint can be expressed by the condition

$$A_n \geq \max_{1 \leq m \leq n-1} \left\{ A_m + \frac{n - m + 1 - K}{\rho} \right\} \tag{40}$$

In [24], Jiang says that a flow is (λ, ν) constrained if for all $m < n$

$$A_n - A_m \geq \frac{1}{\lambda} (n - m - \nu) \tag{41}$$

It is straightforward to see that this is equivalent to Eq.(40) with $\lambda = \rho$ and $\nu = K - 1$, in other words, Jiang's (λ, ν) constraint is the same as the PB($\lambda, \nu + 1$) constraint.

It follows from Eq.(40) that the packet burstiness constraint (hence also the (λ, ν) constraint) is an instance of Pi-regularity, with the operator $\Pi^{PB(\rho,K)}$ given by

$$\Pi^{PB(\rho,K)}(A,L)_n = \max_{1 \leq m \leq n-1} \left\{ A_m + \frac{n - m + 1 - K}{\rho} \right\} \tag{42}$$

Obviously, all of the constraints defined in this section can neither be expressed with arrival curves nor with g-regularity.

E. Combination of Regulation Constraints

Pi-regulation constraints can easily be combined, by taking the maximum of the operators. Indeed, it immediately follows from Definition 1 that a flow (A, L) is both Π^1 and Π^2-regular if and only if it is Π-regular, with Π being the maximum of Π^1 and Π^2, defined by

$$\Pi(A,L)_n = \Pi^1(A,L)_n \vee \Pi^2(A,L)_n \tag{43}$$

for all $n \in \mathbb{N}^+$. It is straightforward to verify that if Π^1 and Π^2 both satisfy C1-C4, then so does Π.

IV. Per-Flow Π−Regulator

After defining Pi-regularity we can now define a per-flow "Π-regulator" as a FIFO system that may delay some or all of the packets of a flow in order to make sure that the resulting output is Π-regular, for some operator Π. A *minimal* Π-regulator is one that outputs the packets of the flow as early as possible. Its existence is shown next.

A. Minimal Per-Flow Π−Regulator

Theorem 2: Consider a single-flow packet sequence (A, L) and let Π be an operator that satisfies C1-C4. The "minimal Π regulator" is defined as the FIFO system that transforms the input packet sequence (A, L) into the output packet sequence (D, L) such that $D_1 = A_1$ and

$$D_n = \max \{ A_n, D_{n-1}, \Pi(D,L)_n \} \tag{44}$$

1) The system defined in this way is a Π-regulator for this flow.
2) (Minimality:) For any other Π-regulator that transforms (A, L) into say (D', L) we have $D'_n \geq D_n$ for all $n \in \mathbb{N}^+$.
3) The flow (A, L) is Π-regular if and only if $D = A$.

The proof is in appendix. Note that the definition in Eq.(44) may appear to be circular, but it is not. This is because Π is assumed to be causal as in condition C2, which implies that $\Pi(D, L)_n$ depends only on $D_1, ..., D_{n-1}$. Therefore, the output sequence D is well defined by the initial condition $D_1 = A_1$ and Eq.(44).

To each of the regulators described in the previous section is thus associated a corresponding minimal per-flow regulator, whose input-output equation is given by Eq.(44). Sometimes this equation can be simplified by removing redundant terms in the maximum operation or in the expansion of $\Pi(D, L)_n$; for example, for the packet spacing constraint PS(τ), Eq.(44) becomes $D_n = A_n \vee D_{n-1} \vee (D_{n-1} + \tau)$, which, because $\tau \geq 0$, can be written more simply as

$$D_n = A_n \vee (D_{n-1} + \tau) \tag{45}$$

We next show that Chang's minimal regulator and the classic packetized greedy shaper [25] are all special cases of minimal per-flow Π-regulators.

B. Chang's Minimal Regulator, LRQ(r)-Regulator.

Chang's minimal g-regulator is the minimal FIFO system that delivers an output that is g-regular [18]. We have seen in Section III-B that g-regularity is equivalent to Pi-regularity, with Π given by Eq.(14). Therefore, the minimal g-regulator is the minimal Π-regulator with Π given by Eq.(14).

In particular for the minimal LRQ(r)-regulator, a simpler formulation of the operator Π is given by Eq.(16), from where we derive the input-output equations $D_n = \max \left\{ A_n, D_{n-1}, D_{n-1} + \frac{L_{n-1}}{r} \right\}$. Observe that $\frac{L_{n-1}}{r} \geq 0$ so the term D_{n-1} can be removed from the max. The input-output equations of the minimal LRQ(r)-regulator are thus $D_1 = A_1$ and for $n \geq 2$:

$$D_n = \max \left\{ A_n, D_{n-1} + \frac{L_{n-1}}{r} \right\} \tag{46}$$

C. Packetized Greedy Shaper.

Given some arrival curve $\sigma()$, a packetized shaper is a system that delivers an output that is packetized and satisfies the arrival curve constraint $\sigma()$. Among all packetized shapers, the packetized *greedy* shaper is the one that delivers its output as early as possible [25]. The input and output of a packetized shaper are cumulative arrival functions such as $R()$ in Eq.(17). Given a sequence of packet lengths L and for inputs and outputs that are packetized, there is a one-to-one mapping between the $R()$ function and the sequence of packet arrival times. Furthermore, by Theorem 1, an arrival curve constraint is a form of Pi-regularity, with Π given by Eq.(25). It is then straightforward to see that a packetized greedy shaper for the arrival curve $\sigma()$ is the same as the minimal Π-regulator with Π given by Eq.(25).

