Performance evaluation of an optically interconnected "scheduling" switch network for Pareto traffic

Kyriakos Vlachos and Emmanuel Varvarigos

Research Academic Computer Technology Institute, University of Patras, GR-26500, Rio Patra, Greece

kvlachos@ceid.upatras.gr

Chris Bintjas

National Technical University of Athens, GR-15773, Zografou, Greece

cbint@cc.ece.ntua.gr

RECEIVED 11 AUGUST 2004; REVISED 9 SEPTEMBER 2004;
ACCEPTED 10 SEPTEMBER 2004; PUBLISHED 05 OCTOBER 2004

A performance analysis of an optically interconnected packet-scheduling switch network is presented. The scheduling switch uses a branch of feed-forward delays for each input port, interconnected with elementary optical switches to resolve contention. The scheduling switch is guaranteed to be lossless under a certain smoothness property condition. We investigate the packet-loss performance of the switch when the smoothness property condition does not hold, as well as the packet delay impairments at the edge of a scheduling switch interconnected network when this property is enforced. © 2004 Optical Society of America

OCIS codes: 060.0060, 060.4510.

1. Introduction

With the explosive growth of data traffic related to Internet applications, optical packet or burst switching [1–3] has been proposed as a method for fully exploiting the advantages of statistical multiplexing because packets make on-demand use of the outgoing capacity while at the same time taking advantage of new optical techniques to overcome limitations related to optical–electrical–optical conversions. Several innovative packet switch architectures have been proposed, including switches with recirculating loops, [4, 5] the staggering switch [6], the switch with large optical buffers (SLOB) [7], the wavelength routing switch (WRS), and the broadcast-and-select switch (BSS) [8]. However, work on new architectural concepts, node performance, and intelligent control has lagged behind progress in transmission speeds.

In this paper we analyze the performance of a packet-scheduling switch interconnected network for self-similar Pareto traffic. The scheduling switch uses a branch of feed-forward delays interconnected with optical switches to resolve packet contention, and it is guaranteed to be lossless when the so-called (n,T) smoothness property condition holds [9].

We first investigate the packet-loss performance of the switch when this smoothness property condition does not hold; we also investigate the delay impairment when this property is enforced at the edge. It is shown that the average holding time or delay induced in order to transform an unconstrained Pareto session into a smoothed one is relatively small and that this transformation can be easily implemented at the edge router (ER) using a store-and-forward traffic-shaping algorithm and a suitable ER architecture. The rest

of the paper is organized as follows. Section 2 presents the scheduling switch architecture and relevant traffic assumptions, while Section 3 presents a Pareto traffic model and the performance of the scheduling switch. Finally Section 4 presents an ER logical architecture along with a traffic-shaping algorithm for guaranteeing lossless communication in an optically interconnected network employing scheduling core switches.

2. Switch Architecture and Traffic Assumptions

The scheduling switch has been designed to provide lossless communication for sessions that have a certain burstiness or that can be transformed into sessions with such a property, tolerating the corresponding delay. It consists of a scheduling unit with k input–output ports and a $k \times k$ nonblocking space switch, as shown in Fig. 1. Each branch delays the incoming packets, assigning incoming packets to outgoing slots, resolving contention and maintaining packet ordering for the same outgoing link. The problem of scheduling packets through a branch of delay blocks to avoid collisions is a problem of routing a permutation between inputs and outputs in the equivalent Benes network, where nonoverlapping paths in the network correspond to collision-free transmission through the delay blocks [9]. Various implementations of the scheduling switch and the corresponding delay blocks have been proposed [9–11].

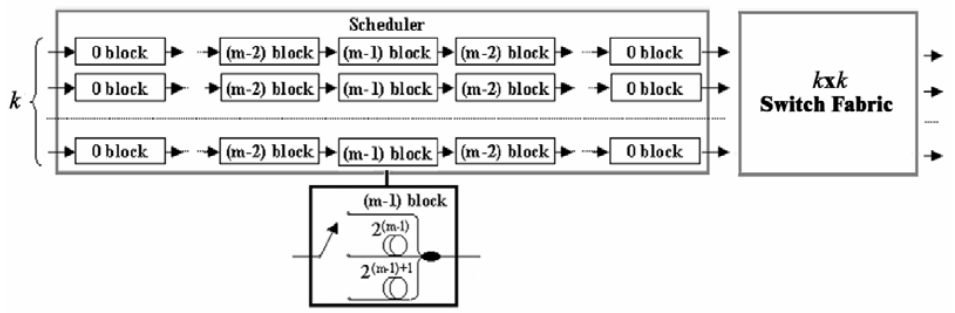

Fig. 1. Scheduling switch architecture, consisting of the scheduler (k inputs) and a $k \times k$ space switch. The scheduler comprises k branches, each with $2\log 2T - 1$ delay blocks. The ith delay block consists of one three-state switch and three delay lines of length 0, with 2^i and 2^{i+1} packet slots.

