Performance evaluation of channel scheduling algorithms with different QoS classes	Đánh giá hiệu suất của các thuật toán lập lịch kênh với các lớp QoS khác nhau 8 h 55
Abstract— This paper compares the performance of different scheduling algorithms, with and without void filling, considering a complete Optical Burst Switched (OBS) network scenario, given by edge and	Tóm tắt-Bài báo này so sánh hiệu suất của các thuật toán lập lịch khác nhau, khi có và không có lấp đầy khoảng trống, xét trường hợp mạng chuyển mạch burst quang (OBS) hoàn chỉnh, đặc trưng bởi các bộ định tuyến biên và lõi dưới các điều kiện

core routers, under real traffic conditions.Performance is investigated taking into account both burst loss probability and the complexity of the algorithms, evaluated in terms of scheduling time. For an effective comparison both aspects have to be considered in order to understand if an algorithm which provides low burst blocking probabilities is actually feasible or not. In this paper the authors propose a scheduling algorithm which represents a good trade-off between burst blocking performance and scheduling time.	lưu lượng mạng thực. Khi đánh giá hiệu suất, chúng ta xét đến cả xác suất mất burst và độ phức tạp của các thuật toán (được đánh giá thông qua thời gian lập lịch). Để việc so sánh có hiệu quả, chúng ta cần phải xét đến cả hai khía cạnh để tìm hiểu xem thuật toán có xác suất chặn burst thấp có thực sự khả thi hay không. Trong bài báo này, tác giả đề xuất một thuật toán lập lịch đảm bảo dung hòa giữa hiệu suất tắc nghẽn burst và thời gian lập lịch.
The employment of Dense Wavelength Division Multiplex-ing (DWDM) has increased the amount of bandwidth available thus meeting the ever increasing bandwidth demands. Many new optical network architectures and transfer modes have been recently studied to effectively exploit this huge amount of bandwidth to develop a fully optical Internet, where signals carried within the network never leave the optical domain [1][2].	[redacted]
In Optical Circuit Switching (OCS) data are carried over optical pipes, called lightpath, that consist of one or more wavelengths used to provide a connection between sender and receiver. OCS is easy to implement, but is not flexible	[redacted]

and does not allow to effectively exploit the huge available bandwidth.	[redacted]
Optical Packet Switching (OPS) is the implementation in the optical domain of the packet switching concept. Theoretically, OPS is the best paradigm that allows to implement the so-called optical transparent network [2]. However, OPS is not easy to implement, as it needs advanced optical technologies, such as optical buffer and optical logic, that are immature.	[redacted]
In recent years a new paradigm has been proposed, called Optical Burst Switching (OBS), that combines the best of circuit and packet switching [3] [4] [5]. OBS can be seen as a middle term solution towards all optical packet switching whose goal is to improve wavelength utilization and sharing by introducing a dynamic wavelength management. In OBS networks data never leave the optical domain: for each data burst assembled at the network edge a reservation request is sent in advance as a separate control packet. There are two kind of nodes: edge and core routers. The main function of edge nodes is the burst assembly: these nodes are the gateways between traditional ”electrical” networks and high speed op-	[redacted]

tical networks, they must collect IP datagrams and assembly them into burst according to proper assembly algorithms.

Core nodes, on the other hand, deal with optical data bursts and the related control packets. They have to set up on the fly internal optical paths for switching bursts and take them hop-by-hop closer to their final destination. In addition, the offset time allows the core router to be bufferless, avoiding then the employment of optical memories, e.g. fiber delay lines, required on the contrary by optical packet switching. The control packet carries relevant forwarding informations, as the next hop, the burst length and the offset time. It precedes the data burst by a basic offset time, that is set to accommodate the non-zero electronic processing time inside the network, and dinamically set up a wavelength path whenever large data flows are identified and need to traverse the network. Only the control packet is converted between optical and electronic domains, therefore is the only information delayed because of the conversion.The most used reservation protocol in OBS network is Just-Enough-

[redacted]

Time (JET) [6]. JET is a delayed reservation protocol which allows to reserve a wavelength channel just for the burst duration, starting at the predicted burst arrival time. If the reservation is successful, the control packet is forwarded to the next node, otherwise the correspondent burst is blocked. The use of this reservation protocol doesn't allow to fully utilize the bandwidth: in every channel there is portion of unused bandwidth between bursts that have made a reservation, called void. In order to get a good utilization of the available resources, an efficient reservation process is required. To this end, effective scheduling algorithms have to be developed [5]. Even if these algorithms have been widely investigated ([4],[7], [8], [9]), in our opinion a more detailed performance comparative investigation would be appropriate.

