Flow Theory: an Enhancement

A. Lombardo, G. Morabito, S. Palazzo, G. Schembra
Istituto di Informatica e Telecomunicazioni
University of Catania
V.le A. Doria 6 - 95125 Catania - ITALY
E-mail: (lombardo, gmorabi, palazzo, schembra)@iit.unict.it

Abstract

Flow Theory is a rich, effective theory introduced in order to study real-time network protocols. It is based on discrete mathematics in which a flow of data is represented by an infinite sequence of numbers and is characterized by its upper bounds. Unfortunately up to now Flow Theory has used a traffic characterization that in some cases can be too loose, thus causing an inefficient network utilization. In this paper we enhance Flow Theory by providing it with a more accurate traffic characterization and with all the tools necessary to deal with it. As we will see, this enhancement provide a more accurate performance evaluation and thus better network utilization.

Keywords: traffic modeling,worst-case analysis, deterministic approach, rate-reservation protocols, MPEG.

1 Introduction

Integrated service networks have to support a large number of sessions with different performance requirements, and at the same time optimize resource utilization. In such a scenario it is of crucial importance to achieve a reliable performance evaluation for each session in terms of end-to-end delay and loss rate.

To this end, two approaches are possible: one is stochastic, the other is deterministic.

The *stochastic* approach is based on modeling the traffic from each session by means of stochastic processes such as Markov-based processes (e.g. [10]) or self-similar processes (e.g. [1]), and, in general, leads to methodologies which are too complex to be utilized in real-time procedures such as call admission control (CAC). Moreover, it is extremely hard to verify whether a traffic source is respecting the statistical characterization it declared at the call set-up time, and it is therefore very difficult to implement a suitable policing mechanism.

The *deterministic* approach was introduced by Cruz in [3] [4]. This approach does not try to match the traffic source behavior, but simply wants to bind it, that is, it takes care of the worst case. Obviously by using this approach, as for example in [12] [9] and [6], we can only obtain upper bounds of the quality of service parameters, but the related methodologies for performance evaluation are computationally much simpler than those achieved with the stochastic approach and thus are more suitable for real-time applications.

Flow Theory, based on a deterministic approach, was introduced in [2] with the aim of studying rate reservation protocols in networks of arbitrary topology. In each of these protocols, at the connection set-up time, the user requests suitable quality of service parameters to be guaranteed by the network and characterizes its behavior; then the network evaluates the amount of resources needed to provide the above quality of service and, if they are available, establishes the connection.

The main merit of Flow Theory, as compared to other performance analysis methodologies using a deterministic approach [3] [4] [12] [9] [6], is that it allows real-time protocols, as well as first-come-first-served protocols, to be modeled and studied.

However, in [2] source characterization is given in terms of the highest average arrival rate in intervals of only one given length, and therefore may be too rough in the case of traffic sources characterized by burstiness in several timescales (MPEG, for an example), consequently it could lead to an overestimation of the amount of resources needed by each session and thus to bad network efficiency.

For the above reason, in this paper we provide Flow Theory with a more accurate traffic model which, according to Knightly and Zhang's D-BIND model [9], considers the highest average arrival rate in intervals of several given lengths. Moreover, we develop the tools needed to deal with this new traffic model and finally we prove that by using them we can achieve a more accurate worst-case performance evaluation and thus better network utilization.

The rest of the paper is organized as follows. In Section 2 we report the definition of flow and of its main two properties - i.e. smoothness and uniformity - introduced in [2]. In Section 3 we recall the definition of flow operator and we describe the main specific flow operators. In Section 4 the definition of linear network is reported. In Section 5 we introduce a more accurate traffic characterization and extend Flow Theory in order to deal with it. In Section 6, through a case study, we demonstrate the power of our approach by comparing the upper bounds for the performance parameters achievable using the original Flow Theory with those achievable using our enhancement. Finally in Section 7 we present our conclusions.

2 Flows

A *flow* r is an infinite sequence of nonnegative real numbers $r_0, r_1, r_2, ...$ In a slotted-time environment r_i could represent the number of bits (or cells) from a session entering a network element at the i-th time instant.

