Jitter-based analysis and discussion of burst assembly algorithms

Javier Aracil*, José Alberto Hernández*, Kyriakos Vlachos†, Emmanouel Varvarigos†
* Universidad Autónoma de Madrid, Spain, e-mail: {javier.aracil,jose.hernandez}@uam.es
† RACTI, University of Patras, e-mail: {kvlachos,manos}@ceid.upatras.gr

***Abstract*— This work provides a jitter analysis of size-based burst assembly algorithms and also discusses other burst assembly algorithms that use the packet delay as the assembly threshold to provide a bound on jitter.**

***Index Terms*— Optical Burst Switching, TAVE, burst-assembly.**

I. INTRODUCTION

In Optical Burst Switched networks (OBS) [1], packets are assembled into large-size optical bursts at the ingress nodes. Such packets traverse all-optically the network, thus suffering only two types of delay apart from propagation delay: burst-assembly delay and offset time. The former comprises the time that packets have to wait until the optical burst is made, whereas the latter relates to the amount of time the Burst Control Packet is sent in advanced of the data bursts [1]. The BCP is sent ahead on attempts to reduce the data burst's blocking probability, via reservation at each intermediate node's scheduler.

The amount of time that packets have to wait until the optical burst is assembled is governed by the particular burst-assembly algorithm employed at the edge node. Several algorithms have been proposed in this light, mainly focusing on either limiting the burst-release time (see the timer-based algorithms [2]), or sizing the outgoing burst to a fixed value (see [3]), or a combination of both [4], [5].

Typically, a single random variable is used to characterize burst assembly delay. Such variable accounts for the delay elapsed from the arrival of the first packet until the burst is finally released. However, the analysis does not take into account the delay experienced on a per-packet basis. In this paper we focus on *jitter* analysis, which is broadly defined as the probability distribution of the average delay experienced by packets in a given burst. From that general definition, one may immediately obtain the variance or coefficient of variation of the delay, which are also well-known jitter measures.

Furthermore, we also discuss burst-assembly algorithms that use the average packet delay [6] as the assembly criterion to limit the delay jitter. In section II we provide preliminary definitions. Section III provides a jitter analysis of size-based algorithms and section IV presents the results and discussion. Then, Section V is devoted to discuss burst-assembly algorithms that are aimed at providing upper bounds for jitter. Finally, we provide the conclusions that can be drawn from this analysis.

This work has been funded by the European Union e-Photon/ONe+ project

II. PRELIMINARIES

Let us assume that packet arrivals follow a Poissonian process at the OBS burst-assembler, as it is the case for highly-multiplexed core Internet traffic [7]. For notation purposes, we shall assume that the first packet arrives at time $t_1 = 0$, the second packet arrives at time $t_2 = x_1$, the third packet arrives at time $t_3 = x_1 + x_2$, and so forth. Clearly, the random variables x_i denote the inter-arrival times between the i-th and the $i-1$-th packets, as shown in figure 1. The x_i values are assumed to be exponentially distributed with rate $\lambda = 1/EX$.

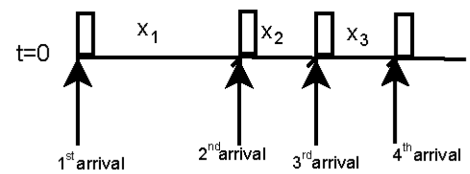

Fig. 1. Notation

Therefore, the i-th packet suffers a burst-assembly delay given by $t_i = \sum_{k=1}^{i-1} x_i$.

Let z_{n+1} denote the average burst-assembly delay suffered by the packets in a burst comprising $n+1$ packets. Taking into account the above, such value is given by:

$$\begin{aligned} z_{n+1} &= \frac{1}{n+1}[(x_1 + \ldots + x_n) + (x_2 + \ldots + x_n) \\ &+ \ldots + (x_{n-1} + x_n) + x_n] = \frac{1}{n+1}\sum_{j=1}^{n} j x_j \quad (1) \end{aligned}$$

The following studies the probability density function (PDF) of the random variable z_{n+1}, that is, $f_{z_{n+1}}(t), t > 0$.

