

Simple On-Line Scheduling Algorithms for All-Optical Broadcast-and-Select Networks

MARCO AJMONE MARSAN, ANDREA BIANCO, EMILIO LEONARDI, FABIO NERI, ANTONIO NUCCI
Dipartimento di Elettronica, Politecnico di Torino, Italy
{ajmone,bianco,leonardi,neri}@polito.it

Abstract. This paper considers all-optical broadcast networks providing a number of slotted WDM channels for packet communications. Each network user is equipped with one tunable transmitter and one fixed receiver, so that full connectivity can be achieved by tuning transmitters to the different wavelengths. Tuning times are not negligible with respect to the slot time. A centralized network controller allocates slots in a TDM/WDM frame according to (long-term) bandwidth requests issued by users. Simple on-line transparent scheduling strategies are proposed, which accommodate bandwidth requests when they are received (on-line approach), with the constraint of not affecting existing allocations when a new request is served (transparency). Strategies that attempt to allocate in contiguous slots all the transmissions of each source on one wavelength reduce overheads, are simple, and provide good performance. Even better performance can be achieved, at the cost of a modest complexity increase, when the transparency constraint is not strictly imposed, i.e., when a full re-allocation of existing connections is performed once in a while.

1 INTRODUCTION

We consider all-optical WDM/TDM broadcast-and-select networks based on a general topology (possibly a star) in which W wavelengths are available for user-to-user packet communications. We assume that user interfaces are equipped with *one* full-duplex transceiver; hence, each user can be source and destination of at most one data flow at any given time. By tuning transmitters and dynamically allocating the W available wavelengths, full connectivity is achieved among end-users. WDM channels are assumed to be slotted and synchronized; each slot can accommodate on one wavelength the transmission of one packet. The time required to tune transceivers is *not* assumed to be negligible with respect to the packet transmission time. This assumption is based on the observation that, with some of the optical components available today, the tuning latency can be quite longer (order of few μ s) than the packet transmission time (order of fractions of μ s).

Simple and efficient access protocols, i.e., protocols that lead to a good exploitation of network resources while limiting the required hardware and software complexity, are necessary to make all-optical broadcast-and-select networks a credible approach for very high-speed communications. A number of access protocol proposals for all-optical WDM/TDM broadcast-and-select networks appeared in the literature: some are based on random access approaches, whereas others are based on a determinis-

tic resolution of conflicts among different sources via fast slot allocation. While random access protocols are generally simple, but do not permit an efficient exploitation of the available bandwidth, protocols based on fast slot allocation algorithms can be rather efficient, but lead to a significant increase in the system complexity, and to the waste of part of the network resources, which must be devoted to control and signalling. Algorithms to compute static WDM/TDM schedules, which allocate transmission resources according to long term user bandwidth needs, were proposed in [1, 2, 3]. This approach is simple, but quite inefficient in the case of bursty traffic. A good trade-off between simplicity and efficiency can be achieved by using these algorithms in a context of slowly-varying traffic patterns, in which scheduling algorithms are executed once in a while (at a much lower rate with respect to individual packet transmissions) to re-compute the WDM/TDM schedule in response to changed user requirements. In this paper we focus on the last class of scheduling protocols: our scheduling algorithms react to slowly-varying user bandwidth request. We shall assume to allocate resources on the basis of average traffic requirements, which can be computed with techniques similar to the ‘equivalent bandwidth’ expressions derived for connection acceptance control in ATM networks.

Two approaches exist to modify the transmission schedule, which are called “off-line”, and “on-line” or “in-

cremental". Most previous studies [1, 2, 3, 4, 5, 6, 7] refer to the *off-line* approach: algorithms to schedule transmissions are designed assuming an exact knowledge of a stable traffic pattern. In this case, the scheduling algorithm is executed by a centralized entity, called network controller, in an off-line fashion with respect to packet transmissions and issuing of bandwidth requests. In this context, a new scheduling is usually obtained by re-allocating network resources for all user-to-user traffic flows. In a dynamic traffic scenario this means that any change in bandwidth allocation results in the rescheduling of all transmission. In this paper, on the contrary, we deal with *on-line* scheduling: the traffic pattern is supposed to be time-varying, and each time a variation in some traffic flows happens, the algorithm is executed at the centralised controller in order to adapt the slot assignments to the traffic pattern change. This scenario naturally leads to scheduling where existing traffic flows are not reallocated. Indeed, in order to reduce the complexity of the scheduling algorithm and the bandwidth devoted to signalling, we often impose to on-line algorithms a property that we call *transparency*: a variation in the allocation of slots for a traffic flow must not change the scheduling of other already established connections. Some heuristic approaches for the identification of a good transparent on-line schedule will be proposed in the paper. However, since we shall show that by strictly imposing the transparency constraint the long-term performance of the whole system can be severely limited, almost transparent approaches are also considered. With the term *almost-transparent* we refer to algorithms where a complete re-allocation is performed once in a while, possibly modifying slot assignments for already existing connections.

