Jitter Regulation for Multiple Streams
(Extended Abstract)

David Hay and Gabriel Scalosub

Computer Science Department, Technion
Technion City, Haifa 32000, Israel
{hdavid,gabriels}@cs.technion.ac.il

Abstract. For widely-used interactive communication, it is essential that traffic is kept as smooth as possible; the smoothness of a traffic is typically captured by its *delay jitter*, i.e., the difference between the maximal and minimal end-to-end delays. The task of minimizing the jitter is done by jitter regulators that use a limited-size buffer in order to shape the traffic. In many real-life situations regulators must handle multiple streams simultaneously and provide low jitter on each of them separately. This paper investigates the problem of minimizing jitter in such an environment, using a fixed-size buffer.
We show that the offline version of the problem can be solved in polynomial time, by introducing an efficient offline algorithm that finds a release schedule with optimal jitter. When regulating M streams in the online setting, we take a *competitive analysis* point of view and note that previous results in [1] can be extended to an online algorithm that uses a buffer of size $2MB$ and obtains the optimal jitter possible with a buffer of size B. The question arises whether such a resource augmentation is essential. We answer this question in the affirmative, by proving a lower bound that is tight up to a factor of 2, thus showing that jitter regulation does not scale well as the number of streams increases unless the buffer is sized-up proportionally.

1 Introduction

Contemporary network applications call for connections with stringent Quality-of-Service (QoS) demands. This gives rise to QoS networks that are able to provide guarantees on various parameters, such as the end-to-end delay, loss ratio, bandwidth, and jitter. The need for efficient mechanisms to provide smooth and continuous traffic is mostly motivated by the increasing popularity of interactive communication and in particular video/audio streaming. The smoothness of the traffic is captured by the notion of *delay jitter* (or *Cell Delay Variation* [2]); namely, the difference between the maximal and minimal end-to-end delays of different fixed-size packets, henceforth referred to as *cells*.

Controlling traffic distortions within the network, and in particular jitter control, has the effect of moderating the traffic throughout the network [3]. This is important when a service provider in a QoS network must meet service level agreements (SLA) with its customers. In such cases, moderating high congestion states in switches along the network results in the provider's ability to satisfy the guarantees to all its customers [4].

Jitter control mechanisms have been extensively studied in recent years (see a survey in [3]). These are usually modelled as *jitter regulators* [1, 5, 6] that use internal buffers in order to shape the traffic, so that cells leave the regulator in the most periodic manner possible. Generally, such regulators calculate a hypothetical periodic schedule, and try to release cells accordingly. Upon arrival, cells are stored in the buffer until their planned release time, or until a buffer overflow occurs. This indicates a tradeoff between the buffer size and the best attainable jitter, i.e., as buffer space increases, one can expect to obtain a lower jitter.

This paper investigates the problem of finding an optimal jitter release schedule, given a predetermined buffer size. This problem was first raised by Mansour and Patt-Shamir [1], who considered only a single-stream setting. However, in practice jitter regulators handle multiple streams simultaneously and must provide low jitter for each stream separately and independently.

In the *multi-stream* model, the traffic arriving at the regulator is an interleaving of M streams originating from M independent abstract sources (see Figure 1). Each abstract source i sends a stream of fixed-size cells in a fully periodic manner, with inter-release time X^i, which arrive at a jitter regulator after traversing the network. Variable end-to-end delay caused by transient congestion throughout the network may result in such a stream arriving at the regulator in a non-periodic fashion. The regulator knows the value of X^i, and strives to release consecutive cells X^i time units apart, thus re-shaping the traffic into its original form. Furthermore, the order in which cells are released by each abstract source is assumed to be respected throughout the network. This implies that the cells from the same stream arrive at the regulator in order (but not necessarily equally spaced), and the regulator should also maintain this order. We refer to this property as the *FIFO constraint*.

Note that the FIFO constraint should be respected in each stream independently, but not necessarily on all incoming traffic. This implies that in the multi-stream model, the order in which cells are released is not known *a priori*. This lack of knowledge is an inherent difference from the case where there is only one abstract source, and it poses a major difficulty in devising algorithms for multi-stream jitter regulation (as we describe in detail in Section 4).

Our Results

This paper presents algorithms and tight lower bounds for jitter regulation in this multiple-streams environment, both in offline and online settings. This answers a primary question posed in [1].

We evaluate the performance of a regulator in the multi-stream environment by considering the maximum jitter obtained on any stream. We show that surprisingly, the offline problem can be solved in polynomial time. This is done by