A flow r is (m, R)-uniform, in which R is a positive real number, if in any time interval of length m the average value of r is lower than or equal to R; formally:

$$\sum_{i=j}^{j+m-1} r_i \leq m \cdot R \qquad \forall j \in \{0, 1, 2, ...\} \qquad (1)$$

Now, let us divide the temporal axis into contiguous intervals of length m; a flow r is (m, R)-smooth, in which R is a positive real number, if the average value of r is lower than or equal to R in each of the above intervals; formally:

$$\sum_{i=j\cdot m}^{(j+1)\cdot m-1} r_i \leq m \cdot R \qquad \forall j \in \{0, 1, 2, ...\} \qquad (2)$$

Let us explicitly observe that in both cases the mean rate of the flow is at most R.

The relationship between smoothness and uniformity is given by Theorem 1 in [2]:

Theorem 1 : Let r be an (m, R)-smooth flow, and r' be an (m, R)-uniform flow.

1. For any $n, n \geq 1$,

   a) r is $[n, (\lceil n/m \rceil + 1) \cdot (m/n) \cdot R]$-uniform.

   b) r' is $[n, \lceil n/m \rceil \cdot (m/n) \cdot R]$-uniform.

2. For any n multiple of m ,

   a) r is (n, R)-smooth.

   b) r' is (n, R)-uniform.

3. For any S larger than or equal to R,

   a) r is (m, S)-smooth.

   b) r' is (m, S)-uniform.

Corollary 1 : r is $(m, 2 \cdot R)$-uniform and r' is (m, R)-smooth.

The second part of Corollary 1 states that uniformity is a stronger property than smoothness; in fact the first implies the second. Unfortunately the second is easier to assess, as is required in real-time control operations like traffic policing. In fact, enforcing smoothness requires Jumping Window algorithms, while enforcing uniformity requires more complex algorithms such as Exponentially Weighted Moving Average (EWMA) [11] [5]. For this reason, although a traffic characterization in terms of uniformity should permit more efficient resource utilization, we will deal with both smoothness and uniformity.

3 Flow operators

While passing through each of the network elements, traffic flows are reshaped; in order to model this occurrence, in [2] the concept of flow operator is introduced.

In general a *flow operator* is an operator manipulating flows.

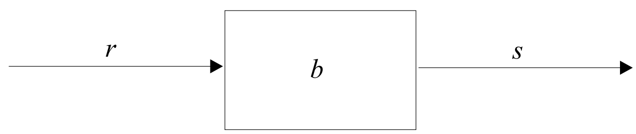

Figure 1. Flow Operator

As in Fig. 1, a flow operator can be represented by an *input flow* r, an *output flow* s, and a third internal flow b which we will refer to as *buffer flow*; at the generic slot i, the flow operator inputs r_i, outputs s_i and stores the remainder in the internal buffer, that is:

$$b_i = b_{i-1} + r_i - s_i \qquad (3)$$

where $b_{-1} = 0$, and in order for $b_i \geq 0$, $s_i \leq b_{i-1} + r_i$ to hold for any i (*flow conservation* property).

The *buffer capacity* B of a flow operator with respect to the input flow r is defined as the highest value taken by b_i. The *delay* D of a flow operator with respect to the input flow r is the smallest nonnegative integer satisfying the following condition:

$$b_i \leq s_{i+1} + s_{i+2} + ... + s_{i+D} \qquad \forall i \in \{0, 1, 2, ...\} \quad (4)$$

In a network environment, both the buffer capacity and the delay of a flow operator can be seen as quality of service (QoS) parameters. Let us explicitly observe that they are defined as upper bounds of both the buffer occupancy and the delay suffered by the bits (or cells) entering a network element.

The relationship between the buffer capacity and the delay of any flow operator is stated by the following theorem:

Theorem 2 : Consider any flow operator whose input is (m, R)-uniform. Let B denote the buffer capacity of this operator, and D denote its delay. Then:

$$B \leq \lceil D/m \rceil \cdot m \cdot R \tag{5}$$

In [2] a number of flow operators are introduced, namely limiters, compactors, expanders, filters, delayers, mergers, splitters and separators.