We do not explicitly consider in this paper the signalling information necessary to inform the centralized controller of the traffic flows to be scheduled; we assume that a separate data path is available for signalling, and we neglect the signalling delay. As a consequence, our results reflect only the performance of the scheduling algorithm, and should be viewed as upper limits to the real system performance.

2 PROBLEM STATEMENT

The problem of finding an optimum transmission schedule for a given traffic pattern in presence of non-negligible tuning latencies was already extensively analyzed [1, 2, 3, 4, 5, 6, 7]. It is usually formulated, for the off-line case, in the following way

Given a traffic matrix R , whose elements r_{sd} are the numbers of (fixed-size) packets that must be transmitted from any source user s to any destination user d , find a time/wavelength assignment that guarantees the delivery of all traffic, while minimizing the time necessary to

accommodate all transmissions (the frame duration), subject to tuning delay constraints.

We refer to this formulation as *variable-frame problem*. In [7] this scheduling problem was shown to be NP-hard. Several heuristic approaches for the determination of good schedules were proposed in [1, 2, 3, 6]. Most of those scheduling algorithms aim at the minimization of the frame duration, assuming that all packets must be transmitted in a single-hop fashion (i.e., packets are directly transferred from their source to their destination with just one transmission). This may not necessarily be the best approach when tuning latencies are long. In [3] we showed that it is possible to significantly improve the system performance by transmitting some packets in a multi-hop fashion, i.e. by routing them through intermediate nodes where they are sent back to the optical network by electronic switching.

All these approaches, being formulated for the off-line case, assume a known and stable traffic pattern. This may be not the case for most application contexts: often the traffic is neither stable nor known a priori. Even if we assume that the traffic can be seen as an aggregation of individual information flows, the aggregation exhibits bandwidth requirements that can be changing with time. We assume therefore in this paper that the network controller reacts (on-line, or incrementally) to variations in the bandwidth requirements of traffic flows by allocating (or de-allocating) in the WDM/TDM frame the appropriate number of slots. In order to reduce the algorithmic complexity and the signalling bandwidth, the operation of slot allocation (and de-allocation) for some flows should not have any impact on the scheduling of those traffic flows whose parameters have not been changed.

The problem of the incremental allocation of resources to traffic flows in a star network configuration was already investigated in the framework of SS-TDMA systems [8]. In such context, however, several aspects are different from the all-optical network scenario that we consider. As an example, in SS-TDMA networks it is assumed that all receivers must tune at the same time; this requirement makes the algorithm design quite different for the two types of networks. In [9] the authors proposed some on-line scheduling algorithms for WDM networks. The proposed strategies, however, imply that the allocation of slots to a novel traffic flow can modify the scheduling of, and possibly the bandwidth reserved to, all existing connections. As a consequence, the algorithms proposed in [9] do not satisfy the transparency constraint. In [10], on-line scheduling algorithms were proposed in a context different from the one pursued in this paper, since tuning latency is assumed negligible and tunability is envisioned both at the transmitter and at the receiver.

The variable-frame problem formalization given at the beginning of this section has the advantage of being well-suited for complexity evaluation, since it can be easily made equivalent to an integer linear programming problem. From the application viewpoint, instead, varying

the frame size without changing the number of slots allocated in the frame means varying the user throughput: normally the user specifies its bandwidth requirements (in bits per second), while the entries r_{sd} of the traffic request matrix R specify packets per frame. If the frame duration is changed without varying the number of slots per frame allocated to the users, also the bandwidth allocated to users changes. The variable-frame problem formalization is therefore suited only for one-shot transmission of the number of packets specified by R (but this is not of interest in the context of this paper), or when the entries stored in matrix R are to be interpreted as relative weights among the different source/destination pairs, and the aim of the scheduling algorithm is to maximize the system throughput while preserving relative weights among connections. Moreover, changing the frame duration in an on-line scenario implies additional complexity and control bandwidth, since the controller must notify all users of the current frame duration after each schedule adjustment, and users must be able to continuously adapt to variations in the frame duration. We therefore propose to re-formulate the off-line scheduling problem as follows:

Given a traffic matrix R , whose elements r_{sd} specify the numbers of packets that must be transmitted from any source user s to any destination user d in a pre-specified time frame comprising F slots, find a time/wavelength assignment satisfying the tuning delay constraints, that minimizes the number of packets that cannot be accommodated in the frame.

