

# Analysis of Cell Spacing on a TCP Flow Mapped onto an ATM Connection

Fabrice Guillemin<sup>1</sup>, Jacqueline Boyer<sup>1</sup>, and Olivier Dugeon<sup>1</sup>,  
and Christophe Mangin<sup>2</sup>

<sup>1</sup> France Télécom, CNET/DAC,

2 Avenue Pierre Marzin, F-22307 Lannion Cedex

<sup>2</sup> ITE/TCL, Mitsubishi Electric, Immeuble Germanium,  
80, Avenue des Buttes de Cosmes, F-35700 Rennes

**Abstract.** We show in this paper that the cell spacing technique can be used to regulate TCP when the traffic of a TCP connection is carried over an ATM connection. For this purpose, we consider a very simple traffic configuration where two workstations exchange TCP data traffic segmented into ATM cells and transmitted over an ATM link traversing a cell spacer. We precisely analyze how the buffer content of the cell spacer evolves over time. The results obtained via very simple mathematical models are then compared with those of a real TCP data transfer experiment.

## 1 Introduction

The so-called transmission control protocol (TCP) is today certainly the transfer protocol the most implemented for data transmission in computer communications. Whereas this situation has been known for a long time in local area networks, it begins to appear for wide area networks. Indeed, with the emergence of the Web in the mid'90s and in parallel with the large scale deployment of the Internet, TCP generates today a huge volume of traffic over wide area backbone networks (see for instance [9] for statistics on Internet traffic).

To estimate the quality of service offered to a data transfer application and hence, the quality of service perceived by a customer, it is essential to model the stochastic behavior of a TCP flow through a wide area network. This task is however made extremely difficult by different factors. First, it is important to note that whatever be the transfer technology chosen, the behavior of TCP greatly depends on the specific implementation considered and in particular on the way acknowledgement segments are returned by the destination to the source. It is hence difficult to draw general conclusions, which hold for all TCP implementations, especially when the results critically depend on the traffic configuration (number of users, number of routers, transmission speed of the network links, etc.).

Different techniques may be envisaged to carry TCP/IP traffic over a wide area network, for instance directly over the transmission layer (IP over SDH or WDM) or by using a connection oriented transfer technology, such as ATM or

Frame Relay. In this paper, we focus on the transport of TCP/IP traffic over ATM. A major difficulty for taking benefit of the high quality transfer of ATM is that there is no direct link between the windowing flow control mechanism at the TCP level and the traffic control procedures at the ATM layer, which are cell oriented (e.g., leaky buckets for traffic parameter enforcement, violation tagging, congestion notification, etc.). It hence appears that analytical models for capturing the dynamics of TCP/IP over ATM in complex traffic configurations seem to be out of reach.

In spite of this difficulty, several simulation studies for the transport of TCP traffic over an ATM connection using a statistical ATM transfer capability have been carried out in the technical literature. It actually appears that the adequation of the ATM transfer capabilities to the transport of TCP critically depends on the fine tuning of control parameters (see [1] for the statistical bit rate capability, [7] for the unspecified bit rate capability with early or partial packet discard, [2] for the guaranteed frame rate capability, [5] for the available bit rate capability).

The different control mechanisms studied so far in the technical literature try to regulate TCP via packet discard. Instead, we show in this paper that a reliable method of monitoring TCP dynamics while guaranteeing no loss is to offer sufficient buffering at network access and to control TCP dynamics via queueing. Specifically, working on the principle that TCP can adapt to any constant bit rate pipe, for which transfer delays have small magnitude variations, we propose in this paper to use the cell spacing technique to regulate TCP traffic carried over an ATM connection.

