NUMERICAL ALGEBRA,
CONTROL AND OPTIMIZATION
Volume **1**, Number **4**, December **2011**

doi:10.3934/naco.2011.1.727

pp. **727–747**

ON THE OPTIMALITY OF PACKET-ORIENTED SCHEDULING IN PHOTONIC SWITCHES WITH DELAY LINES

Wouter Rogiest, Koen De Turck, Koenraad Laevens,
Dieter Fiems, Sabine Wittevrongel and Herwig Bruneel

Department of Telecommunications and Information Processing, Ghent University
St.-Pietersnieuwstraat 41; B-9000 Ghent, Belgium

Abstract. Addressing the bandwidth inefficiency problem of current IP over DWDM backbone switching, Optical Packet/Burst Switching (OPS/OBS) provide viable solutions, capitalizing on statistical multiplexing gain, through packet-oriented scheduling. To resolve packet/burst contention, the involved photonic switches contain wavelength converters and fiber delay lines, controlled through a channel and delay selection (CDS) algorithm. Recently proposed CDS algorithms all rely on heuristics, of which the optimality is unexamined to date.

This paper presents an in-depth analysis of the optimality of CDS algorithms. Methodologically, we rely on Markov chain analysis for performance evaluation, combined with a discrete Markov Decision Process formulation of the optimization problem, optimized for fast calculation, allowing to determine the exact optimum of a specific given setting of the switch, through numerical algebra solution techniques. Results point out that, for the basic switch setting assumed, of all known CDS algorithms, an algorithm called MING (MINimal Gap) is close to optimal, but never strictly optimal. Various graphs support this, showing that an algorithm optimal for any traffic load cannot (in general) be devised. Results for several other switch settings further confirm this, showing how known CDS algorithms might be modified, to attain improved control robustness.

1. Introduction.

1.1. **Context.** To realize the vision of ubiquitous broadband connectivity, the next-generation network has to be ready for the ever-growing bandwidth hunger of its users. The widespread interest in new web applications like High Definition on-demand video streaming is rapidly pushing the current network to its capacity limits. While the current backbone provides transport capacities of well beyond 10 Tbit/s per fiber, this capacity is only available for transmission from node to node. Current end-to-end communication suffers capacity loss from inflexible switching in intermediary nodes, urging for a more flexible approach to optical switching. Addressing this need, both optical burst switching (OBS) [7, 16] and optical packet

2000 *Mathematics Subject Classification.* Primary: 68M20, 90C40; Secondary: 60K25.

Key words and phrases. Performance evaluation, Markov decision processes, optical networks, queueing theory, FDL buffers.

The first and fourth author are Postdoctoral Fellow with the Research Foundation Flanders (FWO-Vlaanderen).

switching (OPS) [17, 29] provide future-proof alternatives for the next-generation network.

Following a packet-based approach, OPS and OBS allow to expand connection capacity by relying on the statistical multiplexing gain brought about by resource sharing for both switching and transmission. This however implies that packets/bursts potentially contend for the same channel at the same time, thus requiring a contention resolution scheme. Since random access memory (RAM) is not available for optical data, light is buffered by sending it through a piece of fiber delay line (FDL) of sufficient length. Since the number of available delay lines is limited to, say, 2 to 5 delay lines per node, the number of corresponding delays is also limited. The resulting contention resolution scheme comes down to a channel and delay selection (CDS) algorithm that has no counterpart in the electronic domain, in which the delay could be chosen arbitrarily, so that the issue is reduced to a channel selection algorithm.

Contention resolution schemes have been studied first in a network scenario with fixed-length packets, which is a usual assumption for OPS [9, 13]. Since the advent of OBS, also the counterpart with variable-length packets/bursts is studied, and often contrasted with the fixed-length case [1, 3–6, 8, 10, 11, 15, 20–22, 24, 28, 30]. The most common description of contention resolution, as discussed in [30], is done along three different dimensions, namely (i) wavelength (wavelength conversion), (ii) time (optical buffering) and (iii) space (deflection or alternate routing).

Firstly, wavelength conversion is enabled by Dense Wavelength Division Multiplexing (DWDM) technology, as it exploits the presence of multiple available wavelengths on the same fiber. The method consists in converting contending packets/bursts from an unavailable wavelength to an available one, with either no restriction (full conversion, so that all available wavelengths can be reached [3, 22]) or a restriction on either wavelength conversion range [14, 15] or number of available converters [8]. Secondly, optical buffering consists in delaying packets/bursts that find a resource unavailable by sending it through a piece of fiber of sufficient length, so that the resource is available again when the packet/burst leaves the delay line [1, 2, 9, 10, 20, 21, 24]. Thirdly, in a multi-fiber setting, deflection or alternate routing consists in routing packets/bursts that find a resource unavailable to another node through an different, available output fiber [5, 12]; this is done preferably topology- and congestion-aware [30]. The interplay of these dimensions is discussed extensively also in [5].

While deflection (or alternate) routing is a matter at network level, both wavelength conversion and optical buffering operate at the level of an individual node. As a result, wavelength conversion and optical buffering can (and should) be integrated consistently in a single contention resolution strategy. This integration has been referred to earlier as wavelength allocation in optical buffers [3, 26, 28], later as the channel and delay selection (CDS) problem [5], and is also the topic of the current paper.

1.2. **The CDS problem.** The core of the CDS problem is the discrete number of assignable delays. Due to this, on a given wavelength, it is in general not possible to schedule a packet/burst just after the transmission of a previous one. Therefore, on each outgoing wavelength, *gaps* occur, which are time periods during which no transmission takes place on the outgoing wavelength, even though packets/bursts are awaiting transmission in the optical buffer [25–28]. These periods are also referred to as *voids*, and can be considered as a strict capacity loss, with an effect

that can be best compared to an increase in traffic congestion [1, 10]. In principle, these gaps can be filled up by means of a void-filling strategy [26], by scheduling later-arriving bursts within the gaps that occur in the schedule. However, since this requires to maintain the relative position of each packet/burst present in the buffer, implementation is costly in terms of hardware. Furthermore, it has poor scalability, since the hardware complexity grows along with the maximal number of packets/bursts that can be processed simultaneously. For these reasons, a horizon-based algorithm is preferred, maintaining only real value per channel, namely the scheduling horizon [3, 4, 6, 15, 22, 25, 27, 28], and is also assumed here.

For horizon-based schedulers, a prime performance measure is the loss probability, since lower loss yields better overall performance. In a very similar setting (an inter-node reservation algorithm, instead of the CDS algorithm), Turner [25] already pointed out that horizon scheduling should be done gap-aware in order to minimize loss. The same observation is reported independently in [27], and termed LAUC (latest-available unused channel). For the setting we consider in this paper, however, Callegati and his co-authors [3] were the first to propose the CDS algorithm minimal gap (MING), always converting packets/bursts to the wavelength which results in the minimal gap size. In [3], it is shown that MING outperforms classical delay-oriented (or, queue-size-oriented) algorithms like Join-The-Shortest-Queue. In more recent work [4–6], taking into account other requirements such as preservation of packet/burst order resulted in "softer" versions of this algorithm, referred to as gap-oriented algorithms. However, always, it has been reasonably assumed that MING minimizes packet/burst loss, and so maximizes the throughput of the switching matrix.

1.3. **Motivation.** The aim of this paper is to show that not MING, but a different CDS algorithm realizes minimal loss in an optical packet/burst switch, and varies over different settings. As our results point out, the CDS algorithm for minimal loss hardly ever (if ever) coincides with MING. For the exact same state space size (and thus, comparable hardware complexity) and same FDL and channel set, the algorithm we obtain yields better overall performance; in several cases, the loss reduction is over 10 percent when compared to MING. As such, the main claim of this paper is straightforward: for the same hardware cost and hardware constraints, the obtained algorithm yields better performance, and is therefore preferred over MING as implementation variant. Initial results were presented in [19], but only under limiting assumptions (fixed burst size, degenerate buffer setting, buffer size $N = 2$). Opposed to this, [18] (5 page conference version) and the current contribution (full length version) consider general distribution for inter-arrival times and burst sizes, general (non-degenerate) buffer setting, and also two novel stochastic mechanisms that, together with the known technique of preventive dropping (introduced in [11]), allow to further mitigate loss. Both techniques refine the CDS algorithm by not only taking into account the scheduling horizon of the different wavelengths (the current best solution), but also the current traffic load and, in case of varying packet/burst size, the size of packets/bursts awaiting allocation. Further, we also consider preventive dropping, a third stochastic mechanism that was introduced for the single-channel case in [11], and now also proves useful in the multi-channel case. While one would reasonably expect that these additional features bring about increased implementation cost, we argue that simple embedding of a pre-calculated action table suffices to allow for optimized and robust contention resolution.

