

Network Controller Design for SONATA— A Large-Scale All-Optical Passive Network

Andrea Bianco, *Member, IEEE*, Emilio Leonardi, *Member, IEEE*, Marco Mellia, *Student Member, IEEE*, and Fabio Neri, *Member, IEEE*

Abstract—This paper describes the network architecture and provides a performance analysis of a passive optical network named SONATA, which has been proposed and demonstrated in the context of the European Union ACTS Program. In this nationwide all-optical network, end terminals access a single passive routing node via PONs using a TDMA/WDMA access scheme based on reservations. The centralized network controller runs resource allocation algorithms in order to avoid conflicts among end terminals. We formally define the resource allocation problem at the network controller, and show that, in general, it is NP-hard. We also provide simple heuristic algorithms to solve the problem. The analysis of the algorithms is performed both via analysis and simulation.

Index Terms—All-optical networks, network control and management, wavelength division multiplexing.

I. INTRODUCTION

THE project SONATA aims at avoiding the need for large and fast switching electronic nodes in a high-speed nationwide network. To reach this goal, the network structure and the layer architecture within the network have been drastically simplified. The new proposal consists of a single-layer network platform for end-to-end optical connections, able to serve a very large number of terminals for both business and residential customers over a nationwide geographical area.

In the SONATA network, a large number of end terminals which, depending on the network usage, can be street cabinets, IP routers, LAN switches, or workstations, are grouped in a passive optical network (PON) infrastructure. The network topology is based on a centralized passive wavelength-routing node (PWRN) with N input and N output ports, which provides full connectivity among PONs via a single dedicated wavelength channel for each pair of PONs. Wavelength converter arrays are used at the PWRN in order to dynamically increase the channel capacity among pairs of PONs, as described below.

This multiple-access network is based on WDMA/TDMA protocols, and exploits time and wavelength agility at terminals in a simple network structure where the single PWRN node provides passive routing functions and actively controlled wavelength conversion. The network control is centralized, and its primary goal is to assign time/wavelength resources to terminals: a signalling protocol that allows terminals to request both connection-oriented and connectionless services is designed.

Manuscript received October 29, 1999; revised May 12, 2000. This work was supported in part by the European Union ACTS Program under project AC351 SONATA.

The authors are with the Dipartimento di Elettronica, Politecnico di Torino, Italy (e-mail: {bianco, leonardi, mellia, neri}@polito.it).

Publisher Item Identifier S 0733-8716(00)09020-X.

The resource assignment performed at the network controller (NC) avoids conflicts among transmitters and receivers when transmitting data.

In SONATA, the physical switching function is removed from the nodes and distributed at the terminals. Although a centralized network control is required, and active wavelength conversion is performed inside the network, we refer to the SONATA network structure as a “switchless” network, in the sense that neither purely electronic switching nodes or cross-connects (telephony, IP, ATM, SDH) are required within the network, nor optical cross-connects (except for the wavelength routing node). Moreover, and most remarkably, the network is completely bufferless. This approach provides major network architecture simplifications and hardware reductions.

We do not tackle issues related to the SONATA network feasibility in this paper, nor do we discuss the components that should be used or the physical limitations that should be taken into account when dimensioning the network. All these issues, together with the specification of the signalling protocol between terminals and the NC, have been deeply analyzed in the SONATA project, and are discussed in [1] and [2].

The focus of the paper is on algorithms that must be executed at the NC to solve the resource allocation problem. We first provide a more detailed description of the switchless network architecture. Then, we discuss the time/wavelength resource allocation problem at the network controller. We provide an integer linear programming (ILP) formulation of the problem, and we show that it is in general NP-hard. Even restricting the problem to make it tractable using polynomial algorithms is impractical given the network size in terms of terminals and PONs. As a consequence, we propose a simple heuristic to solve the problem of resource allocation: we decouple the time dimension from the wavelength dimension, thus splitting the problem into two sub-problems: the scheduling of terminal requests in the time domain given a PON-to-PON channel assignment, and the design of the logical topology, i.e., the assignment of wavelengths to PONs via a proper setting of wavelength converters. We describe two analytical models for both the scheduling and the logical topology design problems. Finally, we analyze the performance of the algorithms used at the network controller both via simulation and analysis.

II. NETWORK ARCHITECTURE

We provide a description of the network architecture in this section. The SONATA switchless network has the target of performing the concentration/distribution, switching, and routing

TABLE I
NETWORK CONFIGURATION AND DIMENSION PARAMETERS

$N_t = 20$ million terminals	$N_p = 400$ PONs
$n_t = 50,000$ terminals per PON	$N_d = 400$ wavelength converter arrays, 400 wavelength converters each
$W = 801$ wavelength channels on each fiber	$N_{wr} = 1$ PWRN (Passive Wavelength Router Node), 801×801 channels
$B_t = 10$ Mbit/s average rate per terminal	$B_p = 622$ Mbit/s maximum rate per terminal
$F = 1,000$ slots per frame	$10\mu s$ of slot time, with $1\mu s$ devoted to guard time
200 Tbit/s maximum network throughput	$D_m = 1000$ km maximum distance between terminals.

functions within a single network layer by providing end-to-end optical connections between a large number of terminals over a large geographical area.

The structure of the switchless network is depicted in Fig. 1. Terminals equipped with fast tunable transmitters and receivers are attached to passive optical network (PON) infrastructures. Each PON is directly connected to an input and an output port of the single passive wavelength-routing node (PWRN). The PWRN has N input and N output ports, i.e., it is an $N \times N$ wavelength multiplexer. Logically, the behavior of the PWRN is such that wavelength λ_k at input i is routed to output $j = |i + k|_N$. Hence, wavelength channels $0, 1, 2, \dots, N - 1$ on input port i lead to output ports $i, i + 1, i + 2, \dots, i - 1$, and, on output fiber j , wavelength channels $0, 1, 2, \dots, N - 1$, transport information originated at input $j, j - 1, j - 2, \dots, j + 1$.

A global synchronization is made available by centralized distribution of reference tones and by the execution of ranging procedures; transmission is organized in WDMA/TDMA frames. Each frame is composed of F fixed-length slots. Any terminal wishing to communicate with another terminal simply tunes, in the allocated time slots, its transmitter and receiver to a known wavelength carrying multiplexed traffic through the PWRN between the pair of PONs to which the transmitting and receiving terminals are attached. Only one tunable transmitter/receiver pair per terminal (plus a fixed transmitter/receiver for signalling) therefore permits data communication between a given terminal and all other terminals in the network, regardless of the network size. Note that multicast transmissions to several terminals belonging to the same PON can be supported.

Each wavelength is shared between all terminals belonging to the same PON. A TDMA control protocol is thus required. Since time is divided into time slots, and each time slot can be used in a particular wavelength channel by only one of the terminals connected to the same PWRN port, a significant complexity is required in the network controller operations. Further constraints on the controller are due to the fact that the single transmitter/receiver pair has to be shared by all the connections which the terminal keeps active at the same time; more precisely, to permit a transmission from terminal s to terminal d in slot k , we must ensure that s is not already transmitting in k and also that d is not already engaged in a reception in k .

Although all connections can be made directly through the PWRN, for unbalanced traffic patterns and when traffic volumes grow toward the capacity limit of the network, a high degree of

flexibility in allocating capacity can be obtained with actively controlled wavelength converter arrays. As shown in Fig. 2, these devices can be added, as required, around the PWRN, and connected to a certain number (N_d) of its auxiliary ports (called “dummy ports”). On each PON, a wavelength channel is available to reach a particular dummy port, hence a wavelength converter array. From each array, one wavelength channel is provided to reach every output PON. Hence, information from PON i to PON j can be routed on the direct channel $|j - i|_N$, and on up to N_d additional channels going through wavelength conversion. In this way a variable number of wavelengths, depending on traffic requirements, can be dynamically allocated between a pair of PONs as traffic demand requires.