In the special case of a single leaky bucket constraint $\text{LB}(r,b)$, the packetized greedy shaper is called the leaky bucket shaper. Here the Π operator can take the simpler form in Eq.(30). The input-output equations of the leaky bucket shaper $\text{LB}(r,b)$ are thus $D_1 = A_1$ and for $n \geq 2$:

$$\begin{aligned} D_n &= A_n \vee D_{n-1} \vee \left(A_m + \frac{\left(\sum_{j=m}^{n} L_j\right) - b}{r} \right) \\ &\quad \vee \ldots \vee \left(A_{n-1} + \frac{\left(\sum_{j=n-1}^{n} L_j\right) - b}{r} \right) \end{aligned} \tag{47}$$

The relation above is the max-plus equation of a leaky bucket shaper. It is not the best formula for a practical implementation (see [25] for a discussion of the different implementations of leaky bucket shapers and their combinations) but it can be used to derive formal properties of such shapers, as we do in Section V.

D. Minimal Per-Flow Regulator Does Not Increase Worst-Case Delay

Regulators are used as a means to avoid burstiness cascades. However, they also add some delay, which needs to be accounted for, when computing end-to-end delay bounds. We show next a "reshaping-for-free" property, which is reminiscent of a similar property of networks that use per-flow queuing [17].

More precisely, assume that a flow (A,L) is fed into a system that is FIFO for this flow (Figure 1). This system can be for example a localized system such as a queuing point inside a switch, router or end-system, which handles multiple flows and respects the order of packets for this flow; or it can be an entire network path that is FIFO for this flow. Assume that, at the input, the flow is Π-regular. This property is, generally, lost by the output flow (D,L). Assume that we place a *minimal* regulator for this flow, just after the output, in order to re-create the regularity that was lost by traversing the FIFO node and let (E,L) be the reshaped flow, i.e., the output sequence of the minimal regulator. The following theorem establishes that the minimal regulator does not add anything to the worst-case delay of the FIFO system.

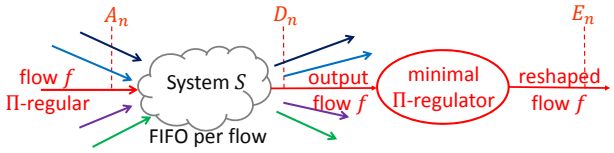

Fig. 1: Configuration for Theorem 3.

Theorem 3: Assume one Π-regular flow is input to a system S that is FIFO for this flow. Assume that the operator Π satisfies conditions C1-C4. The output flow is fed to a minimal regulator with same operator Π. The worst-case delay experienced by this flow through the combination is the same as the worst-case delay experienced through system S alone.

In other words, with the notation above:

$$\sup_{n\in\mathbb{N}^+} (D_n - A_n) = \sup_{n\in\mathbb{N}^+} (E_n - A_n) \tag{48}$$

The proof is illuminating and is given next. It shows that the theorem immediately derives from the minimality of the regulator.

Proof: Let d be the worst-case delay for this flow at system S. If $d = +\infty$ the conclusion is trivially true, thus we can assume that d is finite.

Replace the minimal Π-regulator with a *damper* with parameter d [26], as in Figure 2. The damper, with parameter d, is a (theoretical) device that knows the time of arrival A_n for every packet n and delivers it at time $A_n + d$. The damper input is the flow (D,L) therefore the damper is a system that is FIFO for this flow, as defined in Section II-A, if $D_n \leq A_n + d$, which holds because d is an upper bound to the delay of this flow through S. The output of the damper is $(d + A, L)$; by property C3, it is also Π-regular. Therefore, the damper is a Π-regulator for this flow. By definition of the minimal Π-regulator, it follows that $E_n \leq A_n + d$.

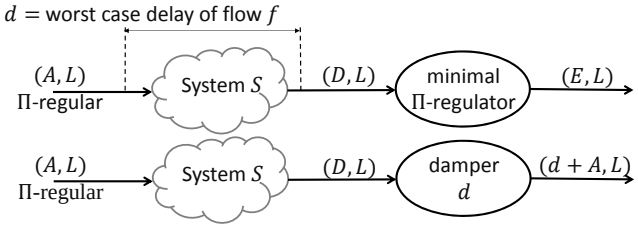

Fig. 2: Proof of Theorem 3.

◇

Theorem 3 extends of a well-known result concerning packetized greedy shapers, which states that re-shaping does not increase the delay bound provided by service curve elements. Here, our result is more general as it applies to any minimal regulator, not just to packetized greedy shapers.

V. Interleaved Regulator

In this section we consider a packet sequence (A,L,F) of several multiplexed flows. Recall here that L_n is the length

of the n^{th} packet, which belongs to flow F_n. Assume that for every flow f we have a regulation constraint, with operator Π^f for flow f. The regulation operators Π^f may be of any kind, as long as they satisfy the conditions C1-C4. For example, the regulation operator may be of the LRQ type, leaky bucket or packet rate limit, or any combination of these operators. Furthermore, the regulation operators of different flows need not be of the same type. Recall that saying that flow f is Π^f-regular means here that $A^f \geq \Pi^f(A^f, L^f)$ where A^f, L^f are the sequences extracted from A and L by keeping only the indices corresponding to packets of flow f.

In this context we define an "interleaved regulator" as a system that is FIFO and may delay some or all of the packets of the input sequence so that every flow f inside the output sequence is Π^f-regular. Note that the FIFO condition imposes that a packet of a flow may not be delivered before a packet of some other flow that arrived before it. Formally, given a collection of regulation operators Π^f that satisfy C1-C4, with one operator per flow, an interleaved regulator is a FIFO system that transforms an input sequence (A, L, F) into an output sequence (D, L, F) such that (D^f, L^f) is Π^f-regular for every flow f.

A. Minimal Interleaved Regulator

The following results establishes the existence of a minimal interleaved regulator, i.e. one that delays the packets as little as possible.

Theorem 4: Consider a packet sequence (A, L, F) with, for every flow f, one regulation operator Π^f that satisfies C1-C4. The "minimal interleaved regulator" is defined as the FIFO system that transforms the input packet sequence (A, L, F) into the output packet sequence (D, L, F) defined by $D_1 = A_1$ and

$$D_n = \max\left\{A_n, D_{n-1}, \Pi^{F_n}\left(D^{F_n}, L^{F_n}\right)_{I(n)}\right\} \qquad (49)$$

Recall that, in the above formula, $I(n)$ is the index of packet n in its flow (namely in flow $f = F_n$).