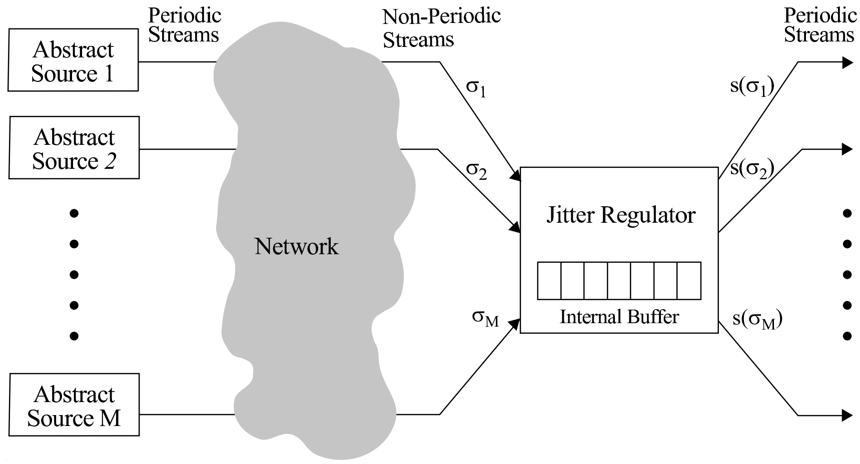


Fig. 1. The multi-stream jitter regulation model

characterizing a collection of optimal schedules, and showing that their properties can be used to devise an offline algorithm that efficiently finds a release schedule that attains the optimal jitter.

We use a *competitive analysis* [7, 8] approach in order to examine the online problem. In this setting, by sizing up the buffer to a size of $2MB$ and statically partitioning the buffer equally among the M streams, applying the algorithm described in [1, Algorithm B] on each stream separately yields an algorithm that obtains the optimal max-jitter possible with a buffer of size B. We show that such a resource augmentation cannot be avoided, by proving that any online algorithm needs a buffer of size at least $M(B-1)+B+1 = \Omega(MB)$ in order to obtain the optimal jitter possible with a buffer of size B. We further show that these tight results also apply when the objective is to minimize the *average* jitter attained by the M streams. These results indicate that online jitter regulation does not scale well as the number of streams increases unless the buffer is sized up proportionally.

Previous Work

Mansour and Patt-Shamir [1] consider a simplified *single-stream* model in which there is only a single abstract source. They present an efficient offline algorithm, which computes an optimal release schedule in these settings. They further devise an online algorithm, which uses a buffer of size $2B$, and produces a release schedule with the optimal jitter attainable with buffer of size B, and then show a matching lower bound on the amount of resource augmentation needed, proving that their online algorithm is optimal in this sense.

This model is later discussed by Koga [9] that deals with jitter regulation of a single stream with delay consideration. An optimal offline algorithm, and a nearly optimal online algorithm are presented for the case where a cell cannot be stored in the buffer for more than a predetermined amount of time.

2 Model Description, Notation, and Terminology

We adopt the following definitions from [1]:

Definition 1. *Given a sequence of cells $\sigma = (p_i^\sigma)_{i=0}^n$ and a non-decreasing arrival function $a : \sigma \to \mathbb{R}^+$ such that cell p_i^σ arrives at time $a(p_i^\sigma)$:*

1. *A* release schedule *for σ is a function $s : \sigma \to \mathbb{R}^+$ satisfying for every $p_i^\sigma \in \sigma$, $a(p_i^\sigma) \leq s(p_i^\sigma)$.*
2. *A release schedule s for σ is B-*feasible *if at any time t,*

$$|\{p_i^\sigma \in \sigma | a(p_i^\sigma) \leq t < s(p_i^\sigma)\}| \leq B.$$

   That is, there are never more than B cells in the buffer simultaneously.

3. *The* delay jitter *of σ under a release schedule s is*

$$J^\sigma(s) = \max_{0 \leq i,k \leq n} \{s(p_i^\sigma) - s(p_k^\sigma) - (i-k)X\}$$

   where X is the inter-release time of σ (i.e., X is the difference between the release times of any two consecutive cells from the abstract source).[1]

We first extend Definition 1 to an arrival sequence σ that is an interleaving of M streams $\sigma_1, \ldots, \sigma_M$. We denote by X^{σ_i} the inter-release time of stream σ_i, and assume for simplicity that all streams have the same inter-release time X; all our results extend immediately to the case where this does not hold. Let p_j^σ denote the j'th cell (in order of arrival) of the interleaving of the streams σ, and let $p_j^{\sigma_i}$ denote the j'th cell of the single stream σ_i. A release schedule should obey a per-stream FIFO discipline, in which cells of the same stream are released in the order of their arrival.

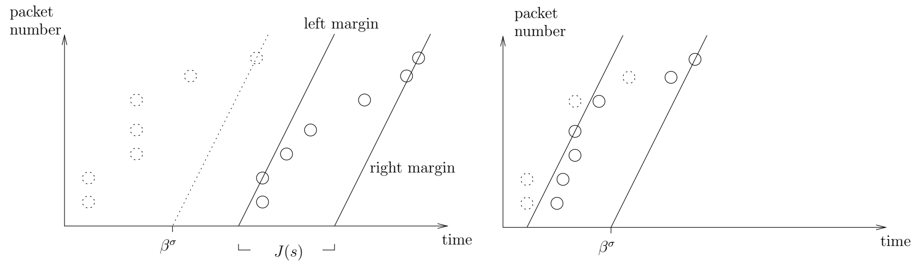
Let $J^{\sigma_i}(s)$ be the jitter of a single stream σ_i obtained by a release schedule s. We use the following metric to evaluate multi-stream release schedhules:

Definition 2. *The* max-jitter *of a multi-stream sequence $\sigma = \{\sigma_1, \ldots, \sigma_M\}$ obtained by a release schedule s is the maximal jitter obtained by any of the streams composing the sequence; that is,* $\mathrm{MJ}^\sigma(s) = \max_{1 \leq k \leq M} J^{\sigma_k}(s)$.