The buffer capacity of the scheduling switch grows logarithmically with the number of delay blocks used. Thus, for building a size T optical buffer, $2m-1$ delay blocks are needed, where $m = \log T$. The ith block consists of a three-state optical switch (or two 2×2 optical switches) and three fiber delay paths, corresponding to delays equal to 0, 2^i and 2^{i+1} packet slots. To ensure that the packets in the incoming frame can be assigned to any slot in the outgoing frame, the latter must start at least $(3T)/2 - 2$ after the incoming frame begins. This modular buffering scheme can be easily expanded to accommodate more burstiness in the traffic, similar to the way electronic buffers can be expanded in a conventional electronic switch.

A corresponding traffic model that can guarantee lossless communication must be based on the aforementioned buffering scheme. To this end, we assume that the time axis on a link is divided into packet slots of equal length and all T slots are virtually grouped to form a frame. This concept is illustrated in Fig. 2.

Packets are grouped in T-size frames before entering the switch, while frame integrity is maintained at the output as well. A session of packets, an active end-to-end network connection, is said to have the (n, T) smoothness property at a node if at most n packets of

the session arrive at that node during a frame of size T. A session can easily be transformed to have the (n, T) smoothness property at the ingress point of the network, and this property can be preserved throughout a network consisting of scheduling switches as a result of frame integrity maintenance. We let n_{ij} be the number of packets that arrive during a frame over incoming link i that have to be transmitted on link j, and we let k be the number of incoming (and outgoing) links of a node. If the connection and flow control protocols guarantee that the number of packets from all active sessions that require the same outgoing link j in a frame is less than or equal to the frame size T, i.e.,

$$\sum_{i=1}^{k} n_{i,j} \leq T \tag{1}$$

for all $j \in \{1, 2, \ldots, k\}$, then all of the incoming packets can be assigned slots in the required outgoing links so that no packets are dropped. Both wait-for-reservation and tell-and-go protocols can be used to ensure that this requirement is met.

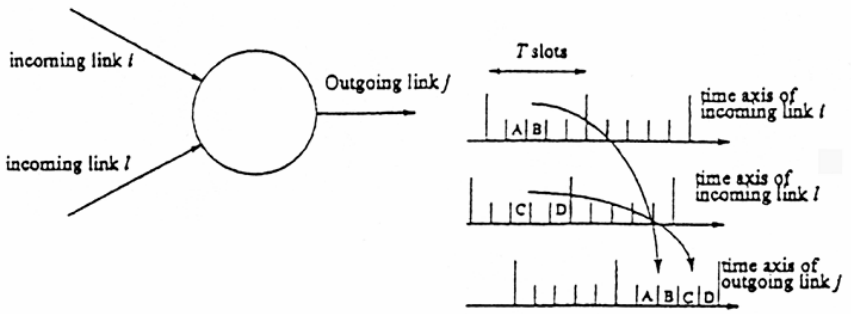

Fig. 2. Incoming and outgoing frames at a node. Packets that arrive in a particular frame of an incoming link and want to use the same outgoing link are sent over the same frame of the outgoing link.

The frame size T is an important parameter and can be viewed as a measure of the traffic burstiness allowed. The larger T is, the less constrained (more bursty) the incoming traffic is allowed to be, and the larger is the flexibility—*granularity*—in assigning rates to sessions. For example, if each link in a network has capacity C and a session has the (n, T) smoothness property, then this session will have an average rate of at most nC/T, implying that capacity can be allocated only in discrete multiples of C/T. It is important to note that this is not circuit-switched data but instead is packet switching with built-in flow control to ensure lossless transmission. Packets from a particular source do not arrive in the same slot, and the number of packets that arrive per frame is not constant but is bounded by n.