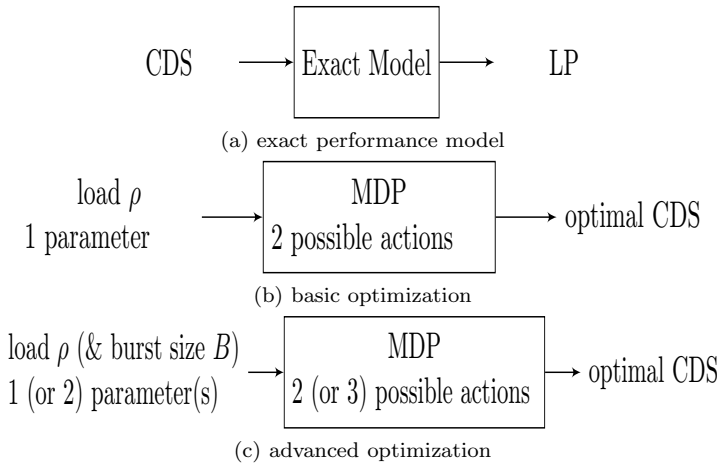
Throughout the paper, we consistently assume two wavelengths, and this because (i) it is the simplest case of a multi-wavelength system, and (ii) it allows for fast numerical evaluation (calculation times in the order of 1-10 seconds). Although we did not include the formulas for more than two wavelengths, it should be feasible to generalize the model to c wavelengths. Further, the case of two wavelengths is probably the most interesting case for wavelength allocation, and this because of the feasibility of its implementation. As explained in detail in [14], wavelength converters with limited range allow converting to adjacent wavelengths only, resulting in a limited number of wavelengths available for allocation (say, 2 to 4), and this despite the typically higher number of wavelengths carried over the physical fiber link (typically 16 to 64). While most studies consider a general number of wavelengths for the conversion range [14], the two-wavelength case assumed here was also studied in [15], with MING as scheduling discipline, but without examining the optimality of the chosen discipline.

1.4. **Methodology.** Methodologically, we present two complementary techniques for performance evaluation: a performance model and an MDP optimization method. For a given CDS algorithm (a so-called action table) and traffic/hardware setting, the performance model generates the exact loss probability. On the other hand, for a given traffic/hardware setting, the MDP method yields the optimal CDS algorithm (or action table).

As for the performance model, we develop it exactly for a broad class of multi-wavelength (or, multi-channel) optical buffers, by exact expression of the state transition probabilities of the embedded Markov chain. The formulas are valid for general independent and identically-distributed (iid) inter-arrival times distribution and packet/burst size distribution, with the assumption of an upper bound on the packet/burst size.

As for the optimization, we rely on Markov decision processes. Since this technique a well-known tool for discrete-time optimization [23], we focus primarily on the way in which to apply it. More precisely, rather than considering the system evolution from slot to slot, we consider embedded points, observing the system only upon arrival instants. This approach allows constructing the action set and involved probabilities for general independent and identically-distributed (iid) inter-arrival time distribution and burst size distribution. As such, the assumptions of the MDP model match those of the performance model. If considered without stochastic mechanisms, we refer to the MDP optimization scheme as basic. If stochastic mechanisms are considered, we speak of advanced optimization.

1.5. **Overview of this paper.** In the following, the CDS algorithm boils down to wavelength allocation among two different wavelengths, with the delay selection implied by the choice of the wavelength (see further). Within a single system description framework, introduced in Sect. 2, and including both basic and advanced optimization, our contribution consists of two complementary techniques. Firstly, we set out a performance model in Sect. 3, allowing to quantify loss performance (more precisely, the loss probability (LP)) in an exact manner, as shown in Fig. 1. Also shown in Fig. 1, we next present the MDP-based optimization technique in Sect. 4, which stands apart from the performance model, but is valid for the same system description, and uses some of the expressions obtained in Sect. 3. Note that we incorporate both basic and advanced optimization within the same model. This optimization method is applied in Sect. 5 to some specific settings, and allows us to



(a) exact performance model

(b) basic optimization

(c) advanced optimization

Figure 1: In Sect. 3, a performance model is presented, which allows to calculate the loss probability of any given CDS (with our without stochastic mechanisms) in an exact manner (a). Apart from this, Sect. 4 presents an MDP-based optimization method, allowing to identify an optimal CDS action table. Basic optimization only takes the traffic load ρ into account (b). In advanced optimization (c), this approach is extended by possible burst-size-dependency, and the possible addition of a preventive drop action.

assess the performance of the optimized algorithm under various conditions of the traffic load, and compare it to the performance of MING. Special attention goes to the performance results of advanced optimization, illustrated in Fig. 1. Conclusions are drawn in Sect. 6. Further, note that we refer to packets/bursts merely as bursts below.

2. System description.

2.1. Horizon scheduling and stochastic mechanisms.

The FDL buffering architecture and related CDS control problem is identified already in [9]; however, the control algorithms of [25, 28] are probably best-known. In [25], Turner introduces *horizon scheduling*, a scheduling algorithm which bases its decision upon the c horizon values of the c available channels. For each channel, the *scheduling horizon* (or, for short, *horizon*) value indicates the earliest time at which all previous bursts on that channel will have left the system. It also referred to as the *unscheduled time*, or the *future available time*. The bulk of research on horizon scheduling has been done on degenerate FDL buffers, with equidistant lengths, equal to multiples of a granularity value D. Choosing equidistant values is a natural choice, mainly because a scenario with non-degenerate FDL lengths failed to demonstrate significant performance advantages over an equidistant one. Although it is shown in [11] that a non-degenerate setting can outperform a degenerate one, the performance difference was rather small, and only occurred for high traffic load. The results in this paper are however also valid for non-degenerate buffers, and therefore, also an example is included further on.

Opposed to horizon scheduling, a contribution revealing the actual complexity of the channel and delay selection algorithm is [24], showing that the optimal CDS algorithm should maintain the position of all voids created on all channels and all delay lines, in order to exploit the (buffering) capacity that is available within the voids. The resulting CDS algorithm is called the void-filling scheduling algorithm, and is rigorously studied in [26], and this for various traffic conditions, buffer sizes and algorithm parameter settings. As illustrated in Fig. 7–10 in [26], void filling allows approaching the performance of a strictly synchronized buffer of the same size (there referred to as synchronous operation) rather close (but not arbitrarily close) as the FDL lengths increase. Reference [24] (which is referred to in [26]) points out that void-filling with degenerate buffer setting is possible, but even better performance is enabled by choosing the FDL lengths non-degenerate (not equal to multiples of D), so improving performance even further, by creating sufficiently large voids, which are then filled up by new arrivals. Such a performance benefit should not be expected from non-degenerate settings for horizon scheduling, mainly because the voids, once created, can never be filled up, and always constitute capacity loss.

For horizon scheduling, typically, very basic CDS algorithms are considered, basing the scheduling decision only on the horizon value of the channels. However, the addition of a stochastic mechanism has been considered earlier in [11] in the case of a single wavelength, and was referred to as preventive drop. Somewhat paradoxically, preventive drop allows to realize performance gain by dropping bursts even when resources are still available. The underlying idea is that, in case of high load situations, it is appropriate to "speculate" and drop those bursts that would cause large gaps, in order to be able to accommodate future bursts with (potentially) smaller gaps. To the best of the authors' knowledge, we are the first to consider preventive dropping now in the multi-wavelength case.

A second stochastic mechanism is load-dependent scheduling, a mechanism which takes into account the current traffic intensity/load, in order to adapt the CDS algorithm to it. This is discussed and motivated later on by means of examples, in Sect. 5; to the best of our knowledge, this is the first time such a mechanism is proposed. Thirdly, we introduce a stochastic mechanism we call burst-size-dependent scheduling. The central idea is to fit bursts within the delay lines in which they (and not necessarily other bursts) fit best; this is particularly useful in case of non-degenerate FDL sets, and is also discussed by means of an example in Sect. 5.

All three mentioned stochastic mechanisms are incorporated into the same performance model and optimization method. As such, they are treated as native to the presented model, rather than as an extension of it.