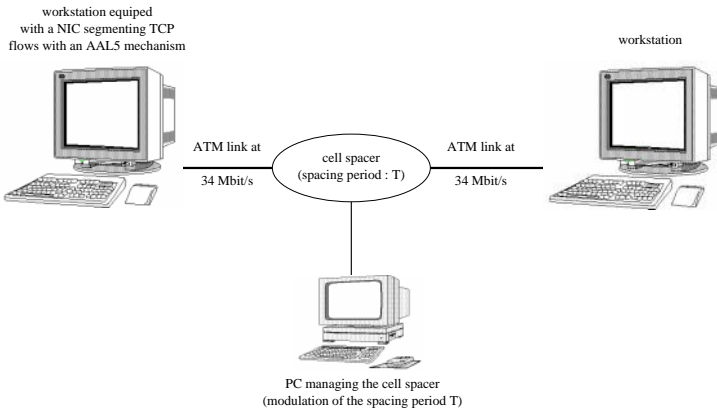
The technique explored in this paper for regulating TCP relies on ATM as transfer technology, but other techniques working directly at the TCP level have also been investigated in the literature (e.g., the Packeteer proposal [6]). Finally, let us note that further techniques consists of delaying acknowledgement segments in the destination terminal so as to limit the amount of data transmitted by the source (see [4] for instance). In that case, we could say that TCP regulation is performed in the backward direction, whereas packet shaping performs the same task on the forward direction.

The organization of this paper is as follows: In Section 2, the reference traffic configuration considered to analyze the impact of cell spacing on a TCP connection is described. The evolution of the buffer content of the cell spacer is studied in Section 3 for the slow start period and in Section 4 in the congestion avoidance regime. We then examine the case in Section 5 when acknowledgment segments are delayed, i.e., an acknowledgement is returned by the destination only when a certain number of packets have been received (namely two in the implementation analyzed). In Section 6, we compare the results obtained via the previous simple mathematical models with experimental data. Some concluding remarks are presented in Section 7. In this paper, we do not detail the description of TCP (see reference books like that by Stevens [8] for an exhaustive presentation of the TCP/IP protocol stack).

## 2 Reference Configuration and Notation

Let us consider the configuration depicted in Figure 1 and representing an idealized TCP data transfer between two workstations. A (full duplex) TCP connection is established between two identical workstations, connected to each other via an ATM link with bit rate  $\mu = 34$  Mbit/s. TCP traffic is segmented into cells in the terminals by using an AAL 5 mechanism and the resulting cell stream is transmitted over a bidirectional ATM connection.

The peak cell rate at the output of the source workstation is denoted by  $h$ . Let  $w$  be the maximum TCP window size, negotiated at the establishment of the TCP connection and expressed in segments; we assume that  $w$  is constant over time and that TCP segments encapsulated in IP packets have all the same size (corresponding to the maximum segment size), giving rise to  $\sigma$  cells.



**Fig. 1.** Experiment for the regulation of TCP via cell spacing

Let  $\delta$  denote the time for a segment issued by the source workstation to be acknowledged by the destination, that is, the corresponding acknowledgement segment is completely received by the source workstation, when the cell spacer is inactive.  $\delta$  depends on the propagation time along the ATM link, the processing time in the receiving workstation, and the bit rate of the reverse ATM connection, since an acknowledgement message gives rise, in general, to 2 cells.  $\delta$  is assumed to be a constant.

In the following, we analyse the transfer of data from one workstation (the source) to the other workstation (the destination). The cell spacer can be modeled as a single server queue fed with the cell stream generated by the source. The service rate is denoted by  $c < h$  and is equal to the spacing rate, which can be modulated over time via a PC. The time to serve a segment is  $\sigma/c$ . Hence, the round trip time  $\delta$  should be replaced with  $\Delta = \delta + \sigma/c$ .

### 3 Analysis of the Slow Start Period

Let us first analyse the slow start period; the slow start threshold is denoted by  $b$  (expressed in segments giving rise to  $b\sigma$  cells). Let  $w_0$  be the initial congestion window size (also expressed in segments and corresponding to  $w_0\sigma$  cells). We assume that  $c\Delta/(w_0\sigma) > 1$ , since  $w_0$  is in general set to a very small value (typically one segment giving rise to  $\sigma$  cells).

The evolution of the buffer content of the cell spacer over time depends on the different parameters  $\Delta$ ,  $w_0$ ,  $\sigma$ ,  $h$ ,  $c$ , and  $b$  and on the fact that the buffer of the cell spacer empties before the slow start threshold is reached or not. Despite this complexity, the outstanding property of the system is that it is possible to determine the content of the buffer of the cell spacer at the end of the slow start period, namely at time  $(n_1 + 1)\Delta$ , where

$$n_1 \stackrel{\text{def}}{=} \lceil \log_2(b/w_0) \rceil, \quad (1)$$

the slow start threshold being reached over the time interval  $[n_1\Delta, (n_1 + 1)\Delta]$ . Let moreover  $n_0$  be defined by

$$n_0 \stackrel{\text{def}}{=} \lceil \log_2(c\Delta/w_0\sigma) \rceil. \quad (2)$$

When  $n_0 \leq n_1$ ,  $[n_0\Delta, (n_0 + 1)\Delta]$  is the first time interval over which the buffer does not empty before the slow start threshold is reached. We can specifically state the following result.