In this paper two main performance figures are taken into account simultaneously: burst blocking probability and scheduling time. Different scheduling algorithms have been implemented and integrated in a object-oriented simulator where they are evaluated in a real network scenario with,g., different traffic profiles. Furthermore, a novel scheduling algorithm is proposed which comes to be a

[redacted]

[redacted]

good trade-off between burst blocking performance and scheduling time. The rest of the paper is organized as follow. In Section II the scheduling algorithms evaluated in this work are presented. In Section III the adopted network topology and the traffic settings are described. Numerical results are then collected in Section IV, while the concluding remarks are in Section V.

[redacted]

SCHEDULING ALGORITHMS

[redacted]

When a core node in a OBS network receives a control packet, it must decide which channel to reserve to forward the incoming data burst. The choice of the wavelength to use is made by a scheduling algorithm. If the reservation is successful, the control packet and the data burst are sent to the next node; otherwise the burst is blocked and eventually dropped if no Fiber Delay Lines (FDLs) are employed. The scheduling algorithm efficiency is an interesting problem for many reasons. It is widely known that a good scheduling implies a reduction of blocking probability.

Gaps between bursts scheduled on a channel are called voids. One of the basic goals of these algorithms is an efficient channel utilization: the problem to exploit the void

[redacted]

intervals is usually called void filling. Furthermore, this task gets more difficult in case of different offset time values, i.e., differ-ent extra-offsets between data burst and control packet, for example in presence of different service classes. Scheduling algorithms can be classified into two groups: without [4] and with [7] void filling.

Moreover, an efficient scheduling algorithm does not have to be too complex, i.e., not be time consuming. When a control packet arrives at a core node, the decision of the scheduling algorithm must be fast enough in order to meet time constraint given by the offset time.

It is worthwhile noting that if JET is employed as reserva-tion method, voids can be actually filled with new data burst only if different extra-offset values are set, i.e., in a scenario with different service classes. As a matter of fact, in presence of equal time offsets it is not possible for a new data burst to be scheduled in a void created by previously arrived bursts. This can be easily seen in Figure 1.

In order to give a brief description of scheduling algorithms investigated in this work, it is useful to introduce the concept of Horizon time [4] of a channel: it is the time after which no reservation has

[redacted]

[redacted]

been made on that channel. First Fit Unscheduled Channel (FFUC) is an algorithm proposed in [10] without void filling. Using FFUC the databurst is sent to the first channel, according to a predefined order e.g. fixed or round robin, among the channels in which the horizon time is smaller than the arrival time of the new data burst. This algorithm is the simplest algorithm for burst scheduling and its main advantage is a very low complexity.	[redacted]
Latest Available Unscheduled Channel (LAUC) is an alter-native algorithm without void filling [4]. It searches among the channels in which the Horizon time is smaller than the arrival time of the new data burst and selects the channel with the longest horizon time in order to minimize the created void. It has a greater complexity than FFUC, yet it is still quite simple and fast.	[redacted]
Latest Available Unscheduled Channel with Void Filling (LAUC-VF) is a channel scheduling algorithm proposed in [7]. Let t be the arrival time of a data burst with a duration L. First, the scheduler finds the outgoing data channels, available for the time interval (t, t + L), and then selects the latest available data channel, i.e. the channel having the smallest gap between t and the	[redacted]

end of last data burst, called starting void . This criterion is known as Minimum Starting Void or Min-SV [9]. To implement LAUC-VF, as well as for all the other algorithms with void filling, information regarding data channels and voids must be stored and kept updated in data structures, not just their horizon time. Consequently, this kind of algorithms have much higher complexity than the ones without void filling.