Additional N_c ports of the PWRN interconnect network terminals with the centralized network control (NC) device, which is responsible for allocating time slots and wavelength channels to terminal requests. This network controller may be duplicated for reliability concerns.

The optimum technical solution for a “switchless” network depends on the size of the network itself. As discussed in [1], a feasible set of characteristics for a large-scale switchless network, which can provide a communication infrastructure to terminals at the customer premises distributed nationwide, is reported in Table I.

A. Logical Topology

The network structure depicted in Fig. 1 provides a full mesh topology between PONs through direct connections, and a large number of additional connections through wavelength converter arrays. The first N_p ports of the PWRN provide a “wired” (fixed) full-mesh connectivity between PONs, while the additional N_d dummy ports add a “programmable” connectivity between PONs. Fig. 2 shows the equivalent logical topology provided in the programmable portion of the PWRN in the case $N_d = N_p = 4$. Terminals in PON i can reach any of the N_d wavelength converter arrays by tuning to the proper wavelength channel. This tuning capability corresponds to the first λ -switching stage in the logical topology. The individual wavelength converters can be configured in order to route input information to any output PON, leading to the second λ -switching stage in Fig. 2. The receivers in each PON can tune to any of the N_d wavelengths routed to the PON by wavelength converter arrays, leading to the third λ -switching

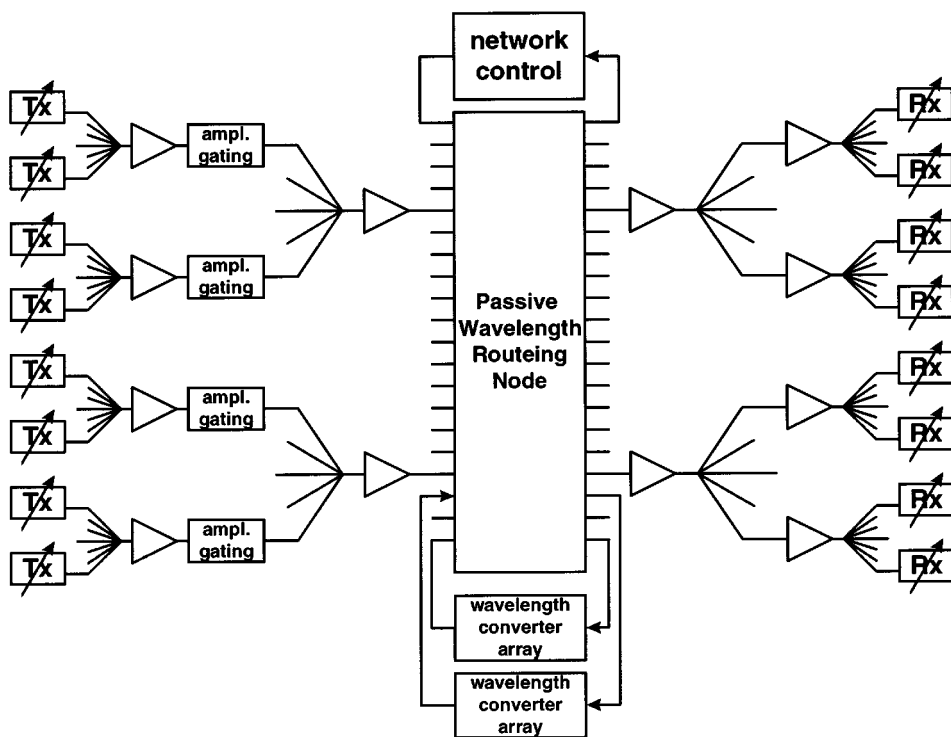


Fig. 1. SONATA network architecture.

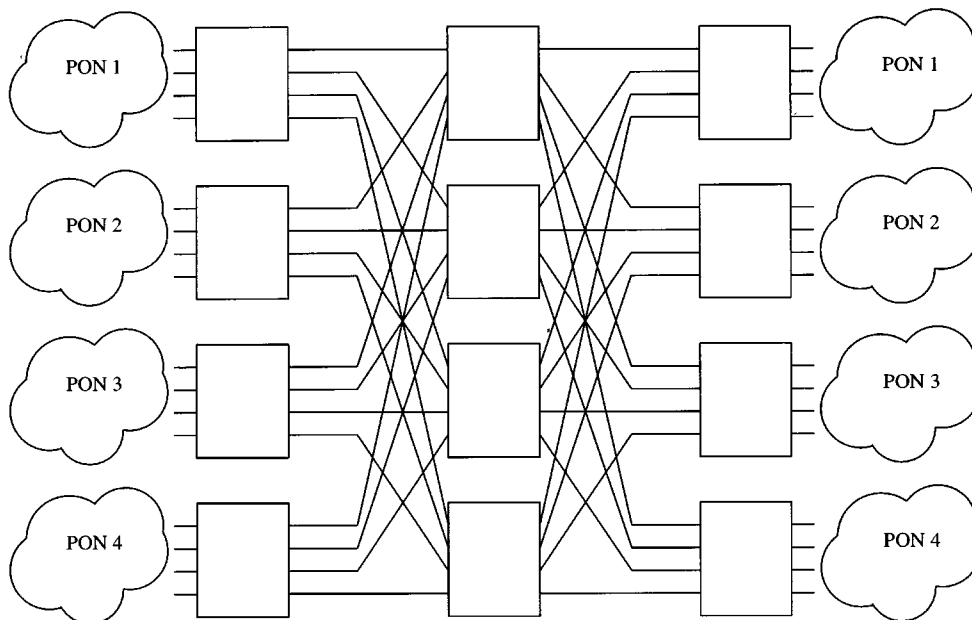


Fig. 2. Three-stage equivalent of the programmable portion of the SONATA network.

stage. The resulting logical topology is a three-stage $\lambda - \lambda - \lambda$ switch, i.e., a Clos interconnection network. It is well known from the Slepian–Duguid theorem [4] that this three-stage interconnection structure is rearrangeable nonblocking when $N_d \geq N_p$, as is normally the case for SONATA. Note that the interconnection between PONs is nonblocking, while the interconnection between terminals is blocking, due to the multiplexing of many terminals in each PON (at most 400 out of 50 000 terminals can receive/transmit in each slot).

If we keep considering only the programmable portion of the network, the possibility of allocating transmission requests in

different slots within the time frame can be viewed as a further switching stage in the time domain. Overall, the programmable portion of the switchless network can be logically described as a five-stage $T - \lambda - \lambda - \lambda - T$ switch, which can be reconfigured in every time slot.

III. NETWORK CONTROLLER

The main task of the network controller is to allocate resources (time slots) to end terminals, choosing a proper wavelength among the available ones. We first formalize the network

controller task as an ILP (integer linear programming) problem. Then, we discuss the feasibility of this approach given the network parameters we should handle in a nationwide network. Later we present our proposal, that is obviously suboptimal, but can be implemented with reasonable complexity in a real network. We will discuss the performance achieved by our algorithms later in the paper.

A. Optimal Resource Allocation: ILP Formulation

We provide in this section an ILP formulation for the resource allocation problem in SONATA.

We first define some indexes:

- s is the index to address a source terminal in the set containing all source terminals;
- d is the index to address a destination terminal in the set containing all destination terminals;
- k is the slot index in the time frame, whose duration is F slots;
- w is the wavelength index in the set of all the possible wavelengths \mathcal{W} .

Recall that N_d is the number of wavelength converters that each source terminal can reach.