**Theorem 1.** *At the end of the slow start period, the buffer content of the cell spacer oscillates between the values*

$$\mathcal{V}_b^- = (b\sigma - c\Delta)^+ \text{ and } \mathcal{V}_b^+ = (b\sigma - c\Delta)^+ + \sigma \left(1 - \frac{c}{h}\right). \quad (3)$$

The oscillating regime starts at time  $t_b$  defined by

$$t_b = (\nu + 1)\Delta + [(b - 2^\nu w_0)^+ + \mathcal{K}] \frac{\sigma}{c}, \quad (4)$$

where

$$\nu = \min\{n_0, n_1\} \quad (5)$$

$$\mathcal{K} = \left[ (b - 2^\nu w_0) \frac{2c - h}{h - c} + 1 + \frac{ch}{\sigma(h - c)} \left( \frac{2^\nu w_0 \sigma}{h} - \Delta \right)^+ \right]^+. \quad (6)$$

*Proof.* From the very definition of the slow start mechanism, a new busy period for the cell spacer buffer begins at time  $n\Delta$  as long as  $n < n_0$ , which condition means that the buffer empties over the time interval  $[n\Delta, (n + 1)\Delta]$ . From time  $n\Delta$ , an acknowledgement packet is received by the source every  $\sigma/c$  time units. For each acknowledgment packet received, the source transmits two segments during a time interval of  $2\sigma/h$ .

If  $2c \leq h$ , the buffer content increases by  $\sigma$  for each acknowledgement packet received by the source. It follows that the maximum value of the buffer content over the time interval  $[n\Delta, (n + 1)\Delta]$ ,  $n \leq \min\{n_0, n_1\}$ , is

$$2^{n-1} w_0 \sigma + \sigma(1 - 2c/h), \quad (7)$$

since two segments of  $\sigma$  cells arrive at the peak rate  $h$  over each time interval  $[n\Delta + k\sigma/c, n\Delta + (k + 1)\sigma/c]$  for  $k = 0, \dots, 2^{n-1}w_0 - 1$ . Thus, the buffer content increases by  $\sigma$  at the end of each time interval  $[n\Delta + k\sigma/c, n\Delta + (k + 1)\sigma/c]$  for  $k = 0, \dots, 2^{n-1}w_0 - 2$  and by  $2\sigma(1 - c/h)$  at the end of the last time interval  $[n\Delta + (2^{n-1} - 1)\sigma/c, n\Delta + 2^{n-1}\sigma/c]$ .

When  $h < 2c$ , the buffer content linearly increases up to the value

$$2^n w_0 \sigma (1 - c/h). \tag{8}$$

To summarize, we can state that when  $n < \nu \stackrel{\text{def}}{=} \min\{n_0, n_1\}$ , the buffer content empties on each time interval  $[n\Delta, (n + 1)\Delta]$  and the maximum buffer content is given by (8) when  $2c > h$  and (7) when  $2c \leq h$ , respectively.

Now, we examine what happens on the time interval  $[n\Delta, (n + 1)\Delta]$  when  $n \geq \nu$ . The two cases  $n_0 \geq n_1$  and  $n_0 < n_1$  must be distinguished because they lead to different behaviors of the cell spacer buffer. In the following, for the sake on conciseness, we analyse only the first case ( $n_0 \geq n_1$ ); similar computations can be carried out when  $n_0 < n_1$ .

When  $n_0 \geq n_1$ , the buffer of the cell spacer may not be empty at the end of the time interval  $[n_1\Delta, (n_1 + 1)\Delta]$  over which the slow start threshold is reached. If  $h > 2c$ , the buffer content increases by  $\sigma$  for the first  $(b - 2^{n_1-1}w_0 - 1)$  acknowledged segments and by  $2\sigma(1 - c/h)$  for the last segment, for which the congestion window hits the slow start threshold at time

$$\tau_b = n_1\Delta + (b - 1 - 2^{n_1-1}w_0) \frac{\sigma}{c}. \tag{9}$$

Then, since an acknowledgement is received by the source every  $\sigma/c$  time units, the buffer content oscillates between the values

$$v_b^- = (b - 2^{n_1-1}w_0)\sigma \text{ and } v_b^+ = (b - 2^{n_1-1}w_0)\sigma + \sigma \left(1 - \frac{c}{h}\right), \tag{10}$$

until the whole congestion window has been completely acknowledged. After this oscillating period, the buffer content goes to the value  $(b\sigma - c\Delta)^+$ .