Let R be a positive real number. An *R-limiter* is a flow operator that at each slot inputs an input flow element and outputs the most it is allowed by the conservation property but no more than R:

$$s_i = \begin{cases} R & \text{if } b_{i-1} + r_i > R \\ b_{i-1} + r_i & \text{if } b_{i-1} + r_i \leq R \end{cases} \tag{6}$$

Let m be a positive integer. An *m-compactor* is a flow operator that accumulates the input flow in its buffer during an interval of m time instants, and then outputs the buffer contents in the first time instant of the next interval. Formally:

$$s_i = \begin{cases} 0 & \text{if } i \bmod m \neq 0 \\ b_{i-1} & \text{if } i \bmod m = 0 \end{cases} \tag{7}$$

Let m be a positive integer. An *m -expander* is a flow operator that may output any value at each time instant provided the flow conservation property is satisfied. In addition, to guarantee a bounded delay, the entire buffer content is transferred to the output flow every m time instants. Formally:

$$s_i = \begin{cases} b_{i-1} + r_i & \text{if } i \bmod m = m-1 \\ (b_{i-1} + r_i) \cdot X_i & \text{if } i \bmod m \neq m-1 \end{cases} \tag{8}$$

where each X_i is a real number in the closed interval $[0, 1]$.

Let R be a positive real number. An *R-filter* is a flow operator that at each time instant may output any value equal to or greater than R, provided that the flow conservation property is not violated. Formally:

$$s_i = \begin{cases} X & \text{if } b_{i-1} + r_i > R \\ b_{i-1} + r_i & \text{if } b_{i-1} + r_i \leq R \end{cases} \tag{9}$$

where X is a real number in the interval $[R, b_{i-1} + r_i]$. To explain better how an R-filter operates, let us consider an R-limiter, and let r be the flow entering both the R-limiter and R-filter. The above condition is respected if at each time instant i the sum of the first i elements of the output flow of the R-filter is greater than or equal to the first i elements of the output flow of the R-limiter. Formally, if we call the buffer flow of the R-filter $b^{(f)}$ and that of the R-limiter $b^{(l)}$:

$$s_i = \max[(b^{(f)}_{i-1} + r_i) \cdot X_i, \quad b^{(f)}_{i-1} + r_i - b^{(l)}_i] \qquad \forall i \in \{0, 1, 2, ...\} \tag{10}$$

where X_i is a real number in the closed interval $[0, 1]$.

A *d-delayer*, where d is a positive integer, is a flow operator that delays its input flow in an arbitrary manner by at most d time instants. A d-delayer can be defined recursively as a $(d-1)$-delayer followed by a 1-delayer. The operation of a 1-delayer is:

$$s_i = r_i \cdot X_i + b_{i-1} \qquad \forall i \in \{0, 1, 2, ...\} \tag{11}$$

where X_i is a real number in the closed interval $[0, 1]$.

Moreover, in [2] in order to be able to construct a flow operator network with an arbitrary topology, some flow operators with multiple inputs or multiple outputs are defined.

A *merger* is a flow operator with two input flows, r and q, and one output flow, s. At each time instant it outputs the sum of its two inputs at that instant. Formally:

$$s_i = q_i + r_i \qquad \forall i \in \{0, 1, 2, ...\} \tag{12}$$

Let X be a real number in the closed interval $[0, 1]$. An *X-splitter* is a flow operator with one input flow r and two output flows, s and t, which at each time instant splits its input between its two outputs with a ratio of X and $(1-X)$. Formally:

$$\begin{aligned} s_i &= X \cdot r_i && \forall i \in \{0, 1, 2, ...\} \\ t_i &= (1 - X) \cdot r_i && \forall i \in \{0, 1, 2, ...\} \end{aligned} \tag{13}$$

A *separator* is a flow operator with one input flow, r, and two output flows, s and t; it splits its input between the two outputs with a ratio chosen arbitrarily at each time instant. Formally:

$$\begin{aligned} s_i &= X_i \cdot r_i && \forall i \in \{0, 1, 2, ...\} \\ t_i &= (1 - X_i) \cdot r_i && \forall i \in \{0, 1, 2, ...\} \end{aligned} \tag{14}$$

where X_i is a real number in the closed interval $[0, 1]$.

4 Linear Networks

A combination of flow operators can be used to model data transmission in a network [2]. For example, in the *Stop-and-Go* protocol, time is partitioned into consecutive periods of equal duration called frames. Each computer involved in a session buffers all the data packets that it receives in one frame and then forwards these packets in the next output frame. The protocol makes no guarantees on how the packets are to be arranged in the output frame; for this reason each computer on the path can be modeled as a sequence of an m-compactor and an m -expander, where m

is the number of time instants constituting a frame. Likewise it could be demonstrated that in *Hierarchical Round-Robin* protocols each computer belonging to a session can be modeled as a sequence of an R-limiter, an m -compactor and an m-expander, where m is the number of time instants in each round and $m \cdot R$ is the maximum number of data packets that the computer can forward along the connection, and that in *Timestamp Protocols* each computer belonging to the path of a session can be represented as a sequence of an R-filter and a $(L/R + L_{max}/C)$-delayer, where L is the maximum packet size of the session, R is the rate reserved by the computer for the session, L_{max} is the maximum packet size over all the sessions whose path includes the computer and C is the rate of the output link.