2.1 Geometric Intuition

One can take a geometric view of delay jitter by considering a two dimensional plane where the x-axis denotes time and the y-axis denotes the cell number. We first consider the case of a single stream σ. Given a release schedule s, a point at coordinates (t, j) is marked if $s(p_j^\sigma) = t$ (see Figure 2(a)). The *release band* is the band with slope $1/X$ that encloses all the marked points and has minimal width. The jitter obtained by s is the width of its release band, and therefore our objective is to find a schedule with the narrowest release band.

[1] Since the abstract source generates perfectly periodic traffic, this definition of delay jitter coincides with the notion of Cell Delay Variation.



(a) non-aligned schedule (b) aligned schedule

Fig. 2. Outline of arrivals (dotted circles) and marked releases (full circles).

Under the multi-stream model, we associate every stream σ_i with a different color i. A point at coordinates (t, j) is colored with color i if $s(p_j^{\sigma_i}) = t$. Any schedule s induces a separate release band for each stream σ_i in σ that encloses all points with color i. Schedule s is therefore characterized by M release bands.

3 Online Multi-Stream Max-Jitter Control

As mentioned previously, there exists an online algorithm with buffer size $2MB$, which obtains the optimal max-jitter possible with a buffer of size B. In this section we show that this result is tight up to a factor of 2, by showing that in order to obtain the optimal max-jitter possible with a buffer of size B, any online algorithm needs a buffer of size at least $M(B-1)+B+1$. Hence, in order to maintain the same jitter performance, it is necessary to increase the buffer size in a linear proportion to the number of streams.

Theorem 1. *For every online algorithm* ALG *with an internal buffer of size* $< M(B-1)+B+1$*, there exists an arrival sequence consisting of* M *streams, such that* ALG *attains max-jitter strictly greater than the optimal jitter possible with a buffer of size* B*.*

Proof. Let ALG be an online algorithm with a buffer of size at most $M(B-1)+B+1$. Consider the following arrival sequence σ: For every $0 \le i \le B-1$, M cells arrive at the regulator at time $i \cdot X$, one for every stream. The sequence stops if ALG releases a cell before time $t' = (B+1)X$.

If ALG releases a cell before time t', say of stream σ_i, consider the following continuation for σ: In time $T > t'$ which can be arbitrarily large, one cell of stream σ_i arrives at the regulator.

Since ALG releases the first cell of stream σ_i before time t', and the last cell of stream σ_i cannot be sent prior to time T, then $j^{\sigma_i}(\text{ALG}) \ge T - t' - (B+1)X \ge T - (B+1)X - (B+1)X = T$, which can be arbitrarily large. It follows that $\text{MJ}^\sigma(\text{ALG})$ is strictly greater than zero. On the other hand, note that for any choice of T, the optimal max-jitter possible with a buffer of size B is zero: Every cell of a stream other than σ_i is released immediately upon its arrival, and for every $0 \leq j \leq B$, cell $p_j^{\sigma_i}$ is released in time $T - (B - j)X$. Since every stream other than σ_i does not consume any buffer space, it is easy to verify that at every time t, there are at most B cells in the buffer. Clearly, every stream obtains a zero jitter by this release schedule.

Assume now that ALG does not release any cells before time t', implying that in time t' there are MB cells in the buffer. Consider the following continuation for σ: In time t', $B + 1$ cells of stream σ_1 arrive at the regulator.

Since ALG has a buffer of size at most $M(B-1) + B + 1 = (M+1)B - M$, it must release at least $M + 1$ cells in time t'. By the pigeonhole principle it follows that two of the released cells correspond to the same stream. This stream attains a jitter of at least $t' - t' - (0 - 1)X = X$, and therefore $\mathrm{MJ}^{\sigma}(\mathrm{ALG})$ is strictly greater than zero. On the other hand, the optimal max-jitter possible with a buffer of size B is zero: Every cell of a stream other than σ_1 is released immediately upon its arrival, and for every $0 \leq j \leq B$, cell $p_j^{\sigma_1}$ is released in time $t' - (B - j)X$. Similarly to the previous case, every stream obtains a zero jitter by this release schedule, and no more than B cells are stored simultaneously in the buffer. □

Note that this lower bound for the case $M = 1$ exactly coincides with the result of the single stream model [1]. Theorem 1 further implies that in case the buffer size is $< M(B-1) + B + 1$, there are scenarios in which an optimal schedule attains zero jitter for all streams, while any online algorithm produces a schedule where at least one stream has a strictly positive jitter. This fact immediately implies that even if the objective is to minimize the average jitter obtained by the different streams, the same lower bound holds. Since the online algorithm, which statically partitions the buffer, minimizes the jitter of each stream independently, it clearly minimizes the overall average jitter as well, thus providing a matching upper bound.