This off-line formalization, that we call *fixed-frame problem*, is more easily adapted to the on-line scenario, as discussed above. We formulate the on-line scheduling problem as follows:

Given a time frame comprising F slots, in which a number of user-to-user transmissions are allocated according to a known schedule, and given a matrix R_n of *new* requests or modifications of allocated requests, find a time/wavelength assignment, satisfying the tuning delay constraints, that avoids modifications in existing allocations (except for those resulting from R_n) and minimizes the number of packets of R_n that cannot be accommodated in the frame.

Note that in the on-line scenario both allocations and de-allocations of slots in the TDM/FDM frame must be handled.

Adopting the multi-hop approach in the on-line scenario can lead to performance improvements in presence of non-negligible tuning times for what regards both throughput and delay. This result has been observed in [3] for the off-line approach. With multi-hop scheduling, the successive hops that a packet must traverse towards its destination

are generally scheduled in successive frames. Each user transmits both locally generated packets, and packets that were generated by other users in previous frames, and are being relayed towards their destination. If the multi-hop routing of a connection is changed while packets belonging to that connection are temporarily stored at an intermediate user interface, it may happen that the intermediate user is not able to reduce the temporary backlog caused by those in-transit packets. Furthermore, undesired phenomena typical of dynamic routing strategies in packet networks, such as out-of-order delivery and temporary looping of packets, may occur. It is of course possible to devise graceful (or delayed) transitions between different schedulings, but this entails extra complexity, and is outside the scope of this paper, in which multi-hop strategies will be only considered as a best-case reference.

It should be noted that in the computer science field, problems similar to the one we are considering have been studied for a long time, and solved with a variety of well-understood approaches. As an example, we quote the problem of external fragmentation in dynamic memory allocation, and in storing contiguous files on disks. The algorithms that we propose in this paper are obviously inspired by the knowledge of those approaches; however, it must be emphasized that our main goal was to assess the effectiveness of extremely simple algorithms that can be run with acceptable cost and reasonable speed in very high-speed networking scenarios.

3 TRANSPARENT ON-LINE SCHEDULING

In this section we present three very simple on-line scheduling strategies, that satisfy the transparency constraint, assuming a fixed frame duration.

Slot allocations and de-allocations are performed only at frame boundaries, i.e., all bandwidth requests received by the controller during a frame are handled at the end of the frame.

We consider a network with N users exchanging a uniform amount of traffic (note that in the context that we consider the uniform traffic assumption implies the most difficulty in scheduling) and W wavelengths (for simplicity, N is assumed to be an integer multiple of W). User interfaces are supposed to be equipped with one tunable transmitter and one fixed receiver. All the strategies presented in this paper can however be easily adapted to the dual case in which transmitters are fixed and receivers are tunable.

In the case of tunable transmitters and fixed receivers, the destination address of the traffic flow immediately identifies the wavelength on which data must be transmitted. If $W < N$, more than one receiver share the same wavelength. This implies that the allocation of resources to transmitters only deals with slots, the wavelength being determined by the destination address. We assume that exactly N/W receivers share a given wavelength.

Slots can be identified with a pair (t, w) , representing respectively the slot position inside the frame and the wavelength. In general, slot (t, w) can be assigned by the controller to a traffic flow (i, j) from source i to destination j only if the following two conditions are satisfied: i) slot t on wavelength w is free, i.e., no other user is transmitting on that wavelength at the same time, ii) user i is source-free in slot t , i.e., it is neither transmitting on some other wavelength, nor tuning its transmitter. Note that the first condition implies that the receiver is free, and the second accounts for the tuning time constraints. If the two conditions are satisfied for slot t on wavelength w , we say that slot (t, w) is an (i, j) -eligible slot.

The novel requests contained in R_n are obtained collecting users' allocation and deallocation requests in a time frame. Obviously, when considering the requests in R_n , slot deallocations are handled before allocations. When the allocation of slots to traffic flow (i, j) must be reduced or eliminated, the previously allocated slots are simply freed. After all slot deallocations have been handled, allocations are considered one by one, starting with those concerning the largest numbers of new slots.

The slot allocation process is as follows. When the controller tries to allocate a request of k slots to flow (i, j) , it first determines the wavelength w on which the slots must be allocated. It then searches all (i, j) -eligible slots on w . If the number of (i, j) -eligible slots is less than k , the request allocation is rejected¹; otherwise it is accepted, and k slots among the (i, j) -eligible slots are assigned to the transmission of the flow. The three different strategies presented below use different algorithms to select k slots among the (i, j) -eligible slots that are assigned to the novel request. The three strategies try first to assign k consecutive slots to the novel request in order to reduce the time for which a transmitter is busy and, on the long run, the number of tuning actions that a user interface must perform. If a consecutive allocation is not possible, they all split the request into more than one sequence of consecutive slots. They differ in the criterion used to allocate slots in the first step, i.e., for the contiguous allocation, while they apply the same rules in the second step, i.e., when the allocation must be split.