In order to model sequences of flow operators, in [2] linear networks are introduced. A finite sequence of flow operators $\langle f_0, f_1, ..., f_{n-1}\rangle$ is a *linear network* if for each i, $0 \leq i < n-1$, the output flow of f_i is the input flow of f_{i+1}. The input flow of f_0 is the input flow of the linear network and the output flow of f_{n-1} is the output flow of the linear network.

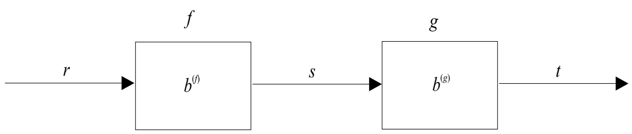

Figure 2. Linear Network $\langle f, g\rangle$

As an example, consider a linear network $\langle f, g\rangle$ of two flow operators f and g, as shown in Fig. 2: the input flow of the network is r, the output flow is t; the buffer flow c of the whole network can be defined as:

$$c_i = c_{i-1} + r_i - t_i \qquad \forall i \in \{0, 1, 2, ...\} \tag{15}$$

where $c_{-1} = 0$.

Once the buffer flow of a linear network is defined, the definitions of its buffer capacity and delay are straightforward and the following theorem can be proved:

Theorem 3 : Let r be the input flow and s the intermediate flow of a linear network $\langle f, g\rangle$ with the operators f and g.

1. if the buffer capacity of f with respect to r is at most $B^{(f)}$, the buffer capacity of g with respect to any intermediate flow s is at most $B^{(g)}$, and the buffer capacity of the network $\langle f, g\rangle$ with respect to r is C, then:

$$C \leq B^{(f)} + B^{(g)} \tag{16}$$

2. if the delay of f with respect to r is at most $D^{(f)}$, the delay of g with respect to any intermediate flow s is at most $D^{(g)}$, and the delay of the network $\langle f, g\rangle$ with respect to r is D, then:

$$D \leq D^{(f)} + D^{(g)} \tag{17}$$

For a particular linear network the following results were obtained in [2]:

Theorem 4 : Let f be an m-expander, and g and h be m-compactors. If the input flow to the linear network $\langle f, g\rangle$ is the same as the input flow of h, then the output flow of $\langle f, g\rangle$ is the same as the output flow of h.

Theorem 5 : Let f and g be two R-filters. If the input flow of f is the same as the input flow of the linear network $\langle f, g\rangle$, then every output flow of f is an output flow of $\langle f, g\rangle$ and vice versa.

Theorem 6 : Let h be a d-delayer, and let f and g be R-filters. If the linear networks $\langle h, g\rangle$ and $\langle f, h\rangle$ have the same input flow:

1. any output flow of $\langle h, g\rangle$ is also an output flow of $\langle f, h\rangle$,

2. the buffer capacity of $\langle f, h\rangle$ is at most the buffer capacity of f plus $d \cdot R$.

5 Extending Flow Theory

In this section we extend Flow Theory in order to provide for more parsimonial resource management. We first introduce a new traffic characterization in which traffic bounds are given in intervals of several lengths, then we adapt the results obtained in [2] to this more accurate traffic characterization.

Let $m_1, m_2, ..., m_J$ be J positive integers such that $m_1 < m_2 < ... < m_J$, and let $R_1, R_2, ..., R_J$ be J positive real numbers. We will say that a flow r is $\{(m_j, R_j)|j = 1, 2, ..., J\}$-smooth if it is (m_j, R_j)-smooth for each j. Analogously it will be $\{(m_j, R_j)|j = 1, 2, ..., J\}$-uniform if it is (m_j, R_j)-uniform for each j. Let us stress that this second characterization in terms of uniformity can be regarded as a discrete-time version of the D-BIND model [9].

