Jitter Regulation for Multiple Streams

DAVID HAY
Ben Gurion University of the Negev
and
GABRIEL SCALOSUB
Tel Aviv University

For widely-used interactive communication, it is essential that traffic is kept as smooth as possible; the smoothness of the 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. Moreover, communication links have limited capacity, and these may pose further restrictions on the choices made by the regulator. 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, in the upcapacitated case, previous results in [Mansour and Patt-Shamir 2001] 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 (and an offline algorithm). 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.

Categories and Subject Descriptors: C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Packet-Switching Network, Store and Forward Networks*; F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Non Numerical Algorithms and Problems—*Computations on Discrete Structures, Sequencing and Scheduling*

General Terms: Algorithms, Design, Performance, Theory

Additional Key Words and Phrases: Jitter Regulation, Buffer Management, Quality of Service, Buffer Overflow and Underflow, Scheduling, Online Algorithms, Competitive Analysis

A preliminary version of this paper appeared in the Proceedings of the 13th Annual European Symposium on Algorithms (ESA), 2005, pp. 496–507.
This work was done while both authors were with the Department of Computer Science at the Technion, Israel.
Author's address: D. Hay, Department of Computer Science, Ben Gurion University of the Negev, Be'er Sheva 84105, Israel (e-mail: davidhay@cs.bgu.ac.il). G. Scalosub, Department of Electrical Engineering, Tel Aviv University, Ramat Aviv 69978, Israel (e-mail: gabriels@eng.tau.ac.il).

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 which provide smooth and continuous traffic is mostly motivated by the increasing popularity of interactive communication and in particular video/audio streaming.[1] The smoothness of such traffic is captured by the notion of *delay jitter* (or *Cell Delay Variation* [The ATM Forum 1999]); 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 [Zhang 1995]. This is important when a service provider in a QoS network must meet service level agreements (SLAs) 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 more customers [Tanenbaum 2003].

Jitter control mechanisms have been extensively studied in recent years (see a survey in [Zhang 1995]). These are usually modelled as *jitter regulators* [Mansour and Patt-Shamir 2001; Keshav 1997; Zhang and Ferrari 1994] 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 [Mansour and Patt-Shamir 2001], 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. Furthermore, in real-life networks, links have finite capacities, which impose further limitations that should be taken into account by the regulator.

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 delays 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, while respecting the capacity constraints of the outgoing link associated with stream i. 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

[1] For example, 6.98 billion video streams were initiated by U.S. users during August 2006, while the U.S. streaming audience increased by 4 percent from July 2006 to reach 110.3 million streamers in August 2006, representing about 64 percent of the total U.S. Internet audience [comScore Networks 2006].

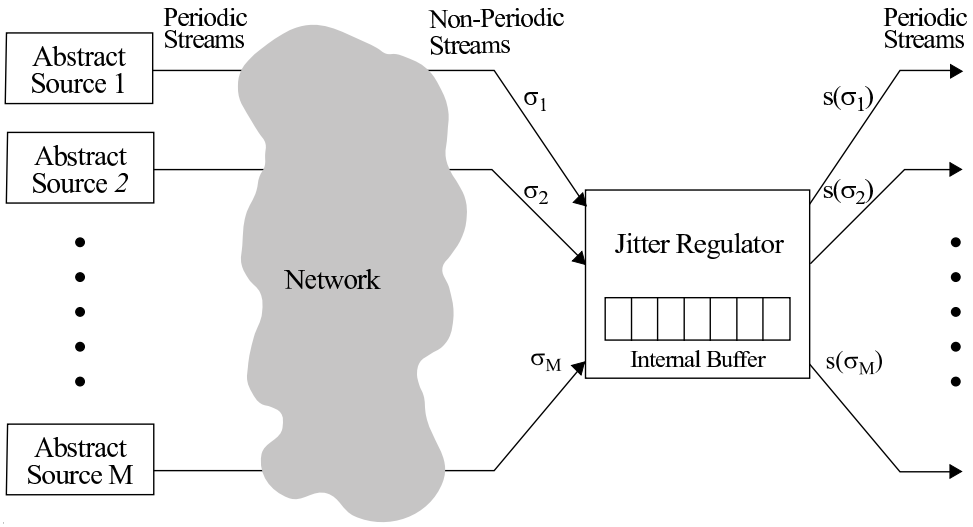


Fig. 1. The multi-stream jitter regulation model

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 [Mansour and Patt-Shamir 2001].

We evaluate the performance of a regulator in the multi-stream model by considering the maximum jitter obtained on any stream. We show that, somewhat surprisingly, the offline problem can be solved in polynomial time. This is done by 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* [Borodin and El-Yaniv 1998; Sleator and Tarjan 1985] approach in order to examine the online problem. In this setting, when there are no capacity constraints on the outgoing links, 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 [Mansour and Patt-Shamir 2001, 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 MB in order to obtain a jitter within a bounded factor from the optimal jitter possible with a buffer of size B. We further show that these tight results (up to a factor of 2) 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

The problem of jitter control has received much attention in recent years, along with the increasing importance of providing QoS guarantees. A prime example is the *Differentiated Services (DiffServ)* architecture, in which there is a specific requirement to maintain low-jitter for *Expedited Forwarding (EF)* traffic [Davie et al. 2002].

Several algorithms have been proposed with the aim of providing traffic jitter control. Generally, all proposed algorithms are not work-conserving, i.e., they might delay releasing a cell even if there are cells in the buffer and the outgoing links are not fully utilized.

A jitter control algorithm which reconstructs the entire sequence at the destination using a predetermined maximum delay bound was proposed in [Partridge 1991]. The *Jitter-Earliest-Due-Date* algorithm proposed in [Verma et al. 1991] uses a predetermined maximum delay bound in order to calculate a deadline for every cell, such that it is released precisely upon its deadline. The *Stop-and-Go* algorithm proposed in [Golestani 1990] uses time frames of predetermined lengths in order to regulate traffic, such that cells arriving in the middle of a frame, are only made available for sending in the following time frame. The *Hierarchical-Round-Robin* algorithm proposed in [Kalmanek et al. 1990] uses a framing strategy similar to the one used in the Stop-and-Go algorithm, but releases are governed by a round robin policy that sometimes allocates non-utilized release time-slots to other streams. For a more thorough survey of jitter control algorithms, see [Zhang 1995]. A slightly different line of research investigated jitter regulation in the Combined Input-Output Queue switch architecture, forcing the jitter regulator to obey additional constraints posed by the switching architecture [Keslassy et al. 2005].