Sequential Search (SS). A sequential search, starting from the beginning of the frame, of k contiguous (i, j) -eligible slots is performed; the search stops when the first k slots are found and assigned to the incoming request. If not enough contiguous slots are found, the sequential search is repeated, allocating the first k (i, j) -eligible slots in the frame.

The complexity of the SS strategy is obviously linear in the frame size, hence $O(F)$.

Best Fit Search (BFS). First a search for all sequences of at least k consecutive (i, j) -eligible slots is per-

formed. If at least one sequence is found, the minimum-size sequence (ties are randomly broken) is chosen; the first k slots in the sequence are devoted to transmit the packets corresponding to the novel request. If no sequences of k consecutive free slots are found, a second sequential search for k (non consecutive) (i, j) -eligible slots is started, as for the SS strategy.

The complexity of the BFS strategy is again linear in the frame size, hence $O(F)$. However, it requires the storage of a significant amount of information in the process of selecting the slots.

Minimum Cost Search (MCS). For each wavelength, a reward function is defined on the space of all possible assignments. The reward function is such that: i) it is positively defined; ii) the maximum value is reached when all slots on wavelength w are (i, j) -eligible; iii) its value is decreased after the assignment of free slots to a flow. In the first search, the set of k consecutive slots for which the global reward function is maximized after performing the novel assignment is chosen. If no sequence of k consecutive free slots is found, a second sequential search for k (non consecutive) (i, j) -eligible slots is started, as for the SS strategy. Note that a maximization of the reward function over all possible (contiguous and non contiguous) assignments of eligible slots would lead to better performance: the optimization of the reward function was restricted to consecutive allocations only for complexity reasons.

The MCS algorithm tries to minimize the probability that subsequent requests will be rejected. For this purpose, it attempts to maximize at the same time the probabilities that other requests involving source i , or wavelength w (i.e., destinations that receive packets on the same wavelength w), will be accepted. In order to maximize the probability of acceptance of other requests involving source i , it is convenient to allocate in the first instance the request on consecutive slots, thereby reducing the number of times in which source i must tune and the number of slots that are made non-eligible by the allocation, and to select among eligible slots those in which the largest number of other wavelengths are already busy. To maximize the probability of acceptance of other requests involving wavelength w , it is convenient to select those eligible slots in which the largest number of other sources are already transmitting or tuning.

Let's denote by $C_{tw} = N_{fs} + N_{fw}$ the reward associated with the (i, j) -eligible slot (t, w) , where N_{fs} is the number of free sources, and N_{fw} is the number of free wavelengths. Let S be the set of sequences of (one or more) contiguous source-free slots, i.e., slots on which source i is neither tuning nor transmitting. Let's denote by $\chi(s)$ the reward associated with each sequence $s \in S$ of contiguous source-free slots. We chose $\chi(s) = k_s \|s\|^{\alpha_s}$, where $\|s\|$

¹For simplicity, we do not consider partial allocation of user requests.

indicates the number of slots in sequence s , and k_s and α_s are suitable constants. Let W be the set of sequences of (one or more) contiguous wavelength-free slots, i.e., slots on which no user is transmitting on wavelength w . Let's denote by $\Psi(f)$ the reward associated with each sequence $f \in W$ of contiguous wavelength-free slots. We chose $\Psi(f) = k_f \|f\|^{\alpha_f}$, where k_f and α_f are suitable constants.

When a new request must be allocated on wavelength w , we first tentatively allocate the new request on eligible slots, then we compute the following reward function

$$M = \sum_{t \in F_{ij}} C_{tw} + \sum_{s \in S} \chi(s) + \sum_{f \in W} \Psi(f)$$

where F_{ij} represents the set of (i, j) -eligible slots. After having computed M for all possible allocations, we choose the allocation that provides the maximum value of M .

With the above definition of the reward function, it is possible to devise an implementation of the MCS strategy for which the complexity is quadratic in the frame size, hence $O(F^2)$.

Note that, under the above hypotheses, the algorithmic complexity of the strategy, i.e. the number of operations that must be executed in order to solve the slot assignment problem for *each* novel request, is a function of neither N nor W . However, the number of novel requests that a user can issue in a time frame can be assumed to be linearly dependent on the number of users N . The number of operations performed by the controller in a time frame to run the strategy for all the novel requests is, as a consequence, $O(N^2 F^2)$.

4 PERFORMANCE RESULTS

4.1 TRANSPARENT SCHEDULING

In this section, results are reported for the three strategies described in the previous section for on-line scheduling with a fixed frame duration.