1) The system defined in this way is an interleaved regulator for this packet sequence.
2) (Minimality:) For any other interleaved regulator that transforms (A, L, F) into say (D', L, F) we have $D'_n \geq D_n$ for all $n \in \mathbb{N}^+$.
3) Every flow f in (A, L, F) is Π^f-regular if and only if $D = A$.

The proof is in appendix. Eq.(49) is the input-output characterization of the minimal interleaved regulator. It has an important consequence: it shows that the minimal interleaved regulator can be implemented as a "head of the line" system, as in [15]. More precisely, a possible implementation of the minimal interleaved regulator is as follows.

- Packets of the multi-flow sequence are queued in FIFO order;
- The packet at the head of the queue is examined against the regulation constraints of its flow; it is released at the earliest time where this is possible without violating the constraints;
- Packets that are not at the head of the queue are not examined until they reach the head of the queue.

The Urgency Based Scheduler of Specht and Samii in [15] is an instance of minimal interleaved regulator, which corresponds to the case where the regulation operator Π^f is either of the form $\Pi^{LRQ(r_f)}$ as in Eq.(16) or $\Pi^{LB(r_f,b_f)}$ as in Eq.(30).

Since the minimal interleaved regulator uses a FIFO queue, there is no need for per-flow queuing. Note that there is per-flow state, but on one hand, this per-fow state can be very simple (a single number) for simple regulation rules such as leaky bucket, LRQ, packet spacing or packet burstiness; on the other hand, per-flow state is typically present in switches and routers for packet forwarding and the per-flow state of the interleaved regulator can be placed there. In contrast, implementing one queue per flow as would be required by per-flow regulators has considerably larger complexity.

It follows from the structure of the minimal interleaved regulator that, when the packet at the head of the queue is not eligible for delivery, all packets behind it are delayed. Therefore a packet may be delayed either because it is too early with respect to the regulation imposed to its flow, or because a packet of some other flow at the head of the queue is being delayed.

B. Minimal Interleaved Regulator Does Not Increase Worst-Case Delay

As discussed in the introduction, interleaved regulators are used in networks that handle multiple flows in the same queue, as a means to avoid burstiness cascades. However, they also add some delay, which needs to be accounted for, when computing end-to-end delay bounds. However, they enjoy the same property as their per-flow counterparts, which is also a consequence of their minimality.

More precisely, assume that a packet sequence (A, L, F) is fed into a FIFO system (Figure 3). The following theorem establishes that the minimal interleaved regulator does not add anything to the worst-case delay of the FIFO system.

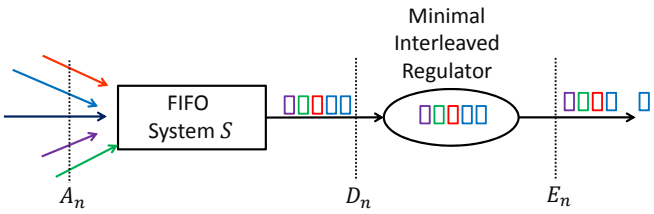

Fig. 3: Configuration for Theorem 5.

Theorem 5: Assume that a packet sequence (A, L, F) is fed into a FIFO system S. Assume that every input flow f is Π^f regular and that the operators Π^f satisfy conditions C1-C4. The output packet sequence (D, L, F) is fed into a minimal interleaved regulator with operator Π^f for flow f; its output is the packet sequence (E, L, F). The worst-case delay of the combination is the same as the worst-case delay of the FIFO system S alone. In other words:

$$\sup_{n\in\mathbb{N}^+}(D_n - A_n) = \sup_{n\in\mathbb{N}^+}(E_n - A_n) \qquad (50)$$

Proof: The proof mimicks the proof of Theorem 3, and consists in comparing the minimal interleaved regulator with a damper with parameter equal to the worst-case delay of system S over all flows. ◇

C. Comparison With Per-Flow Minimal Regulation

Concerning the effect on worst-case delay, we can compare the minimal interleaved regulator with a bank of per-flow minimal regulators, as in Figure 4.

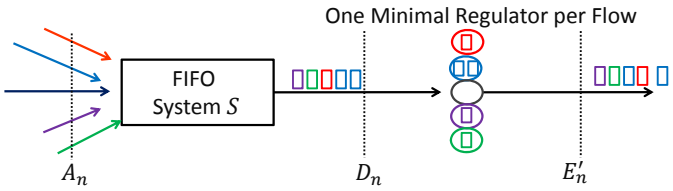

Fig. 4: Configuration for comparing interleaved regulator with a bank of per-flow regulators.

Assume the same setting as in Theorem 5, with the difference that every flow is submitted to its own minimal regulator, with one regulator queue per flow. Let E' be the sequence of output times for the bank of per-flow regulators. The bank of per-flow regulators is not globally FIFO, since the head-of-the-line blocking of the minimal interleaved regulator does not happen here, i.e. the sequence E' is not necessarily wide-sense increasing. For every flow f, let d^f be its worst-case delay at system S. We can apply Theorem 3 to every flow f in this configuration and obtain that

$$\sup_{n\in\mathbb{N}^+} (E'_n - A_n)\,\mathbf{1}_{\{F_n=f\}} = d^f, \quad \forall f \tag{51}$$

Compare with Theorem 5, with which there is a subtle difference. Indeed, Theorem 5 states that the worst-case delay $d = \max_f d^f$ across all flows at system S is not increased by the downstream minimal interleaved regulator. In some cases, the worst-case delay d^f at system S for some flow f may be less than for other flows flows, and thus $d^f < d$ for some flows f. A typical case is when different flows have different maximum packet lengths and when the delay at S includes a transmission delay proportional to packet length. Another case is when different flows require different processing delays. In such cases, the delay bound of Theorem 5 gives

$$\sup_{n\in\mathbb{N}^+} (E_n - A_n)\,\mathbf{1}_{\{F_n=f\}} \le d, \quad \forall f \tag{52}$$

which is weaker than the delay bound in Eq.(51) obtained with per-flow regulators for all flows f such that $d^f < d$. The example in Appendix G shows that it is possible to have equality in Eq.(52) for a flow with $d^f < d$. In other words, interleaved regulation "comes for free" only for the overall worst-case delay across all flows; for the flows that have a worst-case delay less than the overall worst-case, there might be an increase, up to the overall worst-case delay.