When  $2c > h$ , the slow start threshold is still reached at time  $\tau_b$  defined by equation (9). At that time, the congestion window size is equal to  $b$  segments and  $2(b - 2^{n_1-1}w_0)$  segments can be transmitted over the time interval  $[n_1\Delta, \tau_b]$ . At time  $\tau_b + k\sigma/c$ ,  $k \geq 0$ , the source has received credits to transmit, at the peak rate  $h$ ,  $2(b - 2^{n_1-1}w_0) + k$  segments over the time interval  $(\tau_b - n_1\Delta) + k\sigma/c$ . It follows that the buffer content linearly increases as long as

$$k \frac{\sigma}{c} + (\tau_b - n_1\Delta) \leq [2(b - 2^{n_1-1}w_0) + k - 1] \frac{\sigma}{h},$$

that is,

$$k \leq k_0 \stackrel{\text{def}}{=} \left\lceil (b - 2^{n_1-1}w_0) \frac{2c - h}{h - c} + 1 \right\rceil$$

The buffer content ramps up to the value  $[2(b - 2^{n_1-1}w_0) + k_0]\sigma(1 - c/h)$  and then oscillates between the values  $v_b^-$  and  $v_b^+$  defined by equation (10), before going to the value  $(b\sigma - c\Delta)^+$ .

To summarize, when  $n_1 \leq n_0$ , at the end of the slow start period, the buffer content oscillates between the values

$$\mathcal{V}_b^- = (b\sigma - c\Delta)^+ \text{ and } \mathcal{V}_b^+ = (b\sigma - c\Delta)^+ + \sigma \left(1 - \frac{c}{h}\right) \tag{11}$$

in the case  $h < 2c$  as well as  $h > 2c$ . These expressions are consistent with equation (3) when  $n_1 \leq n_0$ . The oscillating regime begins at time  $t_b = (n_1 + 1)\Delta$ , which expression is consistent with equation (4). This completes the proof.

### 4 Analysis of the Congestion Avoidance Period

After the slow start period, the TCP window flow control enters the congestion avoidance regime (when, of course,  $b < w$ ). The congestion window  $cwnd$  increases by  $MSS * MSS / cwnd$  each time an acknowledgement packet is received, up to the the maximum window size  $w$ . To simplify the discussion, we assume in this paper that the congestion window increases by one segment when all the segments of the current congestion window have been acknowledged. Two segments are then sent by the source at the instant when this event occurs. This allows us to greatly simplify the mathematical computations.

Let

$$m_0 = \min\{m : (m + b)\sigma \geq c\Delta\}. \tag{12}$$

To avoid painful computations, we moreover make the following reasonable assumption:

$$(C_1) \quad \tau_b + (b - 1)\frac{\sigma}{c} > t_b$$

where  $t_b$  is defined by (4) and  $\tau_b$  is the time at which the congestion window hits the slow start threshold, given by (9) if  $n_1 \leq n_0$  and  $\tau_b = (n_0 + 1)\Delta + (b - 2^{n_0}w_0 - 1)\sigma/c$  in the case  $n_1 > n_0$ , respectively. This condition means that the buffer content begins to oscillate before the window of  $b$  segments is completely acknowledged.

**Theorem 2.** *At the end of the congestion avoidance period, the buffer content of the cell spacer oscillates between the values:*

$$\mathcal{V}_w^- = [w\sigma - c\Delta]^+ \text{ and } \mathcal{V}_w^+ = [w\sigma - c\Delta]^+ + \sigma \left(1 - \frac{c}{h}\right). \tag{13}$$

*Proof.* For the sake of clarity we consider the cases  $h < 2c$  and  $h > 2c$  separately.

*Case  $h > 2c$*  If  $h > 2c$ , the buffer content of the cell spacer increases by  $2\sigma(1 - c/h)$  when the congestion window increases by one segment.