[redacted]

First Fit Unused Channel with Void Filling (FFUC-VF) [7] is an algorithm based on the same criterion of FFUC but available channels are both the channels in which the Horizon time is smaller than the arrival time of the new data burst and the channels with a void that could contain the data burst. In FFUC-VF the void is chosen using the Min-SV principle and the channels are tested in a round robin manner. In Section 4, comparative performance evaluation is carried on considering both the burst blocking probability and the execution time at runtime as performance figures.

[redacted]

A. Our proposed algorithm

[redacted]

In this paper we propose a scheduling algorithm suitable for two classes of service,

[redacted]

called Latest Available and First Fit Unused Channel with Void Filling (LA-FFVF), whose goal is to provide a good trade-off between burst blocking probability and complexity.

LA-FFVF applies different scheduling criteria to select a channel for arriving bursts depending on their class of service.

As previously mentioned, with JET as reservation mechanism, void filling is possible only when there are different classes of service managed through different extra-offset values. High priority bursts are assigned extra-offset times with the aim to get lower blocking probability values. On the other hand, low priority bursts, which have not extra-offsets, could reduce losses using void filling. Thus, LA-FFVF represents a sort of ”hybrid approach” in which LAUC is used for high priority bursts, to keep low the complexity being the priority guaran-teed by extra-offset, whereas FFUC-VF is employed for low priority bursts in order to exploit voids and get a ”reasonable” complexity.

Figure 2 illustrates the LA-FFVF algorithm. In this case there are three data channels and one control channel. At the beginning, there are two bursts which occupy data

[redacted]

[redacted]

[redacted]

channels 1 and 2 until t2 and t1, respectively. When the control packet A1 arrives, A1 data burst is scheduled on channel 1 because is the latest available channel. Following the same criterion, data burst A2 is scheduled on channel 2. On the arrival of control packet B1, channel 3 is free and B1 data burst is here scheduled following First Fit criterion. On the arrival of A3 control packet, the latest available channel at t12 time is channel 2. Then, B2 data burst is scheduled in channel 2 using FFVF. Finally, B3 data burst is scheduled in channel 1 following first fit criterion.

[redacted]

NETWORK AND TRAFFIC SETTINGS

The network topology investigated in this paper is reported in Figure 3. The reference network topology is composed of 5 nodes: two edge nodes, one core node and two destination nodes. Edge nodes collect traffic from legacy 10GEthernet net-works and generate optical bursts adopting the Tmax assembly algorithm, classified into two QoS classes [11]. OBS nodes support JET.

[redacted]

Incoming traffic into edge nodes is supposed to be M/P areto. Given that x is the mean duration time of the ON periods, on each incoming link the offered load ρ is equal to λ x, where λ is the mean arrival rate. Aiming to model self-

[redacted]

similar traffic, Pareto-distributed ON periods are considered. Indicating by X the random variable which represents the duration of the ON period, its cumulative distribution function

where the constant k is the smallest possible value of the random variable X, expressed as multiple of a basic unit (e.g. the size of the IP datagram); $0 < \alpha < 2$ in order to have a heavy-tailed distribution. Note that under the hypothesis $0 < \alpha < 2$, X has infinite variance. A measure of the degree of self-similarity is the Hurst parameter, $H(\alpha) = 3-\alpha$, adimensionless quantity which ranges between 0.5 (no self-similarity, $\alpha = 2$) and 1 ($\alpha = 1$). Bursts are then classified as A and B, class A is here attributed the highest priority.

In the burst assembly process class A bursts have the highest priority and are thus given an extra-offset time [11]. Let pA and pB represent the occurrence probabilities of the two burst classes.

The OBS core node architecture (Figure 4) is equipped with M × M optical fibers capable of supporting N wavelengths each. This OBS core node mainly consists of input Fiber Delay Lines (FDLs), an optical switching system and a control unit.