4 An Efficient Offline Algorithm

This section presents an efficient offline algorithm that generates a release schedule with optimal max-jitter.

Given a sequence σ that is an interleaving of M streams, consider a total order $\pi = (p'_0, \ldots, p'_n)$ on the release schedule of cells in σ that respects the FIFO order in each stream separately. The release schedule, which attains the optimal max-jitter and respects π, can be found using similar arguments to the ones in [1, Algorithm A]: Cell p'_j can be stored in the buffer only until cell p'_{j+B} arrives, imposing strict bounds on the release time of each cell. In particular, it follows that for every sequence σ, there exists an optimal release schedule. Unfortunately, it is computationally inefficient to enumerate over all possible total orders, hence a more sophisticated approach should be considered.

We first discuss properties of schedules that achieve optimal max-jitter. We then show that these properties allow to find an optimal schedule in polynomial time, based solely on the cells' arrival times, and the parameters X and B.

For every cell p_j^σ, one can intuitively consider $t = a(p_j^\sigma) - jX$ as the time at which p_0^σ should be sent, so that p_j^σ is sent immediately upon its arrival, in a perfectly periodic release schedule. For any stream σ, denote by $\beta^\sigma = \max_j \left\{a(p_j^\sigma) - jX\right\}$. From a geometric point of view, β^σ is a lower bound on the intersection between the time axis and the right margin of any release band (see Figure 2(a)), since otherwise the cell defining β^{σ_i} would have to be released prior to its arrival.

Given a release schedule s for a sequence σ, a stream $\sigma_i \subseteq \sigma$ is said to be *aligned* in s if there is no cell $p_k^{\sigma_i} \in \sigma_i$ such that $s(p_k^{\sigma_i}) > \beta^{\sigma_i} + kX$. Clearly, if σ_i is aligned in s, then the cell $p_j^{\sigma_i}$ that defines β^{σ_i} satisfies $s(p_j^{\sigma_i}) = a(p_j^{\sigma_i})$. Geometrically, the right margin of a release band corresponding to an aligned stream σ_i intersects the time axis in point $(\beta^{\sigma_i}, 0)$ (see Figure 2(b)).

A release schedule s for max-jitter is said to be *aligned*, if every stream is aligned in s. The following simple lemma shows that one can iteratively align the streams of an optimal schedule without increasing the overall jitter:

Lemma 1. *For every sequence σ, there exists an optimal aligned schedule s.*

Proof. Given an optimal schedule s' for sequence σ with *at least* ℓ aligned streams, we prove that s' can be changed into an aligned schedule (i.e. with M aligned streams), maintaining its optimality.

We first show that s' can be altered into an optimal schedule with $\ell + 1$ aligned streams. Let σ_i be one of the non-aligned streams in s', and consider the following schedule $\overline{s}$:

$$\overline{s}(p_j^{\sigma_k}) = \begin{cases} \min\left\{s'(p_j^{\sigma_k}), \beta^{\sigma_k} + jX\right\} & k = i \\ s'(p_j^{\sigma_k}) & k \neq i \end{cases}$$

Clearly for every stream other than σ_i, the schedule remains unchanged, therefore it suffices to consider only stream σ_i. Since $s'(p_j^{\sigma_i}) \geq a(p_j^{\sigma_i})$ and $\beta^{\sigma_i} + jX \geq a(p_j^{\sigma_i})$, $\overline{s}$ is a release schedule and it can easily be verifited that $\overline{s}$ satisfies the FIFO constraint. Schedule $\overline{s}$ is B-feasible, since s' is B-feasible and for any cell $p_j^{\sigma_i}$, $\overline{s}(p_j^{\sigma_i}) \leq s'(p_j^{\sigma_i})$. Stream σ_i is aligned in $\overline{s}$, since clearly every cell $p_j^{\sigma_i}$ satisfies $\overline{s}(p_j^{\sigma_i}) \leq \beta^{\sigma_i} + jX$. Hence, $\overline{s}$ has $\ell + 1$ aligned stream.

In order to prove that $\overline{s}$ is optimal, it suffices to show that $\overline{s}(p_j^{\sigma_i}) - \overline{s}(p_m^{\sigma_i}) - (j-m)X \leq J^{\sigma_i}(s')$ for every two cells $p_j^{\sigma_i}, p_m^{\sigma_i} \in \sigma_i$. First note that $\overline{s}(p_j^{\sigma_i}) - \overline{s}(p_m^{\sigma_i}) - (j-m)X \leq s'(p_j^{\sigma_i}) - \overline{s}(p_m^{\sigma_i}) - (j-m)X$, since $\overline{s}(p_j^{\sigma_i}) \leq s'(p_j^{\sigma_i})$. If $\overline{s}(p_m^{\sigma_i}) = s'(p_m^{\sigma_i})$ then trivially $\overline{s}(p_j^{\sigma_i}) - \overline{s}(p_m^{\sigma_i}) - (j-m)X \leq J^{\sigma_i}(s')$. Otherwise, $\overline{s}(p_m^{\sigma_i}) = \beta^{\sigma_i} + mX = a(p_b^{\sigma_i}) - bX + mX$ for the cell $p_b^{\sigma_i}$ that defines β^{σ_i}. Since s' is a release schedule, then $s(p_b^{\sigma_i}) \geq a(p_b^{\sigma_i})$, which yields $\overline{s}(p_j^{\sigma_i}) - \overline{s}(p_m^{\sigma_i}) - (j-m)X \leq s'(p_j^{\sigma_i}) - s'(p_b^{\sigma_i}) - (j-b)X \leq J^{\sigma_i}(s')$.