The problem of jitter control in an adversarial setting has been discussed by Mansour and Patt-Shamir [Mansour and Patt-Shamir 2001], where they 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 [Koga 2001] 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.

Paper Organization

In Section 2 we define the network model and give some geometric intuition into the problem of jitter regulation for multiple streams. Section 3 presents our results for the online settings and shows that jitter regulation does not scale well as the number of streams increases. In Section 4 we present an efficient algorithm for finding a schedule which minimizes the maximum jitter encountered by any stream in the input. Finally, in Section 5 we conclude with a discussion of our results, and present some open problems.

2. MODEL DESCRIPTION, NOTATION, AND TERMINOLOGY

In this section we define the basic concepts of the multi-stream jitter regulation model, with bounded capacity links.

Definition 2.1. 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 $a(p_i^\sigma)$, we define the following:

(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]

(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) Given a link capacity $\lambda \in \mathbb{N}$, a release schedule s for σ is *capacity-feasible* if for any $p_j^\sigma \in \sigma$,

$$s(p_{j+\lambda}^\sigma) \geq s(p_j^\sigma) + 1.$$

That is, at any time t, at most λ cells are transmitted over the same link simultaneously.

(4) A release schedule is *feasible* if it is B-feasible and capacity-feasible.

(5) 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).[3]

Notice that the arrival sequence a need not be equally spaced due to variable delays in the network.

We first extend Definition 2.1 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 by λ^{σ_i} the capacity of its outgoing link. An essential assumption is that $1/X^{\sigma_i} \leq \lambda^{\sigma_i}$, since otherwise the outgoing link cannot support the source's rate, and therefore cells must be dropped. Note that each stream σ_i has its own outgoing link and therefore should satisfy Condition 3 for its value of λ^{σ_i}, independently of other streams.

We assume for simplicity that all streams have the same inter-release time X, and the same outgoing link capacity λ; 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 schedules:

[2]Note that $s(p_i^\sigma)$ may equal $a(p_i^\sigma)$. This corresponds, for example, to a cut-through switch, in which a cell can start being transmitted through the outgoing link as soon as it arrives at the switch (see [Kermani and Kleinrock 1979; Tassiulas 1999]). This model is also used in [Mansour and Patt-Shamir 2001].

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

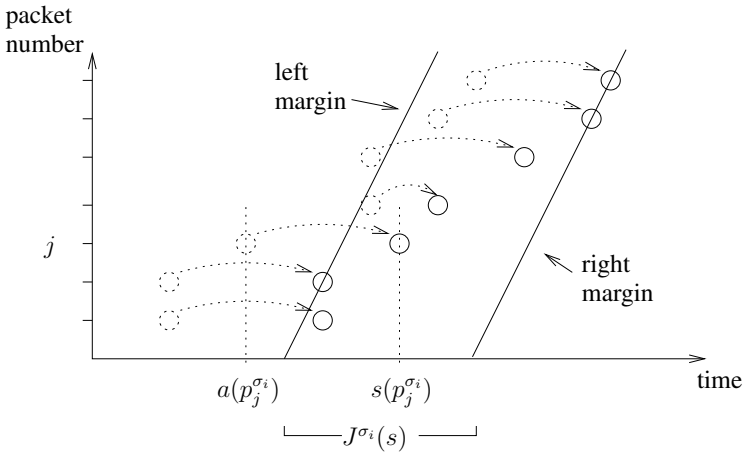


Fig. 2. Outline of arrivals (dotted circles) and marked releases (full circles). The jitter of stream σ_i is the width of the band with slope $1/X$ enclosing all releases.

Definition 2.2. The *max-jitter* of a multi-stream sequence $\sigma = \bigcup\{\sigma_1, \ldots, \sigma_M\}$ obtained by a release schedule s is the maximal jitter obtained by any of the streams composing the sequence; that is,

$$\mathrm{MJ}^{\sigma}(s) = \max_{1 \le k \le M} J^{\sigma_k}(s).$$

In what follows, given any algorithm A, we denote by $J^{\sigma_i}(A)$ ($\mathrm{MJ}^{\sigma}(A)$) the jitter (max-jitter) corresponding to the schedule produced by A given stream σ_i (multi-stream σ). If an algorithm A fails to produce a feasible schedule for stream σ_i, we define $J^{\sigma_i}(A) = \infty$.

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). The *release band* is the band with slope $1/X$ that encloses all the marked points and has minimal *width*, where the width of the band is the maximal difference in the x-axis coordinates between its margins. 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.

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 REGULATION

As mentioned previously, for the uncapacitated case, 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 a max-jitter within a bounded factor from the optimal max-jitter possible with a

buffer of size B, any online algorithm needs a buffer of size at least MB cells. Hence, in order to maintain any reasonable jitter performance, it is necessary to increase the buffer size in a linear proportion to the number of streams. Notice that our lower bounds hold even if there are capacity constraints on the outgoing links.

THEOREM 3.1. *For every online algorithm* ALG *with an internal buffer of size smaller than MB, and for any $T > 0$, there exists an arrival sequence consisting of M streams, forcing* ALG *to have max-jitter at least T, while the optimal jitter possible with a buffer of size B is zero.*

PROOF. Let ALG be an online algorithm with a buffer of size at most $MB-1$. Consider the following arrival sequence σ: For every $0 \leq k \leq B-1$, M cells arrive at the regulator at time $k \cdot X$, one for every stream.

Since the buffer size is at most $MB-1$ and MB cells arrived by time $t' = (B-1)X$, it follows that ALG releases a cell by time t', say of stream σ_i. Consider the following continuation for σ: Given some $T > 0$, in time $T' \geq t' + BX + T$, a single cell of stream σ_i arrives at the regulator.

Note that ALG releases the first cell of stream σ_i by time t', the last cell of stream σ_i cannot be sent prior to time T', and σ_i consists of $B+1$ cells. Let s be the schedule produced by ALG. It follows that

$$\begin{aligned} J^{\sigma_i}(\mathrm{ALG}) &\geq s(p_B^{\sigma_i}) - s(p_0^{\sigma_i}) - (B-0)X \\ &\geq T' - t' - BX \\ &\geq T + t' + BX - t' - BX = T, \end{aligned}$$

which can be arbitrarily large. It follows also that $\mathrm{MJ}^{\sigma}(\mathrm{ALG}) \geq T$. 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. In addition, each stream in σ obeys its capacity constraint since $1/X \leq \lambda$. Clearly, every stream attains zero jitter by this release schedule. □

Theorem 3.1 implies that in case the buffer size is smaller than MB, there are scenarios in which an optimal schedule attains zero jitter for all streams, while any online algorithm can be forced to produce a schedule with arbitrarily large max-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 a buffer, minimizes the jitter of each stream independently, it clearly minimizes the overall average jitter as well, thus providing a matching upper bound (up to a factor of 2).