2.2. **Traffic setting.** We assume a discrete-time setting, with time divided in time slots of fixed (arbitrary) size, and all time-related variables and parameters expressed as multiples of this assumed slot length. Numbering bursts in the order at which they arrive at the buffer, we consider an arbitrary burst with index k, arriving some time T_k after the previous burst (burst $k+1$), with T_k the inter-arrival time. We assume the inter-arrival times general independent and identically distributed (iid) random variables, with values drawn from a discrete probability distribution function $t(n)$, $n \in \mathbb{N}_0$, with

$$t(n) = \Pr[T_k = n]\,, \tag{1}$$

$0 \leq t(n) \leq 1$, $\sum_{n=1}^{\infty} t(n) = 1$ and $\mathrm{E}[T_k] = \sum_{n=1}^{\infty} nt(n) < \infty$. No further restrictions are imposed on this distribution. Similarly, the burst size of burst k is denoted B_k,

and the burst sizes are assumed general iid random variables, with values drawn from a discrete probability distribution $b(n)$, $n \in \mathbb{N}_0$. However, we assume burst sizes upper-bounded by some value B_{max}, so that

$$b(n) = \begin{cases} \Pr[B_k = n] & 0 < n \leq B_M \\ 0 & n > B_M \,. \end{cases} \tag{2}$$

Further, $0 \leq b(n) \leq 1$, $\sum_{n=1}^{B_M} b(n) = 1$ and $\mathrm{E}[B_k] = \sum_{n=1}^{B_M} nb(n) < \infty$. For notational convenience, we also introduce the cumulative distribution of the inter-arrival times F_T, defined as $F_T(n) = \sum_{i=1}^{n} t(n)$, $n \in \mathbb{N}_0$.

The traffic load ρ is defined as

$$\rho = \frac{\mathrm{E}[B_k]}{c\mathrm{E}[T_k]} \,,$$

where c denotes the number of wavelengths, as further explained below.

2.3. **Buffer setting.** In this paper, we assume that the FDL set is general, or non-degenerate [24]. We denote the set by

$$\mathcal{A} = \{a_0, a_1, a_2, \ldots, a_N\} \,,$$

with $a_0 = 0$ zero by definition, and line lengths sorted in ascending order, which results in N lines of non-zero length present in the buffer, so that realizable delays equal a_i, with $i = 0, 1, \ldots, N$. The maximal realizable delay is a_N. Further, since it is used to resolve contention, a useful FDL set never contains the same line length twice, so that $a_0 < a_1 < \ldots < a_N$. In several practical examples, we will consider a degenerate buffer setting. In that case, line lengths are equal to multiples of the granularity D, $a_i = iD$ for $i = 0, 1, \ldots, N$, and the FDL set can be written as function of the granularity, as

$$\mathcal{A} = \{0, D, 2D, \ldots, ND\} \,.$$

This buffer we assume located at the output interface of an optical burst (or packet) switch, and available for contention resolution on two distinct wavelengths λ_1 and λ_2. To enable two-wavelength contention resolution, we assume also that means for full wavelength conversion are present, together with a switching matrix, allowing to switch bursts to any of the lines on either λ_1 or λ_2. Since we do not consider void-filling, the only state information to be kept for basic horizon scheduling is the scheduling horizon of the two wavelengths involved. However, to account for the possibility of burst-size-dependent scheduling (as introduced in Sect. 2.1), we add the burst size of the burst that is to scheduling, so obtaining a three-dimensional state space.

On a given wavelength λ_1 or λ_2, the scheduling horizon is defined as the earliest time at which all previous bursts will have left the system. As said earlier, the CDS algorithm boils down to wavelength allocation among two different wavelengths, with the delay selection implied by the choice of the wavelength. This is so because, given a horizon value n on a certain wavelength (and no further information), there is only one meaningful selection for the delay value, namely $\lceil n \rceil_{\mathcal{A}}$. Here, for notational convenience, we introduced the operator notation

$$\lceil n \rceil_{\mathcal{A}} = \inf\{a_i \in \mathcal{A} : a_i \geq n\}, \; n \in \mathbb{N} \,, \tag{3}$$

which one could call a discrete generalization of the ceiling operation. It reflects the main feature of an optical buffer: rather than providing delays of n time slots, an

optical buffer provides somewhat larger delays $\lceil n \rceil_{\mathcal{A}}$, that correspond to the lengths of the fiber delay lines present in the optical buffer. The difference $\lceil n \rceil_{\mathcal{A}} - n$ accounts for the time during which the outgoing wavelength remains unused, even though the scheduled burst (and possible later-arrived bursts) is still present in the buffer and awaiting transmission on that wavelength. This time period constitutes capacity loss, and is exactly the void [24] or gap [3] mentioned earlier. If $\lceil n \rceil_{\mathcal{A}} \leq a_N$, the required delay can be realized; if not, the burst cannot be accommodated (which is reflected in the fact that the operator (3) returns $+\infty$. In the below, we assume in that case that the given burst is dropped; note however that one could also send the burst to another contention resolution interface (such as another fiber), if such an interface is available.

In case of a degenerate FDL set, the operator becomes

$$\lceil n \rceil_{\mathcal{A}} = \begin{cases} \left\lceil \frac{n}{D} \right\rceil \cdot D & 0 \leq n \leq ND \\ +\infty & n > ND\,. \end{cases}$$

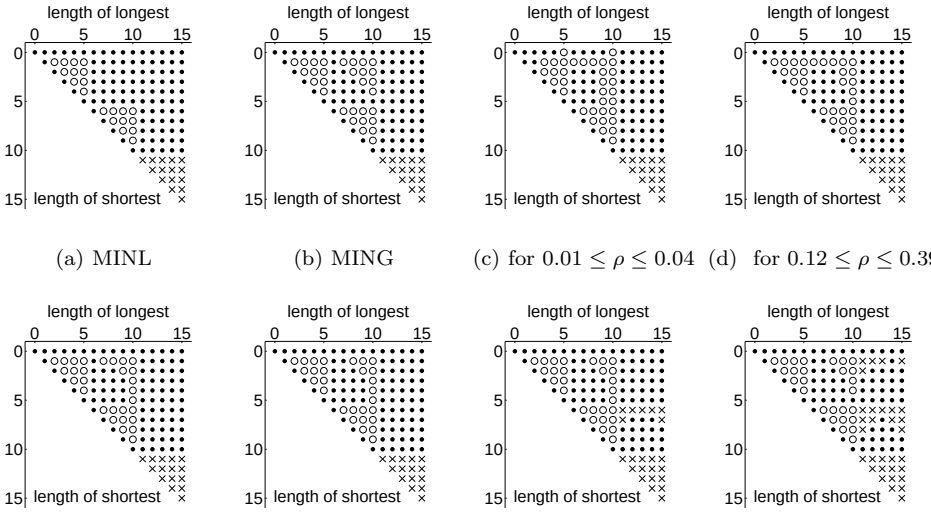
3. **Exact performance model.** Given the assumptions on arrival process, burst sizes and FDL buffer structure, the system can be analyzed in terms of the transition probabilities of a two-dimensional Markov chain. From these, we can extract an exact value for the loss probability (LP) by numerical means.

3.1. **Actions.** The system description is in terms of the scheduling horizon, as seen by an arbitrary arrival k with burst size B_k. Associated with the two wavelengths λ_1 and λ_2 are the scheduling horizon values H_k^1 and H_k^2, gathered in a two-dimensional scheduling horizon vector $\mathbf{H}_k = (H_k^1, H_k^2)$. Upon the arrival of burst k, wavelengths are indexed in order of increasing horizon value, such that $H_k^1 \leq H_k^2$. As such, the index i in H_k^i refers to the relative length of the horizon, and *not* to the index of the wavelength to which the horizon is associated when burst k arrives. The total state space vector $\mathbf{S}_k$ is equal to the combination of the horizon vector $\mathbf{H}_k$, and the burst size of the burst that is to be scheduled,

$$\mathbf{S}_k = (H_k^1, H_k^2, B_k)\,.$$

The reason to include B_k in the state space is to enable burst-size-dependent scheduling, as defined in Sect. 2.1. The process of burst arrival and transmission is governed by the CDS algorithm. More precisely, a scheduling algorithm in this context can be grasped by an action table, associating with each possible $\mathbf{S}_k$ an action c_k. In this paper, we consider three actions: $c_k = 1$, consisting in choosing the wavelength with shortest horizon; $c_k = 2$, consisting in choosing the longest horizon; and $c_k = 3$, consisting in dropping the burst. Note that, if both horizon values exceed the maximum delay a_N, the only possible action is action 3, but that, on the other hand, it may be useful to perform action 3 also if this is not the case, which is the motivation of preventive drop, as defined in Sect. 2.1.