III. ANALYSIS

To obtain the PDF of z_{n+1} as defined by eq. 1, it is first worth noticing that the random variable $(j/(n+1))x_j \sim \exp(\lambda(n+1)/j)$. Thus, it is required to compute the sum of n exponential distributions, with decreasing parameter $\lambda(n+1)/j$, $j = 1, \ldots, n$. The easiest way to proceed makes use of the moment generating function.

Recall that the moment generating function of an exponential distribution with parameter θ is $M_x(s) = (1 - s/\theta)^{-1}$. Hence, the moment generating function of z_{n+1} is the product of the moment generating function of each component in the sum in eq. 1, due to the independence of the x_js, i.e.:

1-4244-0425-8/06/$20.00 ©2006 IEEE

$$M_{z_n}(s) = \prod_{j=1}^{n} \frac{1}{1 - j\frac{s}{(n+1)\lambda}} \quad (2)$$

The above can be decomposed into partial fractions:

$$M_{z_n}(s) = \sum_{j=1}^{n} \frac{A_j}{1 - j\frac{s}{(n+1)\lambda}} \quad (3)$$

whereby the A_j coefficients must be thus computed. By inspection, it can be shown that the A_j coefficients take the following values:

$$A_j = \left(\prod_{k=1, k\neq j}^{n} \left(1 - \frac{k}{j}\right) \right)^{-1} \quad (4)$$

for $j = 1, \ldots, n$. Accordingly, eq. 3 can be tranformed back to

$$f_{z_{n+1}}(t) = \sum_{j=1}^{n} A_j \frac{\lambda(n+1)}{j} e^{-\frac{\lambda(n+1)}{j}t} \quad (5)$$

for $n = 1, 2, \ldots$. With this result, the probability to exceed a given value t_h is straightforward:

$$\begin{aligned} \mathbb{P}(z_n > t_h) &= \int_{t_h}^{\infty} \sum_{j=1}^{n} A_j \frac{\lambda(n+1)}{j} e^{-\frac{\lambda(n+1)}{j}t} dt = \\ &= \sum_{j=1}^{n} A_j \int_{t_h}^{\infty} \frac{\lambda(n+1)}{j} e^{-\frac{\lambda(n+1)}{j}t} dt \\ &= \sum_{j=1}^{n} A_j e^{-\frac{\lambda(n+1)}{j}t_h} \quad (6) \end{aligned}$$

Furthermore, the first and second moments easily arise from the above:

$$\mathbb{E}(z_{n+1}) = \sum_{j=1}^{n} A_j \frac{j}{\lambda(n+1)} \quad (7)$$

$$\mathbb{E}(z_{n+1}^2) = \sum_{j=1}^{n} A_j \frac{2j^2}{\lambda^2(n+1)^2} \quad (8)$$

and the coefficient of variation of z_{n+1}:

$$c_{z_n}^2 = \frac{\mathbb{E}(z_n^2)}{\mathbb{E}^2(z_n)} - 1 \quad (9)$$

IV. Results and discussion

In this experiment, we have simulated the generation of optical bursts with a maximum of $L_{\max} = \in \{1, 3, 5, 7\}$ packets in each burst, assuming the arrival rate of $\lambda = 6$ packets/sec. We have further evaluated the PDFs for z_1, z_3, z_5 and z_7 analitically, following the equations derived in the section above, and plotted them along with the histograms obtained via simulation (see figure 2). Interestingly, as the number of packets in a burst increases the jitter also increases.

Fig. 3 also shows the evolution with n of the mean, standard deviation and coefficient of variation.