We can also easily establish a stronger, per-packet inequality. Consider one flow of interest, say f, and observe only the packets of this flow as they go through the configuration in Figure 3. The output sequence (E^f, L^f) is Π^f-regular and the minimal interleaved regulator is FIFO for this flow (since it is globally FIFO). Therefore, the minimal interleaved regulator is also a per-flow regulator for flow f, though probably not minimal. It follows that $E_n \ge E'_n$ whenever packet n belongs to flow f. Since this holds for every flow f, it follows that

$$E_n \ge E'_n, \quad \forall n \tag{53}$$

In other words, the minimal interleaved regulator cannot beat the per-flow regulator. The example in Appendix G shows that, in general, the above inequality is strict for a non empty set of packets.

D. Application to FIFO Per-Class Networks

This result is quite important for networks that do per-class scheduling as it shows that it is possible to avoid the burstiness cascade while keeping only FIFO queues. Following [15], assume that we place one minimal interleaved regulator per switch input port and per traffic class, before the input to a queuing point, as illustrated in Figure 5. The above theorem can then be applied, where the FIFO system S is the upstream node that feeds the interleaved regulator. Note that in the usptream node, there are other flows that are not fed to the interleaved regulator. Thus, more precisely, the FIFO system S can be defined as the system that transforms the multi-flow packet sequence consisting in all packets of all flows that come from the upstream node.

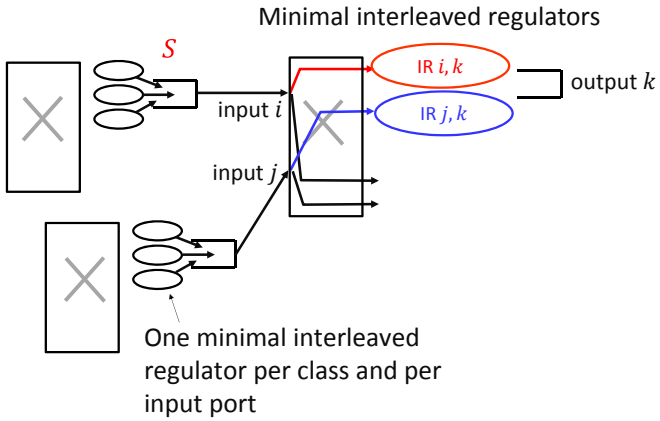

Fig. 5: Use of Interleaved Regulators as proposed by [15]. Minimal interleaved regulators are inserted before the output port schedulers. The figure assumes that there is only one traffic class; at output port k there is one minimal interleaved regulator (such as IR$i,k$ and IT$j,k$) per input port.

When computing end-to-end delay bounds, minimal interleaved regulators can then be ignored, since, by the above theorem, their delay can be absorbed into the previous node delay. This holds for *any* valid delay bounds, because Theorem 5 is about the worst-case delay. Furthermore, since the input flows to any queuing point is the output of an interleaved regulator and thus is regulated, delay and backlog bounds can be computed using network calculus computations; see [27] for a detailed example.

TSN assumes that every source satisfies what we call here the "TSN packet rate regulation" rule in Eq.(36). Therefore, in a TSN network, it would be consistent to use a minimal interleaved regulator where the regulation rule for every flow is "TSN packet rate regulation" (rather than LRQ or leaky bucket, as is currently proposed by UBS). Observe however that the "TSN packet rate regulation" is complex to implement in a regulator, as it requires remembering the dates at which the most recent packets were sent. A simpler alternative is to replace it with the "packet burstiness" constraint (Section III-D), which requires storing only one counter per flow.

VI. Conclusion

Motivated by the Urgency Based Scheduler of [15], we have introduced a new theory of traffic regulators that is able to explain the "reshaping-for free" property of minimal regulators, and can be extended to interleaved regulators, which handle multiple flows in a single FIFO queue. This theory extends the existing, non compatible theories of g-regulators and arrival curve constraints, and also sheds some light on their relationship. It also gives a practical means to avoid burstiness cascades in per-class FIFO networks.

In future research, it might be interesting to explore the service guarantees offered by minimal regulators. Indeed the packetized greedy shaper, which is the minimal regulator for an arrival curve constraints, offers a service guarantee in the form of a service curve (a min-plus concept). Similarly, Chang's minimal g-regulator is a g-server (the max-plus counterpart of a service curve element). Since Pi-regulation subsumes both arrival curve constraints and g-regulation, it will be interesting to find ways of expressing the service guarantees offered by the minimal Pi-regulator and the minimal interleaved regulator.

References

[1] IEEE TSN. https://1.ieee802.org/tsn/, 2018. Accessed: 2018-01-25.

[2] Deterministic networking (detnet), https://datatracker.ietf.org/wg/detnet/about/. Accessed: 2018-09-25.

[3] Anna Charny and Jean-Yves Le Boudec. Delay bounds in a network with aggregate scheduling. In *International Workshop on Quality of Future Internet Services*, pages 1–13. Springer, 2000.

[4] J. C. R. Bennett, Benson K., Charny A., Courtney W. F., and J.-Y. Le Boudec. Delay Jitter Bounds and Packet Scale Rate Guarantee for Expedited Forwarding. *ACM/IEEE Transactions on Networking*, 2002.

[5] Marc Boyer and Christian Fraboul. Tightening end to end delay upper bound for AFDX network calculus with rate latency FIFO servers using network calculus. In *Factory Communication Systems, 2008. WFCS 2008. IEEE International Workshop on*, pages 11–20. IEEE, 2008.

[6] A. Bouillard, L. Jouhet, and E. Thierry. Tight performance bounds in the worst-case analysis of feed-forward networks. In *2010 Proceedings IEEE INFOCOM*, pages 1–9, March 2010.