Let $A^{(r)}(n_1, n_2)$ with $n_1 < n_2$ be the sum of the elements of a flow r from the time instant n_1 to the time instant n_2:

$$A^{(r)}(n_1, n_2) = r_{n_1} + r_{n_1+1} + ... + r_{n_2} \tag{18}$$

As in [9], we define a monotonic increasing function $b^{(r)}(n)$ as a *deterministic flow constraint function* of r if $b^{(r)}(n)$ is a bounding function of the flow r, that is, $A^{(r)}(n_1, n_1 + n - 1) \leq b^{(r)}(n), \forall n_1$ and n.

As we will see later, $b^{(r)}(n)$ plays a crucial role in performance evaluation; so, although there are an infinity of functions bounding a flow, the lower the $b^{(r)}(n)$ values, the more accurate the model and thus the greater the resource utilization.

It could be proved that if a flow r is $\{(m_j, R_j)|j = 1,2,...,J\}$-smooth it can be bounded by the function:

$$b^{(r)}(n) = c(n, m_1, ..., m_J, R_1, ..., R_J) \quad (19)$$

where:
$c(n, m_1, ..., m_j, R_1, ..., R_j) =$

$$= \begin{cases} 0 & \text{if } n \leq 0 \\ m_1 \cdot R_1 & \text{if } n = 1 \\ 2 \cdot m_1 \cdot R_1 & \text{if } n \in]1, m_1+1] \\ \min(2 \cdot m_i \cdot R_i, & \\ \quad c(n, m_1, ..., m_{i-1}, & \text{if } n-1 \in]m_{i-1}, m_i] \\ \quad R_1, ..., R_{i-1})) & \quad \text{and } i = 2, ..., j \\ \min_{l \in [0, \lceil n/m_j \rceil + 1]}(l \cdot m_j \cdot R_j & \\ \quad +c(n-1+\delta(l) - (l-1+ & \\ \quad +\delta(l)) \cdot m_j, m_1, ..., m_{j-1}, & \\ \quad R_1, ..., R_{j-1})) & \text{if } n > m_j + 1 \end{cases} \quad (20)$$

and $\delta(l)$ indicates the Dirac function in the discrete-time domain defined as:

$$\delta(l) = \begin{cases} 1 & \text{if } l = 0 \\ 0 & \text{otherwise} \end{cases} \quad (21)$$

Moreover, if a flow r is $\{(m_j, R_j)|j = 1,2,...,J\}$-uniform it can be bounded by the function:

$$b^{(r)}(n) = d(n, m_1, ..., m_J, R_1, ..., R_J) \quad (22)$$

where:
$d(n, m_1, ..., m_j, R_1, ..., R_j) =$

$$= \begin{cases} 0 & \text{if } n \leq 0 \\ m_1 \cdot R_1 & \text{if } n \in]0, m_1] \\ \min_{l \in \{0,1\}}(l \cdot m_i \cdot R_i + d(n+ & \\ \quad -l \cdot m_i, m_1, ..., m_{i-1}, & \\ \quad R_1, ..., R_{i-1})) & \text{if } n \in]m_{i-1}, m_i] \\ \min_{l \in [0, \lceil n/m_j \rceil]}(l \cdot m_j \cdot R_j + & \\ \quad d(n - l \cdot m_j, m_1, ..., m_{j-1}, & \\ \quad R_1, ..., R_{j-1})) & \text{if } n > m_j \end{cases} \quad (23)$$

Once we have bounded flows with respect to this new characterization, the following theorems, extending the results presented in [2], can be proved:

Theorem 7 : If the input flow r of an S-limiter is $\{(m_j, R_j)|j = 1,2,...,J\}$-smooth (or uniform), then

1. The output flow is $(1, S)$-uniform,
2. The output flow is $\{(m_j, R_j)|\forall j \leq J : R_j \leq S\}$-smooth (or uniform),
3. The buffer capacity is at most $[\max_n(b^{(r)}(n) - n \cdot S)]$
4. The delay is at most: $\lceil[\max_n(b^{(r)}(n) - n \cdot S)]/S\rceil$

where $b^{(r)}(n)$ is the deterministic flow constraint of the flow r obtained by using (19) and (20) (or (22) and (23)).

Theorem 8 : If the input flow r of an n-compactor is $\{(m_j, R_j)|j = 1,2,...,J\}$-smooth (or uniform):

1. The output flow is $(n, b^{(r)}(n)/n)$-uniform
2. The buffer capacity is at most $b^{(r)}(n)$,
3. The delay is at most n.

where $b^{(r)}(n)$ is the deterministic flow constraint of the flow r.