The input FDLs are used to delay incoming data bursts, thus allowing the control unit

[redacted]

[redacted]

[redacted]

to have enough time to process the associated control packet.	[redacted]
We suppose that the optical switching matrix is equipped with a set of full range wavelength converters and that the system is bufferless, i.e., no fiber delay lines are available in order to resolve contention for an output fiber, output wavelength.	[redacted]
The control packets are processed by the control unit. The control packet is converted from optical to electrical domain in order to obtain information related to data burst. The control unit works as an electronic router and keeps the routing information. The scheduling algorithm is implemented into the control unit as well. The scheduler is responsible of scheduling and switching data burst on an output data channel and also of transmitting the control packet. There is a scheduler for each data and control channel pair and each scheduler only needs to keep track of the busy/idle periods of a single outgoing data and control channel. It first reads the arrival time and the burst duration in the control packet and then, using a given scheduling algorithm, it searches for an idle output data channel.	[redacted]
NUMERICAL RESULTS	[redacted]
Numerical investigations are	[redacted]

performed through an object-oriented simulator developed in C++ which simulates a com-plete Optical Burst Switched Network. Basic goal of this study is to compare different scheduling algorithms taking into account at the same time burst blocking probability and scheduling time. The incoming traffic is M/P areto with x = 512 byte, α equal to 1.2, and are assembled in the 2 edge nodes using Tmax assembly algorithm [11]. It is first assumed pA ≡ 0.5 and pB ≡ 0.5.	[redacted]
The assembly time for every service class is 2 µs. Further-more, M = 2 and N = 8 and the core nodes are equipped with 16 wavelength converters. Optical transmission devices allow for transmissions at 10 Gbit/s over each wavelength. Class differentiation is obtained by giving class A an additional extra-offset equal to 18 µs.	[redacted]
Simulations have been performed on a Pentium 4 platform running at 2.8 GHz. We have evaluated and compared the burst blocking probability and the scheduling time for every scheduling algorithm reported in Section II. The scheduling time is given by two components: the search time and the update time. The search time is the average time required to determine the output channel	[redacted]

B on which to schedule the data burst. The update time is B Bthe average time required to update the data structure, for example B B voids and horizon time for ev-ery channel. The information A related to channels utilization is managed using linked lists A Aimplemented in Standard Template Libraries. The values of b A Aurst blocking probability have been computed with a 95 % confidence interval using a t-Student distribution.

[redacted]

In Figure 5 burst loss probability for two different classes of service are reported as a function of the offered load with reference to the core node of Figure 4. As expected, since class A bursts have higher priority than class B, two distinct sets of curves are present, one for class A and the other for class B bursts. More interestingly, definitive and general conclusions can not be drawn concerning scheduling algorithms. If, for instance, the design goal is to provide blocking probabilities 10−3 for class B bursts, LAUC-VF and LA-FFVF seem the best since they support a load up to 0.5 and keep class A losses around 10−5. If, on the other hand, class A performance drive the design, FFUC-VF seems the best since it provides loss values less than 10−5 up to

[redacted]

almost a 0.6 load; however, the trade-off is with class B burst performance, now penalized with 1% loss.

[redacted]

Figure 6 reports the average scheduling time as a function of the offered load for the different algorithms. This figure shows that FFUC has the lowest scheduling time, between 5 and 7 μs, because it has the lowest search time. LAUC needs more time, since its search time takes longer than the simple round robin used by FFUC. The decreasing behavior of LAUC and FFUC is due to the ever decreasing number of free available channels which reduce the search time. Of course, correspondingly, burst blocking increases.

[redacted]

The algorithms with void filling, on the other hand, exhibits longer scheduling times, mostly due to longer update times. Most important, LAUC-VF algorithm requires roughly 10 μs more than FFUC-VF at 0.5 − 0.6 loads. As regards the LA-FFVF algorithm, it requires 6−8 μs less than LAUC-VF. Since void filling is performed for class B bursts only, scheduling of class A bursts, with LAUC, saves time because update time is now remarkably reduced with respect to LAUC-VF. LA-FFVF algorithm then appears as a good trade-off between burst

[redacted]

blocking performance and execution time.

Thus, depending on the design requirements, a solution with good performance in terms of blocking may need too long processing times and this depends on the current technology as well. The extreme case is when there is a strong constraint on the scheduling time, e.g. 6 − 8 μs: here the FFUC only is feasible due to its simplicity, but the burst loss remarkably degenerates in particular for class B bursts. Now, reasonably, the offered load must be limited to 0.3.