Moreover, we are able to prove a more general lower bound:

THEOREM 3.2. *For every online algorithm* ALG *with an internal buffer size smaller than*

$$\max\{MB, M(B-1)+B+1\},$$

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 $M(B-1)+B$. Consider the following arrival sequence σ: For every $0 \leq k \leq B-1$, M cells arrive at the

regulator at time $k \cdot X$, one for every stream. The sequence stops if ALG releases a cell before time $t' = BX$.

If ALG releases a cell before time t', the claim follows from the proof of Theorem 3.1.

Therefore, assume now that ALG does not release any cells before time t', implying that in time t' there are MB cells in the buffer. Note that this implies that ALG has buffer size at least MB. 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 $M(B-1)+B = MB+B-M$, and before time t' ALG has not released any of the $MB+B+1$ cells in σ, it must release at least $M+1$ cells in time t'. By the pigeonhole principle it follows that at least two of the released cells correspond to the same stream, say σ_i. If $\lambda = 1$, ALG cannot produce a feasible schedule, and therefore $\mathrm{MJ}^{\sigma}(\mathrm{ALG}) = \infty$. If $\lambda \geq 2$, the release schedule produced by ALG, denoted by s, can be feasible, but the jitter attained by stream σ_i is at least

$$s(p_0^{\sigma_i}) - s(p_1^{\sigma_i}) - (0-1)X = t' - t' - (0-1)X = X,$$

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. To see this, consider the following release schedule: Every cell of a stream other than σ_1 is released immediately upon its arrival, and for every $0 \leq j \leq 2B$, 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 since every cell, except the last B cells of stream σ_1, is sent immediately upon arrival, and no two cells of the same stream are sent simultaneously. □

Note that Theorem 3.2 implies that a greater resource augmentation is required, compared to Theorem 3.1, while Theorem 3.1 implies unbounded competitiveness whereas Theorem 3.2 only implies no online algorithm can obtain the *optimal* jitter. Furthermore, the lower bound described in Theorem 3.2 exactly coincides with the result of the single stream model (i.e., $M = 1$) with no capacity constraint on the outgoing link [Mansour and Patt-Shamir 2001], where it is shown that in order to obtain the optimal jitter possible with a buffer of size B, any online algorithm must use a buffer of size at least $2B$. Note that both lower bounds still hold even if the online algorithm has prior knowledge of the exact value of the optimal max-jitter of the arrival sequence.

4. AN EFFICIENT OFFLINE ALGORITHM

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

Note that when imposing a capacity constraint on the outgoing link, some arrival sequences may prohibit any schedule from being feasible; For example, if a burst of $B + \lambda + 1$ cells arrive at the regulator simultaneously, the link capacity constraint and the B-feasibility constraint lay in contradiction. The following lemma characterizes arrival sequences with feasible release schedules:

LEMMA 4.1. *There exists a feasible schedule for a sequence $\sigma = \bigcup\{\sigma_1, \ldots, \sigma_M\}$ if and only if the greedy schedule s defined by*

$$s_{greedy}(p_j^{\sigma_i}) = \begin{cases} a(p_j^{\sigma_i}) & i = 1, \ldots, M \quad j = 0, \ldots, \lambda - 1 \\ \max\left\{a(p_j^{\sigma_i}), s_{greedy}(p_{j-\lambda}^{\sigma_i}) + 1\right\} & i = 1, \ldots, M \quad j = \lambda, \ldots \end{cases}$$

is B-feasible.

PROOF. Assume s_{greedy} is B-feasible. Since s_{greedy} clearly satisfies the capacity constraint and the FIFO constraint, s_{greedy} is a feasible schedule.

On the other hand, assume there is a feasible schedule s' for the capacitated problem. For every stream σ_i, we prove by induction on the cell number j that for every $p_j^{\sigma_i}$, $s_{\text{greedy}}(p_j^{\sigma_i}) \leq s'(p_j^{\sigma_i})$. For $j = 0, \ldots, \lambda - 1$ the claim follows from the arrival feasibility of s'. Assume the claim holds for $m < j$. If $s_{\text{greedy}}(p_j^{\sigma_i}) = a(p_j^{\sigma_i})$ then the claim follows from the arrival feasibility of s'. Otherwise, $s_{\text{greedy}}(p_j^{\sigma_i}) = s_{\text{greedy}}(p_{j-\lambda}^{\sigma_i}) + 1$. By our induction hypothesis, $s_{\text{greedy}}(p_{j-\lambda}^{\sigma_i}) \leq s'(p_{j-\lambda}^{\sigma_i})$. Since s' satisfies the capacity constraint we have that $s'(p_j^{\sigma_i}) \geq s'(p_{j-\lambda}^{\sigma_i}) + 1$. Combining these inequalities, the claim follows. The claim shows that for any time t, any set of cells residing in the buffer in time t according to s_{greedy}, also resides in the buffer in time t according to s'. Since s' is B-feasible, it follows that s_{greedy} is B-feasible as well. □

Given a sequence σ that is an interleaving of M streams and has a feasible release schedule, 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 [Mansour and Patt-Shamir 2001, Algorithm A]: Essentially, 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[4]. In particular, it follows that for every sequence σ that has a feasible release schedule, there exists an optimal release schedule. Unfortunately, it is computationally intractable 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 (or decide that no feasible schedule exists) in polynomial time, based solely on the cells' arrival times, and the parameters X and B.

