

# A preemptive scheduling scheme for flexible QoS provisioning in OBS networks

Kostas Ramantas<sup>1</sup>, Tito Raúl Vargas<sup>2</sup>, Juan Carlos Guerri<sup>2</sup> and Kyriakos Vlachos<sup>1</sup>

<sup>1</sup>*Computer Engineering and Informatics Department & Research Academic Computer Technology Institute, University of Patras, Rio, Greece (Tel: +30 2610 996990, email: [kvlachos@ceid.upatras.gr](mailto:kvlachos@ceid.upatras.gr))*

<sup>2</sup>*ITEAM, Instituto de Telecomunicaciones y Aplicaciones Multimedia Universidad Politécnica de Valencia Valencia, Spain*

**Abstract**— In Optical Burst Switching (OBS) networks one of the most challenging problems is to provide service differentiation. The lack of buffering in the intermediate nodes, makes QoS provisioning in OBS networks a difficult problem to address. In this work, we propose a scheduling algorithm that supports QoS based on preemptions, which are controlled by a novel preemption policy. Our claims are validated by analytical modelling and experimental analysis through simulations using data belonging to two different classes of service - best effort and video streaming data. Performance evaluation is based on throughput measurements for best effort class, and perceived quality metrics like PSNR and MOS scores for the video streaming class.

**Index Terms**— Optical Burst Switching, preemptions, Class of Service, Burst Scheduling.

## I. INTRODUCTION

In OBS-related bibliography, LAUC-VF is the scheduling algorithm of choice for OBS networks, due to its good average case performance and relatively small computational complexity. However, LAUC-VF is far from optimal, and it does not support QoS differentiation. QoS provisioning in one-way OBS is a challenging problem, due to lack of buffering at the core, and the best-effort nature of scheduling algorithms. Bursts cannot be stored in intermediate nodes, so contention resolution options are limited, and priority inversions among high-priority and low-priority bursts are difficult to tackle. This is because as an online algorithm, LAUC-VF has no information about future bursts, so inevitably makes non-optimal decisions regarding channel selection and burst dropping. For example, if the control packet, of a low priority burst arrives at an OBS node before the control packet of a higher priority overlapping burst, LAUC-VF has no means to give priority to the latter [1].

Two very effective techniques proposed in bibliography for strict priorities in OBS networks are PJET and PLAUC-VF [2], each one with their pros and cons. PJET implements strict priorities by differentiating the offset time of priority classes, but with a big delay overhead. PLAUC-VF, on the other hand, uses burst preemption, which is a well known technique that adds flexibility to the burst scheduling process. It allows the re-arrangement of already scheduled bursts in order to accommodate reservation requests which would be otherwise blocked, and at the same time it makes QoS differentiation possible.

In this work, we propose a new multi-class preemptive

scheduling scheme that supports QoS differentiation, and perform an in-depth study through mathematical modeling and simulation. The performance evaluation of the proposed scheme is performed on an OBS network modeled in the Network Simulator (ns-2) environment. We study the effectiveness of the proposed scheme in the presence of two traffic types, namely best effort data traffic and real time video traffic, separated in two distinct classes of service. The best effort traffic is generated with fractal traffic generators, and specifically supFRP [3] self-similar process, to mimic the dynamics of real-world internet traffic. Video traffic on the other hand is generated by trace-driven video sources, which are part of the Evalvid framework [4]. Video traffic consists of video streams encoded with MPEG4 codec, the most commonly used at the internet. Through Evalvid framework, we were able to obtain perceptual metrics of video traffic, such as MOS score and PSNR, which can give us a realistic estimation of the video quality which is streamed over a lossy channel. The rest of the paper is organized as follows. Section II reviews existing research on preemptive scheduling focusing on probabilistic preemptions. A novel probabilistic preemption policy is presented, and its blocking probabilities are evaluated analytically and experimentally. Section III presents a variation of the preemptive LAUC-VF, which uses the proposed preemption policy. Its efficiency is evaluated experimentally, through ns-2 and Evalvid Framework. Finally Section IV concludes the paper.