3. Switch Performance and Pareto Traffic Model

The model of independent Bernoulli processes is the simplest model that has been widely considered; it results in a tractable analysis while still yielding an appreciation of switch performance. However, in reality traffic is much more bursty than in that model, and, more specifically, Internet traffic has been shown to be better modeled by Pareto or exponentially distributed statistics. It has been shown in the literature that in principle the multiplexing of several sources of Pareto traffic generates self-similar or long-range-dependent (LRD) network traffic. Nevertheless, in reality traffic in the core depends on many externalities and is still an open research issue [12].

In this section we investigate the performance of the scheduling switch under a heavy-tailed truncated Pareto distribution, which is considered by many researchers to be a good model for bursty traffic in real networks [13, 14].

In our model, packets arrive in bursts (ON periods), which are separated by idle periods (OFF periods). To generate a Pareto-distributed sequence of ON periods, one can generate a Pareto-distributed sequence of burst (packet train) sizes, followed by Pareto-distributed idle periods. The minimum burst size is 1, corresponding to a single packet arrival. The formula to generate a Pareto distribution is

$$X_{\text{PARETO}} = \frac{b}{x^{1/a}}, \tag{2}$$

where x is a uniformly distributed value in the range (0, 1], b is the minimum nonzero value of X_{PARETO}, denoted b_{on} and b_{off} for the packet train and the idle period, respectively, and a is the tail index or shape parameter of the Pareto distribution. However, computer simulations using the above formula generate a truncated Pareto distribution because of the discreet x value. In contrast, any true Pareto distribution of sufficiently great length will have values that exceed the range generated by computer simulations. Thus the question that arises is, what is the minimum idle, b_{off} period so that on average the truncated Pareto distribution yields a specific link utilization factor. To define b_{off}, we have assumed, first, slotted operation and, second, ON and OFF periods equal to an integer multiple of a single slot. Thus the actual minimum ON and OFF periods are kb_{on} and kb_{off} respectively, assuming packets of constant size k. Expressing the utilization factor p as the mean size of the ON period over the mean size of ON and OFF periods,

$$p = \frac{\overline{\text{ON}}_{\text{period}}}{\overline{\text{ON}}_{\text{period}} + \overline{\text{OFF}}_{\text{period}}}; \tag{3}$$

calculating the mean value of the truncated Pareto distribution, which does not exceed the value $X_{\text{Pareto}}^{\max} = b/\overline{x_{\min}^{1/a}}$,

$$E(x) = \int_b^{X_{\text{Pareto}}^{\max}} x f(x)\,\mathrm{d}x = \int_b^{X_{\text{Pareto}}^{\max}} x \frac{ab^a}{x^{a+1}}\mathrm{d}x = \frac{ab}{a-1}\left[1 - x_{\min}^{\frac{a-1}{a}}\right]; \tag{4}$$

and substituting Eq. (4) into Eq. (3) allows us to derive the minimum idle period as a function of link utilization [15]:

$$b_{\text{off}} = \frac{\frac{a_{\text{off}}-1}{a_{\text{off}}}}{\frac{a_{\text{on}}-1}{a_{\text{on}}}} \frac{1 - x_{\min}\frac{a_{\text{on}}-1}{a_{\text{on}}}}{1 - x_{\min}\frac{a_{\text{off}}-1}{a_{\text{off}}}} \left(\frac{1}{p} - 1\right). \tag{5}$$

In the above equations $f(x)$ is the probability density function of the Pareto distribution, $x_{\min}$ is the smallest nonzero value of x that is uniformly distributed in (0,1], and $\alpha_{\text{on}}, \alpha_{\text{off}}$ are the tail indices for the packet train size and the idle period, respectively. Figure **??** summarizes how b_{off} changes with link utilization in the proposed truncated Pareto distribution. It must be noted here that our intention was to generate traffic loads that are very close to the specified load with all combinations of a_{on} and a_{off}.