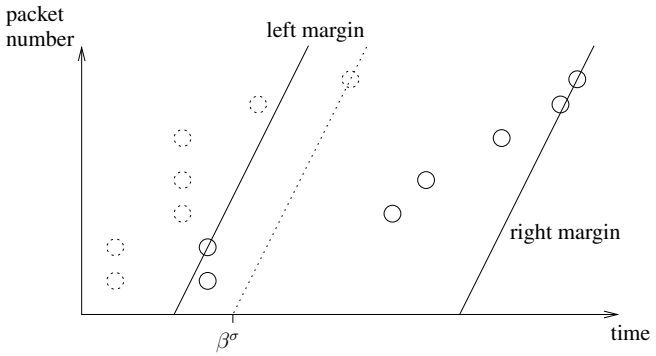
For every cell $p_j^{\sigma_i}$ of any stream σ_i, one can intuitively consider $t = a(p_j^{\sigma_i}) - jX$ as the time at which $p_0^{\sigma_i}$ should be sent, so that $p_j^{\sigma_i}$ is sent immediately upon its arrival, in a perfectly periodic release schedule. For any stream σ_i, denote by $\beta^{\sigma_i} = \max_j \left\{a(p_j^{\sigma_i}) - jX\right\}$. From a geometric point of view, β^{σ_i} is a lower bound on the intersection between the time axis and the right margin of any release band (see Figure 3(a)), since otherwise the cell defining β^{σ_i} would have been 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 any 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 3(b)).

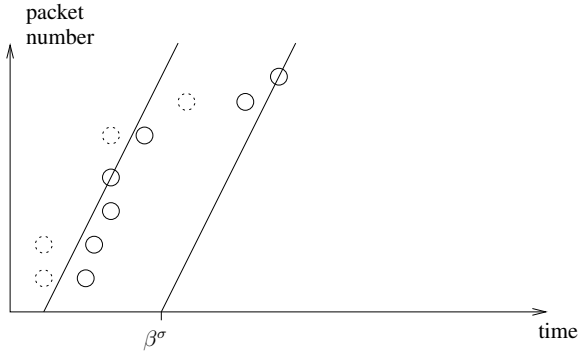
A release schedule s for σ is said to be *aligned*, if every stream is aligned in s. The following lemma shows that one can iteratively align the streams of an optimal schedule without increasing the overall jitter:

LEMMA 4.2. *For every sequence σ, there exists an optimal aligned schedule s.*

[4]The offline algorithm in [Mansour and Patt-Shamir 2001, Algorithm A] does not deal with capacity-constraint, but these can be easily incorporated.



(a) non-aligned schedule



(b) aligned schedule

Fig. 3. Outline of arrivals (dotted circles) and marked releases (full circles).

PROOF. Given an optimal schedule s' for sequence σ with $\ell < M$ 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 verified that $\overline{s}$ satisfies the FIFO constraint. Schedule $\overline{s}$ is B-feasible, since s' is B-feasible and for any cell $p_j^{\sigma_i}$, $\overline{s}(p_j^{\sigma_i}) \leq s'(p_j^{\sigma_i})$.

In order to prove that schedule $\overline{s}$ is capacity-feasible, we consider any two cells $p_j^{\sigma_i}, p_{j+\lambda}^{\sigma_i}$ and show that $\overline{s}(p_{j+\lambda}^{\sigma_i}) \geq \overline{s}(p_j^{\sigma_i}) + 1$. We distinguish between four cases: If $\overline{s}(p_{j+\lambda}^{\sigma_i}) = s'(p_{j+\lambda}^{\sigma_i})$ and $\overline{s}(p_j^{\sigma_i}) = s'(p_j^{\sigma_i})$ the claim immediately follows from feasibility of s'. In case $\overline{s}(p_{j+\lambda}^{\sigma_i}) = \beta^{\sigma_k} + (j+\lambda)X$ and $\overline{s}(p_j^{\sigma_i}) = \beta^{\sigma_k} + jX$ then $\overline{s}(p_{j+\lambda}^{\sigma_i}) - \overline{s}(p_j^{\sigma_i}) = \lambda X \geq 1$, since $\lambda \geq 1/X$. If $\overline{s}(p_{j+\lambda}^{\sigma_i}) = s'(p_{j+\lambda}^{\sigma_i})$ and $\overline{s}(p_j^{\sigma_i}) = \beta^{\sigma_k} + jX$ then $\overline{s}(p_{j+\lambda}^{\sigma_i}) - \overline{s}(p_j^{\sigma_i}) \geq s'(p_{j+\lambda}^{\sigma_i}) - s'(p_j^{\sigma_i}) \geq 1$, by the definition of $\overline{s}$. Finally, if $\overline{s}(p_{j+\lambda}^{\sigma_i}) = \beta^{\sigma_k} + (j+\lambda)X$ and $\overline{s}(p_j^{\sigma_i}) = s'(p_j^{\sigma_i})$ then $\overline{s}(p_{j+\lambda}^{\sigma_i}) - \overline{s}(p_j^{\sigma_i}) \geq \beta^{\sigma_k} + (j+\lambda)X - (\beta^{\sigma_k} + jX) = \lambda X \geq 1$ by

the definition of $\overline{s}$.

Stream σ_i is aligned in $\overline{s}$, since 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 $p_j^{\sigma_i}, p_m^{\sigma_i} \in \sigma_i$. Assume without loss of generality that $\overline{s}(p_j^{\sigma_i}) - jX \geq \overline{s}(p_m^{\sigma_i}) - mX$. If this does not hold then our term is negative, and we can simply switch the roles of $p_j^{\sigma_i}$ and $p_m^{\sigma_i}$. We distinguish between four possible cases: In the first case where $\overline{s}(p_j^{\sigma_i}) = s'(p_j^{\sigma_i})$ and $\overline{s}(p_m^{\sigma_i}) = s'(p_m^{\sigma_i})$ the result follows immediately for the definition of $J^{\sigma_i}(s')$. In the case where $\overline{s}(p_j^{\sigma_i}) = \beta^{\sigma_i} + jX$ and $\overline{s}(p_m^{\sigma_i}) = \beta^{\sigma_i} + mX$ the term is zero, which is not more than $J^{\sigma_i}(s')$ by definition. The third case to consider is when $\overline{s}(p_j^{\sigma_i}) = s'(p_j^{\sigma_i})$ and $\overline{s}(p_m^{\sigma_i}) = \beta^{\sigma_i} + mX$. This implies that $s'(p_j^{\sigma_i}) < \beta^{\sigma_i} + jX$, thus $s'(p_j^{\sigma_i}) - jX < \beta^{\sigma_i}$. Therefore $\overline{s}(p_j^{\sigma_i}) - jX = s'(p_j^{\sigma_i}) - jX < \beta^{\sigma_i} = \overline{s}(p_m^{\sigma_i}) - mX$, contradicting the assumption on $p_j^{\sigma_i}$ and $p_m^{\sigma_i}$. The last case to consider is when $\overline{s}(p_j^{\sigma_i}) = \beta^{\sigma_i} + jX$ and $\overline{s}(p_m^{\sigma_i}) = s'(p_m^{\sigma_i})$: Similarly to the previous case, this implies that $\beta^{\sigma_i} < s'(p_j^{\sigma_i}) - jX$, and therefore $\overline{s}(p_j^{\sigma_i}) - \overline{s}(p_m^{\sigma_i}) - (j - m)X = \beta^{\sigma_i} - (s'(p_m^{\sigma_i}) - mX) < s'(p_j^{\sigma_i}) - jX - (s'(p_m^{\sigma_i}) - mX) \leq J^{\sigma_i}(s')$, as required.