## II. QOS PREEMPTIVE SCHEDULING SCHEME

Research on preemptive scheduling algorithms has focused on multi-class schemes with strict priorities in order to achieve QoS guarantees and class separation. However this approach has a very small flexibility as it does not take into account the significance of preempted bursts and it can lead to severe throughput loss of the low priority class. This happens because large data bursts carrying many TCP sessions can be preempted, leading many TCP sources to halve their transmission windows. In the proposed scheme, we do not enforce strict priorities between classes, but rather employ a probabilistic preemption policy to decide whether to preempt a burst or not. This approach is very flexible, as it allows the combination of a number of parameters in order to derive a preemption probability  $P$ , on which all preemption decisions are based.

In what follows, we consider a buffer-less OBS network employing full wavelength conversion, where burst reservations are performed using JET signaling protocol. The OBS network supports two classes of service, a low priority best effort class, or class 0, for bulk data transmissions and a high priority real time class, or class 1 for video transmissions. At the network ingress, packets of both classes are assembled to bursts and are never mixed in the same burst. Packets of the real time class are transmitted continuously, or “streamed” in UDP frames, without support for retransmissions. As a result, video data are sensitive to frame losses, which are never retransmitted and thus lead to direct quality degradation. In contrast bulk data transmissions use TCP protocol, which adds reliability and flow control through the mechanism of the TCP window and its retransmission scheme. Lost packets are retransmitted, but they lead TCP sources to halve their transmission windows, resulting in a severe throughput loss. Evidently, there is a trade-off between the two types of traffic that is the TCP throughput and video quality degradation, which we will try to capture through the proposed probabilistic preemption policy.

#### A. Preemptive Drop Policy (PDP)

The proposed preemptive scheduling scheme employs a variation of the Preemption Drop Policy (PDP, described in [5]) which is a probabilistic strategy for provisioning flexible priorities. In an OBS node employing PDP, a free wavelength can be reserved by a burst belonging to any service class (no access restriction techniques are employed) but if all wavelengths are occupied, preemptions can be performed, according to a predefined policy. The preemption policy is parameterized by the definition of  $P$ , which expresses the probability of a successful preemption. That is, whenever a reservation request for a new burst arrives, this new burst has a probability  $P$  to preempt an already scheduled overlapping burst. The preemption policy has to consider the sizes of the two overlapping bursts, as well as their priority classes. In a strict priority system, a higher priority burst always preempts an already scheduled one of a lower priority. However, this policy has a very limited flexibility, and can lead to a significant throughput loss of the low priority class. In this work, we propose a flexible preemption scheme, based on the PDP, where the relative burst sizes are considered in the preemption decision too. Our goal is to capture the tradeoff between throughput maximization and quality degradation.

Preemption probability ( $P$ ) as defined in the proposed preemption policy depends on the relative significance of two bursts competing for the same resources. The significance of a burst depends on its length (a large burst has a more significant impact on throughput than a smaller one when dropped) and the priority of the service class it belongs to. So, the probability of a class-1 burst preempting a class-0 burst depends partly on their relative burst size, and partly on a constant preemption probability,  $P_0$ .  $P_0$  parameter leads to class separation, and provides performance guarantees to high priority bursts, regardless of their size. It must be added here that, a class-0 burst never preempts a class 1 burst, whereas between two bursts of the same service class the smaller one is always preempted. The definition of  $P$ , given that  $B_{new}$ ,  $CoS_{B_{new}}$  is the size and class identifier of the incoming burst

respectively and  $B_{old}$ ,  $CoS_{B_{old}}$  is the size and class identifier of an already scheduled overlapping burst, preemption probability  $P$  is defined as follows:

$$P = \begin{cases} P_0 + (1 - P_0) \cdot \frac{B_{new}}{B_{old}}, & CoS_{B_{new}} > CoS_{B_{old}} \\ 1, & B_{new} > B_{old} \\ 0, & B_{new} < B_{old} \end{cases}$$

In what follows, we study, how  $P_0$  selection affects throughput and video quality. In essence,  $P_0$  is expected to provide control on the Throughput / Quality tradeoff.  $P_0$  values close to 1 will tend to favor video bursts over even much larger data bursts, while  $P_0$  values close to 0 will favor larger data bursts, regardless of their service class.