The actions $\{1, 2, 3\}$ suffice to characterize any CDS algorithm considered, regardless whether or not stochastic mechanisms are used. As an example, consider the two CDS algorithms known from literature: MINL (minimum length) (introduced in [28] and named MINL in [3]) and MING (introduced in [3]). Both are burst-size-independent, and do not involve preventive dropping. As a CDS, MINL consists in choosing, of both horizons, the horizon with shortest length n. However, given that not n but $\lceil n \rceil_{\mathcal{A}}$ is assigned as delay (assuming $n \leq a_N$), MINL exploits this to choose the horizon with shortest $\lceil n \rceil_{\mathcal{A}}$, and, as a second criterion,



(a) MINL (b) MING (c) for $0.01 \leq \rho \leq 0.04$ (d) for $0.12 \leq \rho \leq 0.39$

(e) for $0.41 \leq \rho \leq 0.47$ (f) for $0.48 \leq \rho \leq C_f$ (g) for $0.84 \leq \rho \leq 0.89$ (h) for $\rho = 1.00$

Figure 2: Each CDS algorithm can be represented by an action table, with an action for each combination of longest horizon (x-axis) and shortest horizon (y-axis). The actions {1 (join shortest), 2 (join longest), 3 (drop)}) are indicated by $\{\bullet, \circ, \times\}$, respectively. A degenerate buffer is assumed, with $N = 2$, $\mathcal{A} = \{0, 5, 10\}$, a Bernoulli arrival process, and B fixed to 6. The CDS algorithms (c-e) are optimal in the indicated interval, regardless of whether preventive drop is considered or not. The CDS algorithm (f) is optimal for $0.48 \leq \rho \leq C_f$, with $C_f = 0.65$ if preventive drop is considered, and $C_f = 1.00$ if no preventive drop is allowed. The CDS algorithms (g-h) are optimal in the indicated interval only if preventive drop is allowed.

with smallest $\lceil n \rceil_{\mathcal{A}} - n$, as explained in detail in [3]. Given the horizon indexing rule we assumed, MINL can be formalized as Fig. 2a, there in the case of a degenerate buffer with $N = 2$, $\mathcal{A} = \{0, 5, 10\}$, and $B = 6$.

Different from this, MING does not focus on the horizon length, and limits its selection criterion to choosing the horizon n with smallest $\lceil n \rceil_{\mathcal{A}} - n$. This can be formalized in an action table as given in Fig. 2b, also in the case of a degenerate buffer with $N = 2$, $\mathcal{A} = \{0, 5, 10\}$, and $B = 6$. More formally, an action table or *policy* is given by an action table or *policy matrix* $\mathbf{P}$, conditioned on the state space (i, j, n) as seen upon arrival of an arbitrary burst k, with entries

$$p_{ijn} = \begin{cases} 1 \text{ or } 2 \text{ or } 3 & 0 \leq i \leq j \leq a_N \\ 1 \text{ or } 3 & i < a_N + 1 \leq j < a_N + B_M \\ 3 & a_N + 1 \leq i \leq j < a_N + B_M \,. \end{cases}$$

and $0 < n \leq B_M$. If the possibility of preventive drop is excluded, the range of possibilities narrows to

$$p_{ijn} = \begin{cases} 1 \text{ or } 2 & 0 \le i \le j \le a_N \\ 1 & i < a_N + 1 \le j < a_N + B_M \\ 3 & a_N + 1 \le i \le j < a_N + B_M \,. \end{cases}$$

Given this, it is insightful to consider the two-dimensional matrices $\mathbf{P}_n$, which describe the action table to be followed, given a system state $\mathbf{S}_k = (i, j, n)$, and conditioned on the assumption that a burst k of size $B_k = n$ is to be scheduled. Each of these matrices can be split up in block matrices, as

$$\mathbf{P}_n = \left[\begin{array}{c|c} {\mathbf{P}_n}^{123} & {\mathbf{P}_n}^{13} \\ \hline \mathbf{0} & {\mathbf{P}_n}^{3} \end{array}\right], \tag{4}$$

with the super-indices $\{123, 13, 3\}$ referring to the actions available from that state *if* preventive drop is available. Matrices $\mathbf{P}_n^{123}$ and $\mathbf{P}_n^{3}$ are upper-triangular (the latter, with only "3" as entries), whereas $\mathbf{P}_n^{13}$ is a dense stochastic matrix. This block structure can also be distinguished in the case of MINL Fig. 2a and MING Fig. 2b, for which the scheduling is independent of the burst size B_k, and therefore, $\mathbf{P}_1 = \mathbf{P}_2 = \ldots = \mathbf{P}_{B_M}$. However, in the case of burst-size-dependent scheduling, all matrices (B_M in number) in general differ, so allowing for a more refined channel and delay selection. Finally, note that, in the case that no preventive drop is allowed, $\mathbf{P}_n^{123}$ only contains actions 1 and 2, and $\mathbf{P}_n^{13}$ only action 1.

3.2. **Transition probabilities.** For any action table assumed, the system evolution can be gathered in a single Markov chain description. The transition is initiated by the arrival of burst k, and terminated by the arrival of burst $k+1$, T_k slots later. The Markov chain has a three-dimensional state space $\mathbf{S}_k$, and the transition matrices $\mathbf{M}_h$, $h \in 1, 2, 3$, with probabilities m_h describing the transition from $\mathbf{S}_k$ to $\mathbf{S}_{k+1}$, as

$$m_h(l, m, p \,|\, i, j, n) = \Pr[\mathbf{S}_{k+1} = (l, m, p) \,|\, \mathbf{S}_k = (i, j, n), c_k = h] \,,$$

with $0 \le i \le j < a_N + B_M$, $0 \le l \le m < a_N + B_M$, $1 \le n \le B_M$, $1 \le p \le B_M$, $h \in 1, 2, 3$. Depending on whether the first, second or third action is taken, the transition probabilities take on a different form.

1. $c_k = 1$. In this case, regardless of the horizon j of wavelength 2, the horizon of wavelength 1 as seen upon arrival is sufficiently small, $i \le a_N$, so that burst k can surely be buffered on the wavelength with horizon 1. Allocating burst k pushes horizon 1 to $\lceil i \rceil_{\mathcal{A}} + n$ slots just after arrival, whereas horizon 2 remains unaltered, at a value of j slots. Associated is the (six-dimensional) transition matrix $\mathbf{M}_1$, with probabilities m_1 that can be written as the sum of m_1^+ and m_1^-, conditioning on whether the new scheduling horizon of λ_1, $\lceil i \rceil_{\mathcal{A}} + n$, remains below j, or not, respectively. In case that $\lceil i \rceil_{\mathcal{A}} + n \le j$, wavelength indexing remains unchanged for arrival $k+1$, and transitions are made with associated probability $m_1^+(l, m, p \,|\, i, j, n)$ defined as

$$\Pr[\mathbf{S}_{k+1} = (l, m, p) \,|\, \mathbf{S}_k = (i, j, n), c_k = 1, \lceil i \rceil_{\mathcal{A}} + n \le j] \,.$$

Given the assumptions on inter-arrival and burst size distribution, the probabilities $m_1^+(l,m,p\,|\,i,j,n)$ are obtained as

$$\begin{cases} t(q)b(p) & 1 \le q < j \\ & l = [\lceil i \rceil_{\mathcal{A}} + n - q]^+ \\ & m = j - q \\ (1 - F_T(j-1))b(p) & (l,m) = (0,0)\,, \end{cases}$$

and zero elsewhere. In the opposite case, $\lceil i \rceil_{\mathcal{A}} + n > j$, the index of the horizons is swapped in order to have $H^1_{k+1} \le H^2_{k+1}$. The associated transition probability is $m_1^-(l,m,p\,|\,i,j,n$ and represents

$$\Pr[\mathbf{S}_{k+1} = (l,m,p)\,|\,\mathbf{S}_k = (i,j,n), c_k = 1, \lceil i \rceil_{\mathcal{A}} + n > j]\,,$$

with values

$$\begin{cases} t(q)b(p) & 1 \le q < \lceil i \rceil_{\mathcal{A}} + n \\ & l = [j - q]^+ \\ & m = \lceil i \rceil_{\mathcal{A}} + n - q \\ (1 - F_T(\lceil i \rceil_{\mathcal{A}} + n - 1))b(p) & (l,m) = (0,0)\,, \end{cases} \tag{5}$$

and zero elsewhere.