We consider a network with $N = 16$, and $W = \{8, 16\}$, loaded with a uniform traffic pattern. Other values of N and W were also considered, but the results obtained in those cases show no qualitative difference from those reported below. The behavior of each connection (i, j) (i.e., the flow of packets from source i to destination j) is driven by a two-state Markov chain whose states are labelled ON and OFF. When the chain is in the OFF state, the connection is inactive, i.e., no packets are generated at source i for destination j . While in the ON state, a fixed number P_{ON} of packets is generated in each frame at source i with destination j . Note that the specification of the characteristics of the request arrival pattern during a frame is irrelevant, since requests are handled only at frame boundaries. P_{ON} is randomly selected when the chain enters the ON state, and is uniformly distributed between $P_{min} = 1$ and $P_{max} = 16$. The average sojourn time in the OFF state has been fixed to 600 frames, and the average sojourn time in the ON state has been fixed to 400 frames.

The frame duration strongly impacts the system performance, since it drives the network load: as previously noted, the bandwidth corresponding to a request of r slots per frame is inversely proportional to the frame duration. For a given request matrix R , if the frame is very long, the network is very lightly loaded, and all scheduling policies will manage to accommodate all requests. Conversely, if the frame is too short, the network is strongly overloaded and the probability of blocking requests is very large for all strategies.

In a similar manner the scheduling difficulty depends on the tuning latency. If we fix the frame duration and the number of requested slots, longer tuning times make the scheduling more difficult. However, if we allow the frame duration to be chosen proportional to the tuning time value, the effects of the tuning latency are more complex. Long tuning delays imply that the network efficiency is mostly limited by tuning, so that wavelengths are not a bottleneck, and scheduling is relatively easy. On the contrary, short tuning delays make the system bottleneck shift to wavelengths, so that the scheduling problem becomes harder. In order to put in evidence the differences among the considered strategies, it is therefore important to select with care the frame duration, that must be somehow tied to the traffic load and tuning latency. It must be noted that, given a request matrix $R = [r_{ij}]$, the minimum frame duration F_{min} to accommodate all requests is bounded by the following inequality:

$$F_{min} \geq F^* = \max \left[\max_i \left(\sum_j r_{ij} + K_i L \right), \max_w \sum_i \sum_{j \in D_w} r_{ij} \right] \quad (1)$$

where K_i is the number of wavelengths on which source i has packets to transmit, L is the tuning latency, i.e., the number of idle slots between two transmissions of the same source on different wavelengths, and D_w is the set of destinations receiving on w . Hence $\sum_j r_{ij} + K_i L$ is the minimum activity period of user i , including tuning actions, and $\sum_i \sum_{j \in D_w} r_{ij}$ is the minimum activity period on wavelength w .

In our case, the request matrix is not deterministically known. On the average, however, only 40% of connections will be active, due to the ratio between the average duration in frames of the ON and the OFF states. Under this assumption, the lower bound (1) was evaluated to obtain F^* , and the frame duration F was chosen to be $F = \beta F^*$. Results will be reported only for the case $\beta = 1.4$. This approach makes the frame duration F change with the tuning latency in the presentation of results: given the system setup in terms of offered load and tuning time constraints, we select a frame duration which is reasonably far from the minimum, i.e., for which an optimal allocation strategy should be very close to a complete accommodation of all information flows. This also means that performance curves in the sequel are plotted with different frame sizes

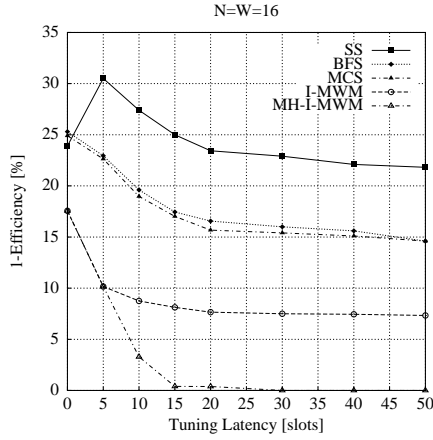


Figure 1: Network inefficiency $1 - \eta$ when $W = 16$, $N = 16$.

for different values of the tuning latency L , and that performance indices cannot be compared in their absolute values: while the different strategies are correctly compared for a fixed value of L , the values taken by the same strategy at different values of L must be compared with care. Note that this does not mean that we consider networks in which the frame duration is adaptive; this is contrary to the spirit of the scheduling strategies that we are considering (see the definition of the fixed-frame problem in Section 2). Rather, since we wish to compare the effectiveness of different scheduling algorithms for variable tuning latencies, we need to select a “critical” frame size to eliminate the effect of the growing scheduling difficulty for growing tuning times, in order to highlight different behaviors.