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

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

LEMMA 4.3. *For any 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 $s(p_k^{\sigma_i}) \geq a(p_k^{\sigma_i})$,

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

which is a contradiction. □

Lemma 4.3 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:

COROLLARY 4.4. *For any aligned schedule s for sequence σ and every stream $\sigma_i \subseteq \sigma$,*

$$J^{\sigma_i}(s) = \max_j \left\{ \beta^{\sigma_i} - s(p_j^{\sigma_i}) + jX \right\}.$$

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 4.5. *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*

$$s'(p_j^{\sigma_k}) = \begin{cases} \max\left\{a(p_j^{\sigma_k}), \beta^{\sigma_k} - J + jX\right\} & k = i, j < \lambda \\ \max\left\{a(p_j^{\sigma_k}), \beta^{\sigma_k} - J + jX, s'(p_{j-\lambda}^{\sigma_k}) + 1\right\} & k = i, j \geq \lambda \\ s(p_j^{\sigma_k}) & k \neq i \end{cases}$$

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, capacity-constraint and jitter of each stream other than σ_i. Furthermore, by its definition, schedule s' clearly satisfies the capacity constraint for σ_i.

We first show that s' respects the FIFO order of cells in σ_i. Assume by contradiction that $p_j^{\sigma_i}$ is the first cell in σ_i such that $s'(p_j^{\sigma_i}) > s'(p_{j+1}^{\sigma_i})$. If $s'(p_j^{\sigma_i}) = a(p_j^{\sigma_i})$ then its release time cannot be later than $a(p_{j+1}^{\sigma_i}) \leq s'(p_{j+1}^{\sigma_i})$. If $s'(p_j^{\sigma_i}) = \beta^{\sigma_i} - J + jX$ then $s'(p_j^{\sigma_i}) \leq \beta^{\sigma_i} - J + (j+1)X \leq s'(p_{j+1}^{\sigma_i})$. It follows that $s'(p_j^{\sigma_i}) = s'(p_{j-\lambda}^{\sigma_i}) + 1$. However, $s'(p_{j+1}^{\sigma_i}) \geq s'(p_{j+1-\lambda}^{\sigma_i}) + 1 \geq s'(p_{j-\lambda}^{\sigma_i}) + 1 = s'(p_j^{\sigma_i})$, where the first inequality follows from the capacity constraint and the second inequality follows from the assumption that $p_j^{\sigma_i}$ is the first cell to violate the FIFO order.

In order to bound the max-jitter of s', it suffices to show that $J^{\sigma_i}(s') \leq \mathrm{MJ}^{\sigma}(s)$. We first show by induction on j that $s'(p_j^{\sigma_i}) \leq \beta^{\sigma_i} + jX$. For the base case $j = 0$, both $\beta^{\sigma_i} - J$ and $a(p_j^{\sigma_i})$ are at most β^{σ_i} by the definition of β^{σ_i}. For $j > 0$, $a(p_j^{\sigma_i}) \leq \beta^{\sigma_i} + jX$ and $s'(p_{j-\lambda}^{\sigma_i}) + 1 \leq \beta^{\sigma_i} + (j-\lambda)X + 1 \leq \beta^{\sigma_i} + jX$ by the induction hypothesis and the fact that $\lambda > 1/X$.

Consider any pair of cells $p_a^{\sigma_i}, p_b^{\sigma_i} \in \sigma_i$. By the definition of s', $s'(p_a^{\sigma_i}) \geq \beta^{\sigma_i} - J + aX$. As we proved, $s'(p_b^{\sigma_i}) \leq \beta^{\sigma_i} + bX$. Hence, $s'(p_b^{\sigma_i}) - s'(p_a^{\sigma_i}) \leq J + (b-a)X$, which implies that $J^{\sigma_i}(s') = \max_{a,b}\left\{s'(p_b^{\sigma_i}) - s'(p_a^{\sigma_i}) - (b-a)X\right\} \leq J \leq \mathrm{MJ}^{\sigma}(s)$.

In order to show that s' is B-feasible, we first show that for every cell $p_j^{\sigma_i} \in \sigma_i$, $s'(p_j^{\sigma_i}) \leq s(p_j^{\sigma_i})$. The proof is by induction on the cell index j. By Lemma 4.3, the claim holds for $j = 0$, since $J \geq J^{\sigma_i}(s)$. For $j > 0$, we consider the following three cases. If $s'(p_j^{\sigma_i}) = a(p_j^{\sigma_i})$, then the claim follows from the arrival feasibility of s. If $s'(p_j^{\sigma_i}) = s'(p_{j-\lambda}^{\sigma_i}) + 1$, then by the induction hypothesis, $s'(p_j^{\sigma_i}) \leq s(p_{j-\lambda}^{\sigma_i}) + 1$, which is at most $s(p_j^{\sigma_i})$, by the capacity-feasibility of s. The last case to consider is when $s'(p_j^{\sigma_i}) = \beta^{\sigma_i} - J + jX$. In this case

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

We now turn to show that s' is B-feasible. Assume the contrary, 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. Since for every cell $p_j^{\sigma_i} \in \sigma_i$, $s'(p_j^{\sigma_i}) \leq s(p_j^{\sigma_i})$, all cells $p_j^{\sigma_i} \in P$ are stored in the buffer at time t under schedule s as well. This contradicts the assumption that s is B-feasible.