#### B. Theoretical Model

In this section, we provide a mathematical model to derive the blocking probabilities of priority classes. Our goal is to derive burst loss ratio as a relation to preemption probability  $P$ , and show how the variation of  $P$  affects the loss ratio of both priority classes, namely a best-effort class (or class-0) and a real time class (or class-1). In what follows, we model an OBS node as an  $M/G/k/k$  queuing system, and evaluate the blocking probabilities of priority classes. Suppose that both priority classes generate bursts according to a Poisson process, with rates  $\lambda_0$  and  $\lambda_1$  respectively and exponential burst size distribution. In [6] it is argued that conservation law is satisfied in a multi class OBS system, where mean burst length is equal for all classes. Therefore, the overall loss probability can be approximated by the well-known ErlangB formula. In the two-class OBS node, modeled with  $W$  wavelengths, offered load  $\rho_0, \rho_1$  (for class-0 and class-1 respectively), the overall burst loss ratio of the traffic mix can be approximated as  $P_B(0,1) = \text{ErlangB}(\rho_0 + \rho_1, W)$ , or:

$$P_B(0,1) = \frac{(\rho_{0,1}^W / W!)}{\left( \sum_{m=0}^W \frac{\rho_{0,1}^m}{m!} \right)} \quad \text{Eq. 1}$$

The question remains how to calculate loss probabilities of individual classes. Unlike the queuing model used in [7], we do not assume strict priorities among priority classes and therefore ErlangB formula does not provide an accurate estimation of class-1 burst loss ratio. ErlangB formula expresses the probability that all wavelengths are reserved when a new burst arrives, which essentially leads to a burst loss in an  $M/G/k/k$  system due to the lack of buffering. However, this is not always true for a high priority burst, which can preempt an already scheduled lower priority class-0 burst. Thus, class-0 bursts face a higher blocking probability than ErlangB, whereas class-1 bursts a smaller one. The relation between class-0 and class-1 burst loss ratio using total probability theorem can be expressed as:

$$P\{B \text{ is lost}\} = p_0 \cdot P\{B \text{ is lost} | B \in \text{Class-0}\} + p_1 \cdot P\{B \text{ is lost} | B \in \text{Class1}\} \quad \text{Eq. 2}$$

In the above formula  $p_0$  and  $p_1$  are the probabilities of a burst belonging to class-0 and class-1 respectively. As long as burst

preemptions are relatively rare, a good approximation of  $p_0$ ,  $p_1$  which was also used in the multi-class model in [7] is:

$$p_i = \lambda_i / (\lambda_0 + \lambda_1), i = 0, 1 \quad \text{Eq. 3}$$

In what follows, we approximate the blocking probabilities of class-0 and 1 bursts, as  $P_B(1) = P\{B \text{ is lost} | B \in \text{Class-1}\}$  and  $P_B(0) = P\{B \text{ is lost} | B \in \text{Class-0}\}$ .

We note here that calculating the loss rate of class-1 bursts can be directly applied in Eq. 2 to provide the loss rate of class-0 bursts, since the loss ratio of the traffic mix is easily calculated using ErlangB formula. A new burst that belongs to class-1 is lost if it finds all wavelengths reserved, which happens with probability  $P_B(0,1)$  and when one of the following applies:

- All wavelengths are reserved with class-1 bursts (so preemption is not possible). If we accept that channel reservations are independent, then it directly follows from Eq. 3 that this happens with probability  $p_1^W$ .
- There is at least one wavelength reserved by class-0 burst, which can be preempted to serve a class-1 burst but the reservation fails. In the proposed model, preemptions are performed according to the preemption probability  $P$ , which depends on burst size distributions. Thus, the preemption fails with a probability  $(1 - P)$ . Therefore, the blocking probability of a class-1 burst is expressed as:

$$P_B(1) = P_B(0,1) \cdot [p_1^W + (1 - p_1^W) \cdot (1 - \bar{P})] \quad \text{Eq. 4}$$

### C. Model Validation

In order to confirm the validity of the proposed model, we perform simulation over a single OBS node, which employs the proposed scheduling scheme. We obtain class-0 and class-1 blocking probabilities both via simulation and through our theoretical model, using Eq. 2 and Eq.4. Burst arrivals are assumed exponentially distributed, with the same inter-arrival time distribution for both classes. The burst size is also exponentially distributed, with the same mean burst size for both classes, leading to an M/M/k/k system. We can obtain class-1 blocking probability directly from Eq. 1 and 4 substituting  $p_{0,1} = \lambda/\mu$ , where parameter  $\lambda$  stands for the aggregated arrival rate of bursts belonging to both classes, and  $\mu$  stands for the burst service rate. Class-0 blocking can be indirectly calculated through class-1 blocking and ErlangB, through Eq. 2. In our simulation scenario, we have assumed an OBS node with 2 wavelengths per fiber, and 2.5Gbps capacity per channel. Figure 1 shows the blocking probabilities for both priority classes, obtained both analytically and through simulation. As expected, the theoretical model accurately predicts the blocking probability of both priority classes.