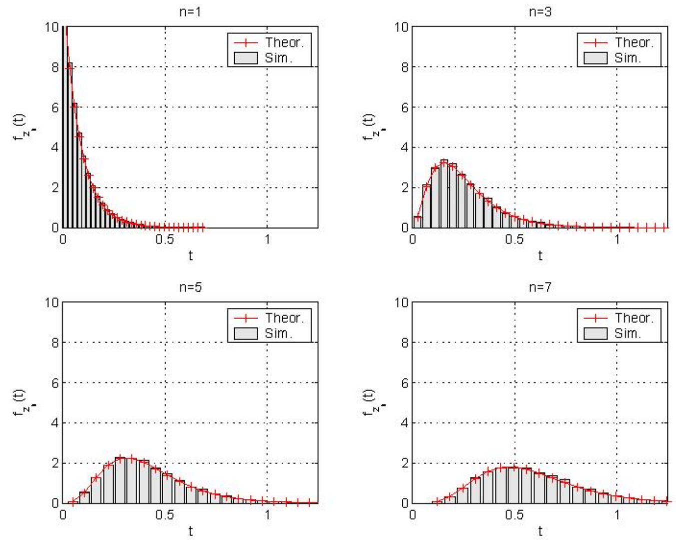

Fig. 2. Probability distribution of z_1 (top-left), z_3 (top-right), z_5 (bottom-left), z_7 (bottom-right).

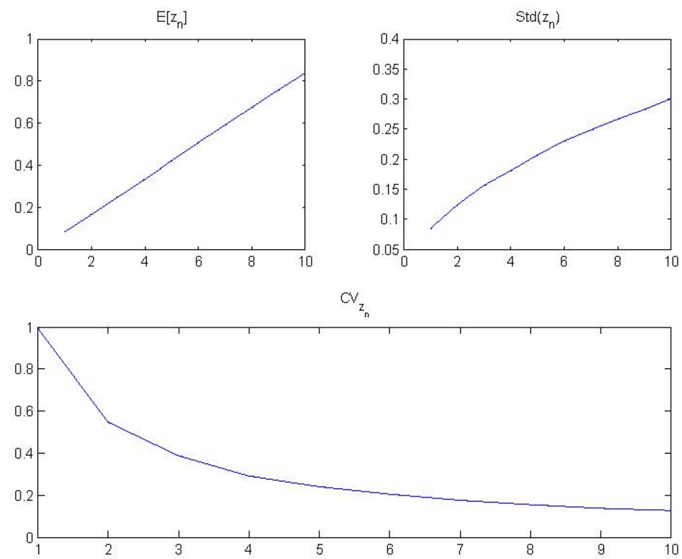

Fig. 3. $\mathbb{E}(z_n)$ (top-left), $Std(z_n)$ (top-right), and CV_{z_n} (bottom).

As shown, as the number n of packets in a burst increases, both the average and standard deviation increase. However, the ratio at which the standard deviation grows is smaller than the growth ratio of the average, thus leading to a decreasing coefficient of variation with n.

At this point, the conclusions obtained can be seen from two different perspectives: On the one hand, the benefits of aggregating packets following a size-based policy with large values of L is straightforward. The more packets assembled into the same burst, the better since, although jitter mean and standard deviation increase, the latter grows more slowly than the former, thus resulting to "some short of" small global jitter (low coefficient of variation). However, large-size bursts have the handycap of long and variable delay suffered especially by the early packet arrivals in each burst. Hence, the network designer must trade-off these two aspects.

V. BURST-ASSEMBLY ALGORITHMS FOR JITTER-LIMITED SERVICES

In this section, we present a new burst assembly algorithm that uses the average delay of the packets comprising the burst as the assembly criterion. More specifically, when a packet that belong to certain burst assembly queue arrives, then the average packet delay, of eq. 1 is updated. When it reaches a threshold denoted here as T_{AVE}, the assembly process stops and a burst is generated. This burst assembly algorithm guarantees that the average delay of the packets belonging to a given assembly queue is set to the desired value. As a result, the packet delay jitter in the assembled bursts can be significantly improved compared to that of the timer-based and length-based algorithms. It must be noted here that the average packet delay does not vary monotonically with time but may decrease or increase depending on the packets' arrival times.