If  $b\sigma \geq c\Delta$  (and then  $m_0 = 0$ ), the buffer of the cell spacer never empties and the buffer content oscillates between two values over some time intervals. Specifically, when the congestion window size is between  $b + m$  and  $b + m + 1$ ,

$m = 0, \dots, w - b - 1$ , the buffer content oscillates over a time interval of length  $(b + m - 1)\sigma/c$  between the values

$$b\sigma - c\Delta + m\sigma \quad \text{and} \quad b\sigma - c\Delta + m\sigma + \sigma \left(1 - \frac{c}{h}\right). \tag{14}$$

Two consecutive time intervals are separated by a transition interval of length  $\sigma/c$ , where the buffer content increases before starting an oscillation period. The buffer content eventually oscillates between the values

$$\mathcal{V}_w^- = w\sigma - c\Delta \quad \text{and} \quad \mathcal{V}_w^+ = w\sigma - c\Delta + \sigma \left(1 - \frac{c}{h}\right), \tag{15}$$

when the maximum window size is reached. This expression is consistent with equation (13).

When  $b\sigma < c\Delta$ , the buffer of the cell spacer empties over the time intervals  $[(n_1 + m)\Delta, (n_1 + m + 1)\Delta]$  for  $m = 1, \dots, m_0 - 1$ . Assume first that  $b + m_0 > w$  (i.e., the buffer always empties). On the successive time interval  $[(n_1 + k)\Delta, (n_1 + k + 1)\Delta]$ ,  $k \geq 1$ , the buffer content oscillates between 0 and  $\sigma(1 - c/h)$  before a transient period where the buffer reaches the  $2\sigma(1 - c/h)$  before returning to 0. This happens until the maximum window size is reached, and then, the buffer content eventually oscillates between the values 0 and  $\sigma(1 - c/h)$ . This result is consistent with equation (13).

In the case  $b + m_0 < w$ , the buffer of the cell spacer does not empty from time  $(n_1 + m_0)\Delta$  on, and oscillates stepwise between the values  $(b + m - 1)\sigma$  and  $(b + m - 1)\sigma + \sigma(1 - c/h)$  over the  $m$ th time interval of length  $(b + m - 1)\sigma/c$ ,  $m = m_0, \dots, w - b - 1$ . The successive oscillation intervals are separated by a transition interval of length  $\sigma/c$ , where the buffer increases before oscillating. The buffer content eventually oscillates between the values

$$\mathcal{V}_w^- = w\sigma - c\Delta \quad \text{and} \quad \mathcal{V}_w^+ = w\sigma - c\Delta + \sigma \left(1 - \frac{c}{h}\right). \tag{16}$$

This expression is consistent with equation (13).

*Case  $h < 2c$*  Now, we consider the case when  $h < 2c$  and assume in a first step that  $b\sigma < c\Delta$ . If  $b + m_0 > w$ , the buffer always empties and its behavior is identical to that described above under the same assumptions, except during some time intervals of length  $\sigma/c$  when two segments arrive at the cell spacer. If  $b + m_0 < w$ , the buffer does not empty any more from time  $(n_1 + m_0)\Delta$  on. To understand what happens when the congestion window increases by one, assume that such an event occurs at time  $t_0$ . The buffer content linearly increases at rate  $(h - c)$  over the time interval  $[t_0, t_0 + k\sigma/c]$  as long as

$$(1 + k)\frac{\sigma}{c} \leq (2 + k - 1)\frac{\sigma}{h},$$

that is,

$$k \leq \left\lfloor \frac{c}{h - c} \right\rfloor$$

The buffer content increases up to the value  $v(t_0) + (2 + k)\sigma(1 - c/h)$ , where  $v(t_0)$  is the buffer content at time  $t_0$ , and then oscillates between the values  $v(t_0) + \sigma$

and  $v(t_0) + \sigma + \sigma(1 - c/h)$ . Hence, we see that except for transition periods, the behavior of the buffer content is roughly the same as in the case  $h > 2c$ . The same conclusions hold under the assumption  $b\sigma > c\Delta$ . This completes the proof.