[7] Anne Bouillard and Giovanni Stea. Exact worst-case delay in FIFO-multiplexing feed-forward networks. *IEEE/ACM Transactions on Networking (TON)*, 23(5):1387–1400, 2015.

[8] Jens B Schmitt, Frank A Zdarsky, and Ivan Martinovic. Improving performance bounds in feed-forward networks by paying multiplexing only once. In *Measuring, Modelling and Evaluation of Computer and Communication Systems (MMB), 2008 14th GI/ITG Conference-*, pages 1–15. VDE, 2008.

[9] Steffen Bondorf and Jens Schmitt. Calculating accurate end-to-end delay bounds-you better know your cross-traffic. In *Proceedings of the 9th EAI International Conference on Performance Evaluation Methodologies and Tools*, pages 17–24. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2016.

[10] Steffen Bondorf, Paul Nikolaus, and Jens B Schmitt. Catching corner cases in network calculus–flow segregation can improve accuracy. In *International Conference on Measurement, Modelling and Evaluation of Computing Systems*, pages 218–233. Springer, 2018.

[11] Steffen Bondorf and Jens B Schmitt. The DiscoDNC v2: a comprehensive tool for deterministic network calculus. In *Proceedings of the 8th International Conference on Performance Evaluation Methodologies and Tools*, pages 44–49. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2014.

[12] Ahlem Mifdaoui and Hamdi Ayed. Wopanets: a tool for worst case performance analysis of embedded networks. In *Computer Aided Modeling, Analysis and Design of Communication Links and Networks (CAMAD), 2010 15th IEEE International Workshop on*, pages 91–95. IEEE, 2010.

[13] Hamid Ayed, Ahlem Mifdaoui, and Christian Fraboul. Hierarchical traffic shaping and frame packing to reduce bandwidth utilization in the AFDX. In *Industrial Embedded Systems (SIES), 2014 9th IEEE International Symposium on*, pages 77–86. IEEE, 2014.

[14] Sven Kerschbaum, Kai-Steffen Hielscher, and Reinhard German. The need for shaping non-time-critical data in profinet networks. In *Industrial Informatics (INDIN), 2016 IEEE 14th International Conference on*, pages 160–165. IEEE, 2016.

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

[16] Ahmed Saeed, Nandita Dukkipati, Vytautas Valancius, Carlo Contavalli, Amin Vahdat, et al. Carousel: Scalable traffic shaping at end hosts. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, pages 404–417. ACM, 2017.

[17] L. Georgiadis, R. Guérin, V. Peris, and R. Rajan. Efficient support of delay and rate guarantees in an internet. In *Proceedings of Sigcomm'96*, pages 106–116, August 1996.

[18] Cheng-Shang Chang and Yih Haur Lin. A general framework for deterministic service guarantees in telecommunication networks with variable length packets. In *Quality of Service, 1998. (IWQoS 98) 1998 Sixth International Workshop on*, pages 49–58, May 1998.

[19] C.S. Chang. *Performance Guarantees in Communication Networks*. Springer-Verlag, New York, 2000.

[20] Jörg Liebeherr. Duality of the max-plus and min-plus network calculus. *Foundations and Trends in Networking*, 11(3-4):139–282, 2017.

[21] Jean-Yves Le Boudec and Patrick Thiran. *Network Calculus*. Springer Verlag Lecture Notes in Computer Science volume 2050 (available online at http://infoscience.epfl.ch/record/282), July 2001.

[22] Jean Dieudonné. *Foundations of Modern Analysis*. Read Books Ltd, 2013.

[23] R.L. Cruz. A calculus for network delay, part i: Network elements in isolation. *IEEE Trans. Inform. Theory, vol 37-1*, pages 114–131, January 1991.

[24] Yuming Jiang. A basic result on the superposition of arrival processes in deterministic networks. In *Proceedings of Globecom*. IEEE, 2018.

[25] Jean-Yves Le Boudec. Some properties of variable length packet shapers. *ACM/IEEE Transactions on Networking*, August 2002.

[26] Dinesh Verma. Guaranteeing delay jitter bounds in packet switching networks. *Proceedings of IEEE TriCom'91, April*, 1991.

[27] Ehsan Mohammadpour, Eleni Stai, Maaz Mohiuddin, and Jean-Yves Le Boudec. End-to-End Latency and Backlog Bounds in Time-Sensitive Networking with Credit Based Shapers and Asynchronous Traffic Shaping. In *ITC 30 Network Calculus Worskhop, Wien, Sept 2018*, 2018.

Appendix

A. Proof of Lemma 1

- $\Leftarrow$ is Eq.(8).
- $\Rightarrow$: We have $t \geq f^{\downarrow}(x)$. If $t > f^{\downarrow}(x)$, the conclusion follows from Eq.(9). Else we have $t = f^{\downarrow}(x)$. For any $s > t$, we have $f(s) \geq x$ by Eq.(9). Therefore $\lim_{s \to t, s > t} f(s) \geq x$. Since $f()$ is right-continuous, $\lim_{s \to t, s > t} f(s) = f(t)$.

B. Proof of Lemma 2

We do the proof for the statement $f^{\downarrow} = (f^{+})^{\downarrow}$. The proof for $f^{\uparrow} = (f^{-})^{\uparrow}$ is exactly similar and is left to the reader.

Note that $f(t) \leq f^+(t)$ for all t therefore $\{t : f(t) \geq x\} \subseteq \{t : f^+(t) \geq x\}$, thus $f^\downarrow(x) \geq (f^+)^\downarrow(x)$.