Applying the same arguments repeatedly alters schedule s' into an aligned schedule and preserves its optimality. □

Next we show that the optimality of a schedule s is maintained even if cells that are stored in the buffer are released earlier, as long as their new release time satisfies FIFO order and remains within a release band of width $\mathrm{MJ}^\sigma(s)$:

Lemma 2. *Let s be an optimal schedule for sequence σ. Then, for every stream $\sigma_i \subseteq \sigma$ and for every $J \in [J^{\sigma_i}(s), \mathrm{MJ}^{\sigma}(s)]$, the new schedule*

$$s'(p_j^{\sigma_k}) = \begin{cases} \max\left\{a(p_j^{\sigma_k}), \beta^{\sigma_k} - J + jX\right\} & k = i \\ s(p_j^{\sigma_k}) & k \neq i \end{cases}$$

is B-feasible and $\mathrm{MJ}^{\sigma}(s') = \mathrm{MJ}^{\sigma}(s)$. Furthermore, if s is aligned then so is s'.

Proof. Since s' only changes the release schedule of stream σ_i, it clearly preserves the FIFO order and jitter of each stream other than σ_i.

We first show that s' respects the FIFO order of cells in σ_i. Let $p_j^{\sigma_i}$ be any cell in σ_i. If $s'(p_j^{\sigma_i}) = a(p_j^{\sigma_i})$ then its release time is $\leq a(p_{j+1}^{\sigma_i}) \leq s'(p_{j+1}^{\sigma_i})$. Otherwise, $s'(p_j^{\sigma_i}) = \beta^{\sigma_i} - J + jX \leq \beta^{\sigma_i} - J + (j+1)X \leq s'(p_{j+1}^{\sigma_i})$.

In order to bound the max-jitter of s', it suffices to show that $J^{\sigma_i}(s') \leq \mathrm{MJ}^{\sigma}(s)$. Consider any pair of cells $p_a^{\sigma_i}, p_b^{\sigma_i} \in \sigma_i$. By the definition of s', $s'(p_a^{\sigma_i}) \geq \beta^{\sigma_i} - J + aX$. On the other hand, $s'(p_b^{\sigma_i}) = \max\{a(p_b^{\sigma_k}), \beta^{\sigma_k} - J + bX\} \leq \beta^{\sigma_i} + bX$ since $a(p_b^{\sigma_i}) \leq \beta^{\sigma_i} + bX$ by the definition of β^{σ_i}. Hence, $s'(p_b^{\sigma_i}) - s'(p_a^{\sigma_i}) \leq J + (b-a)X$, which implies that $J^{\sigma_i}(s') = \max_{a,b}\{s'(p_b^{\sigma_i}) - s'(p_a^{\sigma_i}) - (b-a)X\} \leq J \leq \mathrm{MJ}^{\sigma}(s)$.

Assume by way of contradiction that s' in not B-feasible, and let t be any time in which a set P of more than B cells are stored in the buffer. Since the release schedule of any stream σ_k other than σ_i is identical under both s and s', every cell $p_j^{\sigma_k} \in P$, for $k \neq i$, is also stored in the buffer at time t under schedule s. Note first that any cell in P is not released upon its arrival. Hence,

$$\begin{aligned}
s'(p_j^{\sigma_i}) &= \beta^{\sigma_i} - J + jX && \text{by the definition of } s' \\
&\leq \beta^{\sigma_i} - J^{\sigma_i}(s) + jX && \text{since } J \in [J^{\sigma_i}(s), \mathrm{MJ}^{\sigma}(s)] \\
&= a(p_k^{\sigma_i}) - kX - J^{\sigma_i}(s) + jX && \text{for } p_k^{\sigma_i} \text{ defining } \beta^{\sigma_i} \\
&\leq s(p_k^{\sigma_i}) - (k-j)X - J^{\sigma_i}(s) && \text{since } a(p_k^{\sigma_i}) \leq s(p_k^{\sigma_i}) \\
&\leq s(p_k^{\sigma_i}) - (k-j)X - \\
&\qquad \big(s(p_k^{\sigma_i}) - s(p_j^{\sigma_i}) - (k-j)X\big) && \text{by definition of } J^{\sigma_i}(s) \\
&\leq s(p_j^{\sigma_i})
\end{aligned}$$

Therefore, all cells $p_j^{\sigma_i} \in P$ are stored in the buffer at time t under schedule s as well, contradicting the B-feasibility of s.