After defining b_{off}, we performed computer simulations for a $k = 2$ and $k = 4$ scheduling switch with $\alpha_{\text{on}} = 1.7$, $\alpha_{\text{off}} = 1.2$ and $X_{\min} = 10^{-4}$. In our simulation, we have selected a_{on} to be larger that a_{off}, since in real traffic, the probability of having extremely large OFF periods is higher than the probability of having extremely large ON periods. Figure 3 shows the corresponding loss ratio results for $T \in [2 \ldots 64]$ and $T = 1024$. Again, packet destinations were evenly distributed. From Fig. 3 it can be seen that the scheduling switch

loss ratios in the case of a Pareto distribution differ significantly from the ones shown in Ref. [16] for random Bernoulli traffic. In both cases the scheduling switch is regarded as an optical switch with T available buffer slots per input port. This is more evident for small T values and is attributed to the bursty nature of the Pareto distribution. More specifically, since the mean size of the ON periods, $\sim$2.4, is close to T during an ON period, all slots of the frame are filled, independent of the resulting workload. This results in increased packet-loss ratios, and, especially in the case $T = 2$, it can be noted that loss varies slightly for all p. This is because T is smaller than the mean value of the ON periods. Nevertheless, as T increases, the packet-loss ratio drops fast, and a loss ratio smaller than 10^{-6} can be obtained for T values higher than or equal to 64. In particular, for $T = 128$ and $k = 2$ loss is $\sim 10^{-6}$, whereas for $T = 1024$ loss is far below 10^{-9}.

Table 1. Calculated b_{off} for Various Link Utilization Factors

Link Utilization, p	Min. Idle period, b_{off}	Truncated b_{off} (for fixed 100-byte packets)	Truncated b_{off} (for fixed 200-byte packets)
0.1	879.1254131	800	800
0.2	390.7224058	400	400
0.3	227.9214034	200	200
0.4	146.5209022	100	200
0.5	100.6806015	100	200
0.6	65.12040097	100	0
0.7	41.86311491	0	0
0.8	24.42015036	0	0
0.9	10.85340016	0	0

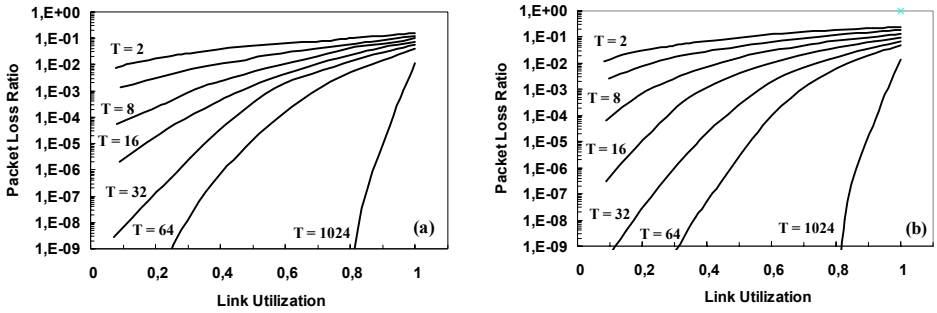

Fig. 3. Packet-loss ratio for (a) $k = 2$ and (b) $k = 4$ versus link utilization for $T \in [2 \ldots 64]$ and $T = 1024$. Packet arrivals follow a truncated Pareto distribution of ON periods with a tail index of 1.7, while OFF periods follow a truncated Pareto distribution as well, with a tail index of 1.2.

4. Delay Impairments Enforcing (n, T) Smoothness in an Optically Interconnected Scheduling Switch Network

The scheduling switch has been designed to be lossless for properly shaped traffic that conforms to the (n, T) smoothness property. In this section the delay impairments that arise because of traffic transformation at the network ingress point are investigated. This per-packet delay is important, since not all sessions can tolerate excessive delays. Furthermore, excessive packet delays can cause buffer overflow at the network edge, and this in turn may result in high packet-dropping ratios or complete service denial.

To investigate this delay, we have used the Pareto packet arrival model presented previously and modeled a simple ER architecture that consists of an input buffer that first stores incoming packets before forwarding them to the output and positioning them in outgoing frames. It is assumed that the underlying core network consists of scheduling switches and is guaranteed to be lossless.

Figure 4 shows the logical structure of the modeled ER. The ER manages one first-in-first-out (FIFO) queue, virtually separated in N FIFOs, one per output port. Hence, a total of $N \times N$ queues are present. This queue separation makes it possible to avoid performance degradations of the scheduling switch due to head-of-the-line blocking [17] and is called virtual output queuing. Within each ER FIFO—and thus in all N virtual output queues—the packet arrival position is maintained, so that processing always starts from the first packet stored in the FIFO. If the first packet stored in the ith input FIFO requests output j of the next scheduling switch and T packets have already been selected for that specific output, only then is the FIFO property relaxed and the second packet in the FIFO processed.