It must be noted here that to simplify our model we do not take into account offsets between bursts and control packets, since they do not directly affect blocking probability, but they can affect preemption probability. In [8] it is argued that preemption probability depends on the ratio of the burst offset time to the average burst size. The authors prove analytically and through simulation that when this ratio is 5 or more (equivalently: burst offset time is at least 5 times the average

burst size) then the preemptions always succeed. In our model, we assume that the offsets are large enough, so that preemption probability only depends on the selection of  $P_0$ .

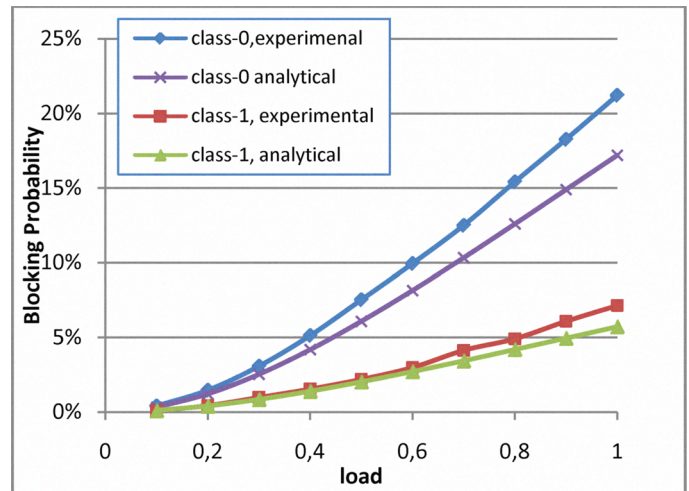


Fig. 1: Model validation. Blocking probabilities for class-0 and class-1 bursts classes obtained both experimentally (exp) and analytically (anal).

### III. PREEMPTIVE LAUC-VF BASED CHANNEL SCHEDULING

In this section, we describe the implementation of a variation of LAUC-VF termed as PLAUC-VF, using preemptions based on the proposed preemption policy, detailed in Section II. Several scheduling algorithms have been proposed in the OBS related bibliography, due to their vital role in OBS performance. Scheduling algorithms can be classified in the ones without Void filling (like LAUC and Horizon) and with void filling (like LAUC-VF) and finally, the ones with preemption (like PLAUC-VF).

Fig. 2 illustrates the principle of operation of LAUC-VF scheduling algorithm, which forms the basis of PLAUC-VF. In LAUC-VF, the scheduler keeps track the free periods in each channel. Since, it is vital to keep computational complexity low, the scheduler only stores two time intervals, the semi-unbounded interval ( $\text{sched\_end}, \infty$ ) as well as the void period ( $\text{void\_start}, \text{void\_end}$ ), between the two most recent burst reservations. Thus, LAUC-VF only stores three time values per channel, and it only makes two comparisons to decide whether a new burst can be scheduled in a channel or not.

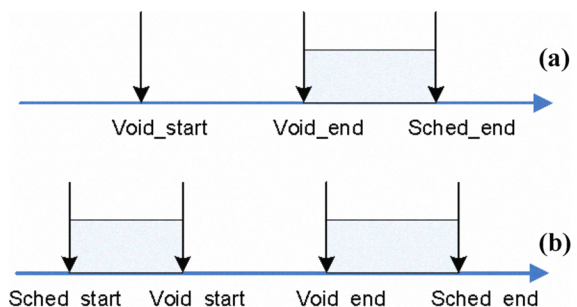


Fig. 2: Principle of operation, for (a) LAUC-VF and (b) preemptive LAUC-VF



If a reservation request can be served by more than one channel, then the channel that minimizes the remaining idle period is chosen. When a new reservation request arrives, the scheduling unit performs the following three steps:

- It scans all channels for a free interval to schedule the forthcoming data burst, either into the void or after `sched_end` time value.
- It finds the “best fit” if multiple channels were found (where “best fit” is considered the option that minimizes the remaining idle period) or reports failure if none was found
- It updates the `void_start`, `void_end` and `sched_end` values of the selected channel, if one was found, and reports success.