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'. As shown above, for every cell $p_j^{\sigma_i} \in \sigma_i$, $s'(p_j^{\sigma_i}) \leq s(p_j^{\sigma_i})$. Since s is aligned, we are guaranteed that for every cell $p_j^{\sigma_i} \in \sigma_i$, $s(p_j^{\sigma_i}) \leq \beta^{\sigma_i} + jX$. Combining these two inequalities we obtain that σ_i is also aligned in s'. □

By iteratively applying Lemma 4.5 with $J = \mathrm{MJ}^{\sigma}(s)$ on all streams, we get:

COROLLARY 4.6. *Given an optimal aligned schedule s for sequence σ, the schedule defined by*

$$s'(p_j^{\sigma_k}) = \begin{cases} \max\left\{a(p_j^{\sigma_k}), \beta^{\sigma_k} - \mathrm{MJ}^{\sigma}(s) + jX\right\} & j < \lambda \\ \max\left\{a(p_j^{\sigma_k}), \beta^{\sigma_k} - \mathrm{MJ}^{\sigma}(s) + jX, s'(p_{j-\lambda}^{\sigma_k}) + 1\right\} & j \geq \lambda \end{cases}$$

is an optimal aligned schedule.

Corollary 4.6 also implies the following important observation:

COROLLARY 4.7. *For every $J > 0$ and sequence σ, if the schedule defined by*

$$s'(p_j^{\sigma_k}) = \begin{cases} \max\left\{a(p_j^{\sigma_k}), \beta^{\sigma_k} - J + jX\right\} & j < \lambda \\ \max\left\{a(p_j^{\sigma_k}), \beta^{\sigma_k} - J + jX, s'(p_{j-\lambda}^{\sigma_k}) + 1\right\} & j \geq \lambda \end{cases}$$

is not B-feasible, then there is no B-feasible schedule for sequence σ attaining max-jitter J.

PROOF. Assume that s' is not B-feasible and that there is a schedule $\overline{s}$ that attains a max-jitter J. It follows that the optimal schedule for sequence σ, denoted by s, attains max-jitter $\mathrm{MJ}^{\sigma}(s) \leq J$. By Corollary 4.6, it follows that the schedule defined by

$$s''(p_j^{\sigma_k}) = \begin{cases} \max\left\{a(p_j^{\sigma_k}), \beta^{\sigma_k} - \mathrm{MJ}^{\sigma}(s) + jX\right\} & j < \lambda \\ \max\left\{a(p_j^{\sigma_k}), \beta^{\sigma_k} - \mathrm{MJ}^{\sigma}(s) + jX, s''(p_{j-\lambda}^{\sigma_k}) + 1\right\} & j \geq \lambda \end{cases}$$

is B-feasible. Since schedule s' releases every cell before its release time under schedule s, this implies that schedule s' is also B-feasible, contradicting the assumption. □

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 (possibly of another stream), which necessitates some cell of σ_i to be released earlier than desired.

LEMMA 4.8. *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 $p_\ell^{\sigma} \in \sigma$.*

PROOF. If $\mathrm{MJ}^{\sigma}(s) = 0$ then, for every stream σ_i, the cell defining β^{σ_i} is sent upon its arrival (since s is aligned), and the claim follows. Thus, assume next that $\mathrm{MJ}^{\sigma}(s) > 0$.

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 $\Sigma = \{\sigma_i \mid J^{\sigma_i}(s) = \mathrm{MJ}^{\sigma}(s)\}$, and for every $\sigma_i \in \Sigma$, 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 $T = \bigcup_{\sigma_i \in \Sigma} T_i$. Assume by contradiction that for every $p_j^{\sigma} \in T$, there is no cell p_ℓ^{σ} such that $s(p_j^{\sigma}) = a(p_\ell^{\sigma})$.

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, capacity-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

$$\delta(p_j^\sigma) = \begin{cases} \min\limits_{p_\ell^\sigma : a(p_\ell^\sigma) > s(p_j^\sigma)} \left\{a(p_\ell^\sigma) - s(p_j^\sigma)\right\} & \text{if } \left\{p_\ell^\sigma : a(p_\ell^\sigma) > s(p_j^\sigma)\right\} \neq \emptyset. \\ \infty & \text{otherwise.} \end{cases}$$

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

$$\varepsilon(p_j^\sigma) = \begin{cases} s(p_{k+1}^{\sigma_i}) - s(p_k^{\sigma_i}) & \text{if } p_j^\sigma \text{ is not the last cell in } \sigma_i \\ \infty & \text{otherwise.} \end{cases}$$

(3) *Maintaining the capacity constraint:* Recalling that $p_j^\sigma = p_k^{\sigma_i}$, do not postpone further than $s(p_{k+\lambda}^{\sigma_i}) - 1$, unless $p_{k+\lambda}^{\sigma_i} \in T$. This constraint is captured by

$$\mu(p_j^\sigma) = \begin{cases} (s(p_{k+\lambda}^{\sigma_i}) - 1) - s(p_k^{\sigma_i}) & p_{k+\lambda}^{\sigma_i} \notin T \\ \infty & p_{k+\lambda}^{\sigma_i} \in T \text{ or } p_{k+\lambda}^{\sigma_i} \text{does not exist.} \end{cases}$$

The geometric intuition underlying each of these constraints is depicted in Figure 4.

Let $\delta = \min_{p_j^\sigma \in T} \delta(p_j^\sigma)$, $\varepsilon = \min_{p_j^\sigma \in T} \varepsilon(p_j^\sigma)$, and $\mu = \min_{p_j^\sigma \in T} \mu(p_j^\sigma)$, capturing the amounts of time for which these constraints are satisfied for all cells in T.

It is important to notice that δ, ε and μ are strictly greater than zero: $\delta > 0$ by its definition; $\varepsilon > 0$ since if there is a cell $p_k^{\sigma_i} \in T_i$ such that $s(p_k^{\sigma_i}) = s(p_{k+1}^{\sigma_i})$ then $s(p_{k+1}^{\sigma_i}) = \beta^{\sigma_i} - \mathrm{MJ}^\sigma(s) + kX < \beta^{\sigma_i} - \mathrm{MJ}^\sigma(s) + (k+1)X$, contradicting Lemma 4.3. Finally, note that by Lemma 4.3, and the fact that $\lambda X \geq 1$, we have for any $p_k^{\sigma_i} \in T$,

$$\begin{aligned} s(p_{k+\lambda}^{\sigma_i}) &\geq \beta^{\sigma_i} - J^{\sigma_i}(s) + (k+\lambda)X \\ &= \beta^{\sigma_i} - J^{\sigma_i}(s) + kX + \lambda X \\ &= s(p_k^{\sigma_i}) + \lambda X \\ &\geq s(p_k^{\sigma_i}) + 1. \end{aligned}$$

Since equality can only hold if $p_{k+\lambda}^{\sigma_i} \in T$, in which case $\mu(p_k^{\sigma_i}) = \infty$, we are guaranteed to have $\mu(p_k^{\sigma_i}) > 0$, which implies that $\mu > 0$.