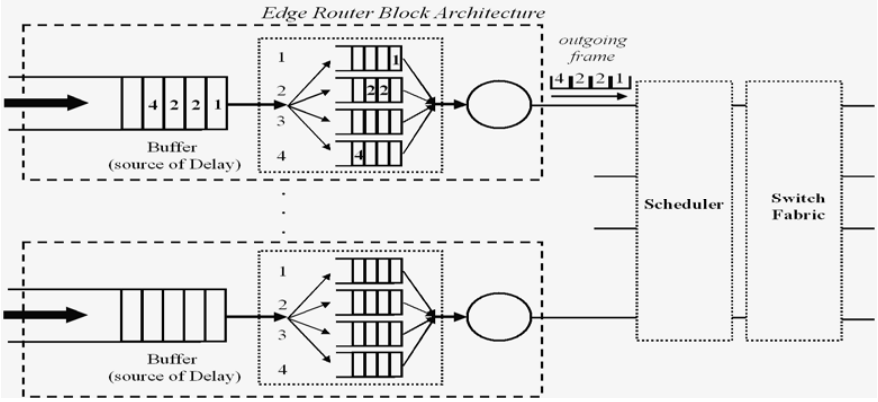

Fig. 4. ER logical structure and simulated experimental setup.

Figure 5 shows the scheduling–traffic-shaping algorithm in pseudocode, assuming N connecting ERs for each scheduling core switch. ERs are inspected in a simple round-robin way. The constraint of Eq. (1) is enforced to all ERs by logging the output destination of the forwarded (per outgoing frame) packets. The basic concept of the traffic-shaping algorithm is to increase the link load to the maximum allowed, so that all outgoing slots of the size T outgoing frame are filled without violating Eq. (1) and minimizing holding times on a packet basis. Thus, each input FIFO is searched thoroughly if outgoing slots are still empty, and the FIFO property is relaxed only when Eq. (1) is violated and there are still empty outgoing slots. Alternatively, a window size of w packets could be used to reduce long processing times at the expense of lower link utilization.

We have simulated four ERs, each with an input load $p \in [0 \ldots 1]$ and $T \in [T \ldots 1024]$. Packet arrival follows the truncated Pareto model, presented in Section 3. Figure 6 dis-

plays the average packet delay (holding time) for a workload per ER equal to 1 and for $T \in [4 \ldots 128]$, versus the number of the outgoing frames (number of iterations) in the simulation. From Fig. 6 it can be seen that the edge packet delay tends to an upper bound value for all T, denoting the stability of the system. This is a significant feature for the network and is due to the traffic-shaping algorithm that relaxes the FIFO property only when Eq. (1) is violated.

For T values higher than 8, packet delay reaches its upper bound faster, since from the first iterations it is possible to fill all the outgoing time slots of the current frame. For example, for $T = 1024$ it has been found that the edge packet delay is close to 0.1 frames.

As has been already mentioned, the packet delay is a critical factor that determines the necessary buffer size to avoid packet dropping at the network edge due to possible buffer overflows. Figure 7 displays the corresponding instant size of the buffer per outgoing frame of the simulation. Again, each ER offers a workload of 1, while T is varied from 4 to 1024. It is worth noting, from Fig. 7, that for T values equal to or higher than 128, from the very first iterations, the number of packets that are stored in the FIFO is constant and the incoming–outgoing packet process enters steady-state operation. This is consistent with the almost constant packet delay per outgoing frame for the same T values. The opposite is valid for smaller T values such as 4 or 8, for which the buffer size reaches its steady state after 5000 outgoing frames, and therefore the packet delay is significantly higher as shown by Fig. 6. The confidence level of our measurements is expected to be high, since less than 10^{-5} deviation was noticed in the experiments.

```
input_FIFO_size is the FIFO capacity for its input port
fifo_slot is a pointer that goes through each input  FIFO checking stored packets
output_frame is pointer that maintain packets output port so that to check that eq.
(1)      is not violated
TRANSMIT is the function when a packet is found that does not violate eq. (1)
N is the number of the Edge Router connected to the scheduling switch
packet_destination the destination of the packet
input [i, fifo_slot] denotes the fifo_slot position of the i input port FIFO

output_frame[j] := 0;                (for all outputs j ϵ [1…N])
fifo_slot [i] : = 1;                 (for all inputs i ϵ [1…N])

WHILE (output_frames[j] < T; for all j ϵ [1…N] ) OR
          (fifo_slot [i] = input_FIFO_size; for all i ϵ [1…N])  )
DO
   fifo_slot [j] : = 1;     (for all inputs j ϵ [1…N])
   FOR i IN 1 TO N LOOP
        WHILE  (fifo_slot != null) AND
                  (Output _frame[packet_destination(input[i, fifo_slot]] = T) AND
                  (fifo_slot[i]<= input_FIFO_size)
        DO
            fifo_slot : = fifo_slot  + 1;
        END DO;
        IF        (fifo_slot <= input_FIFO_size)
        THEN (packet found)
                   TRANSMITT packet [fifo_slot] from input(i) ;
                  output _frame[packet_destination [input[i, fifo_slot]]] :=
                  output _frame[packet_destination [input[i, fifo_slot]]] + 1;
        END IF;
   END LOOP;
END DO;
```