From the above results, we see that at the end of the congestion avoidance period, the source adapts to the spacing rate  $c$  by sending cells at a rate at most equal to  $c$ . Hence, the cell spacing technique is capable of regulating a TCP source so that it adapts its transmission rate to a prescribed spacing rate. Loosely speaking, by means of the cell spacing technique, a TCP connection can be assigned a bit rate, whereas by definition, there is no concept of bit rate in the TCP flow control mechanism, handling amounts of data only. This is a key point of TCP over ATM regulated by cell spacing.

Besides, when we consider the equations (3) and (13), we can observe the following somewhat paradoxical phenomenon. Indeed, for a fixed round trip time, when  $c$  is large, the source transmits a large amount of data, and then, we may expect that the buffer is quite full. Surprisingly, it turns out that the larger the spacing rate, less data are buffered in the cell spacer, even if the slow start mechanism is enabled. This phenomenon is due to the fact that a significant part of the congestion window is buffered in the so-called “channel memory”, which appears in the equations via the term  $c\Delta$ . In fact, two parameters have a major impact on the buffer content: the spacing rate  $c$  and the round trip time  $\Delta$ . In parallel, we can also note that, the larger the round trip time, the smaller the buffer content is.

## 5 Delayed Acknowledgements

We now assume that one acknowledgement packet is returned to the source when two segments are received by the destination. During the slow start phase, an acknowledgement packet received by the source generates three credits (two for the two acknowledged segments plus one outstanding credit due to slow start). We use the same arguments as in the previous sections to study the evolution over time of the buffer content of the cell spacer. The congestion window starts with two segments, since two segments should be received by the destination before returning an acknowledgement packet.

Assume that the first segment arrives at the buffer of the cell spacer at time 0. Then, on the time interval  $[0, \Delta']$ , where  $\Delta' = \delta + 2\sigma/c$ , two segments arrive at the cell spacer. In the time intervals  $[\Delta', 2\Delta']$  and  $[2\Delta', 2\Delta' + \Delta]$ , where  $\Delta = \delta + \sigma/c$ , three segments arrive. On the time interval  $[2\Delta' + \Delta, 3\Delta' + \Delta]$ , two bursts of three segments arrive, the distance between the beginning of the two consecutive bursts is equal to  $2\sigma/c$ . On the time interval  $[3\Delta' + \Delta, 4\Delta' + \Delta]$ , nine segments arrive, and so on.

In general, as long as the buffer of the cell spacer empties, we can note that a certain number of segments arrive in bursts of three segments over a time interval with a certain length, equal to either  $\Delta'$  or  $\Delta$ . To be more specific, let  $p_n$ ,  $L_n$ , and  $c_n$  denote the number of segments and the length of the  $n$ th time interval, and the number of segments of the  $(n - 1)$ th time interval, which have



not been acknowledged, respectively. It is clear that  $c_n = 0$  if all the segments have been acknowledged and  $c_n = 1$  if one segment has not been acknowledged. If  $c_n = 0$ , then  $L_n = \Delta'$  and if  $c_n = 1$ , then  $L_n = \Delta$ . It can moreover be shown that as long as the buffer of the cell spacer empties,  $p_n$  and  $c_n$  and  $L_n$  satisfy the following recursion:

$$n \geq 1, p_{n+1} = 3c_n + 3 \left\lfloor \frac{p_n - c_n}{2} \right\rfloor \text{ and } c_{n+1} = p_n - c_n - 2 \left\lfloor \frac{p_n - c_n}{2} \right\rfloor \quad (17)$$

with  $p_1 = 2$  and  $c_1 = 0$ .

Over the  $n$ th time interval for  $n \geq 2$ ,  $p_n/3$  bursts of three segments at the peak cell rate  $h$  arrive at the cell spacer. Each segment is composed of  $\sigma$  cells and the distance between the beginning of two consecutive bursts is equal to  $2\sigma/c$ . It follows that the maximum buffer size over the  $n$ th time interval is equal to

$$v_n = \left( \frac{p_n}{3} - 1 \right) \sigma + 3\sigma \left( 1 - \frac{c}{h} \right) \quad (18)$$

if  $3c < 2h$ , and

$$v_n = p_n \sigma \left( 1 - \frac{c}{h} \right) \quad (19)$$

if  $3c > 2h$ . The server of the buffer of the cell spacer is busy over a time interval of length  $p_n \sigma / c$ . As long as  $p_n \sigma \leq cL_n$ , the buffer of the cell spacer empties over the  $n$ th time interval.