2. $c_k = 2$. In this case, the horizon value j of wavelength 2 as seen upon arrival is sufficiently small, $j \le a_N$, so that burst k can be buffered on the wavelength associated with horizon 2. Allocating burst k pushes the horizon 2 to $\lceil j \rceil_{\mathcal{A}} + n$, while horizon 1 remains at i. Since $i \le j$, a fortiori, $i \le \lceil j \rceil_{\mathcal{A}} + B$ and therefore, the index of the wavelengths is never switched in case of action 2, and the associated probability m_2 entirely characterizes the corresponding transition, with the expression directly related to (5), through

$$m_2(l,m,p\,|i,j,n) = m_1^-(l,m,p\,|\,j,i,n)\,.$$

3. $c_k = 3$. The buffer is found in blocking state, with both i and j larger than a_N. Since the burst size is upper-bounded by B_M, and the minimal inter-arrival time is assumed equal to 1, the scheduling horizon value is smaller than or equal to $a_N + B_M - 1$, resulting in $a_N < i \le j < a_N + B_M$. Action three corresponds to discarding arriving burst k, or forwarding it to another contention resolution interface. The scheduling horizon remains unaltered by the arrival, and the involved transition probabilities $m_3(l,m,p\,|\,i,j,n)$ are as follows,

$$\begin{cases} t(q)b(p) & 1 \le q < j \\ & l = [i - q]^+ \\ & m = j - q \\ (1 - F_T(j-1))b(p) & (l,m) = (0,0)\,, \end{cases}$$

and zero elsewhere.

As such, all transition probabilities are known as soon as one assumes a certain CDS (and corresponding policy matrix $\mathbf{P}$). From there, one can calculate the sparse transition matrix $\mathbf{M}$ associated with them. From the obtained $\mathbf{M}$, using standard numerical means (for instance, the linear equation solving command "\" in Matlab), one can extract the left eigenvector associated with eigenvalue 1, known as the Perron-Frobenius eigenvector, yielding the steady-state distribution of the system state as seen upon arrival. Denoting these probabilities by $\Pr[H^1 = i\,, H^2 = j,\, B =$

$n] = s(i,j,n)$, $0 \le i \le j < a_N + B_M$, $1 \le n \le B_M$, one obtains the probability that an arbitrary arriving burst is lost by evaluating the probability that action 3 is taken. The loss probability (LP) is thus obtained as

$$\mathrm{LP} = \sum_{n=1}^{B_M} \sum_{i=0}^{a_N+B_M-1} \sum_{j=i}^{a_N+B_M-1} s(i,j,n)\delta_{p_{ijn},3}\,,$$

where $\delta_{p_{ijn},3}$ denotes the Kronecker delta, which equals one if $p_{ijn} = 3$, and zero elsewhere.

While the above allows to calculate the loss probability in an exact manner, it is important to bear in mind that the computational burden of calculating transition and steady-state probabilities grows with $\mathcal{O}((a_N + B_M)^2 B_M)$, and therefore, feasible calculations require to keep the involved parameters small, to values for which $a_N + B_M$ is smaller than, say, 100 slots. Note, however, that a more sophisticated modeling approach would allow to limit numerical complexity to $\mathcal{O}(N(a_N + B_M)B_M)$, by considering a heterogeneous Markov state space, consisting of the last-assigned waiting time of one wavelength, completed with the scheduling horizon of the other wavelength. This is discussed in more detail in [15] but is considered out of the scope of the present contribution.

4. **Markov decision process.** In principle, the MDP optimization stands apart from the analysis of the previous section; however, the "actions" introduced there fittingly bear the name of a classic ingredient of MDP optimization. More precisely, we apply an MDP technique to determine a policy matrix $\mathbf{P}$, in a way that is similar to the one described in [23], as it is based on the policy iteration algorithm described there. Each policy matrix is obtained for a given value of the traffic load ρ, which can be varied through the parameter $p = 1/\mathrm{E}[T_k]$.

We imagine an agent that has at its disposition the set of three actions $\{1,2,3\}$, and desires to maximize a reward function by choosing appropriate actions. Each action constitutes a way to handle arriving bursts, and the choice for a given action is conditioned on the scheduling horizon as seen by the arriving bursts. Since we aim to minimize the loss probability, the rewards are negative, and we associate a reward that is proportional to the size of the burst that is lost, namely $-B$ (or, equivalently, a cost of B) to each lost burst with burst size B, and a reward of zero to each accepted burst. This reward function trivially maps on the set of actions: action one and two correspond to zero reward, action 3 yields a reward of $-B$.

The policy iteration algorithm now consists in choosing an arbitrary initial (three-dimensional) policy array $\mathbf{P}$, with an arbitrary choice among the actions possible in a given state. In the case that preventive drop is allowed, conditioned on burst size n, the allowed actions are 1,2 or 3 for $\mathbf{P}_n^{123}$, 1 or 3 for $\mathbf{P}_n^{13}$, and 3 for $\mathbf{P}_n^{3}$. In the case no preventive drop is allowed, the allowed actions are 1,2 for $\mathbf{P}_n^{123}$, 1 for $\mathbf{P}_n^{13}$, and 3 for $\mathbf{P}_n^{3}$.

Given the (three-dimensional) policy $\mathbf{P}$, a value is determined for each state $\mathbf{S} = (i,j,n)$, consisting of the immediate reward, and all rewards to be earned in the future, taking into account the possible state evolution (as dictated by the probabilities m_1^+, m_1^-, m_2 and m_3 obtained in Sect. 3). Then, each policy iteration consists in (i) determining the new policy $\mathbf{P}'$ that maximizes the expected reward, given the computed values of the previous steps and the probabilities m_1^+, m_1^-, m_2 and m_3; and next (ii) computing the new values, given the new policy $\mathbf{P}'$. The policy is reiterated until no change takes place in the policy in step (i). We

refer to [23] for further details, noting the difference in our approach: while [23] suggests basing the description on transitions from time slot to time slot, we prefer to consider embedded transition points. In our approach, each arrival triggers an iteration, which yields the advantage that our approach is immediately applicable to the case of general inter-arrival time distribution $t(n)$ and general burst size distribution $b(n)$, whereas a slot-based iteration would only allow applicability to the case of a Bernoulli arrival process.

5. **Optimization examples.** In this section, we apply the optimization techniques developed in the previous sections. As outlined earlier, and illustrated in Fig. 1, we first consider basic optimization Fig. 1b (without stochastic mechanisms), to then contrast its output with results from advanced optimization (with stochastic mechanisms). A variety of parameter and traffic settings is considered in the following: deterministic and non-deterministic burst size distributions, degenerate and non-degenerate FDL sets, smaller and larger buffer sizes, and this first without and then with the application of stochastic mechanisms. To further limit the vast amount of possible combinations, we assume only one arrival process: a Bernoulli arrival process, which is the discrete-time counterpart of a Poisson arrival process. This arrival process is often assumed for performance modeling of backbone network traffic. At the beginning of each slot, either one or no arrival occurs, with probability p or $1-p$, respectively. Given this, the inter-arrival time T_k between the arrival of burst k and burst $k+1$ (as introduced in general in (1)) follows a geometric distribution, with probability density function

$$t(n) = p \cdot \bar{p}^{n-1}\,,\ n \in \mathbb{N}_0\,, \tag{6}$$

cumulative distribution function $F_T(n) = 1-\bar{p}^n$, $n \in \mathbb{N}_0$, and expected value $\mathrm{E}[T_k] = 1/p$. In the below, different values of the traffic load $\rho = p\mathrm{E}[B_k]/c$ (with c the number of wavelengths, $c=2$) are considered; the difference is obtained by varying p, with the burst size distribution unaltered.

5.1. **Deterministic burst size distribution.** Probably the simplest instance of the CDS problem is obtained for deterministic burst size distribution, with burst sizes fixed to some integer value of B slots, and corresponding burst size distribution (as introduced in general in (2)),

$$b(n) = \delta_{n,B}\,,$$

(where $\delta_{i,j}$ again denotes the Kronecker delta), $B_M = B$, and $\mathrm{E}[B_k] = B$. This distribution, combined with geometric inter-arrival times (6), is an often-studied combination in case of single-wavelength optical buffers. Previous studies revealed that a degenerate buffer setting, with $D = B-1$ is almost always (but not always, see [11]) the optimal choice if the load ρ remains below some threshold load $\rho_{th} \approx 0.6$ [20]. For multiple wavelengths, fewer results are available; for a degenerate setting however, $D = B-1$ is also an advantageous choice in terms of performance, see [22]. As such, we also consider this choice here, in several specific setting of modest numerical complexity, and $B = 6$ slots. Using the formulas of the previous sections, for each load value $\rho = i \cdot 0.01$, $i = 1,\ldots,100$, we performed an independent MDP optimization, each yielding a separate policy matrix $\mathbf{P}$, optimized for minimal loss at exactly the load assumed. Given the limited set of states, $(ND+B-1)(ND+B)/2 = 136$, each with a maximum of three possible actions (or less), it comes as no surprise that several load values yielded exactly the same policy as optimal policy.