Assume now, by contradiction, that $f^\downarrow(x) > (f^+)^\downarrow(x)$. For any t such that $(f^+)^\downarrow(x) < t < f^\downarrow(x)$, we have:

1) $t < f^\downarrow(x)$ and since $f^\downarrow(x)$ is the infimum of the set $\{t : f(t) \geq x\}$, it follows that t is not in this set, i.e. $f(t) < x$
2) $t > (f^+)^\downarrow(x)$ and by Eq.(9) it follows that $f^+(t) \geq x$

Thus, for any $t \in \left((f^+)^\downarrow(x),\ f^\downarrow(x)\right)$ we have $f^+(t) > f(t)$ i.e. t is a point of discontinuity of f. But this is impossible because this interval is not a countable set. Thus it is not possible that $f^\downarrow(x) > (f^+)^\downarrow(x)$, which proves that $f^\downarrow(x) = (f^+)^\downarrow(x)$.

C. Proof of Lemma 3

We do the proof for the statement $(f^\uparrow)^- = f^\downarrow$. The proof for $(f^\downarrow)^+ = f^\uparrow$ is exactly similar and is left to the reader.

We can rewrite the definitions in Eq.(5) and Eq.(6) as

$$f^\downarrow(y) = \sup_{s\in\mathbb{R}^+} \left(s\mathbf{1}_{\{f(s)>y\}}\right) \quad (54)$$

$$f^\uparrow(y) = \sup_{s\in\mathbb{R}^+} \left(s\mathbf{1}_{\{f(s)\leq y\}}\right) \quad (55)$$

Therefore, using associativity of sup:

$$(f^\uparrow)^-(x) \stackrel{\text{def}}{=} \sup_{0\leq y<x} f^\uparrow(y) \quad (56)$$

$$= \sup_{0\leq y<x} \left(\sup_{s\in\mathbb{R}^+} \left(s\mathbf{1}_{\{f(s)\leq y\}}\right)\right) \quad (57)$$

$$= \sup_{s\in\mathbb{R}^+} \left(\sup_{0\leq y<x} \left(s\mathbf{1}_{\{f(s)\leq y\}}\right)\right) \quad (58)$$

$$= \sup_{s\in\mathbb{R}^+} \left(s\ \varphi(s,x)\right) \quad (59)$$

with $\varphi(s,x) \stackrel{\text{def}}{=} \sup_{0\leq y<x} \left(\mathbf{1}_{\{f(s)\leq y\}}\right)$. Now if $x > f(s)$ then $\varphi(s,x) = 1$ and if $x \leq f(s)$ then $\varphi(s,x) = 0$. Therefore $\varphi(s,x) = \mathbf{1}_{\{x>f(s)\}}$. Thus

$$(f^\uparrow)^-(x) = \sup_{s\in\mathbb{R}^+} \left(s\ \mathbf{1}_{\{x>f(s)\}}\right) = f^\downarrow(x) \quad (60)$$

where the last equality is by Eq.(54).

D. Proof of Theorem 1

• 1)⇒ 3):

Consider some packet numbers $1 \leq m \leq n$. If $m = n$ then 3) trivially holds because $\sigma^\downarrow(0) = 0$. Assume now that $m < n$. Let $T()$ be the arrival time function defined by Eq.(19). Take $y = L_1 + ... + L_n - \epsilon$ with $0 < \epsilon \geq L_n$ and $x = L_1 + ... + L_m$. Because L_n is integer, we have $T(y) = A_n$ and $T(x) = A_m$.

By [20], the max-plus traffic envelope condition Eq.(20) also holds. Thus

$$A_n - A_m = T(y) - T(x) \geq \sigma^\uparrow(y - x) = \sigma^\uparrow(L_m + ... + L_n - \epsilon) \quad (61)$$

Take the limit of the above equation as $\epsilon \to 0$ and obtain

$$A_n - A_m \geq (\sigma^\uparrow)^-((L_m + ... + L_n) \quad (62)$$

By Lemma 3, $(\sigma^\uparrow)^- = \sigma^\downarrow$, which concludes this part of the proof.

• 3)⇒ 2):

Consider some packet numbers $1 \leq m \leq n$. If $m = n$ then 2) trivially holds because $\sigma(0) \geq 0$ and $\sigma^+(0) \geq 0$. Assume now that $m < n$. Eq.(21) follows from Lemma 1 applied to $f() = \sigma^+()$.

• 2)⇒ 1):

Part 1: no simultaneous arrivals.

We first prove this case assuming that there cannot be simultaneous arrivals, namely we assume $A_n < A_{n+1}$ for all $n \in \mathbb{N}^+$.

Consider $s, t \in \mathbb{R}^+$ with $0 \leq s \leq t$. If $s = t$ then Eq.(18) trivially holds. Assume therefore that $0 \leq s < t$. Let $\mathcal{A} = \{A_1, A_2, ...\}$. We consider several cases:

Case 1: $\mathcal{A} \cap [s,t]$ is empty. In this case. $R(t) - R(s) = 0$ and Eq.(18) is trivially satisfied.

Case 2: $\mathcal{A} \cap [s,t]$ is nonempty, $s \notin \mathcal{A}$ and $t \notin \mathcal{A}$. Let m be the smallest packet number such that $s < A_m$ and let n be the largest packet number such that $A_n < t$, so that $\mathcal{A} \cap [s, A_m)$ and $\mathcal{A} \cap (A_n, t]$ are empty. Therefore

$$R(t) - R(s) = \sum_{j=m}^{n} L_j \quad (63)$$

and by Eq.(21)

$$R(t) - R(s) \leq \sigma^+(A_n - A_m) \quad (64)$$

We must also have

$$s < A_m \leq A_n < t \quad (65)$$

and thus $A_n - A_m < t - s$; by Eq.(11)

$$\sigma^+(A_n - A_m) \leq \sigma(t - s) \quad (66)$$

This concludes the proof in this case.

Case 3: $s \in \mathcal{A}$ and $t \notin \mathcal{A}$. Thus $s = A_m$ for some m. Let n be the largest packet number such that $A_n < t$. We have therefore

$$R(t) - R(s) = \sum_{j=m}^{n} L_j \quad (67)$$

and

$$s = A_m \leq A_n < t \quad (68)$$

The rest of the proof in this case is as in Case 2.