As in the previous sections, let us introduce the integers  $n_0$  and  $n_1$  defined by

$$n_0 = \min\{n : p_n \sigma > cL_n\}, \quad (20)$$

$$n_1 = \min\{n : p_n > b\}. \quad (21)$$

The  $n_0$ th time interval is the first interval over which the buffer of the cell spacer does not empty and the  $n_1$ th time interval is the first interval over which the slow start threshold is reached. Let  $m_0$  be defined as in equation (12). The analysis of the previous section can be extended and we can state the following result, whose proof is omitted.

**Theorem 3.** *At the end of the slow start period, the buffer content oscillates between the values*

$$\mathcal{V}_b^- = (b\sigma - cL_{n_0})^+ \text{ and } \mathcal{V}_b^+ = (b\sigma - cL_{n_0})^+ + 2\sigma \left( 1 - \frac{c}{h} \right). \quad (22)$$

*Once the maximum congestion window size  $w$  has been reached, the buffer content oscillates between the values*

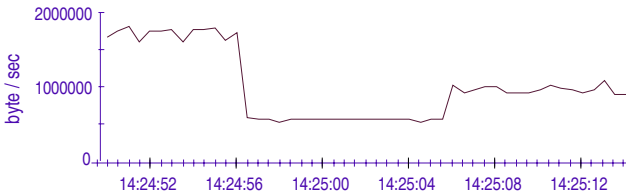
$$\mathcal{V}_w^- = [w\sigma - cL_{n_1+m_0}]^+, \mathcal{V}_w^+ = [w\sigma - cL_{n_1+m_0}]^+ + 2\sigma \left( 1 - \frac{c}{h} \right). \quad (23)$$

The above formulae will be compared in the next section with what is observed via an experimentation with a real TCP/IP protocol stack.

## 6 Experimental Results

To show the capability of cell spacing to regulate TCP dynamics without information loss in a real situation, the configuration depicted in Figure 1 has been used with two Silicon Graphics workstations connected to each other through a cell spacer via a 34 Mbit/s link. The operating system on the workstations is Irix 6.3, the TCP protocol stack is BSD 4.4, the slow start mechanism is enabled, and the TCP retransmission time out value is initially set to 1 second. The TCP information flow generated by a workstations is segmented in a NIC (network interface card) implementing an AAL 5 mechanism. The maximum segment size MSS is equal to 1516 bytes and the maximum window size is equal to  $60 \times 1024$  bytes, corresponding to 41 segments with maximum size.

The cell spacer is managed via a PC, which allows the cell spacing period to be modulated over time. A buffer space of 1300 cells is dedicated to the connection in the cell spacer. The dynamics of TCP are observed by using the Silicon Graphics Performance Co-Pilot (v. 2.0) tool. The TCP application consists in displaying from the server workstation a graphical animation sequence on the receiving workstation through an Xwindow client server connection.



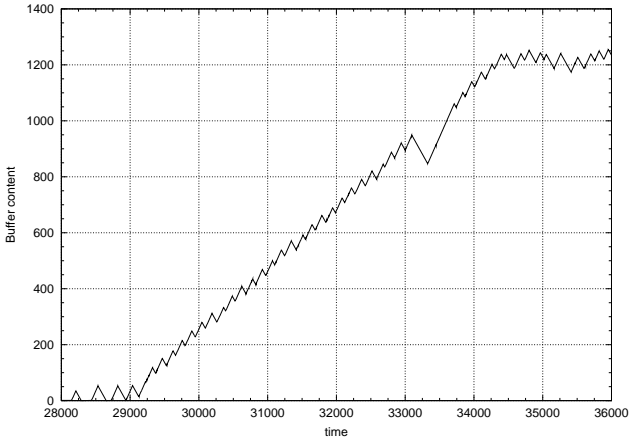
**Fig. 2.** Instantaneous bit rate of the TCP application when the cell spacing period is modulated

Figure 2 shows the adaptation of TCP to changes of the spacing period, that is, how the bit rate of the TCP connection fits the spacing rate. At the beginning the spacing rate is set equal to 33.920 Mbit/s and the TCP application is not constrained (its free run rate is then about 2 Mbyte/s). The spacing rate is then set equal to 5 Mbit/s and TCP adapts its transmission rate to 0.5 Mbytes/s, without information loss. In the last part of Figure 2, the spacing rate is set equal to 10 Mbit/s and TCP regulates at 1 Mbyte/s. This clearly shows that TCP can be regulated via cell spacing .