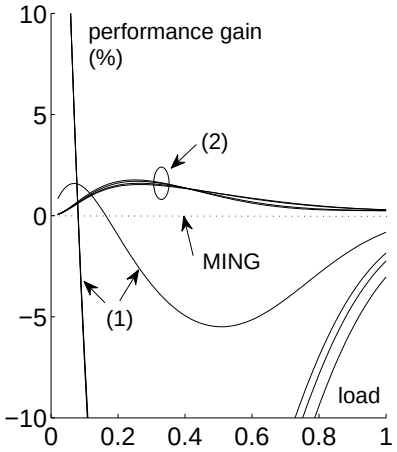


Figure 3: The loss performance of the CDS algorithms obtained as function of the traffic load, without allowing preventive drop. When compared to MING, four CDS algorithms (belonging to group (2)) consistently outperform MING, showing that MING is never strictly optimal.

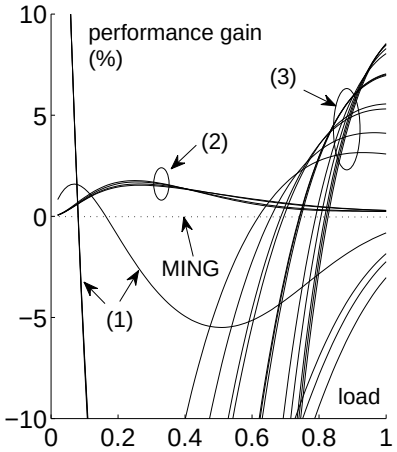


Figure 4: The loss performance of the CDS algorithms obtained as function of the traffic load, without allowing preventive drop. Of the algorithms, only the ones belonging to group (3) apply preventive drop.

First, we consider the case of $N = 2$ and $D = 5$ (and thus, $\mathcal{A} = \{0, 5, 10\}$), and a basic optimization, without the possibility of preventive drop. As said, even though we performed a total of 100 MDP optimizations, only 8 non-identical CDS algorithms were obtained, typically optimal for certain intervals of the traffic load,

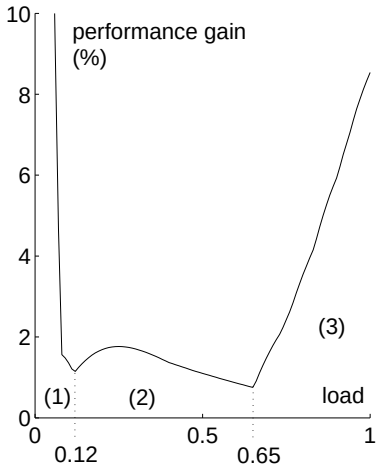


Figure 5: When compared to MING, a load-dependent mechanism yields much better loss performance when the load is low (zone (1) in the figure). For intermediary load ($0.12 \leq \rho \leq 0.65$, zone (2)), loss reduction is limited to 1–2 %, whereas for high load (zone (3)), the performance gain is again larger.

rather than for a single value. The following intervals for the load ρ are obtained as output of our optimization:

$$\begin{aligned}(1)&[0.01, 0.04][0.05, 0.06][0.07][0.08, 0.11]\,,\\ (2)&[0.12, 0.39][0.40][0.41, 0.47][0.48, 1.00]\,.\end{aligned} \tag{7}$$

Four of the eight obtained CDS algorithm are displayed on Fig. 2, more precisely Fig. 2c ($0.01 \leq \rho \leq 0.04$), Fig. 2d ($0.12 \leq \rho \leq 0.39$), Fig. 2e ($0.41 \leq \rho \leq 0.47$) and Fig. 2c ($0.48 \leq \rho \leq 1.00$). The load intervals fall apart in two groups (1) and (2), which correspond to a low-traffic regime (1), with $0.01 \leq \rho \leq 0.11$, and a moderate-to-high-traffic regime (2), with $0.12 \leq \rho \leq 1.00$. The rationale of this grouping can be best understood when considered along with the performance output, as it can be obtained exactly, by using the 8 policies as input to the exact model of Sect. 3, as depicted in Fig. 1c. In Fig. 3, the performance of these 8 policies is compared in an exact manner to that of MING (Fig. 2b). Clearly, the four CDS algorithms of group (2) outperform MING for any value of the load, but the performance gain is only moderate (1–2%). This contrasts with the situation for low load, for which the algorithms of group (1) realize a loss reduction of over 30 %. One of the algorithms (the one optimal for $0.1 \leq \rho \leq 0.04$) is displayed in Fig. 2c. One can see that the policy is different from MING (Fig. 2b) in several ways; for instance, when finding horizon $(0, 5)$ or $(0, 10)$, it prefers to join the wavelength with longest horizon, even though the gap size is equal for both wavelengths.

In other words, in the case of Fig. 2c, counter-intuitively, the optimal policy is to join the longest queue even if the shortest queue is empty. This can be understood as follows. Assume horizon $\mathbf{H}_k = (0, 5)$, and burst k (with burst size fixed to 6 slots) joins the shortest queue. Then, upon arrival, burst $(k+1)$ will find $\mathbf{H}_{k+1} = ([5 - T_k]^+, [6 - T_k]^+)$, which in general will result in non-zero gap size for burst

$(k+1)$. Opposed to this, if burst k joins the longest queue, burst $(k+1)$ will find $\mathbf{H}_{k+1} = (0, [11 - T_k]^+)$, which always allows for zero gap size for burst $(k+1)$, by joining the shortest queue. As such, striving for minimal average cost (and thus, minimal average gap size) yields a counter-intuitive decision. However, it is optimal in this specific setting, since it is generated by an exact optimization method.

Returning to Fig. 3, note that part of the curve for (1) is not shown to avoid improper scaling: for $\rho = 0.01$, the performance gain rises to 37.9%. Further, note that the results for group (1) would have been hard to obtain with sufficient accuracy through simulation, whereas they can be calculated instantly and exactly with our method. For instance, for $\rho = 0.01$, the LP for MING is $3.76 \cdot 10^{-14}$, while it is $2.33 \cdot 10^{-14}$ for the CDS policy displayed in Fig. 2c.

We reconsider the case of $N = 2$ and $D = 5$ ($\mathcal{A} = \{0, 5, 10\}$), now allowing the MDP optimization to generate policies with preventive drop if they realize minimal loss. The MDP optimization was again performed independently for each load value $\rho = i \cdot 0.01$, $i = 1, \ldots, 100$, and now yields 21 different CDS algorithms. The corresponding intervals for the load ρ are as follows,

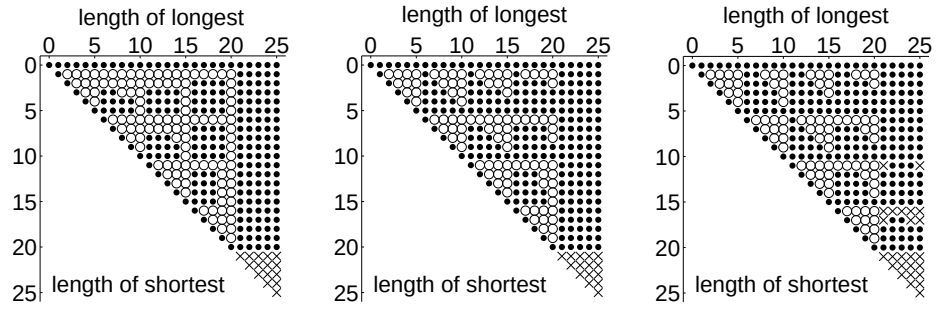
$$\begin{aligned}
&(1)[0.01, 0.04][0.05, 0.06][0.07][0.08, 0.11]\,,\\
&(2)[0.12, 0.39][0.40][0.41, 0.47][0.48, 0.65]\,,\\
&(3)[0.66, 0.73][0.74, 0.76][0.77, 0.81][0.82, 0.83]\\
&\quad [0.84][0.85, 0.89][0.90][0.91, 0.93], [0.94]\\
&\quad [0.95][0.96, 0.97], [0.98, 0.99][1.00]\,.
\end{aligned} \tag{8}$$

Closer inspection of the results showed that 8 of the 21 algorithms obtained without preventive drop were also obtained for the optimization without preventive drop. More precisely, the CDS algorithms of (8) in group (1) and (2) coincide with those reported in (7) (and, with one exception, also the load intervals); opposed to this, the 13 algorithms in group (3) are new, and all involve some preventive drop. Two of the 13 algorithms are displayed on Fig. 2: Fig. 2g ($0.85 \leq \rho \leq 0.89$), and Fig. 2h ($\rho = 1$). In this regard, the optimization with preventive drop allows for "'richer" results than an optimization without preventive drop, since it finds 13 additional algorithms for $\rho \geq 0.66$, instead of none. The loss performance of all 21 algorithms is set out as a function of the load in Fig. 4. While the curves of group (1) and (2) are identical to those of Fig. 3, the curves of group (3) show that, for high traffic load ($\rho \geq 0.66$), preventive drop does allow for additional loss reduction. As can be seen on the plots of some of the corresponding policies, on Figs. 2g and 2h, this performance gain is mainly due to actions of preventive drop, the "'speculative" technique mentioned earlier, which is typical for high load situations, dropping those bursts that would cause large gaps, in order to be able to accommodate future bursts with (potentially) smaller gaps. When comparing 2g and Fig. 2h, note that, while the number of preventive drop actions increases, the other actions remain largely the same as in the policy of Fig. 2f.