Case 4: $s \notin \mathcal{A}$ and $t \in \mathcal{A}$. Thus $t = A_n$ for some n. Let m be the smallest packet number such that $s < A_m$. We have therefore

$$R(t) - R(s) = \sum_{j=m}^{n-1} L_j \quad (69)$$

and

$$s < A_m \leq A_n = t \tag{70}$$

Thus

$$R(t) - R(s) \leq \sum_{j=m}^{n} L_j \leq \sigma^+(A_n - A_m) \tag{71}$$

where the last inequality is by Eq.(21). Now

$$A_n - A_m < t - s \tag{72}$$

$$R(t) - R(s) \leq \sigma(A_n - A_m) \tag{73}$$

Now $A_n - A_m \leq t - s$ and $\sigma()$ is wide-sense increasing, thus $\sigma(A_n - A_m) \leq \sigma(t - s)$, which concludes the proof in this case.

Case 5: $s \in \mathcal{A}$ and $t \in \mathcal{A}$. Thus $s = A_m$ and $t = A_n$ for some $m \leq n$. We have therefore

$$R(t) - R(s) = \sum_{j=m}^{n-1} L_j \tag{74}$$

If $m = n$ then $R(t) - R(s) = 0$ and Eq.(18) is trivially verified. We can therefore assume $m < n$. It follows that $0 \leq A_{n-1} - A_m < A_n - A_m$ therefore

$$\sigma^+(A_{n-1} - A_m) \leq \sigma(A_n - A_m) \tag{75}$$

By Eq.(21)

$$R(t) - R(s) \leq \sigma^+(A_{n-1} - A_m) \leq \sigma(A_n - A_m) \tag{76}$$

Part 2: with simultaneous arrivals.

We now allow simultaneous arrivals in the flow (A, L). We assume that there is a finite number of packet arrivals in every bounded interval. Indeed, if this does not hold, the conditions in the theorem are false and the equivalence holds.

We derive from (A, L) another packet sequence, (A', L') obtained by aggregating all packets that arrive at the same time under (A, L). Formally, (A', L') is defined by:

$$\begin{aligned} A'_1 &= \min\{A_m, m \in \mathbb{N}^+\} \\ A'_n &= \min\{A_m, m \in \mathbb{N}^+, A_m > A'_{n-1}\} \\ L'_n &= \textstyle\sum_{j \in \mathbb{N}^+} L_j \mathbf{1}_{\{A_j = A'_n\}} \end{aligned}$$

Note that L'_n is finite for every n by our assumption and L'_n is the sum of all packet sizes of all packets that arrive at the same instant. Note that $A' \in \mathcal{F}_{inc}$ and there are no simultaneous arrivals in the flow (A', L').

We next show that, for $i = 1, 2$, condition i of the theorem holds for (A, L) if and only if it holds for (A', L'), which will conclude the proof.

Condition 1): The cumulative arrival function $R()$ is the same for (A, L) and (A', L') so Condition 1 of the theorem holds for (A, L) if and only if it holds for (A', L').

Condition 2): Assume first that Condition 2) holds for (A, L). Consider some fixed $m', n' \in \mathbb{N}^+$ and let m be the index of the first packet such that $A_m = A'_{m'}$ and n the index of the last packet such that $A_n = A'_{n'}$. We have

$$\sum_{j=m'}^{n'} L'_j = \sum_{i=m}^{n} L_i \tag{77}$$

and $A_n - A_m = A'_{n'} - A'_{m'}$. Since Eq.(21) holds for (A, L), it follows that it also holds for (A', L').

Conversely, assume that Condition 2) holds for (A', L') and consider some fixed $m, n \in \mathbb{N}^+$. Let Define and m', n' by $A'_{m'} = A_m$ and $A'_{n'} = A_n$. We have

$$\sum_{j=m}^{n} L_j \leq \sum_{i=m'}^{n'} L'_i \tag{78}$$

and $A_n - A_m = A'_{n'} - A'_{m'}$. Since Eq.(21) holds for (A', L'), it follows that it also holds for (A, L).

This concludes the proof in this case.

E. Proof of Theorem 2

1) We first prove that the system defined by $D_1 = A_1$ and Eq.(44) is a Π-regulator. We obviously have $D_n \geq D_{n-1}$ i.e. $D \in \mathcal{F}_{inc}$ and $D_n \geq A_n$ for all $n \in \mathbb{N}^+$ thus this is a FIFO system. Also $D_n \geq \Pi(D, L)_n$ by construction.

2) Next, we show by induction that $D'_n \geq D_n$.

Base Step: We have $D_1 = A_1 \leq D'_1$ because the Π-regulator is a FIFO system.

Induction Step: Assume $D'_m \geq D_m$ for $1 \leq m \leq n-1$. Let $\bar{D}$ be the sequence defined by $\bar{D}_m = D'_m$ for $1 \leq m \leq n-1$ and $\bar{D}_m = -\infty$ for $m \geq n$. We have $D' \geq \bar{D}$ by induction hypothesis and by Condition C4, $\Pi(D', L) \geq \Pi(\bar{D}, L)$. By Condition C2, $\Pi(\bar{D}, L)_n = \Pi(D', L)_n$. Therefore $\Pi(D', L)_n \geq \Pi(D, L)_n$. But since D' is Π-regular, we also have $D'_n \geq \Pi(D', L)_n$. Thus

$$D'_n \geq \Pi(D, L)_n \tag{79}$$

Now

$$D'_n \geq D'_{n-1} \geq D_{n-1} \text{ and } D'_n \geq A_n \tag{80}$$

because the Π-regulator is a FIFO system. Combining the last two inequalities gives

$$D'_n \geq \max\{A_n, D_{n-1}, \Pi(D, L)_n\} = D_n \tag{81}$$

3) If $D = A$ then since D is Π-regular by 1) obviously A is Π-regular.

Conversely, if (A, L) is Π-regular then the identity system, which maps (A, L) into itself, is a Π-regulator for this flow. By item 2), we have $D \leq A$. But since $D \geq A$ by construction it follows that $D = A$.