With regard to the quality of service, when the cell spacing rate is decreased, the perceived quality of the graphical animation sequence is of course not as good as in the case of free run, but is still satisfying. In particular, there is no interruptions due to packet loss. The movement of the objects is just slowed down.

To conclude, let us examine the behavior of the buffer content of the cell spacer. In the TCP/IP protocol stack considered, one acknowledgement packet

is returned to the source when two segments are received. The evolution of the buffer content (expressed in cells) is depicted in Figure 3, when the spacing rate is about  $c = 15.264$  Mbit/s. The time unit is the cell transmission time, namely 12,5 microseconds. The round trip time is  $\Delta = 3.6$  milliseconds. A TCP segment gives rise to  $\sigma = 32$  cells. The slow start threshold is equal to the maximum window size, equal to  $w = 1312$  cells.



**Fig. 3.** Content of the buffer of the cell spacer plotted from experimental data.

In Figure 3, we can distinguish the first time intervals where the buffer of the cell spacer empties. Then, when the buffer does not empty anymore, the buffer content increases almost linearly at rate  $c/2$  (because  $3\sigma$  cells are transmitted every  $2\sigma/c$  and the buffer release rate is equal to  $c$ ) until it hits the maximum window size and then oscillate between some values, which are not very far from the theoretical values

$$w - c\Delta = 118 \quad \text{and} \quad w - c\Delta + 2\sigma(1 - c/h) = 1217. \quad (24)$$

The difference between the theoretical and the actual values is due to the real behavior of TCP and to round off errors.

## 7 Conclusion

In this paper, we have shown that the cell spacing technique can be used to efficiently regulate a TCP connection without packet loss, provided that the cell spacer can buffer at least a whole TCP window per connection. In fact, cell spacing acts in such a way that TCP automatically adapts to the offered constant bit rate pipe, which bit rate is equal to the spacing rate.

Such a regulation technique can then be combined to a bandwidth reservation and renegotiation scheme to transport TCP traffic over variable bit rate connections. Among all the resource reservation techniques, which have been specified for ATM networks, the resource management (RM) procedures, which operate on the same time scale as the round trip time across the network, offer the best response time with minimal overhead. Their responsiveness allows in particular a coupling between the establishment of a TCP connection with that of the underlying ATM connection [3].

Such an approach is very close to that followed by the MPLS group within the IETF, but it includes in addition traffic management aspects, which are currently not covered by MPLS, such as the regulation of TCP via cell spacing and the explicit reservation of resources (namely bandwidth in the network and buffer space in the cell spacer) for carrying TCP traffic. In this respect, it may be envisioned to use the label distribution protocol to set up underlying ATM connections. This issue will be addressed in future work.

## References

1. Bonaventure O., Klovning E., and Danthine A (1996) Is VBR a solution for an ATM LAN ? In *Proc. IFIP 5th International Workshop on Protocols for High Speed Networks*, pp. 60–74.
2. Bonaventure O. (1997) A simulation study of TCP with the proposed GFR service category. In *Proc. High Performance Networks for Multimedia Applications*.
3. Boyer J., Boyer P. , Dugeon O., Guillemin F., and Mangin C. (1999) Accelerated signalling for the Internet over ATM. *European Transactions on Telecommunications and Related Technologies* 10(2): 153-164.
4. Ghani N. (1999) Enhanced TCP/IP ACK pacing for high speed networks. In *Proc. ITC'16*.
5. Johansson P., Wedlund E., and Karlsson J. (1997). Interaction between TCP flow control and ABR rate control. In *Proc. IEEE ATM'97 Workshop*.
6. The Packeteer Technical Forum (1998) TCP/IP bandwidth management series.
7. Romanow A. and Floyd S. (1994) Dynamics of TCP traffic over ATM networks. In *Proc. ACM Sigcom'94*, pp. 79–88.
8. Stevens W.R. (1994) *TCP/IP illustrated*. Vol. 1. Addison Wesley Professional Computing Series.
9. Thompson K., Miller G.J., and Wilder R. (1997) Wide area Internet traffic patterns and characteristics. *IEEE Network Magazine*, November/December issue.