Fig. 5. Pseudocode of the scheduling algorithm at the network edge.

It is worth noting here that the aforementioned results were obtained for an offered

workload per ER equal to 1, which underscores the advantage of combining the scheduling switch in the core and the aforementioned ER model at the network ingress point. To this end, properly configuring the FIFO limit at the edge can achieve a zero packet dropping ratio at the ingress point of the network as well as a zero packet-loss ratio in the core. For other workload values lower than 1, the curves obtained for packet delay and FIFO size per outgoing frame have a similar slope, but steady-state operation is obtained even faster (after 1×10^3 frames), and FIFO size reaches a significantly lower bound.

5. Conclusions

We have performed an evaluation of a scheduling-switch, optically interconnected network for bursty Pareto traffic. The scheduling switch is guaranteed to provide lossless communication when the incoming traffic has the so-called (n, T) smoothness property. We have evaluated the performance of the switch when this condition does not hold, and we have investigated the delay impairments of a scheduling-switch-based network when this property is enforced at the network edge. For this purpose we have modeled an ER architecture along with a traffic-shaping algorithm that guarantees lossless communication in the underlying core network. Based on simulations carried out, we have found that the induced delay is relative small and that the incoming–outgoing packet process enters its steady state within a few thousand outgoing frames. This feature is important and guarantees a zero packet-drop ratio at the edge as well as a zero packet-loss ratio in the core network, with a worst-case finite holding time. In ongoing work we will investigate and optimize the performance of various traffic-shaping algorithms for uniform and nonuniform traffic.

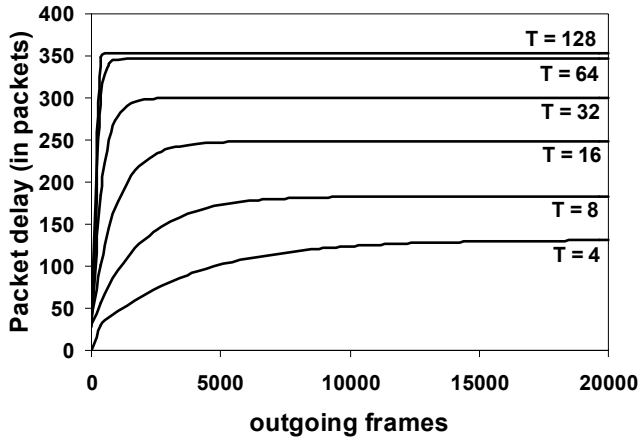

Fig. 6. Average edge packet delay (holding time) per outgoing frame. Simulations have been carried out for a workload per source value of 1.

References and Links

[1] S. Yao, B. Mukherjee, and S. Dixit, "Advances in photonic packet-switching: an overview," IEEE Commun. Mag. **38**, 84–94 (2000).

[2] C. Qiao, "Labeled optical burst switching for IP-over-WDM integration," IEEE Commun. Mag. **38**, 104–114 (2000).

[3] C. Guillemot, M. Renaud, P. Gambini, C. Janz, I. Andonovic, R. Bauknecht, B. Bostica, M. Burzio, F. Callegati, M. Casoni, D. Chiaroni, F. Clerot, S. L. Danielsen, F. Dorgeuille, A. Dupas, A. Franzen, P. B. Hansen, D. K. Hunter, A. Kloch, R. Krähenbühl, B. Lavigne, A. Le Corre, C. Raffaelli, M. Schilling, J.-C. Simon, and L. Zucchelli, "Transparent optical packet switching: the European acts KEOPS project approach," J. Lightwave Technol. **16**, 2117–2134 (1998).