Let \mathcal{S} be the set of all source terminals that belong to PON S , and \mathcal{D} the set of all destination terminals that belong to PON D .

Finally, let $r_{s,d}$ be the number of slots that source terminal s has to transmit to destination terminal d during a frame. We consider this request as atomic: either all the slots are scheduled or the request is rejected.

Now, we define the variables that are used in the ILP formulation.

- $T_{s,d,w}(k)$ is a binary variable that takes the value 1 if terminal s is scheduled to transmit to terminal d in the k th slot using wavelength w .
- $C_{S,D,w}(k)$ is a binary variable that takes the value 1 if PON S is connected to PON D during the k th slot via wavelength w . Note that $C_{S,D,w}(k)$ is fixed (not a variable) if $w \in \mathcal{W}_S^{fix}$, where \mathcal{W}_S^{fix} is the subset of wavelengths that connects directly PON S to any PON via the fixed portion of the network, while $C_{S,D,w}(k)$ is a state variable if $w \in \mathcal{W}_S^{prg}$, where \mathcal{W}_S^{prg} is the subset of wavelengths connecting PON S to a wavelength converter array in the programmable portion of the network.
- $X_{s,d}$ is a binary variable that takes the value 1 if all slots from source terminal s to destination terminal d are scheduled in the current frame.

Our objective function is to maximize the total number of transmissions per frame:

$$\max \sum_{s,d,w,k} T_{s,d,w}(k)$$

subject to the following constraints.

- 1) The number of slots allocated from each source s to each destination d is either equal to the number of requested slots or zero (atomicity constraint):

$$\sum_{w,k} T_{s,d,w}(k) = r_{s,d} X_{s,d} \quad \forall s, d. \quad (1)$$

- 2) In each slot, a source terminal can only transmit one packet:

$$\sum_{w,d} T_{s,d,w}(k) \leq 1 \quad \forall k, s. \quad (2)$$

- 3) In each slot, a destination terminal can only receive one packet:

$$\sum_{w,s} T_{s,d,w}(k) \leq 1 \quad \forall k, d. \quad (3)$$

- 4) At most one transmitter can use a particular slot in a given wavelength channel, provided that a path exists from source PON S to destination PON D :

$$\sum_{s \in \mathcal{S}, d \in \mathcal{D}} T_{s,d,w}(k) \leq C_{S,D,w}(k) \quad \forall k, w. \quad (4)$$

- 5) The w wavelengths can be used only once by a single source PON S , for all the slots k :

$$\sum_{S,D} C_{S,D,w}(k) \leq 1 \quad \forall k, w \in \mathcal{W}_S^{prg}. \quad (5)$$

- 6) The number of wavelength converters that source PON S can reach in each slot k is always smaller than N_d :

$$\sum_{D,w \in \mathcal{W}_S^{prg}} C_{S,D,w}(k) \leq N_d \quad \forall S, k. \quad (6)$$

- 7) The number of wavelength converters that destination PON D can use in each slot k is always smaller than N_d :

$$\sum_{S,w \in \mathcal{W}_S^{prg}} C_{S,D,w}(k) \leq N_d \quad \forall D, k. \quad (7)$$

Note that we could relax the atomicity constraint, i.e., atomic allocation of terminals requests, by modifying constraint (1) as:

$$\sum_{w,k} T_{s,d,w}(k) \leq r_{s,d} \quad \forall s, d.$$

In this case, the variables $X_{s,d}$ are not needed in the formulation.

B. Optimal Resource Allocation: Feasibility

We have formulated the problem as an ILP problem. It is well known that the general ILP problem is NP-hard. However, the ILP formulation by itself does not imply that the problem is NP-hard; we are able to show that, under the atomicity constraint for terminal requests, the problem is NP-hard, whereas, when relaxing the above constraint, and considering only the programmable portion of the SONATA network, polynomial algorithms can be used to solve the problem. We are unable to formally prove that for the complete network (i.e., considering both fixed and programmable portions) without the atomicity constraints on terminal requests the problem is still NP-hard, although we provide some considerations suggesting that this is a reasonable assumption.

Note that, given the very large number of terminals, even with polynomial algorithms it would be impractical to implement an optimal solution. Moreover, to make the problem practical, the

resource allocation must be done on-line (while the ILP formulation assumes off-line operation), i.e., the network controller decides whether or not to accept slot allocation requests given an allocation state without modifying previously allocated requests. Otherwise, the signalling bandwidth required to notify from network controller the time/wavelength assignment to end terminals would be very large, as discussed in [2]. For all the above reasons, we propose later in this paper a simplified approach to the resource allocation problem.

Let us assume that the NC is subject to the atomicity constraint, i.e., either all or none of the requested slots are allocated. In this context, if we consider only a source and a destination PON, the NC task is to allocate terminal requests given a number of available slots ranging from a minimum of F , equal to the frame size on the fixed wavelength connecting S to D , to a maximum of $F \times (N_d + 1)$, equal to the frame size multiplied by the maximum number of channels available from S to D . This is equivalent to a knapsack problem which is known to be an NP-hard problem [7]. As a consequence, the SONATA NC resource allocation problem is also NP-hard.

Let us now relinquish the atomicity constraint: the NC could partially allocate slot requests. In this context, if we consider only the programmable portion of the network, the SONATA network can be seen as a hierarchical switching system (HSS, see [5]), with N_t inputs and outputs. In [6], it has been shown that the time slot assignment (TSA) in an HSS is equivalent to a TSA in a simple TDM switching system with a significantly larger number of input and outputs. The TSA in a TDM switching system can be solved using single commodity network flow algorithms which have polynomial complexity [8].

If we consider the complete SONATA network, i.e., also the fixed portion of the network, we can model the system by using a multicommodity network flow integer formulation, which is known to be NP-hard [8]. However, we are not able to formally prove that the problem is equivalent to a well-known NP-hard problem.

C. The SONATA Resource Allocation Algorithm

Although better algorithms to solve the problem of resource allocation at the network controller are possible, we propose in this paper a simple approach based on the decoupling of the time dimension from the wavelength dimension. This means that we split the resource allocation problem into two subproblems: *scheduling* of terminal requests in the time domain given a PON-to-PON channel assignment, and *logical topology design* of the network connectivity by properly assigning wavelengths to PONs via wavelength converter arrays.

While scheduling is a relatively simple task which must be done slot by slot, the more complex logical topology design is assumed to be performed at a lower rate. Aiming at a suboptimal approach with a limited complexity, we assume that the wavelength converters are reconfigured only once in a while (e.g., at frame boundaries) and that the scheduling algorithm operates on a given fixed logical topology.

We analyze in the sequel the two-steps of the resource allocation algorithm:

- *scheduling*: first-fit allocation of slots given a fixed network connectivity;
- *logical topology design*: reconfiguration of the network connectivity via wavelength converter arrays.

1) *Scheduling*: The scheduling algorithm is executed at the network controller on the basis of terminal requests received via signalling messages. Terminals can request slots according to two different modes:

- “persistent” request—an integer number of slots in each frame is assigned at connection setup; this number can be changed (either increased or decreased) during the connection life;
- “nonpersistent” request—a possibly large amount of slots is assigned to a datagram communication, without time constraints in terms of number of involved frames.

Obviously, the two modes refer mainly to CBR (or slowly varying VBR) connection-oriented services, and to datagram services, respectively.

When terminal s belonging to PON S sends a request to the NC in order to communicate with terminal d belonging to PON D , the sequence of steps executed at the NC is the following.