Keeping the average packet delay constant is worthwhile, since it reduces delay jitter at the receiver end. This is important for transport protocols like TCP that use estimations of the round-trip-time to increase or decrease their congestion window and thus their throughput. Large variations of the delay jitter result in timeouts, which in turns decrease the efficient throughput.

We have performed simulation experiments over a single link, to measure the density function of end-to-end packet delay for both a timer-based and the proposed average-delay assembly algorithm. Figure 4 shows the corresponding results. For the simulation experiments, we have set the average packet delay threshold equal to 6 time units and the timer-based threshold T_{MAX} equal to 20 time units.

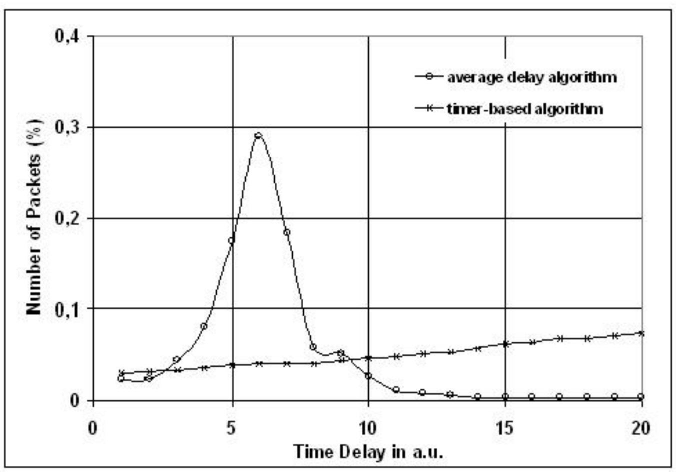

Fig. 4. packet delay distribution for both a timer-based and the proposed average-delay assembly algorithm

From figure 4, it can be seen that when applying the average packet delay algorithm, around 30% of all the packets experience the same average delay of 6 time units, while 80% of all the packets experience a delay within +/- 1 time unit of that value. On the other hand, when the timer-based algorithm is enforced, then the packet delay is spread across the entire time span. This is as expected, since packet arrival time, and thus delay, may span from 0 to T_{MAX} time. Thus, the variance of packet jitter is very high, resulting in a poor performance, when applied in TCP traffic.

VI. CONCLUSIONS AND FURTHER WORK

In this paper we have investigated the burstification delay at the packet level, with emphasis in the packet jitter. A novel burstification algorithm is reported, that takes into account the packet jitter as the burstification criterion.

REFERENCES

[1] C. Qiao and M. Yoo, "Optical Burst Switching (OBS) – A new paradigm for an optical Internet," *J. High Speed Networks*, vol. 8, pp. 69–84, 1999.

[2] V. Vokkarane, K. Haridoss, and J. P. Jue, "Threshold-based burst assembly policies for QoS support in optical burst-switched networks," in *Proceedings of SPIE/IEEE OPTICOMM*, 2002, pp. 125–136.

[3] A. Ge, F. Callegati, and L. S. Tamil, "On Optical Burst Switching and self-similar traffic," *IEEE Comm. Letters*, vol. 4, no. 3, pp. 98–100, March 2000.

[4] X. Cao, J. Li, Y. Chen, and C. Qiao, "Assembling TCP/IP packets in Optical Burst Switched networks," in *Proceedings of IEEE GLOBECOM*, 2002, pp. 2808– 2812.

[5] X. Yu, Y. Chen, and C. Qiao, "Study of traffic statistics of assembled burst traffic in optical burst switched networks," in *Proceedings of SPIE/IEEE OPTICOM*, 2002, pp. 149–159.

[6] Emmanouel Varvarigos K. Christodoulopoulos and K. Vlachos, "A new burst assembly scheme based on the average packet delay and its performance for TCP traffic," *submitted to Elsevier Optical Switching and Networks*.

[7] T. Karagiannis, M. Molle, M. Faloutsos, and A. Broido, "A nonstationary Poisson view of Internet traffic," in *IEEE INFOCOM*, 2004.