#### A. PLAUC-VF Channel Scheduling Algorithm

In PLAUC-VF the state information stored per channel is augmented, to include burst length, class of service, as well as unique burst identifier for scheduled bursts. This information is typically required in preemptive algorithms to base preemption decisions, and make the reconfiguration of OBS nodes possible. At the same time complexity has to remain low, to keep the reservation process efficient. In our implementation, we make use of an extra time value, `Sched_start` (see figure 1b). This gives the scheduler the ability to keep track of and preempt the two most recently scheduled bursts. For each one of them the class of service and the burst identifier has to be stored, while burst sizes are derived from the time values. The channel selection phase remains identical to the one used in LAUC-VF, using the same time values `void_start`, `void_end` and `sched_end`. If the channel selection fails, then the more computationally demanding preemption phase follows. But, due to the fact that preemptions are relatively rare, average computational complexity remains low, at the same levels as LAUC-VF. Preemption decisions in our implementation of PLAUC-VF are based on the probabilistic preemptive scheme detailed in Chapter II. When a new reservation request arrives, the preemption-capable scheduling unit follows these steps (see Figure 3 for an algorithm illustration):

1. First, it scans all channels for an idle period to schedule the burst. If voids are found in more than one channel, the one that minimizes the remaining idle period is chosen, as in LAUC-VF.
2. If no voids are found, we conclude that there is at least one overlapping burst in each channel. Then, the scheduling algorithm iterates over overlapping bursts, and decides whether one of them has to be preempted, in order to free resources for the newly arrived burst. The decision is based on the Preemption Policy, as described in section II. As an example, if we accept that all bursts in Figure 3 belong to the same class of service, burst 0 would be preempted, since it is the smallest among all overlapping bursts and the reservation request, and the new burst would be scheduled in its place (see Figure 3)
3. If a burst was preempted in step 2, as an optional step the scheduler can attempt to reschedule the preempted burst at another unused channel. In what follows, the PLAUC-VF variation that performs this optional step is denoted as PLAUC-VF-ext. If more than one channel is available for rescheduling, the one that minimizes the remaining

idle period is chosen by PLAUC-VF-ext. On the other hand, if the rescheduling step fails, the preempted burst is dropped. In our example (see Figure 3), preempted burst 0 can be moved to data channel 1.

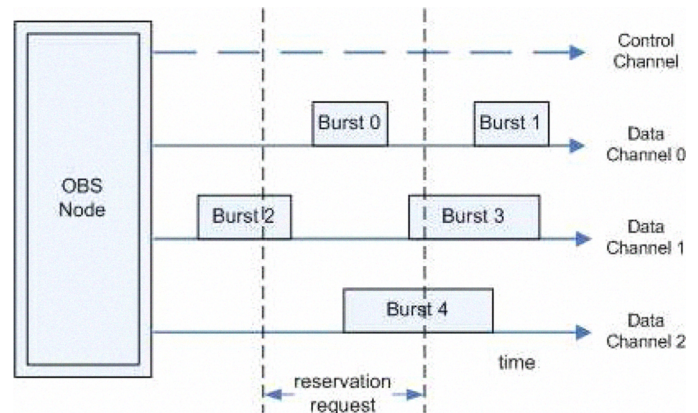


Fig. 3: Algorithm illustration

Next, we provide a formal description of the PLAUC-VF. The function `search_channel(i)` searches for an idle period in channel `i` to schedule the newly arrived burst. The void found or the overlapping burst(s) are stored in the arrays `void[]` and `ovburst[]` respectively. Then, the function `find_best_fit` scans the voids (if more than one was found) and returns the one that minimizes the remaining idle period, or an empty set if none was found. If no void was found, `preempt()` function scans all overlapping bursts in `ovburst[]` array and based on the preemption policy may select one of them to be preempted, so that the newly arrived burst can be scheduled. If an overlapping burst is preempted, the `find_best_fit` function rescans the `void[]` array to reschedule the preempted burst (`pburst`), or it has to be dropped. Fig.4 summaries the pseudo code for the proposed PLAUC-VF scheduling algorithm.

```

for i=0 to num_channels
    [void[i], ovburst[i]] =
    search_channel(i)
end i
result = find_best_fit(void, newburst);
if result != empty
    return result;
else
    [result, pburst] = preempt(ovburst, P)
    if result != empty
        find_best_fit(void, pburst)
        return result;
    end if
end if
return failure