We conclude the proof by showing that if s is aligned then s' is also aligned. Assume s is aligned. For any stream $\sigma_k \neq \sigma_i$ schedules s and s' are identical on σ_k, and therefore σ_k is aligned in s'. Assume by contradiction that σ_i is not aligned, therefore there is a cell $p_j^{\sigma_i}$ such that $s'(p_j^{\sigma_i}) > \beta^{\sigma_i} + jX$. Note that the definition of β^{σ_i} is independent of s and s'. By the definition of s', $\max\left\{a(p_j^{\sigma_i}), \beta^{\sigma_i} - J + jX\right\} > \beta^{\sigma_i} + jX$. It follows that $a(p_j^{\sigma_i}) > \beta^{\sigma_i} + jX$, contradicting the maximality of β^{σ_i}. □

The new schedule obtained in the above lemma is illustrated by the circled cells in Figure 3. By iteratively applying Lemma 2 with $J = \mathrm{MJ}^{\sigma}(s)$ on all streams, we get:

Corollary 1. *Given an optimal aligned schedule s for sequence σ, the schedule defined by*

$$s'(p_j^{\sigma_k}) = \max\left\{a(p_j^{\sigma_k}), \beta^{\sigma_k} - \mathrm{MJ}^{\sigma}(s) + jX\right\}$$

is an optimal aligned schedule.

The following lemma bounds from below the release time of cells in an aligned schedule. Intuitively, this lemma defines the left margin of the release band.

Lemma 3. *For any aligned schedule s for sequence σ, every stream $\sigma_i \subseteq \sigma$, and every cell $p_j^{\sigma_i}$, $s(p_j^{\sigma_i}) \geq \beta^{\sigma_i} - J^{\sigma_i}(s) + jX$.*

Proof. Assume by contradiction that there exists a stream σ_i and a cell $p_j^{\sigma_i}$ such that $s(p_j^{\sigma_i}) < \beta^{\sigma_i} - J^{\sigma_i}(s) + jX$. Let $p_k^{\sigma_i}$ be the cell defining β^{σ_i}. Since s is aligned, it follows that $s(p_k^{\sigma_i}) = a(p_k^{\sigma_i})$. Hence,

$$\begin{aligned} J^{\sigma_i}(s) &\geq s(p_k^{\sigma_i}) - s(p_j^{\sigma_i}) - (k-j)X \\ &> a(p_k^{\sigma_i}) - (\beta^{\sigma_i} - J^{\sigma_i}(s) + jX) - (k-j)X \\ &= a(p_k^{\sigma_i}) - (a(p_k^{\sigma_i}) - kX) + J^{\sigma_i}(s) - jX - kX + jX = J^{\sigma_i}(s), \end{aligned}$$

which is a contradiction. □

Lemma 3 indicates an important property of aligned optimal schedules. In such schedules, the jitter of any stream can be characterized by the release time of a single cell, as depicted in the following corollary: (proof omitted)

Corollary 2. *For any aligned schedule s for sequence σ and every stream $\sigma_i \subseteq \sigma$, $J^{\sigma_i}(s) = \max_j\left\{\beta^{\sigma_i} - s(p_j^{\sigma_i}) + jX\right\}$.*

The following lemma shows that at least one of the widest release bands, corresponding to some stream σ_i attaining the max-jitter, has its left margin determined by the following event: An arrival of a cell causing a buffer overflow, which necessitates some cell of σ_i to be released earlier than desired.

Lemma 4. *Let s be an aligned optimal schedule for sequence σ. There exists a stream $\sigma_i \subseteq \sigma$ that attains the max-jitter, and a cell $p_j^{\sigma_i}$ such that $s(p_j^{\sigma_i}) = \beta^{\sigma_i} - \mathrm{MJ}^{\sigma}(s) + jX$ and $s(p_j^{\sigma_i}) = a(p_\ell^{\sigma})$ for some cell p_ℓ^{σ}.*

Proof. We show by contradiction that if the claim does not hold for an optimal aligned schedule, then such a schedule can be altered into a new schedule with max-jitter strictly less than the original schedule. Formally, consider an aligned optimal schedule s for σ. Let $M = \{\sigma_i \mid J^{\sigma_i}(s) = \mathrm{MJ}^{\sigma}(s)\}$, and for every $\sigma_i \in M$, let $T_i = \left\{p_j^{\sigma_i} \mid s(p_j^{\sigma_i}) = \beta^{\sigma_i} - \mathrm{MJ}^{\sigma}(s) + jX\right\}$. From a geometric point of view, T_i consists of all the cells in σ_i, whose release time lies on the left margin of σ_i's release band. Finally, let $T = \bigcup_{\sigma_i \in M} T_i$. Assume by contradiction that for every $p_j^{\sigma} \in T$, there is no cell p_ℓ^{σ} such that $s(p_j^{\sigma}) = a(p_\ell^{\sigma})$.

Note first that in such a case, $\mathrm{MJ}^{\sigma}(s) > 0$. Otherwise, since s is aligned, for each stream σ_i the cell $p_k^{\sigma_i}$ defining β^{σ_i} satisfies both $s(p_k^{\sigma_i}) = a(p_k^{\sigma_i})$ and $s(p_k^{\sigma_i}) = \beta^{\sigma_i} - 0 + jX$.

The altered schedule s' is obtained by postponing the release of all the cells in T for some positive amount of time. As we shall prove, schedule s' is B-feasible, and has a max-jitter strictly less than $\mathrm{MJ}^\sigma(s)$, contradicting the optimality of s.