Finally, as introduced in Sect. 2.1, the stochastic mechanism of load-dependent scheduling enables to reduce loss probability even further. Assumed that we have a perfect estimation of the current traffic load, and schedule bursts accordingly, one can obtain an improved loss performance curve, as displayed in Fig. 5. While this is the ideal case, inspection of Figs. 3 and 4 shows that it should suffice to dynamically switch between three algorithms, in order to approach the loss performance curve of Fig. 5 rather close. The three algorithms would be drawn from group (1), (2) and

ρ	0.20	0.40	0.60	0.80	1.00
$N = 2$	1.69	1.37	0.86	3.55	8.54
$N = 4$	5.36	2.92	1.49	6.31	17.86

Table 1: The loss reduction (in %) of an optimized load-dependent CDS algorithm over MING is assessed, for $N = 2$ and $N = 4$, and with preventive drop enabled. A larger number of delay lines allows for refined optimization, and therefore, the performance improvement over MING grows as the number of delay lines increases.

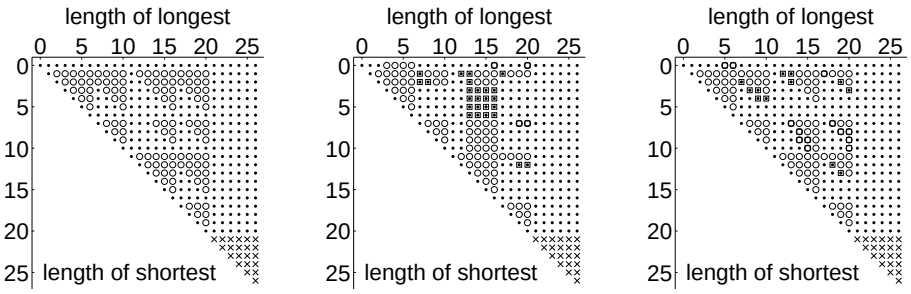


(a) for $\rho = 0.01$ (b) for $0.33 \leq \rho \leq 0.39$ (c) for $0.81 \leq \rho \leq 0.82$

Figure 6: While the algorithms in Fig. 2 were obtained for $N = 2$, these are three CDS algorithms obtained for $N = 4$. As in Fig. 2, the actions {1 (join shortest), 2 (join longest), 3 (drop)}) are indicated by $\{\bullet, \circ, \times\}$, respectively. Clearly, the larger number of possible actions better reveals the symmetry of the CDS algorithm, and this in the case of both low and high traffic load.

(3), in order to deal with low, medium and high traffic load, respectively. Note that such a dynamic algorithm can be used in practice on the same hardware as MING, by adapting the action tables of the CDS algorithm to the currently measured traffic load. The implementation aspects of such setting however go beyond the scope of this contribution. Further, note that the comparison of such a dynamic algorithm with the (static) MING algorithm is not strictly fair, and merely serves to show the potential performance gain enabled by a stochastic mechanism like load-dependent scheduling.

To verify whether the optimization for deterministic burst size distribution for $N = 2$ is representative also for larger FDL sets, we performed the same calculations for the case of $N = 4$, $B = 6$, $D = 5$ and $\mathcal{A} = \{0, 5, 10, 15, 20\}$. Apart from being similar, the optimization is somewhat richer, since the set of possible actions is larger. With preventive drop allowed, this resulted in 46 different algorithms, which realize somewhat more performance gain over MING than in the case of $N = 2$. Rather than repeating the same figures, we choose to sum up the performance gain over MING in a load-dependent scenario (with preventive drop allowed), and gather results in Table 1. Further, in Fig. 6 we display the optimal CDS algorithm



(a) MING (b) for $\rho = 0.01$ (c) for $0.49 \leq \rho \leq 0.59$

Figure 7: Opposed to the algorithms in Fig. 2 and Fig. 6, these algorithms were obtained for a non-degenerate FDL set $\mathcal{A} = \{0, 6, 10, 16, 20\}$, which alters the shape of MING to (a) (instead of the shape in Fig. 2b). In (b) and (c), burst-size-dependent algorithms are considered. Still, the actions {1 (join shortest), 2 (join longest), 3 (drop)}) are indicated by $\{\bullet, \circ, \times\}$, respectively; but this only for the conditioned policy matrix $\mathbf{P}_1$. The policy matrix $\mathbf{P}_2$ contains the same actions as $\mathbf{P}_1$ except for the actions marked with a "□"; a marked "•" implies action 1 for $\mathbf{P}_1$ and action 2 for $\mathbf{P}_2$; a marked "○" vice versa. As can be seen, the policy matrices $\mathbf{P}_1$ and $\mathbf{P}_2$ differ substantially, and in a non-trivial manner.

for $\rho = 0.01$ (Fig. 6a), $0.33 \leq \rho \leq 0.39$ (Fig. 6b), and $0.81 \leq \rho \leq 0.82$ (Fig. 6c) (the latter, with preventive drop). Especially in the case of very low traffic load, (Fig. 6a), more symmetry is revealed than for the case of $N = 2$. However, it is not possible/obvious to extract a simple rule from this.

Finally, for deterministic bust size distribution, burst-size-dependent scheduling does not allow further loss mitigation. To examine the latter possibility, we consider a non-deterministic distribution in the next section.

5.2. **Non-deterministic burst size distribution.** For varying burst sizes, the number of possible optimal scheduling algorithms increases spectacularly, as the possibility of burst-size-dependent allows for further refinement of the CDS algorithm, especially when considered in combination with preventive drop and load-dependent scheduling. Intuitively, it seems that such burst-size-dependent scheduling algorithms are of crucial importance in case of non-degenerate buffers. Therefore, rather than an exhaustive optimization study, we limit ourselves here to stating a clear example with a non-degenerate buffer setting, that motivates the usefulness of burst-size-dependent scheduling.

The specific setting we assume has a burst size distribution

$$b(n) = 0.5 \cdot \delta_{n,B_1} + 0.5 \cdot \delta_{n,B_2} \,,$$

(with $\delta_{i,j}$ again the Kronecker delta), $B_1 < B_2$, $B_M = B_2$, and $\mathrm{E}[B_k] = (B_1 + B_2)/2$. We assume $B_1 = 5$ and $B_2 = 7$. Further, rather than choosing a degenerate FDL set, we choose a non-degenerate FDL set $\mathcal{A} = \{0, 6, 10, 16, 20\}$ with thus $N = 4$. Note that this set is by no means optimized for this purpose; rather, it is composed of values that are sufficiently near to a degenerate buffer setting (with

ρ	0.20	0.40	0.60	0.80	1.00
loss reduction (%)	44.65	21.00	11.86	5.59	1.70

Table 2: The loss reduction (in %) of an optimized burst-size-dependent (and load-dependent) CDS algorithm over MING is assessed, in case no preventive drop is applied. Clearly, the performance improvement over MING is significant, especially for low values of the load.

$\mathcal{A} = \{0, 5, 10, 15, 20\}$), as it can be expected that non-degenerate FDL sets with large fluctuation in line lengths are of little practical relevance (at least in the case of horizon scheduling, as opposed to void-filling [24]).

In a non-equidistant FDL set, also the MING algorithm looks different; its shape for $\mathcal{A} = \{0, 6, 10, 16, 20\}$ is displayed in Fig. 7a. Further, Figs. 7b and 7c provide the optimal CDS algorithm for $\rho = 0.01$ and $0.49 \leq \rho \leq 0.59$, respectively. Further, we recall that, for burst-size-dependent scheduling, the policy matrices are conditioned on the burst size, see (4). As such, Figs. 7b and 7c each display both $\mathbf{P}_1$, the policy matrix in case a burst of size $B_1 = 5$ is scheduled, and $\mathbf{P}_2$, in case a burst of size $B_2 = 7$ is scheduled. Further, burst-size-dependent optimization without preventive drop yields the results listed in Table 2. It illustrates that burst-size-dependent scheduling performs much better than MING in a non-degenerate setting, especially for low traffic loads. Note that the introduction of preventive drop (not applied here) would also enable to realize loss reduction for high traffic load; this would however not have a direct connection with the benefits of burst-size-dependent scheduling. Other (initial) results not included here confirm that the large improvement reported in Table 2 is less pronounced in degenerate buffer settings, but can still be significant. This question is however beyond the scope of the current contribution, in which we limit ourselves to introducing the concept of burst-size-dependent scheduling, and connecting it to non-degenerate settings, for which it seems particularly relevant.