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

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

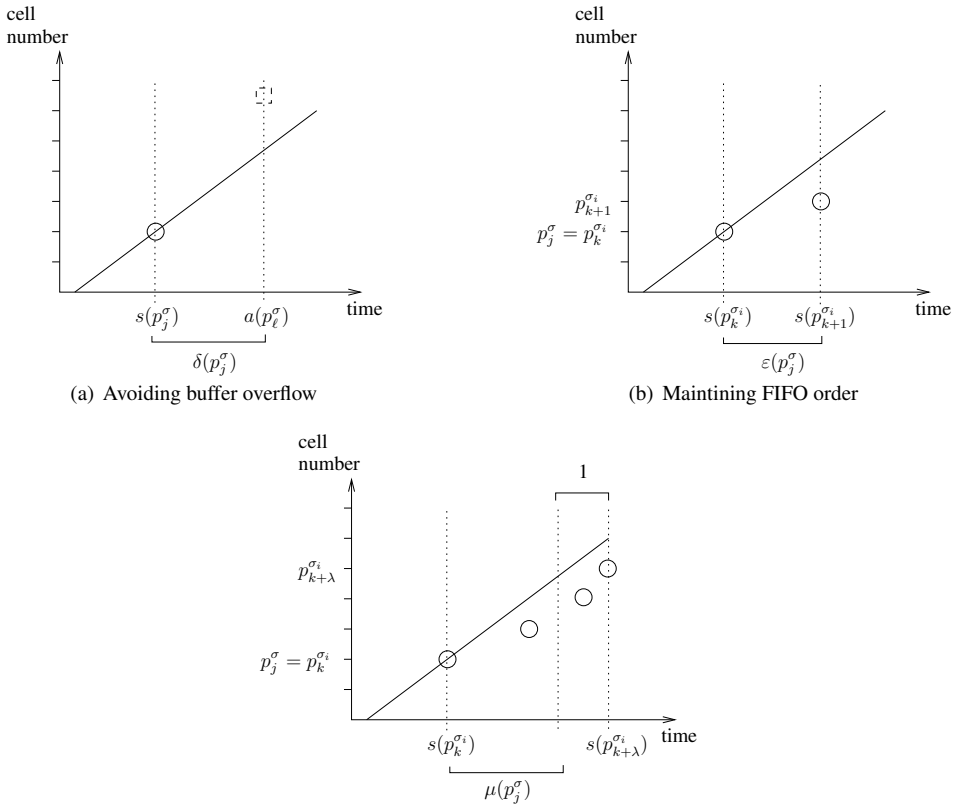
$\rho(\sigma_i)$ comes to capture how far is the rest of the stream from the left margin. Since for any $\sigma_i \in \Sigma$, $J^{\sigma_i}(s) > 0$, then $\sigma_i \setminus T_i$ is not empty and $\rho(\sigma_i) > 0$ and finite. Let $\rho = \min_{\sigma_i \in \Sigma} \rho(\sigma_i)$. It follows that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

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

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

Note that this schedule is well-defined since $\Delta > 0$ and finite.

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



(a) Avoiding buffer overflow

(b) Maintining FIFO order

(c) Maintaining the capacity constraint

Fig. 4. Outline of arrivals and releases defining the values of $\delta(p_j^\sigma)$, $\varepsilon(p_j^\sigma)$ and $\mu(p_j^\sigma)$.

$\Delta \leq \delta(p_j^\sigma)$, 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 $\Delta \leq \varepsilon(p_j^\sigma)$ for every $p_j^\sigma \in T$. In order to prove that s' is capacity-feasible we distinguish between two cases for each $p_k^{\sigma_i} \in T$: If $p_{k+\lambda}^{\sigma_i} \in T$ then both cells are postponed for the same duration, and therefore do not violate the capacity constraint. On the other hand, if $p_{k+\lambda}^{\sigma_i} \notin T$, capacity feasibility is maintained since $\Delta \leq \mu(p_k^{\sigma_i})$.

We conclude the proof by showing that $\mathrm{MJ}^\sigma(s') < \mathrm{MJ}^\sigma(s)$. Consider any $\sigma_i \in \Sigma$, and any $p_k^{\sigma_i}$. If $p_k^{\sigma_i} \in T$ then by the definition of s' and Lemma 4.3, $s'(p_k^{\sigma_i}) = s(p_k^{\sigma_i}) + \Delta/2 \geq \beta^{\sigma_i} - \mathrm{MJ}^\sigma(s) + kX + \Delta/2$. The same holds also for $p_k^{\sigma_i} \notin T$: 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 4.4, $J^{\sigma_i}(s') < J^{\sigma_i}(s)$ for any stream $\sigma_i \in \Sigma$. The jitter of any other stream

remains unchanged, therefore $\mathrm{MJ}^{\sigma}(s') < \mathrm{MJ}^{\sigma}(s)$, contradicting the optimality of s. □

Finally, we conclude this section by showing that there exists a polynomial-time algorithm that finds an optimal schedule for the multi-stream max-jitter problem (or returns NULL if no feasible schedule exists). Algorithm 1 depicts the pseudo-code of this algorithm.

THEOREM 4.9. *There exists a polynomial-time algorithm that finds an optimal schedule for the multi-stream max-jitter problem (if a feasible schedule exists).*

PROOF. Assume a feasible schedule exists. Lemma 4.8 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 (See Algorithm 1, lines 18-19).

It follows that by enumerating over all possible choices of pairs $(p_k^{\sigma_i}, p_l^{\sigma})$, one can find the collection of possible values of the optimal jitter (See Algorithm 1, function OFFLINE).

For every such value J, one can compute the aligned release schedule defined in Corollary 4.6 assuming $\mathrm{MJ}(s) = J$ (See Algorithm 1, function COMPUTESCHEDULE). This schedule satisfies the capacity constraints but may not be B-feasible. However, its B-feasibility can be verified in linear time (See Algorithm 1, function ISFEASIBLE); if this schedule is not B-feasible then Corrolary 4.7 implies that there is no B-feasible schedule attaining jitter J. □

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 [Mansour and Patt-Shamir 2001]. In addition we further extend the model to the case where each outgoing link is associated with a single stream, and has a bounded capacity.

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, for the uncapacitated case, 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.