- If terminal s requests the set up of a new connection, the NC looks for a candidate wavelength channel by scanning a linked list of wavelength channels assigned to the pair of PON $S \rightarrow D$; the linked list is ordered in increasing order of channel utilization, and the first element is always the “wired” channel among PONs $S \rightarrow D$. The new allocation is attempted on the first channel on the list that satisfies the condition that the number of free slots exceeds by a given percentage b the number of requested slots.
- If s is requiring new slots for an already set-up connection, the NC considers the first wavelength channel used for this connection. The NC counts the available slots, excluding among free slots those in which either the source terminal or the destination terminal are busy (possibly for transmitting/receiving data associated to different connections).
- If the amount of free slots resulting from the previous operation is less than the terminal request, the NC looks for another wavelength channel (only for a new connection request).
- If none of the wavelength channels already used for the communication from PON S to PON D is usable, the request is not accepted.
- If the number of free slots matches the terminal request, the NC selects the first available slots in the selected channel (first-fit allocation).
- In the case of a datagram request, if the number of available free slots is less than the number of data slots requested by the end terminal, the NC could also decide to operate a partial resources allocation: a threshold can be defined for this goal.

Fig. 3 provides a block diagram of the proposed scheduling algorithm.

For connection-oriented (or persistent) requests, resources are allocated until one of the two involved terminals signals

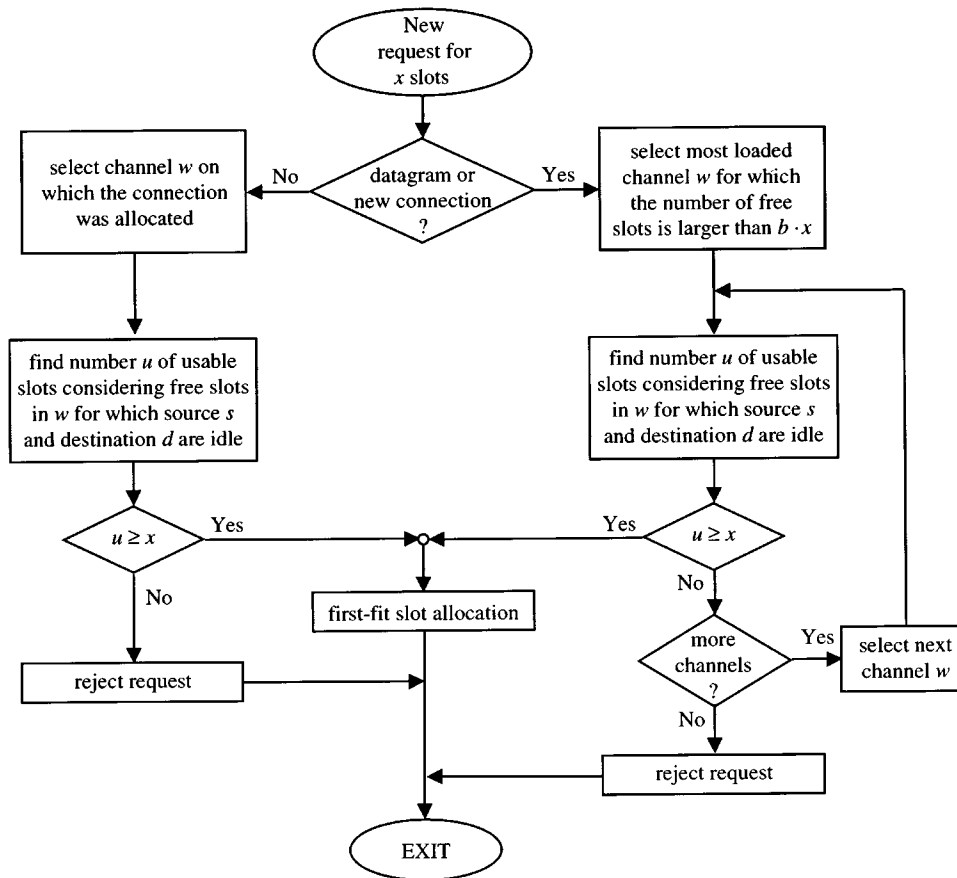


Fig. 3. Block diagram for the scheduling algorithm.

that it wants either to release some slots for this connection or to close it. For datagram (or nonpersistent) requests, we assume that resources are allocated only in a single frame; in the next frame, the NC will release all the resources previously allocated to datagram traffic.

2) *Logical Topology Design*: The logical topology design aims at obtaining an efficient allocation of available channels among PONs. The idea is quite simple: in order to reduce the slot allocation failure rate, the NC can reassign channels. A channel experiencing a load below a given threshold can be freed, obviously rerouting connections previously assigned to that channel; free channels can be assigned to pairs of PONs experiencing a load above a given threshold.

In every frame, a pair of PONs (a source PON S and a destination PON D) is selected using a round-robin algorithm. We compute L_{SD} , the number of free slots on all the channels from PON S to PON D . If L_{SD} is larger than threshold L , we try to release a single channel, choosing the least loaded channel. If L_{SD} is below threshold $H < L$, we try to assign another channel to the pair of PONs, provided that at least a free channel is available.

In order to avoid user service interruptions, the process of channel release is subject to the reallocation of all used slots in the selected channel. Only if this process is successful, i.e., all the slots can be allocated on other channels, is the selected channel freed. We perform a complete search on all other channels allocated to the pair of PONs in order to reallocate every single slot, starting from the “wired” channel and continuing

on all the other channels in the ordered list. If the reallocation process is successful, the channel can be freed and added to a list of available channels. Otherwise, if even a single slot is not successfully reassigned, the channel cannot be released. Note that, to be more precise, releasing a channel means releasing a wavelength on the transmitter side, and a wavelength on the receiver side.

When trying to add a channel we consider only a single channel addition between the PONs selected according to the round robin scheme. We first look for an available wavelength converter, i.e., a wavelength converter that is not used by both PON S and PON D ; if found, we compute the wavelength that should be used at the transmitter and at the receiver, and we logically add the channel to the pair of PONs.

If an available converter is not directly found, we check whether an available wavelength at both the transmitter and the receiver exists. If this is true, we know by the Slepian–Duguid theorem [4] that we can rearrange the logical topology so that at the end of the process we obtain an available wavelength converter. Paull’s algorithm [4] can be used to obtain the new logical topology, i.e., the new wavelength assignment. The latter algorithm is based on an iterative process that, after reassigning at most N_p wavelength converters, guarantees the availability of a wavelength converter among the two PONs.

Fig. 4 provides a block diagram of the proposed logical topology design algorithm.

Note that each reassignment made while running the algorithm implies updating a significant amount of data structures

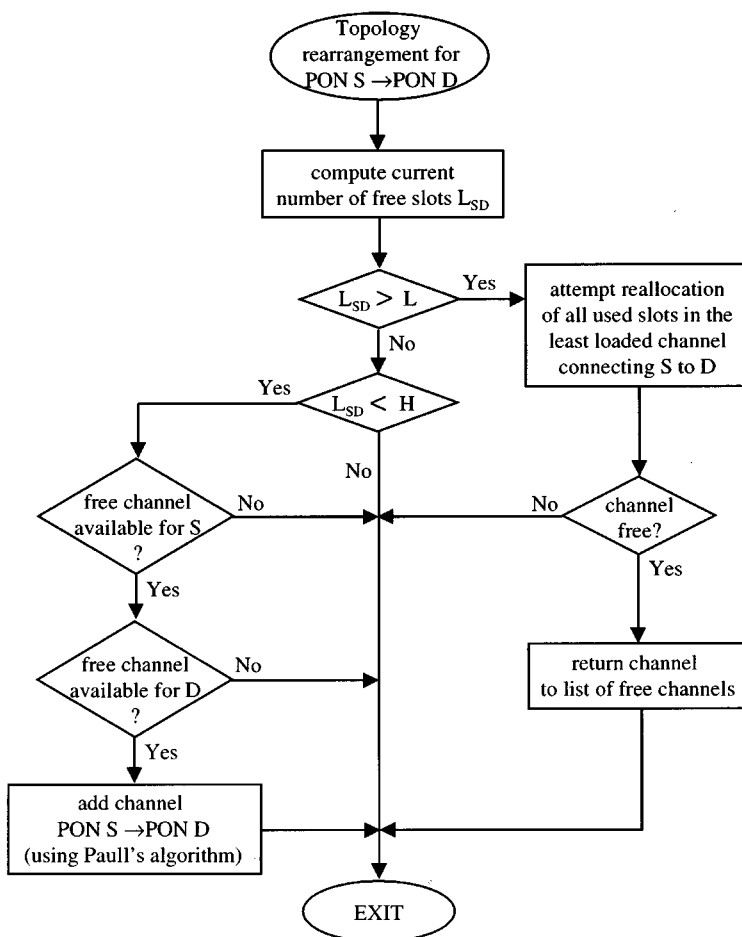


Fig. 4. Block diagram for the logical topology design algorithm.