```

Fig. 4: Pseudo code for PLAUC-VF Channel Scheduling Algorithm

#### B. Experimental analysis

The preemptive scheduling algorithm was evaluated through experimentation, in the Network Simulator (ns-2) environment. We modeled an OBS network over the 14-node NSF network topology. The modeled OBS network supports two classes of service, one best effort class for data transmissions and one real time class for video transmissions.

The burst assembly process is performed in the OBS network's ingress nodes, using a timer-based aggregation algorithm. Packets from different classes arriving at the ingress node are allocated in distinct burst assembly queues, and are never mixed in the same burst. Our goal is to investigate the trade-off between throughput maximization and QoS provisioning, and to demonstrate the ability of our scheduling algorithm to provide service differentiation. Additionally, we will also show how the selection of  $P_0$  parameter affects throughput and video quality. In our simulations, we consider real-world performance metrics, which are specific to each priority class. For low priority traffic (class-0), the performance metric considered is packet loss ratio. For video traffic (class-1), we consider perceived video quality metric, which accurately describes the impact of burst losses on video streams. The video specific metrics were obtained with Evalvid framework [4], which is a set of tools for the generation of video traffic, and the evaluation of video transmissions. Combined with our ns-based OBS simulator it results in a powerful video-over-OBS modeling tool. Our simulator's architecture consists of the following modules:

- An OBS network module for ns-2 simulator, implemented in C++ and partly based on OIRC obs-ns simulator.
- A supFRP [3] traffic generator, which uses superposition of fractal renewal processes to generate self-similar traffic.
- An MPEG-4 encoder and decoder (ffmpeg open source tool)
- A video sender utility, which generates a video trace file with the intrinsic details of the mpeg4 file (specifically video frames, packets within frames, and frame transmission times)
- A video traffic generator, which generates video traffic according to the aforementioned video trace file.
- A set of tools that evaluate some perceptual-based metrics of video transmission, and specifically MOS score.

In our simulation scenario, video traffic is generated through trace-driven traffic sources, which are part of the Evalvid framework. They take as an input the video trace file which is produced by the Video Traffic Generator module, after parsing the reference MPEG4 video file. The reference video file is a 2-minute video clip, with a video resolution of 352x266 and a constant bitrate of 1Mbps. Low priority internet traffic is modeled with supFRP traffic sources, which generate self-similar traffic, using superposition of fractal renewal processes. In every OBS edge node, we employ one supFRP traffic generator with a Hurst parameter of 0.75, and multiple video streams with random destinations.

During each simulation cycle, we obtain detailed log files of burst transmissions as well as burst and frame losses. At the end of the simulation cycle, these are used to calculate the loss ratio, as well as the video perceptual quality. In the video quality calculation phase, we focus on a specific video stream, reconstructing the reference video file by taking into account frame losses that occurred during transmission over the OBS network. The perceptual video quality is a subjective metric, and it is described by MOS (Mean Opinion Score). MOS is the (subjective) human impression of video quality as

described in Table 1, and it is normally assigned by a group of experts, but it can be approximated with PSNR calculations on video frames. PSNR score express the degree of noise introduced through burst losses. PSNR calculations are performed by Evalvid framework, by comparing reference video frames with reconstructed video frames. The conversion table used to obtain MOS metric from PSNR values is the following:

TABLE I. PSNR to MOS conversion table

PSNR [dB]	MOS	Impairments
> 37	5 (Excellent)	Imperceptible
31- 37	4 (Good)	Perceptible, but not annoying
25 – 31	3 (Fair)	Slightly annoying
20 – 25	2 (Poor)	Annoying
< 20	1 (Bad)	Very annoying