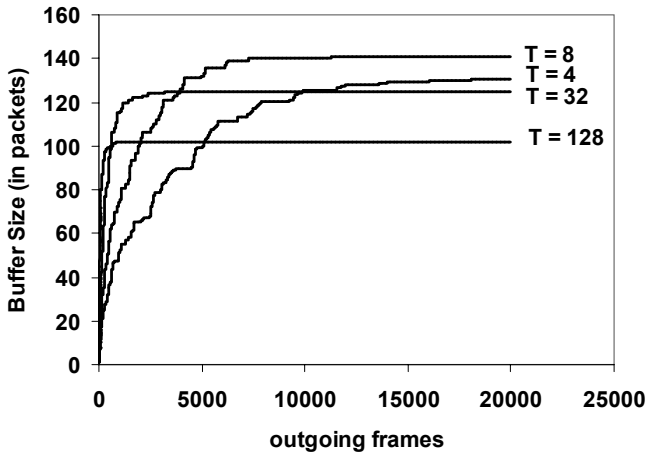

Fig. 7. Instant buffer size of the ERs for a workload per source value equal to of 1 and $T = 4, 8, 32, 128$.

[4] A. Huang and S. Knauer, "Starlight: a wideband digital switch," in *Proceedings of IEEE Global Communication Conference* (IEEE, New York, 1984), pp. 121–125.

[5] K. Vlachos, I. T. Monroy, A .M. J. Koonen, C. Peucheret, and P. Jeppesen, "STOLAS: switching technologies for optical label signals," IEEE Commun. Mag. **41**, 43–49 (2003).

[6] Z. Haas, "The 'staggering switch': an electronically controlled optical packet switch," J. Lightwave Technol. **11**, 925–936 (1993).

[7] D. Hunter, W. Cornwell, T. Gilfedder, and A. Franzen, and I. Andonovic, "SLOB: A switch with large optical buffers for packet switching," J. Lightwave Technol. **16**, 1725–1736 (1998).

[8] M. Renaud, F. Masetti, C. Guillemot, and B. Bostica, "Network and system concepts for optical packet switching," IEEE Commun. Mag. **35**, 96–102, (1997).

[9] E. Varvarigos, "The 'packing' and the 'scheduling packet' switch architectures for almost all-optical lossless networks," J. Lightwave Technol.**16**, 1757–1767, (1998).

[10] G. Theophilopoulos, M. Kalyvas, K. Yiannopoulos, K. Vlachos, E. Varvarigos, and H. Avramopoulos, "An alternative implementation technique for the scheduling switch architecture," J. Lightwave Technol. (to be published).

[11] J. P. Lang, E. A. Varvarigos, and D. J. Blumenthal, "The lambda-scheduler: a multiwavelength scheduling switch," J. Lightwave Technol. **18**, 1049–1063 (2000).

[12] T. Karagiannis, M. Molle, M. Faloutsos, and A. Broido, "A nonstationary Poisson view of Internet traffic," in *Proceedings of IEEE International Conference on Computer Communication* (IEEE, New York, 2004) (to be published).

[13] M. E. Crovellaand and A. Bestavros, "Self-similarity in World Wide Web traffic: evidence and possible causes," IEEE/ACM Trans. Netw. **5**, 835–846 (1997).

[14] M. Zukerman, T. D. Neame, and R. G. Addie, "Internet traffic modelling and future technology implications," in *Proceedings of IEEE International Conference on Computer Communication* (IEEE, New York, 2003), pp. 587–596.

[15] G. Kramer, "On generating self-similar traffic using pseudo-Pareto distribution," Technical brief, Department of Computer Science, University of California, Davis, `http://wwwcsif.cs.ucdavis.edu/~kramer/papers/self_sim.pdf`.

[16] K. Vlachos, K. Seklou, and E. Varvarigos, "Performance evaluation of an optical packet scheduling switch," in *Proceedings of IEEE Global Communication Conference* (IEEE, New York) (to be published).

[17] M. Karol, M. Hluchyj, and S. Morgan, "Input versus output queueing on a space division switch," IEEE Trans. Commun. **35**, 1347–1356 (1987).