(which are not reported here for the sake of conciseness), thus resulting in some computational complexity. Moreover, some bandwidth must be devoted to signal all the modifications to end terminals.

In the SONATA project, schemes where all existing connections are continuously reallocated in subsequent frames were proposed and studied. These schemes have much stronger requirements in terms of signalling bandwidth and of computational complexity at the NC. However, the results presented in [3, Ch. II, Sect. 3] show that signalling in this worst case can use as little as less than 2% of the available network capacity. Moreover, sophisticated parallel algorithms and architectures can be used at the NC controller to cope with the computational complexity of schedule computation.

The approach described in this section is much less demanding in terms of complexity at the NC and of signalling bandwidth, since the reconfiguration process is enabled only for one source/destination PON per frame.

Note that no service degradation is perceived by the users, since active connections are transparently moved in the time-wavelength frame at frame boundaries.

IV. MODELING THE NC RESOURCE ALLOCATION

We describe in this section two analytical models: the first refers to the scheduling problem and allows us to compute

the call blocking probability; the second describes the logical topology design algorithm, and provides performance indexes such as the distribution of the number of wavelength channels used between any pair of PONs, the probability that a new channel that should be allocated between two PONs is not available when requested, and the average channel holding time.

A. Modeling the Scheduling Algorithm

The model estimates the call blocking probability obtained with our scheduling algorithm, and allows us to compare the performance achieved with our scheme to the one obtained with an ideal scheduling algorithm.

We focus on a specific transmitter, a specific receiver, and a specific wavelength channel to describe the model. We assume that the transmitter, the receiver, and the wavelength channel occupancies can be modeled as independent random variables. This assumption is obviously closer to reality as the number of PONs and users increases. We further assume that the traffic originated by end terminals inside a PON can be modeled as Poisson traffic with average rate λ , and that the call duration is exponentially distributed with average $1/\mu$.

Under these hypotheses, neglecting the effect of blocking probability on the call arrival rate, we can model the evolution of the number of slots per frame in which the terminal is transmitting as a M/M/F/0 queue.

Thus, at the transmitter:

$$\pi_t(i) = \frac{\frac{\rho_t^i}{i!}}{\sum_{i=0}^F \frac{\rho_t^i}{i!}} \quad (8)$$

where $\pi_t(i)$ represents the probability that transmitter t is sending i slots per frame, and ρ_t is the transmitter load.

Similarly, for the receiver:

$$\pi_r(j) = \frac{\frac{\rho_r^j}{j!}}{\sum_{j=0}^F \frac{\rho_r^j}{j!}} \quad (9)$$

where $\pi_r(j)$ represents the probability that receiver r is receiving j slots per frame, and ρ_r is the receiver load.

Finally, considering the wavelength channel:

$$\pi_c(k) = \frac{\frac{\rho_c^k}{k!}}{\sum_{k=0}^F \frac{\rho_c^k}{k!}} \quad (10)$$

where $\pi_c(k)$ represents the probability that k slots per frame are transmitted on channel c , and ρ_c is the channel load.

The blocking probability given i , j , and k can be computed by an analogy: consider red, green, and yellow balls representing, respectively, busy slots at the transmitter, busy slots on the wavelength channel, and busy slots at the receiver. Assume to distribute the balls into cups, corresponding to slots in the frame, under the constraint of having no more than one ball of the same color in the same cup. The blocking probability can be easily computed as the probability of finding no empty cups, i.e., as the ratio between the number of ways in which i red balls, j green balls, and k yellow balls can be distributed in F cups such that no cup will remain empty, and the total number of ways in which the balls can be distributed in F cups.

A simple combinatorial computation gives:

$$P_b(i, j, k) = \begin{cases} \frac{\sum_{l=\max(0, j-i)}^{\min(F-i, j)} \binom{F}{i} \binom{F-i}{l} \binom{i}{j-l} \binom{i+l}{i+l+k-F}}{\binom{F}{i} \binom{F}{j} \binom{F}{k}} & i+k+k \geq F \\ 0 & \text{otherwise.} \end{cases} \quad (11)$$

As a consequence, the call blocking probability can be evaluated by averaging $P_b(i, j, k)$ over all possible states:

$$P_b = \sum_i \sum_j \sum_k P_b(i, j, k) \pi_t(i) \pi_r(j) \pi_c(k).$$

Relations (8)–(10) were derived assuming that the rate at which end terminals offer calls to the system is constant, i.e., it is independent from the system state. This is an approximation,

since the offered traffic rate should be dependent on the blocking probabilities (11). The model could be improved in order to take into account the effect of the blocking probability on the arrival rates and on the state distribution. In this case, we should describe the state evolution of the receiver, of the transmitter, and of the wavelength channel as continuous time Markov chains whose transition rates depend on the blocking probability value. A fixed point technique can be used to obtain the call blocking probability, as described in [9].

The perturbation due to the blocking probability can be considered as a second-order effect until the blocking probability remains small, and we decided to neglect it, given the huge number of terminals in the real network. In addition, by neglecting this effect we obtain an upper bound on the blocking probability of our scheme. We can therefore perform a conservative comparison with an ideal scheduling algorithm.

B. Modeling the Logical Topology Design

In this section we present a model based on a continuous time Markov chain (CTMC) to evaluate the performance of the SONATA logical topology design algorithm. The model allows us to compute the distribution of the number of wavelength channels used between any pair of PONs, along with the probability that a new channel that should be allocated between two PONs is not available when requested, and the average channel holding time.

We suppose that all calls request the same bandwidth, equal to one slot per frame. In the model, the duration of every call is exponentially distributed with average $1/\mu$. The aggregate call arrival process at each PON is assumed to be a Poisson process.

Let us consider a pair of PONs (source PON, destination PON); the CTMC state is defined by (s, w) , where s is the number of calls allocated between the PONs, and w is the number of channels allocated between the PONs.

Denote by $C = N_d + 1$ the maximum number of available channels among each pair of PONs. F is the length of the frame in slots, H is the channel load threshold above which a new channel between PONs is allocated, and L is the channel load threshold below which a channel between PONs is deallocated.

From state (s, w) , the following states are reachable.

- $(s+1, w)$: a new call request arrives and it is allocated; it must be $s < Fw$; the rate of the transition is λ .
- $(s-1, w)$: a call ends; it must be $s > 0$; the rate of the transition is $s\mu$.
- $(s, w+1)$: a new channel is allocated; it must be $s > wF - H$; the rate of the transition is ω_w^+ .
- $(s, w-1)$: a channel is deallocated; it must be $s < wF - L$; the rate of the transition is ω_w^- .