It should be noted here that burst loss ratio is not directly related to throughput, since variable burst sizes are used, and burst losses are not uniformly distributed due to preemptions. Thus, larger bursts have a smaller blocking probability. Therefore, in what follows, packet loss ratio is used as a performance metric for the low priority class. In Figure 5, we compare standard LAUC-VF with PLAUC-VF and PLAUC-VF-ext. The latter, in addition to preemption supports re-scheduling of preempted bursts to an available channel (if one is found). It must be noted here, that since LAUC-VF does not support multiple priority classes, we set  $P_0 = 0$ , and therefore preemptions are only based on burst size criterion. From Figure 5, it can be seen that the preemption of small data bursts by larger ones does pay off (see PLAUC-VF curve), yielding a smaller blocking probability. The extra complexity of re-scheduling preempted bursts results in a further (though marginal) improvement of the scheduling efficiency (see PLAUC-VF-ext curve).

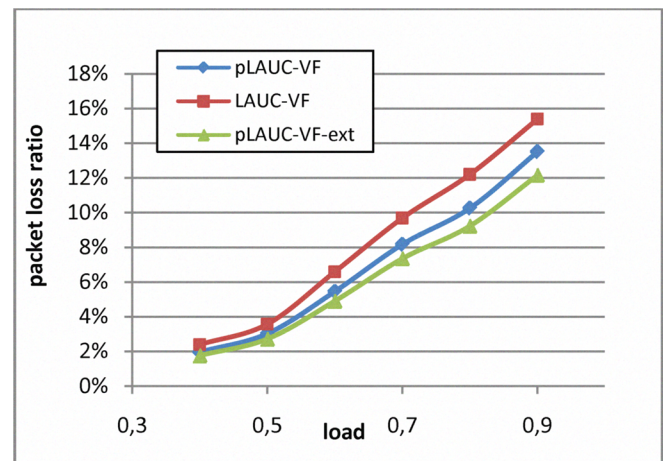


Fig. 5: Packet loss probability comparison of standard LAUC-VF with PLAUC-VF and PLAUC-VF-ext for  $P_0 = 0$

Figures 6 and 7 display how  $P_0$  selection affects throughput and video quality. It can be clearly seen as expected, when  $P_0$  approaches 1 video quality is favored against loss rate (and equivalently throughput). The opposite occurs when  $P_0$  tends to 0. To this end, we may argue that the proposed scheduling



scheme can actually support QoS differentiation over different classes of service, in a very flexible and efficient way. It is parameterized by a simple parameter, that is  $P_0$ , which expresses the guaranteed preemption probability and effectively controls class separation.

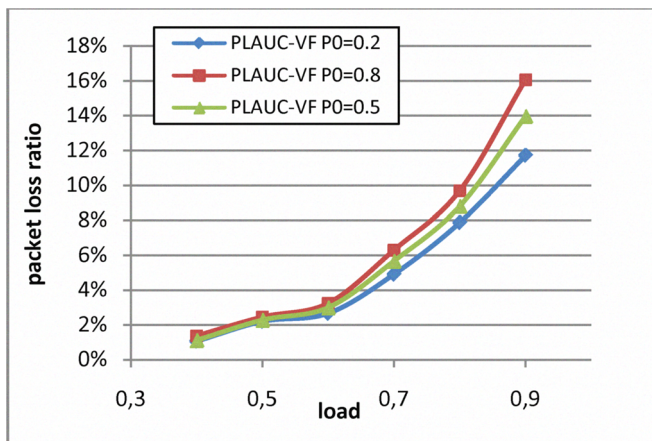


Fig. 6: Packet loss probability comparison for low priority class bursts for PLAUC-VF-ext with  $P_0 = 0.2, 0.5$  and  $0.8$ .