For each cell $p_k^{\sigma_i} \in T$ which is the j'th cell of σ (i.e, $p_k^{\sigma_i} = p_j^\sigma$), the exact amount of postponing time is determined by the following constraints:

1. *Avoiding buffer overflow:* Do not postpone further than the first arrival of a cell after $s(p_j^\sigma)$. This constraint is captured by

$$\delta(p_j^\sigma) = \min_{p_\ell^\sigma : a(p_\ell^\sigma) > s(p_j^\sigma)} \left\{ a(p_\ell^\sigma) - s(p_j^\sigma) \right\} .$$

2. *Maintaining FIFO order:* Do not postpone further than $s(p_{k+1}^{\sigma_i})$. This constraint is captured by $\varepsilon(p_j^\sigma) = s(p_{k+1}^{\sigma_i}) - s(p_k^{\sigma_i})$.

If p_j^σ is the last cell in σ, $\delta(p_j^\sigma) = \varepsilon(p_j^\sigma) = \infty$. Let $\delta = \min_{p_j^\sigma \in T} \delta(p_j^\sigma)$ and $\varepsilon = \min_{p_j^\sigma \in T} \varepsilon(p_j^\sigma)$, capturing the amounts of time that satisfy these constraints for all cells in T. Since $\mathrm{MJ}^\sigma(s) > 0$ and by using the previous lemmas and the assumption, it can be verified that both δ and ε are finite and strictly greater than zero.

For the purpose of analysis, define for every stream $\sigma_i \in M$,

$$\rho(\sigma_i) = \min_{p_k^{\sigma_i} \in \sigma_i \setminus T_i} \left\{ s(p_k^{\sigma_i}) - (\beta^{\sigma_i} - \mathrm{MJ}^\sigma(s) + kX) \right\} .$$

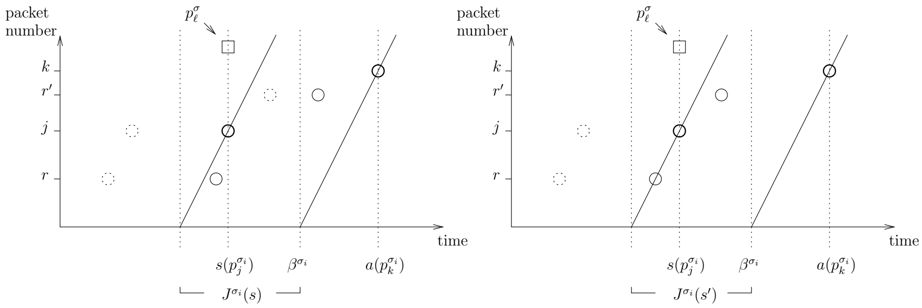
$\rho(\sigma_i)$ comes to capture how far is the rest of the stream from the left margin. Since $J^{\sigma_i}(s) > 0$, $\sigma_i \setminus T_i$ is not empty and $\rho(\sigma_i) > 0$. Let $\rho = \min_{\sigma_i \in M} \rho(\sigma_i)$. It follows that $\rho > 0$.

Let $\Delta = \min\{\delta, \varepsilon, \rho\}$, and consider the following schedule that, as we shall prove, attains a jitter strictly smaller than $\mathrm{MJ}^\sigma(s)$:

$$s'(p_j^\sigma) = \begin{cases} s(p_j^\sigma) + \Delta/2 & p_j^\sigma \in T \\ s(p_j^\sigma) & \text{otherwise} \end{cases}$$

We first prove that s' is B-feasible and maintains FIFO order. Assume by way of contradiction that s' is not B-feasible, and let t be the first time the number of cells in the buffer exceeds B. By the minimality of t, there exists a cell that arrives at time t. For every cell $p_j^\sigma \in T$, no cells arrive to the buffer in the interval $[s(p_j^\sigma), s(p_j^\sigma) + \Delta/2]$ because $\Delta \le \delta(p_j^\sigma)$, implying that t is not in any such interval. But the definition of s' yields that the content of the buffer in such a time t is the same under schedules s and s', thus contradicting the B-feasibility of s. The FIFO order of s' is maintained since $\Delta \le \varepsilon(p_j^\sigma)$ for every $p_j^\sigma \in T$.

We conclude the proof by showing that $\mathrm{MJ}^\sigma(s') < \mathrm{MJ}^\sigma(s)$. Consider any $\sigma_i \in M$, and any $p_k^{\sigma_i}$. If $p_k^{\sigma_i} \in T$ then by the definition of s' and Lemma 3, $s'(p_k^{\sigma_i}) = s(p_k^{\sigma_i}) + \Delta/2 \ge \beta^{\sigma_i} - \mathrm{MJ}^\sigma(s) + kX + \Delta/2$. The same holds also for



(a) schedule s (b) schedule s'

Fig. 3. Outline of arrivals (dotted circles) and releases (full circles) for cells of the stream σ_i that attains the max-jitter, in an aligned release schedule, as discussed in Corollary 1 and in Lemma 4. The square represents an arrival of some cell in σ causing buffer overflow.