Algorithm 1 Algorithm OFFLINE

```
1: boolean function IsFeasible(sequence σ, schedule s, buffer size B)
2:     BufferOccupancy ← 0
3:     σ' ← Merge(σ, s)                ▷ for identical values, those of s appear before those of σ
4:     for every e ∈ σ' in increasing order do
5:         if e ∈ σ then
6:             BufferOccupancy ← BufferOccupancy + 1
7:         else                                                          ▷ i.e., e ∈ s
8:             BufferOccupancy ← BufferOccupancy − 1
9:         end if
10:        if BufferOccupancy > B then
11:            return FALSE
12:        end if
13:    end for
14:    return TRUE
15: end function

16: schedule function ComputeSchedule(sequence σ, jitter J)
17:    s ← NULL
18:    for every stream σ_i ∈ σ do
19:        β^{σ_i} ← max_j {a(p_j^{σ_i}) − jX}                    ▷ defines the right margin
20:    end for
21:    for every cell p_j^{σ_i} ∈ σ do
22:        s(p_j^{σ_i}) ← max{a(p_j^{σ_i}), β^{σ_i} − J + jX, s(p_{j−λ}^{σ_i}) + 1}   ▷ For j < λ, s(p_{j−λ}^{σ_i}) ≜ −1
23:    end for
24:    return s
25: end function

26: schedule function Offline(sequence σ, buffer size B)
27:    Array of jitter values MinJitter[M]. Initially all ∞
28:    Array of schedules MinSchedule[M]. Initially all NULL
29:    for every stream σ_i ∈ σ do
30:        β^{σ_i} ← max_j {a(p_j^{σ_i}) − jX}
31:        for every p_j^{σ_i} ∈ σ_i do
32:            for every p_k^σ ∈ σ do
33:                if a(p_k^σ) ≥ a(p_j^{σ_i}) and a(p_k^σ) ≤ β^{σ_i} + jX then
34:                    α^{σ_i} ← a(p_k^σ) − jX
35:                    s ← ComputeSchedule(σ, β^{σ_i} − α^{σ_i})
36:                    if IsFeasible(σ, s, B) and MinJitter[i] > β^{σ_i} − α^{σ_i} then
37:                        MinJitter[i] ← β^{σ_i} − α^{σ_i}
38:                        MinSchedule[i] ← s
39:                    end if
40:                end if
41:            end for
42:        end for
43:    end for
44:    ℓ ← arg min MinJitter
45:    return MinSchedule[ℓ]
46: end function
```

In addition, our offline algorithm can be used to evaluate whether a buffer of a given size can provide a certain jitter guarantee on a given traffic, or whether a larger buffer size is necessary. This implies that when designing the network, one can sample long enough periods of the incoming traffic and employ our offline algorithm to check whether a resource augmentation might be needed.

Sometimes, only few "misbehaved" cells significantly increase the delay jitter. Furthermore, real-life regulators may be allowed to drop cells. Therefore, it is appealing to examine the correlations between buffer size, optimal jitter, and drop ratio. In addition, it is of interest to examine situations in which the different streams share the same outgoing link; that is, the capacity constraint is imposed on the interleaving of streams rather than on each stream separately. We conjecture that this latter problem is NP-hard.

ACKNOWLEDGMENTS

We would like to thank Hagit Attiya, Seffi Naor, Adi Rosén, and Shmuel Zaks for their useful comments on the preliminary version of this paper. We would also like to thank Keren Censor for her suggestions on the current version, as well as the anonymous referees for their helpful comments.

REFERENCES

BORODIN, A. AND EL-YANIV, R. 1998. *Online Computation and Competitive Analysis*. Cambridge University Press.

COMSCORE NETWORKS. 2006. comScore releases August U.S. Video Metrix rankings. Available online at: `http://www.comscore.com/press/release.asp?press=1035`.

DAVIE, B., CHARNY, A., BENNETT, J., BENSON, K., BOUDEC, J. L., COURTNEY, W., DAVARI, S., FIROIU, V., AND STILIADIS, D. 2002. An expedited forwarding PHB (per-hop behavior). RFC 3246. March.

GOLESTANI, S. 1990. A stop-and-go queueing framework for congestion management. In *ACM SIGCOMM*. 8–18.

KALMANEK, C., KANAKIA, H., AND KESHAV, S. 1990. Rate controlled servers for very high-speed networks. In *Proceedings of the Conference on Global Communications (GLOBECOM)*. 12–20.

KERMANI, P. AND KLEINROCK, L. 1979. Virtual cut-through: A new computer communication switching technique. *Computer Networks 3*, 267–286.

KESHAV, S. 1997. *An Engineering Approach to Computer Networking*. Addison-Wesley Publishing Co.

KESLASSY, I., KODIALAM, M., LAKSHMAN, T., AND STILIADIS, D. 2005. On guaranteed smooth scheduling for input-queued switches. *IEEE/ACM Transactions on Networking 13,* 6 (December), 1364–1375.

KOGA, H. 2001. Jitter regulation in an internet router with delay constraint. *Journal of Scheduling 4,* 6, 355–377.

MANSOUR, Y. AND PATT-SHAMIR, B. 2001. Jitter control in QoS networks. *IEEE/ACM Transactions on Networking 9,* 4 (August), 492–502.

PARTRIDGE, C. 1991. Isochronous applications do not require jitter-controlled networks. RFC 1257. September.

SLEATOR, D. AND TARJAN, R. 1985. Amortized efficiency of list update and paging rules. *Communications of the ACM 28,* 2, 202–208.

TANENBAUM, A. 2003. *Computer Networks*, Fourth ed. Prentice Hall.

TASSIULAS, L. 1999. Cut-through switching, pipelining, and scheduling for network evacuation. *IEEE/ACM Transactions on Networking 7,* 1, 88–97.

The ATM Forum 1999. *Traffic Management Specification*. The ATM Forum. Version 4.1, AF-TM-0121.000.

VERMA, D., ZHANG, H., AND FERRARI, D. 1991. Guaranteeing delay jitter bounds in packet switching networks. In *Proceedings of TriComm*. 35–46.

ZHANG, H. 1995. Service disciplines for guaranteed performance service in packet switching networks. *Proceedings of the IEEE 83,* 10 (October), 1374–1396.

ZHANG, H. AND FERRARI, D. 1994. Rate-controlled service disciplines. *Journal of High-Speed Networks 3,* 4, 389–412.

Received Month Year; revised Month Year; accepted Month Year