F. Proof of Theorem 4

1) We first prove that the system defined by $D_1 = A_1$ and Eq.(49) is an interleaved regulator. We obviously have $D_n \geq D_{n-1}$ i.e. $D \in \mathcal{F}_{inc}$ and $D_n \geq A_n$ for all $n \in \mathbb{N}^+$ thus this is a FIFO system. Also by construction

$$D_n \geq \Pi^{F_n}\left(D^{F_n}, L^{F_n}\right)_{I(n)} \tag{82}$$

which is the same as

$$D^{F_n}_{I(n)} \geq \Pi^{F_n}\left(D^{F_n}, L^{F_n}\right)_{I(n)} \tag{83}$$

which shows that $D^f \geq \Pi^f\left(D^f, L^f\right)$ for every flow f, i.e. every flow at the output is Π^f-regular.

2) Next, we show by induction that $D'_n \geq D_n$.

Base Step: We have $D_1 = A_1 \leq D'_1$ because the interleaved regulator is a FIFO system.

Induction Step: Assume $D'_m \geq D_m$ for $1 \leq m \leq n-1$. Let $\bar{D}$ be the sequence defined by $\bar{D}_m = D'_m$ for $1 \leq m \leq n-1$ and $\bar{D}_m = -\infty$ for $m \geq n$. We have $D' \geq \bar{D}$ by induction hypothesis and thus $D'^{F_n} \geq \bar{D}^{F_n}$. By Condition C4, $\Pi^{F_n}(D'^{F_n}, L^{F_n}) \geq \Pi^{F_n}(\bar{D}^{F_n}, L^{F_n})$. By Condition C2, $\Pi^{F_n}(\bar{D}^{F_n}, L^{F_n})_{I(n)} = \Pi^{F_n}(D^{F_n}, L^{F_n})_{I(n)}$. Therefore $\Pi^{F_n}(D'^{F_n}, L^{F_n})_{I(n)} \geq \Pi^{F_n}(D^{F_n}, L^{F_n})_{I(n)}$. But since D' is Π^{F_n}-regular, we also have $D'_n = (D'^{F_n})_{I(n)} \geq \Pi^{F_n}(D'^{F_n}, L^{F_n})_{I(n)}$. Thus

$$D'_n \geq \Pi^{F_n}(D^{F_n}, L^{F_n})_{I(n)} \tag{84}$$

Now

$$D'_n \geq D'_{n-1} \geq D_{n-1} \text{ and } D'_n \geq A_n \tag{85}$$

because the interleaved regulator is a FIFO system. Combining the last two inequalities gives

$$D'_n \geq \max\left\{A_n, D_{n-1}, \Pi^{F_n}(D^{F_n}, L^{F_n})_{I(n)}\right\} = D_n \tag{86}$$

3) Since the system is an interleaved regulator by item 1, every flow f in the output sequence is Π^f regular. Thus, if $D = A$, the input is equal to the output and every flow f in the input sequence is also Π^f regular.

Conversely, if every flow in (A, L, F) is Π^f-regular then the identity system, which maps (A, L, F) into itself, is an interleaved regulator for this packet sequence. By item 2), we have $D \leq A$. But since $D \geq A$ by construction it follows that $D = A$.

G. A Numerical Example

We choose per unit values where 1 data unit $= 1200$ bytes and 1 time unit $= 12\mu$sec. Consider a scenario with 2 flows. For flow 1, all packets have length equal to 2 data units, and for flow 2 it is 1 data unit. Flow 1 is subject to a packet spacing regulation with $\tau_1 = 5$ time units and flow 2 to a packet spacing regulation with $\tau_2 = 10$ time units.

We use the notation of Figures 3 and 4. The input packet sequence to the FIFO systems S is (A, L, F) with

$$\begin{aligned} A &= (0, 5, 5, 10, 15, 15, 20, 25, 25, ...) \\ L &= (2, 2, 1, 2, 2, 1, 2, 2, 1, ...) \\ F &= (1, 1, 2, 1, 1, 2, 1, 1, 2, ...) \end{aligned}$$

In other words, flow 1 sends one packet every 5 time units and flow 2 sends one packet every 10 time units, which arrives immediately after an even-numbered packets of flow 1. At the input, both flows are conforming to their regulation constraints.

The output of the FIFO system S is (D, L, F) with

$$D = (5, 7, 8, 15, 17, 18, 25, 27, 28, ...)$$

i.e., the odd packets of flow 1 have a response time of 5 time units, the even packets of 2 time units, and the packets of flow 2 have a response time of 3 time units. This corresponds to the case where S is a simple priority queue, where flows 1 and 2 are served with low priority at a rate equal to 1 p.u. (i.e. 800 Mb/s) and where a high priority packet preempts the server during time intervals $[0; 3], [10; 13], ...$ The worst-case delay at S for flow 1 is $d^1 = 5$ time units and for flow 2 it is $d^2 = 3$ time units. The overall worst-case delay at S is $d = 5$ time units.

The output of the minimal interleaved regulator is (E, L, F) with

$$E = (5, 10, 10, 15, 20, 20, 25, 30, 30, ...)$$

Indeed the odd packets of flow 1 are not delayed by the minimal interleaved regulator, but the even packets are delayed because they arrive too early with respect to a spacing constraint of 5 time units. Packets of flow 2 are delayed because they stand behind the even packets of flow 1. For flow 1, the worst-case delay at the combination of S and the interleaved regulator is $d^1_{tot} = 5$ time units and for flow 2 it is $d^2_{tot} = 5$ time units as well. The overall worst-case delay is $d_{tot} = d = 5$ time units. The overall worst-case delay is not increased by the minimal interleaved regulator, but the worst-case delay of flow 2 *is* increased.

The output of the bank of per-flow regulators is (E', L, F) with

$$E' = (5, 10, 8, 15, 20, 18, 25, 30, 28, ...)$$

The difference with the interleaved regulator is that packets of flow 2 are not delayed (as a result, the bank of per-flow regulators is not globally FIFO). For packets of flow 2 we have $E'_n < E_n$.