Theorem 9 : If the input flow r of an n-expander is $\{(m_j, R_j)|j = 1,2,...,J\}$-smooth (or uniform):

1. The output flow is $(n, b^{(r)}(n)/n)$-smooth
2. The buffer capacity is at most $b^{(r)}(n)$,
3. The delay is at most n.

where $b^{(r)}(n)$ is the deterministic flow constraint of the flow r

Theorem 10 : If the input flow r of an S-filter is $\{(m_j, R_j)|j = 1,2,...,J\}$-smooth (or uniform), then

1. The buffer capacity is at most $\max_n(b^{(r)}(n) - n \cdot S)$
2. The delay is at most $\lceil[\max_n(b^{(r)}(n) - n \cdot S)]/S\rceil$

Theorem 11 : For a merger if an input flow is $\{(m_j, R_j)|j = 1,2,...,J\}$-smooth (or uniform) and the other is $\{(m_j, S_j)|j = 1,2,...,J\}$-smooth (or uniform), then the output flow is $\{(m_j, R_j + S_j)|j = 1,2,...,J\}$-smooth (or uniform).

Theorem 12 : For an X-splitter, if the input flow is: $\{(m_j, R_j)|j = 1,2,...,J\}$-smooth (or uniform), then the first output flow is $\{(m_j, X \cdot R_j)|j = 1,2,...,J\}$-smooth (or uniform) and the other is $\{(m_j, (1-X) \cdot R_j)|j = 1,2,...,J\}$-smooth (or uniform).

Theorem 13 : For a separator, if the input flow is $\{(m_j, R_j)|j = 1,2,...,J\}$-smooth (or uniform), then both the output flows are $\{(m_j, R_j)|j = 1,2,...,J\}$-smooth (or uniform).

6 Case study

In this section we demonstrate that our extension to Flow Theory is very useful in cases in which the traffic sources we are studying are characterized by burstiness on multiple time scales. As an example we will deal with an MPEG-video source which, because of the particular coding technique, gives a typical case of traffic with burstiness on multiple scales.

The basic idea behind MPEG is, in fact, to remove spatial redundancy within a video frame and temporal redundancy between subsequent video frames [8] [7] and, as a consequence, the coder output is a deterministic periodic sequence called *Group of Pictures* (GoP) achieved with three types of coded frames:

- *I-frames*, which are coded using only information present in the picture itself, in order to provide potential random access points in compressed video sequences. The coding is based on the discrete-cosine transform according to the JPEG coding technique;
- *P-frames*, which are coded using a coding algorithm similar to the one used for I-frames, but with the addition of motion compensation with respect to the previous I- or P- frame (forward prediction);
- *B-frames*, which are coded with motion compensation with respect to the previous I- or P-frame, and the next I- or P-frame, or an interpolation between them (bidirectional prediction).

A widely-used GoP structure, which we use in this case study, is constituted by 12 frames alternated as follows: *IBBPBBPBBPBB* . Typically, I-frames require more bits than P-frames, while B-frames have the lowest bandwidth requirement. For this reason the burstiness of an MPEG-video traffic source strongly depends on the observation interval. In fact in an interval as long as a GoP the presence of three P-frames and eight B-frames will smooth the peak due to the I-frame. If we consider an interval as long as the time covered by three frames, the possible presence of an I-frame will be smoothed only by two B-frames and we will therefore have an higher average value than in the case of GoP long intervals.

For these reasons, an MPEG-video traffic trace can be characterized by its upper bounds in GoP-long intervals, in three-frame-long intervals and in one-frame-long intervals.

Let us apply the above source characterization to an MPEG-video trace of ”*The Simpsons*”.

Along with the definition in Section 5, the *Simpsons* trace is: $\{(1\cdot\Delta, 148496), (3\cdot\Delta, 83.978), (12\cdot\Delta, 69.824)\}$-smooth, where Δ indicates the number of time instants covered by a frame interval. In Fig. 3 we show the deterministic flow constraint function evaluated as in Section 5.

In Figs. 4 and 5, we show the upper bounds for the buffer capacity and the delay respectively, achieved using the results we obtained in the last section for an R-limiter versus the value of R (solid lines), compared with those obtained using the results presented in [2] (dashed lines).