In order to simplify the model and to reduce the state space cardinality, we assume an ideal slot scheduling algorithm. (This assumption is close to reality as demonstrated by the results reported in Fig. 6, where the SONATA and the ideal scheduler are compared.) By ideal scheduler, we refer to a system which is always able to deallocate a channel between a pair of PONs if enough free slots are available on the remaining channels, and for which w is decreased by one as soon as the load reaches threshold L . Note that, in the real system, the deallocation of

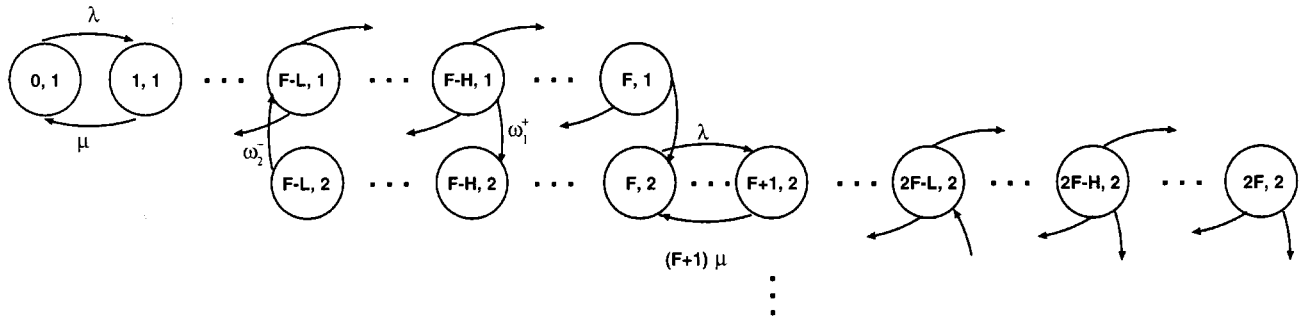


Fig. 5. State transition diagram of the CTMC.

channel w_{ij} can take place only provided that the scheduler is able to reallocate on other wavelength channels all calls active on channel w_{ij} .

The state transition diagram of the CTMC is sketched in Fig. 5.

In the CTMC, the channel allocation rates are unknown. They are obtained by applying the following fixed point technique.

- 1) New channel allocation rates are initialized to a very large value.
- 2) The CTMC models are solved for each pair of PONs.
- 3) The distribution of the number of channels used between each pair of PONs is evaluated.
- 4) The global distribution of the number of used channels is computed.
- 5) The values of the probabilities of new channel allocation are computed. These provide also a new set of channel allocation rates.
- 6) Steps 2–5 of the procedure are iterated until convergence.

When the load grows above threshold H , a new channel is allocated either immediately, if available, or as soon as a channel becomes available. The inverse of the channel allocation rate (i.e., the average time that should pass before the new channel can be allocated) is therefore obtained as the product of two terms: the average channel holding time, i.e., the average duration of the time between the allocation of a channel and its subsequent deallocation; and the probability that a new channel is not available when it is needed.

While the first quantity can be easily obtained by the solution of the CTMC, in order to obtain the second a further assumption is needed: we assume that the numbers of channels used between each pair of PONs are independent random variables. Let $N_{ij}(w)$ be the distribution of the number of channels between PON i and PON j . The number of wavelength channels used from i to all the destinations except j , $N_{i|j}(w)$, is thus the sum of independent random variables. Hence, $N_{i|j}(w)$ is the convolution of the distributions:

$$N_{i|j}(w) = N_{i,1}(w) * N_{i,2}(w) * \dots * N_{i,j-1}(w) \\ * N_{i,j+1}(w) * \dots * N_{i,N_p}(w)$$

where N_p is the number of PONs. The tail of the distribution is folded onto $N_{i|j}(C)$.

The probability that no new wavelength channel is available at PON i , given that k channels are already allocated for the

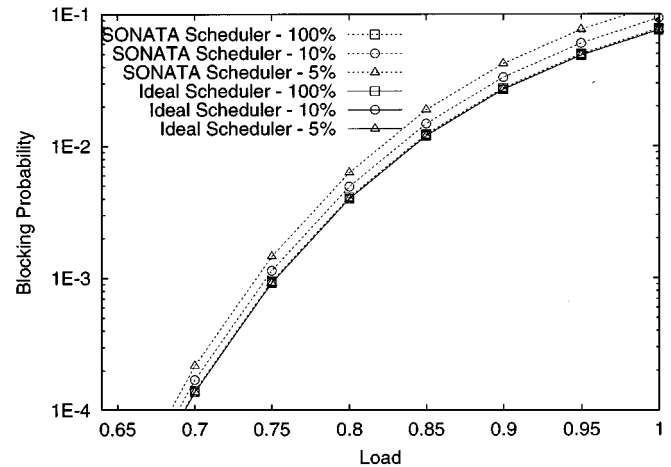


Fig. 6. Blocking probability for scheduling algorithms.

communication to PON j , is

$$P_i[\text{block}|k \text{ to } j] = \sum_{w=C-k}^C N_{i|j}(w).$$

The probability that no additional wavelength channel is available at PON j , given that k channels are already allocated for the communications from PON i , can be computed in the same way:

$$P_j[\text{block}|k \text{ from } i] = \sum_{w=C-k}^C N_{j|i}(w). \quad (12)$$

Using the distributions $N_{ij}(w)$ to uncondition (12) with respect to k , we can obtain the average probability $P_j[\text{block}]$, i.e., the probability that no additional wavelength channel is available at PON j .

Finally, the probability that a channel is not available when it is needed for establishing a connection from PON i to PON j is equal to

$$P_{ij}[\text{unavailability}|k] = P_i[\text{block}|k] + P_j[\text{block}|k] \\ - P_i[\text{block}|k]P_j[\text{block}|k]$$

V. PERFORMANCE RESULTS

In this section we present some performance results to assess the merits of our proposal. We first concentrate on the scheduling algorithm and compare our algorithm to an ideal algorithm

TABLE II
CHANNEL DISTRIBUTION P_n , CHANNEL UNAVAILABILITY PROBABILITY P_u , AND AVERAGE CHANNEL HOLDING TIME τ , UNDER UNIFORM TRAFFIC

n	$\lambda = 0.6$			$\lambda = 1.4$			$\lambda = 1.6$			$\lambda = 2.4$		
	1	2	3	1	2	3	1	2	3	1	2	3
P_n	0.99	0.01	$< 10^{-6}$	$2 \cdot 10^{-4}$	0.99	10^{-6}	$1 \cdot 10^{-3}$	0.99	10^{-3}	0.12	0.76	0.12
P_u	–	10^{-6}	10^{-6}	–	$1.8 \cdot 10^{-2}$	0.98	–	0.18	0.99	–	0.99	0.99
τ	∞	$3.4 \cdot 10^{-1}$	$8.6 \cdot 10^{-2}$	∞	$7.4 \cdot 10^5$	0.2	∞	$2.0 \cdot 10^4$	0.3	∞	$4.3 \cdot 10^{11}$	$3.2 \cdot 10^8$

by using the previously described analytical model. Then we concentrate on the analysis of the logical topology design algorithm. Finally, we examine a small-scale network by simulation, taking into account both the scheduling and the logical topology design algorithms.

A. Scheduling Algorithm

We examine the SONATA network under a uniform traffic distribution. We consider $N_t = 20\,000\,000$ end terminals, 400 PONs, 400 wavelength converter arrays, and $F = 1000$ slots. We compute the blocking probability of an ideal scheduler, i.e., a scheduler able to allocate a new request if there are enough time slots available at the source terminal, at the receiver terminal, and on the wavelength channel, independently of the position of these time slots in the frame, and of the allocation used for the already existing competing traffic. We compare the blocking probability of our scheduler (using the analytical model presented in Section IV-A) to the ideal scheduler, varying the PON offered load. We consider as a parameter the percentage of active terminals in a PON, i.e., the percentage of users that can generate a call request among the 50 000 end terminals.