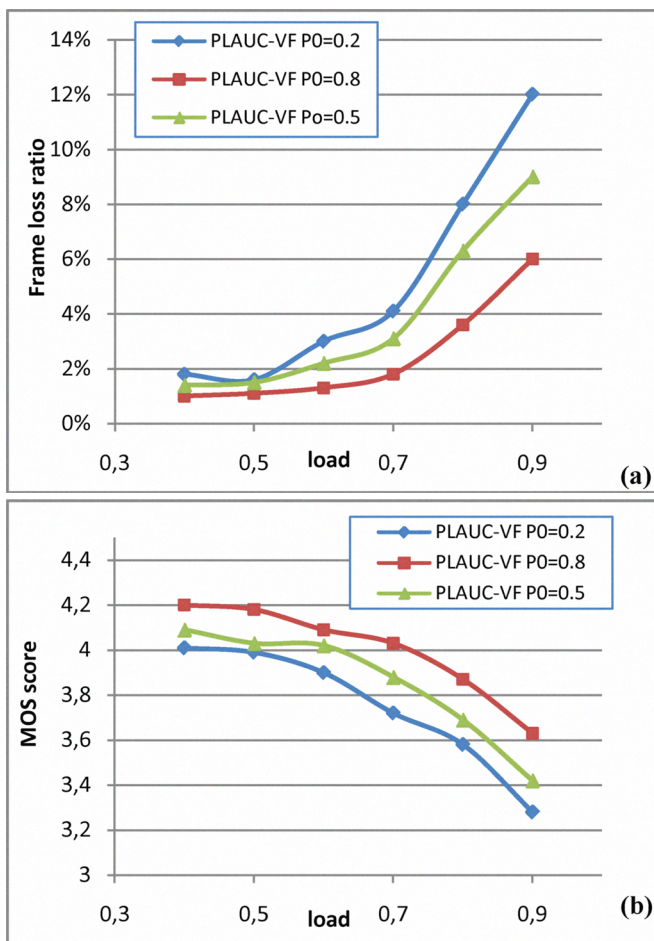


Fig. 7: Comparison of PLAUC-VF-ext with  $P_0 = 0.2, 0.5$  and  $0.8$  in terms of (a) video frame loss ratio, and (b) video perceptual quality (MOS score).

#### IV. CONCLUSIONS

In this work, a new multi-class preemptive scheduling scheme was proposed. It was demonstrated that proposed scheduling scheme can actually provide QoS differentiation in different service classes, in a very flexible and efficient way, parameterized by a simple parameter, which expresses guaranteed preemption probability. Proposed scheduling scheme clearly performs better than LAUC-VF with a slightly increased complexity.

#### ACKNOWLEDGEMENTS

The work described in this paper was carried out with the support of the BONE-project ("Building the Future Optical Network in Europe"), a Network of Excellence and the DICONET-project, funded by the European Commission through the 7th ICT-Framework Tito Raúl Vargas work was performed, while visiting University of Patras, through a BONE mobility action.

#### REFERENCES

- [1] Y. Xoing, M. Vandenhoute, and C. Cankaya, "Control architecture in optical burst-switched WDM networks", IEEE J. on selected areas in communications, 18:1838–1851, Oct. 2000
- [2] A Kaheel, H Alnuweiri, "A strict priority scheme for quality-of-service provisioning in optical burst switching networks", in proceed. of ISCC 2003
- [3] M. Yuksel, B. Sikdar, K.S. Vastola, and B. Szymanski, 'Workload generation for ns Simulations of Wide Area Networks and the Internet', in proceedings of CNDS Conference, 2000
- [4] J. Klaue, B. Rathke, and A. Wolisz, "EvalVid - A Framework for Video Transmission and Quality Evaluation", In Proc. of the 13th International Conference on Modelling Techniques and Tools for Computer Performance Evaluation, Urbana, Illinois, USA, September 2003.
- [5] H. Øverby, N. Stol, Providing quality-of-service in optical packet/ burst switched networks with the preemptive drop policy, In: Proceedings of the Third International Conference in Networking, (ICN), vol. 1, pp. 312–319, 2004
- [6] M. Yoo, C. Qiao, and S. Dixit, "QoS performance of optical burst switching in IP-Over-WDM networks," IEEE J. Select. Areas Commun., vol. 18, pp. 2062–2071, Oct. 2000.
- [7] H Le Vu, M Zukerman, "Blocking probability for priority classes in optical burstswitching networks", IEEE Communications Letters, 2002
- [8] M. Yoo and C. Qiao. A new optical burst switching protocol for supporting quality of service. In *SPIE Proceedings, All Optical Networking: Architecture, Control and Management Issue*, 1998
- [9] J Li, C Qiao, J Xu, D Xu, "Maximizing throughput for optical burst switching networks", IEEE/ACM Transactions on Networking (TON), 2007
- [10] C. Qiao and M. Yoo, "Optical burst switching (OBS)-A new paradigm for an optical internet," J. High Speed Networks, vol. 8, no. 1, pp. 69–84, 1999.