As can be observed by using our extension to Flow Theory, for the same value of R we get smaller upper bounds for delay and buffer capacity: this means that we know that less bandwidth is needed to guarantee a certain maximum

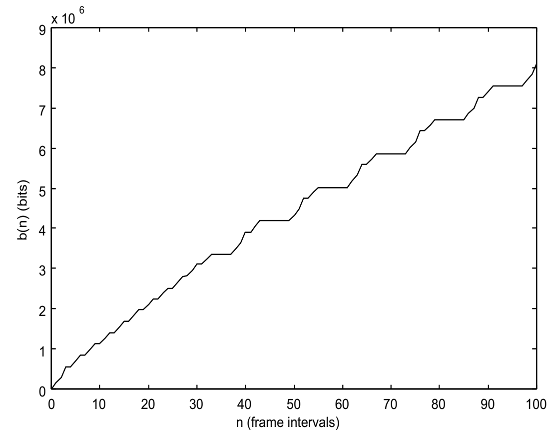

Figure 3. Deterministic constraint flow function

buffer capacity and delay in the session and therefore better resource utilization can be achieved.

Analogously, in Figs. 6 and 7 we compare the upper bounds for the buffer capacity achievable using our extension (solid lines) and using [2] (dashed lines) for an m-compactor and an m-expander, respectively, when m varies. There again, by using our extension smaller values are found for the upper bounds of the buffer capacity. We do not present figures referring to delay because by using the two methodologies for both the m-compactor and the m-expander the maximum delay is always m.

Let us explicitly observe that in each of the above figures all the values for the upper bounds related to the results in [2] have been evaluated as the minimum of the upper bounds achievable by considering the three traffic flow parameter characterizations separately, namely, $(1\cdot\Delta, 148496)$-smooth, $(3\cdot\Delta, 83.978)$-smooth and $(12\cdot\Delta, 69.824)$-smooth.

7 Conclusions

In this paper, in order to achieve efficient resource utilization, we have enhanced Flow Theory, extending its results in a framework in which a more accurate traffic model is used.

More specifically, we have refined the source model by characterizing traffic in terms of smoothness and uniformity in more than one interval, and we have provided the theory with all the tools necessary to study real-time protocols for networks with an arbitrary topology when the above traffic characterization is used.

Through a case study we have shown that by using our extension it is possible to achieve smaller upper bounds for

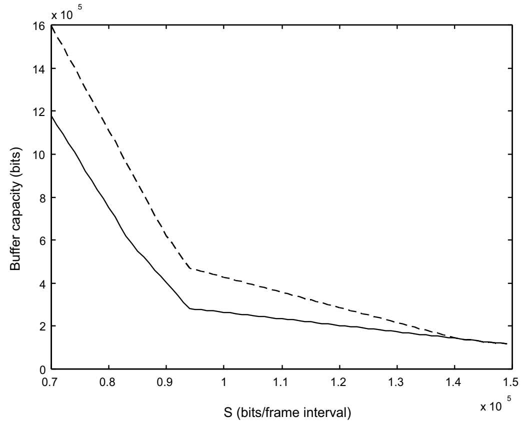

Figure 4. Upper bounds for the buffer capacity of an R-limiter versus the value of R achievable using our extension (solid line) and [2] (dashed line)

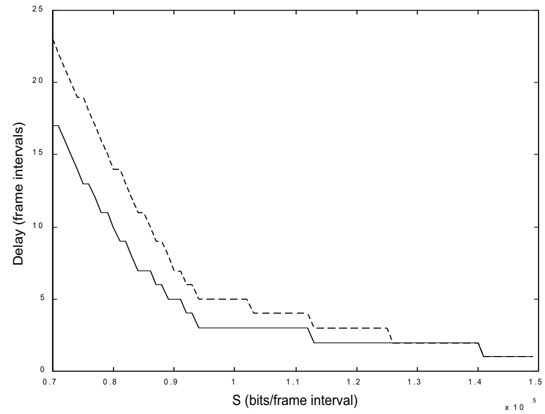

Figure 5. Upper bounds for the delay of an R-limiter versus the value of R achievable using our extension (solid line) and [2] (dashed line)

quality of service parameters, such as buffer capacity and delay, than those achievable using the theory presented in [2], thus permitting better resource management.