The results are plotted in Fig. 6, where dashed lines refer to the real SONATA scheduler, and continuous lines refer to the ideal scheduler. The curves for the ideal scheduler are not distinguishable in the figure because they overlap among themselves, and also in part with the lowest curve of the real scheduler. Our SONATA scheduler performs very close to the ideal scheduler when all users (100%) are active. Only when the percentage of active users is reduced to 10% or 5% is it possible to see a slight performance degradation; indeed, having a smaller number of active users per PON decreases the multiplexing effect, thus the blocking probability increases due to the unavailability of source or destination slots.

These results, together with the fact that for the ideal scheduler no performance difference exists when reducing the percentage of active users, support the assumption that the blocking probability is mainly addressable to the unavailability of slots on wavelength channel, and not to the users occupation level.

Moreover, the analysis of the logical topology design algorithm discussed in Section IV-B, that assumes an ideal scheduler, is supported by the small differences shown in this section between the ideal and the real scheduler.

B. Topology Design Algorithm

We analyze the “steady-state” behavior of the algorithm by using the analytical model described in Section IV-B.

The performance indexes we obtain with the analytical model are the channel distribution among each pair of PONs, the probability that a channel is not available when required, and the average channel holding time.

A network of 100 PONs is considered with $N_d = 100$ dummy ports, and the frame length F is set equal to 100 slots. The average call duration has been set to 1 time unit, and each call requires 1 slot. The channel allocation threshold H has been set to 10 slots, while the channel deallocation threshold L has been set to 20 slots. Note that such a large-scale system can be hardly analyzed by simulation. Moreover, although in the real network the frame length is equal to 1000 slots, envisioning single connection-oriented calls of 1 slot is reasonable only for telephone calls. An allocation request of 10 slots would correspond to a bandwidth of about 5 Mbit/s, a reasonable value for video traffic; the ratio 10 slots over 1000 slots is the same ratio we use in this section, where connections require 1 slot and the frame length is set to 100 slots.

Table II reports the channel distribution P_n (probabilities of having n channels allocated to a pair of PONs), the channel unavailability probability P_u , and the average channel holding time τ obtained when the traffic is uniformly distributed among each PON pairs, for variable values of traffic load λ . If $\lambda = 0.6$, the programmable portion of the network is marginally exploited. Thus, the probability of channel unavailability is negligible, and the average channel holding time is very small for the extra channels.

When the PON-to-PON load is increased to $\lambda = 1.4$ or to $\lambda = 1.6$, the programmable portion of the network is fully exploited; the probability of unavailability of the third channel approaches 1, while its value is smaller when the second channel is required.

When the PON-to-PON load is further increased to $\lambda = 2.4$, so that more than two channels between each PONs pair are required on average to transfer all the traffic, the channel unavailability probability approaches 1, and the channel holding time becomes really large. In this case, the channel distribution shows a significant variance. This is mainly due to the fact that the topology design algorithm can very hardly converge to the optimal topology due to the high network traffic overload.

Table III refers to a scenario in which the amount of traffic originated in a PON and directed to receivers belonging to the same PON is larger than the amount directed to any other PON. Intra-PON traffic load is $\lambda_{in} = 6.0$ while inter-PON traffic load is $\lambda_{out} = 1.0$. Data reported refer only to intra-PON connections. In this case, the probability of channel unavailability

TABLE III
CHANNEL DISTRIBUTION P_n , CHANNEL UNAVAILABILITY PROBABILITY P_u ,
AND AVERAGE CHANNEL HOLDING TIME τ , UNDER NONUNIFORM TRAFFIC

n	5	6	7	8
P_n	$2 \cdot 10^{-4}$	0.83	0.17	$1 \cdot 10^{-6}$
P_u	$2.53 \cdot 10^{-5}$	$1.45 \cdot 10^{-4}$	$6.77 \cdot 10^{-4}$	$2.51 \cdot 10^{-3}$
τ	$2.87 \cdot 10^9$	$2.80 \cdot 10^2$	0.333	0.091

is negligible, since on average only few programmable channels are used to carry inter-PON traffic. The intra-PON channel distribution has an average value close to 6 as expected. The channel holding time of the first 5 channels is really large since they are rarely deallocated, while it decreases more than six orders of magnitude for channels 6, 7, and 8.

The model also provides a rough estimation of the call blocking probability, evaluated as the probability of finding the system in the state (kF, k) , where $k = 1, \dots, C$. This allows us to perform a first setting of threshold values. Although optimal threshold values are dependent on the traffic pattern, it can be shown that, in general, when thresholds H and L are decreased, the probability of channel unavailability slightly decreases, since we better exploit the network bandwidth. However, by decreasing H , the call blocking time experienced while waiting for the channel to become available also increases.

C. Network Controller Resource Allocation

We examine by simulation the algorithm in terms of its speed of convergence to the optimal logical topology, and its stability. By optimal topology, we define the topology that maximizes the network throughput in a given traffic scenario.

We concentrate on a smaller system, with 3 PONs, $N_t = 150$ end terminals, i.e., 50 users per PON, frame length $F = 100$, $N_d = 3$ dummy ports connected to a bank of 3 wavelength converter arrays, and 6 wavelengths per fiber. We study both network controller resource allocation algorithms, i.e., logical topology design and scheduling, under uniform traffic pattern. We discuss the speed of convergence of the algorithm to the optimal logical topology as a function of the system load. Note that in this case the optimal topology is obtained by allocating the same number (2) of channels to each PON pair.

At time 0, all the $n_p \times N_d = 9$ additional channels provided by the programmable portion of the network are allocated between PONs connected to input and output ports with the same index, i.e., 4 channels are available between source PON I and destination PON I and 1 (wired) channel is available between PON I and all other PONs. Thus, the topological distance¹ is 12, since each PON has 2 additional channels allocated to reach

¹We define as topological distance between two topologies the following quantity:

$$\sum_S \sum_D |C_{SD}^1 - C_{SD}^2|$$

where indexes S and D span over PONs, and where C_{SD}^1 and C_{SD}^2 represent the number of channels allocated from PON S to PON D in the first and second logical topology, respectively.

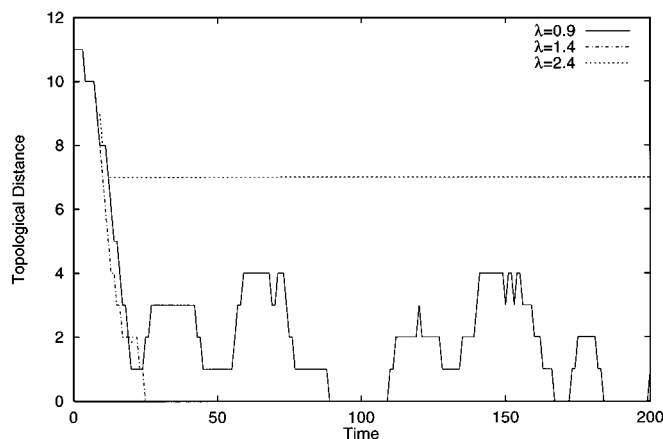


Fig. 7. Convergence of the current logical topology to the reference topology.

its end terminals, and one channel missing toward the other two paths. Note that this is a logical topology that maximizes the topological distance from the optimal topology under uniform traffic.