Simulation was used to study the behavior of the proposed algorithms. Each simulation experiment approximately lasts 500,000 frame times, hence each user-to-user flow goes through approximately 500 ON/OFF cycles in each experiment. Performance results will be presented in terms of efficiency η (or inefficiency $1 - \eta$) and of the percentage $P\{\text{split}\}$ of accepted requests that cannot be allocated in contiguous slots. The system efficiency η is defined as the ratio between the average network throughput and the average offered load. $1 - \eta$ is therefore the percentage of packets that could not be scheduled in the TDM/WDM frame. Note that this performance index is meaningful only for the fixed-frame formalization of the scheduling problem; in the case of variable frame size, the frame duration should be considered instead.

A rough optimization of the parameters k_s , k_f , α_s and α_f for minimum cost strategies was carried out by repeated simulation experiments. Results presented in the sequel refer to the values $k_s = 1.5$, $\alpha_s = 1.2$ and $k_f = 0$, which proved to yield good results. It must be said that our investigations showed a weak sensitivity of the system performance to the values taken by these parameters.

Figure 1 plots $1 - \eta$ for the three on-line strategies described in section 3 in the case $W = 16$, $N = 16$. In addition to the SS, BFS, and MCS strategies, the figure also shows the performance achieved by first releasing the

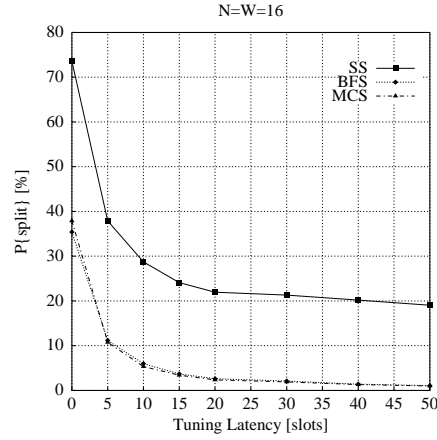


Figure 2: Percentage of requests allocated in non consecutive slots for transparent on-line strategies when $W = 16$, $N = 16$.

transparency constraint (curve labeled I-MWM), and then by also permitting multi-hop transmissions (curve labeled MH-I-MWM). For the first reference curve, i.e., when the transparency constraint is not enforced, we consider the case in which at every request a complete re-allocation of all existing flows is performed according to the I-MWM strategy described in [3]. I-MWM, which is based on the repeated application of a standard maximum weight matching algorithm, proved to be one of the best performing off-line strategies. The asymptotic complexity of I-MWM is $O(WN^4 \log_2 N)$, i.e., larger than that of the on-line strategies proposed in this paper, which is, in the worst case, i.e. for MCS, $O(N^2 F^2)$. For the multi-hop case, at each request a complete re-allocation of existing flows is again applied. We implemented the MH-I-MWM algorithm, a multi-hop variation of I-MWM, again proposed in [3]. Obviously, the algorithmic complexity and signalling bandwidth requirements of these two off-line strategies heavily limit their applicability in high-speed networking on-line contexts.

The first general observation on the curves of Figure 1 is that, as explained before, efficiency grows for increasing tuning latencies. As we noted, this is due to the fact that we chose frame durations that are tied to the scheduling latency in order to eliminate the negative impact of the tuning delay on the system performance. A significant performance improvement (efficiency grows from about 77% to about 84% for intermediate values of L) can be achieved by using BFS instead of SS for non-null values of the tuning time L (the case $L = 0$, in which SS performs slightly better than BFS and MCS, is atypical for the assumptions of this paper). The better performance of BFS with respect to SS is mainly due to the fact that BFS leads to a strong reduction of the probability of non consecutive slot allocations, as shown in Figure 2. It is quite interesting to note that the MCS strategy, despite the remarkable complexity increase, leads only to marginal improvements over BFS. By completely releasing the transparency constraint, i.e. by

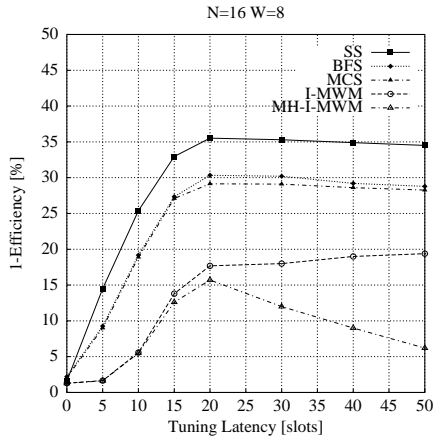


Figure 3: $1 - \eta$ when $W = 8$, $N = 16$.

performing a complete re-allocation of all active connections at each new request (strategy I-MWM), a significant gain (an additional 8% for intermediate values of L) in the system efficiency can be achieved. Further gains, which increase for increasing values of the tuning time, can be achieved by introducing the multi-hop option, for which no losses were observed for large tuning times.