References

[1] J. Beran, R. Sherman, M. S. Taqqu, and W. Willinger. Long-range dependence in variable-bit-rate video traffic. *IEEE Transactions on Communication*, 43(2/3/4):1566–1579, 1995.

[2] J. A. Cobb and M. G. Gouda. Flow theory. *IEEE/ACM Transactions on Networking*, 5(5):661–674, Oct. 1997.

[3] R. L. Cruz. A calculus for network delay, part I: Network elements in isolation. *IEEE Transactions on Information Theory*, 37(1):114–131, Jan. 1991.

[4] R. L. Cruz. A calculus for network delay, part II: Network analysis. *IEEE Transactions on Information Theory*, 37(1):132–141, Jan. 1991.

[5] L. Dittmann, S. B. Jacobsen, and K. Moth. Flow enforcement algorithms for ATM networks. *IEEE Journal on Selected Areas in Communications*, 9(3):343–350, Apr. 1991.

[6] L. Georgiadis, R. Guerin, and A. Parekh. Optimal multiplexing on a single link: delay and buffer requirements. *IEEE Transactions on Information Theory*, 43(5):1518–1536, Sept. 1997.

[7] D. I. S. ISO/IEC DIS 13818-2. *Coding of moving picture and associated audio Part. 2, Video*. 1993.

[8] I. S. ISO/IEC IS 11172-2. *Coding of moving picture and associated audio for digital storage media up to 1.5 Mbit/s Part. 2, Video*. 1993.

[9] E. W. Knightly and H. Zhang. D-BIND: an accurate traffic model for providing qos guarantees to VBR traffic. *IEEE/ACM Transactions on Networking*, 5(2):219–231, Apr. 1997.

[10] A. Lombardo and G. Schembra. An analytical paradigm to compare routing strategies in an atm multimedia environment. *IEEE/ACM Transactions on Networking*, 5(6):958–969, Dec. 1997.

[11] E. Rathgeb. Modeling and performance comparison of policing mechanisms for ATM networks. *IEEE Journal on Selected Areas in Communications*, 9(3):325–334, Apr. 1991.

[12] D. E. Wrege, E. W. Knightly, H. Zhang, and J. Liebeherr. Deterministic delay bounds for VBR video in packet-switching networks: fundamenatl limits and practical trade-offs. *IEEE/ACM Transactions on Networking*, 4(3):352–362, June 1996.

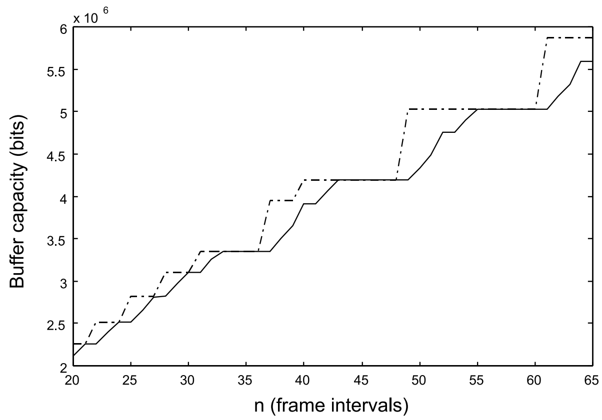

Figure 6. Upper bounds for the buffer capacity of an m-compactor versus the value of m achievable using our extension (solid line) and [2] (dashed line)

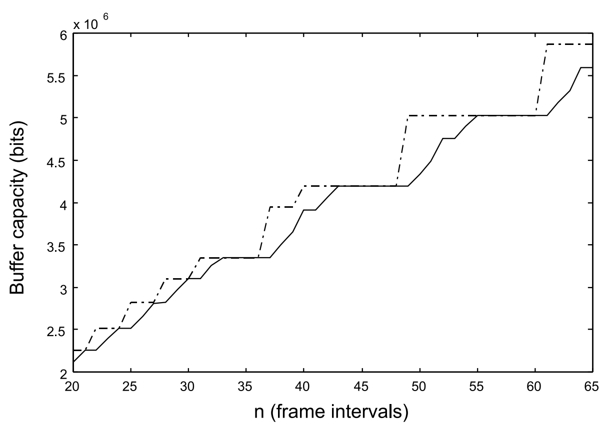

Figure 7. Upper bounds for the buffer capacity of an m-expander versus the value of m achievable using our extension (solid line) and [2] (dashed line)