We show in Fig. 7 the topological distance of the current topology from the optimal topology as a function of time (one unit refer to a time frame), for variable loads. For medium-high loads ($\lambda = 0.9$), the system converges rapidly to the optimal topology; once the optimal topology is reached, due to traffic fluctuations, the algorithm sometimes modify the logical topology, but the topological distance from the optimum remains small. Results for smaller loads are not reported for the sake of conciseness and show a very similar behavior.

For moderate overload ($\lambda = 1.4$), the optimal topology is reached quite quickly; once all the channels are correctly allocated to pair of PONs, the system does not modify the topology since no channels become available at any time.

For sustained overload ($\lambda = 2.4$), the optimal topology is never reached; similarly to the previous scenario, once all the channels are allocated to pair of PONs, the system is not able to modify this topology since no channel becomes available at any time. However, the system does not converge in general to the optimal topology but to a topology that depends on the initial stochastic behavior of call requests.

Note the good agreement between the results presented in this section and those reported in Section V-B.

VI. CONCLUSION

The paper described the network architecture and provided a performance analysis of a passive optical network named SONATA, which has been proposed and demonstrated in the context of the European Union ACTS Program. SONATA aims at avoiding the need for large and fast switching electronic nodes in a high-speed nationwide network. To reach this goal, the network structure and the layer architecture within the network have been drastically simplified: the transport infrastructure consists of a single-layer of end-to-end optical connections. End terminals fully exploit time and wavelength agility to exchange packetized information in the multiple-access network.

The control of the network is centralized at a network control device, whose primary goal is to assign time/wavelength resources to terminals in a way such that conflicts among transmitters and receivers are avoided. The paper focused on the algorithms that must be executed at the network controller to solve the resource allocation problem.

We analyzed and formally defined the resource allocation problem at the network controller as an ILP problem, showed that it is in general NP-hard, and provided simple heuristic algorithms that divide the solution into scheduling and logical topology design subproblems, aiming at a suboptimal approach with a limited complexity.

The performance evaluation of the considered algorithms was mainly based on analytical models, although simulation was also considered to assess the transient behavior of our proposals. In particular, we provided analytical models both for the slot scheduling algorithm, and for the logical topology design algorithm. These models proved to be very effective in predicting the general behavior of the algorithms, and to evaluate the impact of design tradeoffs, in a context where the network complexity makes simulation impractical for nontrivial system dimensions. We observed, for example, that for a large number of terminals, the impact of transmitter and receiver contentions is negligible with respect to the blocking due to the limited number of transmission resources.

The design effort reported in this paper shows that, although the considered network is of daunting dimension, the architectural simplifications proposed by the SONATA project lead to the possibility of implementing network control procedures of acceptable complexity, whose behavior is very close to that of ideal control strategies, which would require an unbearable complexity to implement. As such, SONATA can be considered as a meaningful step toward the provision (through fast circuit switching) of quality of service in packet networks of geographical span.

REFERENCES

- [1] A. M. Hill, N. P. Caponio, F. Neri, and R. Sabella, "Single layer optical platform based on WDM/TDM multiple access for large scale 'switchless' networks," *European Trans. Telecommun., Special Issue on WDM Networks*, vol. 11, no. 1, pp. 73–82, Jan./Feb. 2000.
- [2] A. Bianco, E. Leonardi, M. Mellia, M. Motisi, and F. Neri, "Specification of signalling functions in SONATA," ACTS Project SONATA AC-351, Working Document, Politecnico di Torino, also available in [3, Ch.II, Sect. 2], 1999.
- [3] SONATA (Switchless Optical Network for Advanced Transport Architecture), "Second report on system studies," Deliverable D9, 1999.
- [4] J. Hui, *Switching and Traffic Theory for Integrated Broadband Networks*. Norwell, MA: Kluwer Academic, 1990.
- [5] K. Eng and A. Acampora, "Fundamental conditions governing TDM switching assignments in terrestrial and satellite networks," *IEEE Trans. Commun.*, vol. TOC-37, pp. 187–189, Feb. 1987.
- [6] A. Varma and S. Chalasani, "An incremental algorithm for TDM switching assignments in satellite and terrestrial networks," *IEEE J. Select. Areas Commun.*, vol. JSAC-10, pp. 364–377, Feb. 1992.
- [7] C. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*. Englewood Cliffs, NJ: Prentice-Hall, 1982.
- [8] G. Nemhauser, A. Rinnooy Kan, and M. Todd, *Optimization*. Amsterdam, The Netherlands: North-Holland, 1989, vol. 1.
- [9] A. Bianco, G. Galante, E. Leonardi, and M. Mellia, "Analysis of call blocking probability in TDM/WDM networks with transparency constraint," *IEEE Commun. Lett.*, vol. 4, Mar. 2000.



Andrea Bianco (M'99) was born in Torino, Italy, on January 31, 1962. He received the Dr.Ing. degree in electronics engineering in 1986 and the Ph.D. degree in telecommunications engineering in 1993, both from Politecnico di Torino.

He has been an Assistant Professor at the Dipartimento di Elettronica of Politecnico di Torino since December 1994. In 1993 he visited the Hewlett-Packard Laboratory, Palo Alto, CA, for one year. In Summer 1998 he visited the Electronics Department of Stanford University. His current research interests are in the fields of algorithms and architectures for all-optical networks, and performance analysis of input-buffered packet switches. He has coauthored about 50 papers published in international journals and presented in leading international conferences.



Emilio Leonardi (M'99) received the Dr.Ing. degree in electronics engineering in 1991 and the Ph.D. degree in telecommunications engineering in 1995, both from Politecnico di Torino.

He is currently an Assistant Professor in the Electronics Department of Politecnico di Torino. In 1995, he spent one year at the Computer Science Department of the University of California, Los Angeles (UCLA), where he was involved in the Supercomputer-SuperNet (SSN) project aimed at the design of an high capacity optical network architecture. During Summer 1999, he joined the "High-Speed Research Group" of Lucent Technologies, where he worked on the design of scheduling algorithms for high capacity switch architectures. His research interests are in the fields of all-optical networks, switching architectures, queueing theory, and wireless communications.



Marco Mellia (S'97) was born in Torino, Italy, in 1971. He received the degree in electronic engineering from the Politecnico di Torino in February 1997.

Between February and October 1997, he was a Researcher supported by CSELT, the Italian Public Telephone Research Company, developing a call admission control algorithm for ATM networks and computer tools for simulation and performance evaluation. Since November 1997, he has been with the Electronics Department of Politecnico di Torino, Turin, Italy, as a Ph.D. student. From March to October 1999 he was with the CS department at Carnegie Mellon University as Visiting Scholar. His research interests are in the fields of all-optical networks, switching architectures, and QoS routing algorithms.



Fabio Neri was born in Novara, Italy, in 1958. He received the Dr.Ing. and Ph.D. degrees in electrical engineering from Politecnico di Torino in 1981 and 1987, respectively.

He is an Associate Professor at the Electronics Department of Politecnico di Torino, Turin, Italy. From 1991 to 1992 he was with the Information Engineering Department at University of Parma, Parma, Italy, as an Associate Professor. From 1982 to 1983 he was a Visiting Scholar at George Washington University, Washington, DC. In Summer 1995 he was Visiting Researcher at the Computer Science Department, the University of California at Los Angeles (UCLA). In Summer 1998 he visited Bell Laboratories/Lucent Technologies in Holmdel, NJ. His research interests are in the fields of performance evaluation of communication networks, high-speed and all-optical networks, packet switching architectures, discrete event simulation, and queueing theory. He has coauthored over 100 papers published in international journals and presented in leading international conferences.

Dr. Neri is a member of the IEEE Communications Society. He has served IEEE conferences, including Globecom and Infocom, and was the Technical Program Co-Chair of 1999 IEEE Workshop on Local and Metropolitan Area Networks.