In Figures 3 and 4, performance curves are reported when $W = 8$, $N = 16$, i.e. when two receivers share each wavelength. Also in this case, the advantage of using BFS instead of SS is evident for non-null values of L , while the differences between BFS and MCS are negligible. For large values of L , the advantages of the strategies that perform a complete re-allocation of all the existing connections every time a request is received by the controller are even more significant. This is mainly due to the fact that policies that re-allocate all existing connections allow users to transmit on the same wavelength in consecutive slots all packets directed to all receivers reachable on that wavelength, leading to a reduction of the number of tuning actions that must be performed. On-line scheduling strategies, instead, allocate the requests coming from the same user and directed to the different receivers reachable on the same wavelength at different times, hence in general they are not able to aggregate the different flows. As a consequence, even when each request is allocated in consecutive slots, the number of tuning actions is not smaller than the number of destinations that must be reached.

The gain achieved when $W = 8$ by allowing flows to be routed in a multi-hop fashion almost disappears for small values of the tuning latency. This is due to the fact that, for small values of the tuning time, the system performance is limited by congestion on wavelengths, i.e., by the reduced availability of free slots: the load on each wavelength on the average is $N/W = 2$ times the source load. In these conditions, the multi-hop optimization of the scheduling algorithms cannot pay off in terms of performance. Only for large values of L , the network performance becomes limited by the source inefficiency due to tuning actions (the average busy period of the transmitters, including tuning

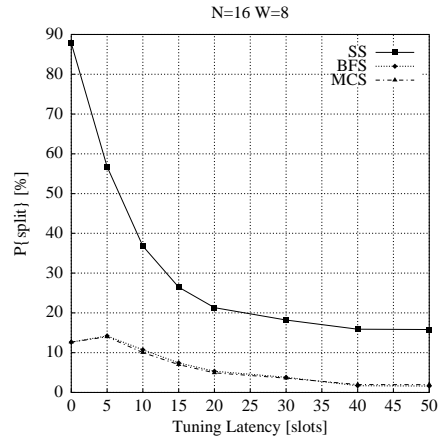


Figure 4: Percentage of requests allocated in non consecutive slots for transparent on-line strategies with $W = 8$, $N = 16$.

actions, becomes closer to the wavelength busy periods), and multi-hop routing becomes more effective in reducing tuning time overheads. These same arguments explain why the curves of $1 - \eta$ are decreasing for increasing L in the case $W = 16$, $N = 16$, in which performance is only limited by tuning times overheads, while they first increase and then decrease for increasing L in the case $W = 8$, $N = 16$, in which wavelength congestion dominates performance at low values of L .

4.2 ALMOST-TRANSPARENT SCHEDULING

In order to improve the performance of on-line scheduling algorithms, it is possible to occasionally perform a complete re-allocation of active flows. This means that a non transparent scheduling algorithm is run by the network controller at some points in time, and that the novel allocation pattern must be distributed to every user. The non-transparent re-allocation is not performed at every received request: between two consecutive re-allocations an on-line transparent strategy is used to accommodate incoming requests. We call these strategies *almost-transparent* schedulings. Almost-transparent schedulings offer a trade-off between complexity required at the controller and achievable performance.

Note that the ideas behind the almost-transparent approach are very similar to those used by algorithms that reduce segmentation in memory managers, and by (usually off-line) de-fragmentation tools for computer disks.

Figure 5 plots, for $N = W = 16$, $1 - \eta$ for the almost-transparent approach, in which the best performing on-line scheduling strategy MCS is used in conjunction with the I-MWM non transparent strategy. The complete re-allocation is performed *at a fixed rate*, i.e., I-MWM is executed instead of MCS once every $X = \{10, 50, 100\}$ allocation requests received by the controller. The figure also reports results for the transparent on-line strategy MCS alone (corresponding to $X = \infty$), and the non transparent strategy I-MWM alone (corresponding to $X = 1$). A per-

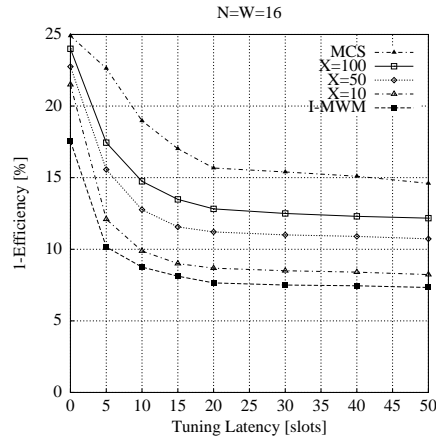


Figure 5: $1 - \eta$ for almost-transparent strategies when $W = 16$, $N = 16$.

formance gain around 5% with respect to the purely transparent on-line strategy can be achieved using an almost-transparent scheduling at rate $1/X = 1/50$. As expected, the gain is reduced for larger values of X .