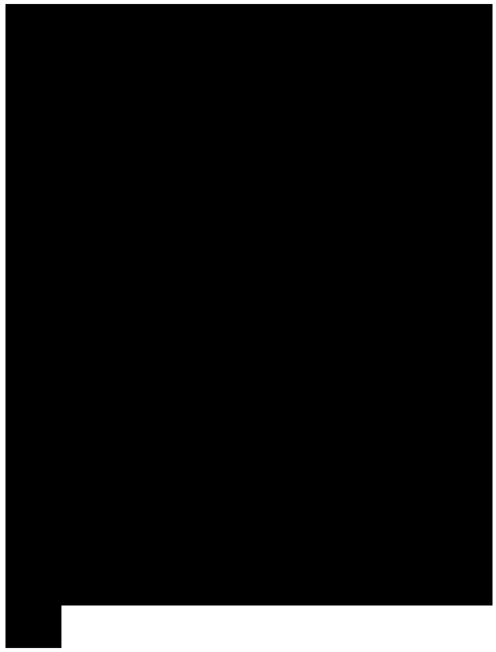

Figure 7 reports the burst loss probability for two different class of service as a function of the offered load with now pA = 0.2 and pB = 0.8. Again, two set of curves are present, one for class A and the other for class B bursts. Performance given by LAUC-VF and LA-FFVF are very similar also in this case with unbalanced traffic distribution and provide basically the same outputs in terms of system design previously discussed: for a target of loss for class B bursts equal to 10−3 a load up to 0.5 can be supported, which provide also class A loss values less than 10−5.

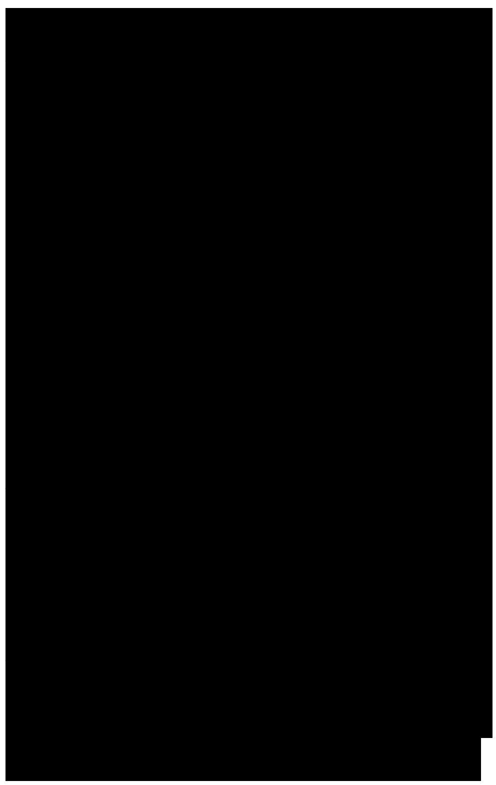

Figure 8 reports the average scheduling time as a function of the offered load for LAUC, LAUC-VF and LA-FFVF with pA = 0.2 and pB = 0.8. Again,

the decreasing behavior of LAUC appears and the difference between LAUC-VF and LA-FFVF is remarkable, between 5 and 7 μs. Their increasing behavior, on the other hand, is due to the ever increasing update time as a consequence of the void filling management.	[redacted]
V. CONCLUSIONS This paper has compared the performance of different scheduling algorithms with void filling, FFUC-VF and LAUC-VF, and without, FFUC and LAUC. A novel scheduling algorithm, LA-FFVF, has also been presented.	[redacted]
A complete Optical Burst Switched (OBS) network scenario has been considered. Edge nodes implement the Tmax assem-bly algorithm. Incoming IP datagrams have been modelled as M/P areto process and the outgoing bursts have been classified into two service classes, with different extra-offset values and different traffic profiles, balanced and unbalanced. Core routers have been assumed to be bufferless and equipped with a set of full-range wavelength converters.	[redacted]
The performance of different scheduling disciplines has been investigated taking into account both the burst loss probability and the complexity of the algorithms, evaluated in terms of scheduling time as	[redacted]

summation of search time and update time. Both aspects have to be considered in order to understand whether an algorithm that provides low burst blocking probabilities is also feasible.	[redacted]
The proposed LA-FFVF algorithm, which performs LAUC for high priority bursts and FFUC-VF for low priority bursts, has shown the best trade-off between burst blocking perfor-mance and complexity.	[redacted]