$p_k^{\sigma_i} \notin T$: Since $\rho(\sigma_i) \geq \Delta > \Delta/2$, it follows that $s'(p_k^{\sigma_i}) = s(p_k^{\sigma_i}) \geq \beta^{\sigma_i} - \mathrm{MJ}^{\sigma}(s) + kX + \rho(\sigma_i) > \beta^{\sigma_i} - \mathrm{MJ}^{\sigma}(s) + kX + \Delta/2$. Hence, for every $p_k^{\sigma_i}$,

$$\begin{aligned}\beta^{\sigma_i} - s'(p_k^{\sigma_i}) + kX &\leq \beta^{\sigma_i} - (\beta^{\sigma_i} - \mathrm{MJ}^{\sigma}(s) + kX + \Delta/2) + kX \\ &= \mathrm{MJ}^{\sigma}(s) - \Delta/2 \;<\; J^{\sigma_i}(s).\end{aligned}$$

By Corollary 2, $J^{\sigma_i}(s') < J^{\sigma_i}(s)$ for any stream $\sigma_i \in M$. The jitter of any other stream remains unchanged, therefore $MJ^{\sigma}(s') < MJ^{\sigma}(s)$, contradicting the optimality of s. □

Lemma 4 implies that there is an optimal schedule s and a stream σ_i, such that $\mathrm{MJ}(s) = \beta^{\sigma_i} - a(p_l^{\sigma}) + kX$, for some cells $p_k^{\sigma_i} \in \sigma_i$ and $p_l^{\sigma} \in \sigma$. Note that for any stream σ_i, the value of β^{σ_i} can be computed in linear time using only the arrival sequence σ_i. It follows that by enumerating over all possible choices of the pair $(p_k^{\sigma_i}, p_l^{\sigma})$, one can find the collection of possible values of the optimal jitter. For every such value J, verifying that there is a B-feasible release schedule attaining jitter J can be done in linear time by checking the feasibility of the schedule defined in Corollary 1 assuming $\mathrm{MJ}(s) = J$. This yields the following result:

Theorem 2. *There exists a polynomial-time algorithm that finds an optimal schedule for the multi-stream max-jitter problem.*

5 Discussion

This paper examines the problem of jitter regulation and specifically, the tradeoff between the buffer size available at the regulator and the optimal jitter attainable using such a buffer. We deal with the realistic case where the regulator must handle many streams concurrently, thus answering a primary question posed in [1].

We focus our attention on regulating the jitter of multiple streams with the objective of minimizing the maximum jitter attained by any of these streams. We show that the offline problem of finding a schedule that attains the optimal max-jitter can be solved in polynomial time, by a time-efficient algorithm which produces an optimal schedule. We observe that existing single-stream online algorithms can be used to devise an online algorithm for the multi-stream jitter regulation problem, at a cost of multiplying the buffer size linearly by the number of streams. We prove that such a resource augmentation is essential by providing an asymptotically matching lower bound. Our results for the online setting apply also to the problem of finding a release schedule with optimal average jitter.

Note that our offline algorithm suggests an interesting heuristic for improving the value of the jitter for an online algorithm using a buffer of size considerably less than MB, compared to the optimal jitter attainable with a buffer of size B. One can calculate an optimal schedule of a prefix of the traffic using our offline algorithm, and then prolong the schedule by attempting to send consecutive cells as equally spaced as possible. Although there are traffics in which this approach fails, as shown by our lower bound, it may prove a useful heuristic in situations where the overall traffic in the network does not change radically over time.

Since real-life networks clearly have finite capacity links, it is also interesting to investigate the behavior of a jitter regulator that handles multiple streams simultaneously and its outgoing link is of bounded capacity. In addition, since regulators might be allowed to drop cells, it is of interest to examine the correlations between buffer size, optimal jitter, and drop ratio.

Acknowledgements: We would like to thank Hagit Attiya, Seffi Naor, Adi Rosén, and Shmuel Zaks for their useful comments.

References

1. Mansour, Y., Patt-Shamir, B.: Jitter control in Qos networks. IEEE/ACM Transactions on Networking **9** (2001) 492–502
2. The ATM Forum: Traffic Management Specification. (1999) Version 4.1, AF-TM-0121.000.
3. Zhang, H.: Service disciplines for guaranteed performance service in packet-switching networks. Proceedings of the IEEE **83** (1995) 1374–1396
4. Tanenbaum, A.: Computer Networks. Fourth edn. Prentice Hall (2003)
5. Keshav, S.: An Engineering Approach to Computer Networking. Addison-Wesley Publishing Co. (1997)
6. Zhang, H., Ferrari, D.: Rate-controlled service disciplines. Journal of High-Speed Networks **3** (1994) 389–412
7. Borodin, A., El-Yaniv, R.: Online Computation and Competitive Analysis. Cambridge University Press (1998)
8. Sleator, D., Tarjan, R.: Amortized efficiency of list update and paging rules. Communications of the ACM **28** (1985) 202–208
9. Koga, H.: Jitter regulation in an internet router with delay constraint. Journal of Scheduling **4** (2001) 355–377