6. **Conclusions.** In this paper, a CDS performance model and optimization method were presented. By means of a Markov decision process modeling, it is possible to generate CDS algorithms that outperform MING, as well as all other commonly studied heuristic algorithms for CDS. If we assume a static CDS algorithm, some of the obtained CDS algorithms outperform MING for any value of the load, showing that MING is never strictly optimal for the given setting, and is thus in general suboptimal. By combining the obtained policies in a load-dependent algorithm, it is possible to obtain much better overall loss performance, for the same hardware requirements as MING.

Further, the performance evaluation of CDS algorithms with preventive drop shows that this mechanism allows for a richer optimization process, and allows to further mitigate loss, but only when the load is high. Finally, also the notion of burst-size-dependent scheduling was introduced; initial results point out that the loss probability for non-degenerate FDL settings is lowered significantly by means of this stochastic mechanism, especially when the traffic load is low.

Concluding, an algorithm optimal for any traffic load cannot (in general) be devised, but a load-dependent CDS algorithm should allow to attain improved control robustness. Further research is needed in order to study (and potentially, develop)

algorithms that take into account the (instantaneously varying) traffic load, so being more robust to its variations.

Acknowledgments. This paper is dedicated to Professor Yutaka Takahashi in celebration of his 60th birthday. During fall 2011, the first author is a visiting researcher at Takahashi Laboratory; he hereby expresses his gratitude to Professor Takahashi and his team for the valuable interaction and kind hospitality.

REFERENCES

[1] F. Callegati, *Optical buffers for variable length packets*, IEEE Communications Letters, **4** (2004), 292–294.
[2] F. Callegati, *Approximate modeling of optical buffers for variable length packets*, Photonic Network Communications, **3** (2001), 383–390.
[3] F. Callegati, W. Cerroni and G. Corazza, *Optimization of wavelength allocation in WDM optical buffers*, Optical Networks Magazine, **2** (2001), 66–72.
[4] F. Callegati, D. Careglio, W. Cerroni, G. Muretto, C. Raffaelli, J. Solé-Pareta and P. Zaffoni, *Keeping the packet sequence in optical packet-switched networks*, Optical Switching and Networking, **2** (2005), 137–147.
[5] F. Callegati, W. Cerroni and G. S. Pavani, *Key parameters for contention resolution in multi-fiber optical burst/packet switching nodes*, Proceedings of the Fourth IEEE International Conference on Broadband Communications, Networks and Systems, Broadnets (Raleigh), (2007), 217–223.
[6] F. Callegati, G. Muretto, C. Raffaelli, P. Zaffoni and W. Cerroni, *A framework for performance evaluation of OPS congestion resolution*, Proceedings of the Ninth Conference on Optical Network Design and Modelling, ONDM (Milan), (2005), 242–249.
[7] Y. Chen, C. Qiao and X. Yu, *Optical burst switching: A new area in optical networking research*, IEEE Network, **18** (2004), 16–23.
[8] C. M. Gauger, *Optimized combination of converter pools and FDL buffers for contention resolution in Optical Burst Switching*, Photonic Network Communications, **8** (2004), 139–148.
[9] Z. Haas, *The staggering switch: an electronically controlled optical packet switch*, IEEE/OSA Journal of Lightwave Technology, **11** (1993), 925–936.
[10] K. Laevens and H. Bruneel, *Analysis of a single-wavelength optical buffer*, Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM (San Francisco, CA), (2003), 1–6.
[11] J. Lambert, B. Van Houdt and C. Blondia, *Single-wavelength optical buffers: Non-equidistant structures and preventive drop mechanisms*, Proceedings of the 2005 Networking and Electronic Commerce Research Conference, NAEC (Riva del Garda), (2005), 545–555.
[12] G. Muretto and C. Raffaelli, *Combining contention resolution schemes in WDM optical packet switches with multi-fiber interfaces*, OSA Journal of Optical Networking, **6**, (2007), 74–89.
[13] F. Masetti, M. Sotom, D. De Bouard, D. Chiaroni, P. Parmentier, F. Callegati, G. Corazza, C. Rafaelli, S. L. Danielsen and K. E. Stubkjaer, *Design and performance of a broadcast and select photonic packet switching architecture*, Proceedings of the 1996 European Conference of Optical Communication, ECOC (Oslo), (1996), 15–19.
[14] T. Phung-Duc, H. Masuyama, S. Kasahara and Y. Takahashi, *Performance analysis of optical burst switched networks with limited-range wavelength conversion, retransmission and burst segmentation*, Journal of the Operations Research Society of Japan (JORSJ), **52** (2009), 58–74.
[15] J. F. Pérez and B. Van Houdt, *Wavelength allocation in an optical switch with a fiber delay line buffer and limited-range wavelength conversion*, Telecommunication Systems, **41** (2009), 37–49.
[16] C. Qiao and M. Yoo, *Optical burst switching–a new paradigm for an optical internet*, Journal on High-Speed Networks, **8** (1999), 69–84.
[17] R. Van Caenegem, D. Colle, M. Pickavet, P. Demeester, K. Christodoulopoulos, K. Vlachos et al., *The design of an all-optical packet switching network*, IEEE Communications Magazine, **45(11)** (2007), 52–61.

[18] W. Rogiest, K. De Turck, D. Fiems, K. Laevens, S. Wittevrongel and H. Bruneel, *Optimized channel and delay selection for contention resolution in optical networks*, Proceedings of the IEEE International Conference on Communications (ICC2011, Kyoto), (2011), 1–5.
[19] W. Rogiest, K. De Turck, K. Laevens, S. Wittevrongel and H. Bruneel, *Contention resolution for optical switching: tuning the channel and delay selection algorithm*, Proceedings of the Ninth IARIA International Conference on Networks (ICN, Les Menuires), (2010), 1–6.
[20] W. Rogiest, J. Lambert, D. Fiems, B. Van Houdt, H. Bruneel and C. Blondia, *A unified model for synchronous and asynchronous FDL buffers allowing closed-form solution*, Performance Evaluation, **66** (2009), 343–355.
[21] W. Rogiest, K. Laevens, D. Fiems and H. Bruneel, *A performance model for an asynchronous optical buffer*, Performance Evaluation, **62** (2005), 313–330.
[22] W. Rogiest, K. Laevens, D. Fiems and H. Bruneel, *Modeling the performance of FDL buffers with wavelength conversion*, IEEE Transactions on Communications, **57** (2009), 3703–3711.
[23] H. C. Tijms, "Stochastic Modelling and Analysis: A Computational Approach," J. Wiley and Sons, 1986.
[24] L. Tancevski, S. Tamil and F. Callegati, *Non-degenerate buffers: A paradigm for building large optical memories*, IEEE Photonic Technology Letters, **11** (1999), 1072–1074.
[25] J. Turner, *Terabit burst switching*, Journal on High-Speed Networks, **8** (1999), 3–16.
[26] L. Tancevski, S. Yegnanarayanan, G. Castanon, L. Tamil, F. Masetti and T. McDermott, *Optical routing of asynchronous, variable length packets*, IEEE Journal on Selected Areas in Communications, **18** (2000), 2084–2093.
[27] Y. Xiong, M. Vandenhoute and H. Cankaya, *Design and analysis of optical burst-switched networks*, Proceedings of SPIE (Boston, MA), **3843** (1999), 112–119.
[28] Y. Xiong, M. Vandenhoute and H. Cankaya, *Control architecture in optical burst-switched WDM networks*, IEEE Journal on Selected Areas in Communications, **18** (2000), 1838–1851.
[29] S. Yao, B. Mukherjee and S. Dixit. *Advances in photonic packet switching: An overview*, IEEE Communications Magazine, **38** (2000), 84–94.
[30] S. Yao, B. Mukherjee, S. J. B. Yoo and S. Dixit. *A unified study of contention-resolution schemes in optical packet-switched networks*, Journal of Lightwave Technology, **21** (2003), 672–683.

Received June 2011; revised August 2011.

E-mail address: wrogiest@telin.ugent.be
E-mail address: kdeturck@telin.ugent.be
E-mail address: kl@telin.ugent.be
E-mail address: df@telin.ugent.be
E-mail address: sw@telin.ugent.be
E-mail address: hb@telin.ugent.be