Similar behaviors were observed when the non-transparent allocation is triggered by the observation of a decrease in the system efficiency $1 - \eta$.

5 CONCLUSIONS

This paper considered all-optical broadcast-and-select networks providing W slotted wavelengths for packet communications, in which each user interface is equipped with one tunable transmitter and one fixed receiver, and tuning times are not negligible with respect to slot times. The access to network resources is ruled by a network controller, to which requests of (long-term) bandwidth variations are issued by the users over a dedicated channel, whose performance is assumed to be ideal. While proposals in the technical literature mainly refer to off-line and non-transparent scheduling strategies, we proposed in the paper on-line and transparent approaches. On-line means that requests of bandwidth variations reach the controller and are accommodated over time by varying the TDM/WDM scheduling. Transparency means that the accommodation of an incoming request does not affect the behavior of previously allocated information flows.

The proposed strategies are very simple to implement with the technology available today. Although we considered the case of a centralized controller, the scheduling function can also be distributed over all user interfaces, provided that allocation requests are broadcasted in the network.

We compared by simulation the proposed strategies with a non-transparent off-line strategy that proved to yield good performance. In addition to simplicity, our simulation results show that good performance can be achieved, specially by strategies that attempt to allocate in contiguous slots all the transmissions of each source on one wave-

length, thereby reducing tuning overheads. Even better performance can be achieved, at the cost of a modest complexity increase, if the almost-transparent approach is taken, in which the transparency constraint is not strictly imposed, i.e., when a full re-allocation of existing connections is performed once in while.

We also showed that additional performance gains can be achieved if the information flows are routed in a multi-hop fashion through intermediate user interfaces.

Manuscript received on ...

REFERENCES

- [1] A. Ganz, Y. Gao. A Time-Wavelength Assignment Algorithm For a WDM Star Network. In *IEEE INFOCOM'92*, Florence, Italy, 1992.
- [2] M.S. Borella, B. Mukherjee. Efficient Scheduling of Nonuniform Packet Traffic in a WDM/TDM Local Lightwave Network with Arbitrary Transceiver Tuning Latencies. *IEEE Journal on Selected Areas of Communications, Special Issue on Multiwavelength Optical Technology and Networks*, Vol. JSAC-14, No. 5, pages 923-934, June 1996.
- [3] M. Ajmone Marsan, A. Bianco, E. Leonardi, F. Neri, A. Nucci. Efficient Multi-Hop Scheduling Algorithms for All Optical WDM Broadcast-and-Select Networks with Arbitrary Transceivers Tuning Latencies. In *IEEE GLOBECOM'98*, Sydney, Australia, 1998.
- [4] G.P. Pieris, G.H. Sasaki. Scheduling Transmissions in WDM Broadcast and Select Networks. *IEEE/ACM Transactions on Networking*, Vol. TON-2, No. 2, pages 105-110, April 1994.
- [5] M. Azizoglu, R.A. Barry, A. Mokhar. Impact of Tuning delay on Performance of Bandwidth-Limited Optical Broadcast Networks with Uniform Traffic. *IEEE Journal on Selected Areas of Communications*, Vol. JSAC-14, No. 5, pages 935-944, June 1996.
- [6] G.N. Rouskas, V. Sivaraman. Packet Scheduling in Broadcast WDM Networks with Arbitrary Transceivers Tuning Latencies. *IEEE/ACM Transactions on Networking*, Vol. TON-5, No. 3, pages 359-370, June 1997.
- [7] I.S. Gopal, C.K. Wong. Minimizing the Number of Switchings in a SS/TDMA System. *IEEE Transactions on Communications*, Vol. COM-33, No. 6, pages 497-501, June 1985.
- [8] M.A. Bonuccelli, I. Gopal, C.F. Wong. Incremental Time-Slot Assignment in SS/TDMA Satellite Systems *IEEE Transactions on Communications*, Vol. TOC-39, No. 7, pages 1147-1156, July 1991.
- [9] K.M. Sivalingam, J. Wang. Media Access Protocols for WDM Networks with On-Line Scheduling *IEEE Journal on Lightwave Technology, Special Issue on Multiwavelength Optical Technology and Networks*, Vol. JLT-14, No. 6, pages 1278-1286, June 1996.
- [10] B. Wang, C.J. Hou, C.C. Han. On Dynamically Establishing and Terminating Isochronous Message Streams in WDM-based Local Area Lightwave Networks. In *IEEE INFOCOM'97*, Kobe, Japan, April 1997.