Packet Scheduling in Broadcast WDM Networks with Arbitrary Transceiver Tuning Latencies

George N. Rouskas, *Member, IEEE*, and Vijay Sivaraman

***Abstract*—We consider the problem of scheduling packet transmissions in a broadcast, single-hop wavelength-division multiplexing (WDM) network, with tunability provided only at one end. Our objective is to design schedules of minimum length to satisfy a set of traffic requirements given in the form of a demand matrix. We address a fairly general version of the problem as we allow arbitrary traffic demands and arbitrary transmitter tuning latencies. The contribution of our work is twofold. First we define a special class of schedules which permit an intuitive formulation of the scheduling problem. Based on this formulation we present algorithms which construct schedules of length equal to the lower bound provided that the traffic requirements satisfy certain optimality conditions. We also develop heuristics which, in the general case, give schedules of length equal or very close to the lower bound. Secondly, we identify two distinct regions of network operation. The first region is such that the schedule length is determined by the tuning requirements of transmitters; when the network operates within the second region however, the length of the schedule is determined by the traffic demands, not the tuning latency. The point at which the network switches between the two regions is identified in terms of system parameters such as the number of nodes and channels and the tuning latency. Accordingly, we show that it is possible to appropriately dimension the network to minimize the effects of even large values of the tuning latency.**

***Index Terms*— Optical networks, packet scheduling, tuning latency, wavelength-division multiplexing.**

I. Introduction

WAVELENGTH-division multiplexing (WDM) is considered as the most promising approach to fully exploiting the vast information-carrying capacity of single-mode fiber. Our focus in this paper is on a WDM network architecture known as the *single-hop* architecture [14]. Single-hop networks are especially attractive as they are *all-optical* in nature. In other words, any information transmitted into the medium remains in the optical form until it reaches its destination. Critical to the design of single-hop networks is the availability of *tunable* lasers and/or optical filters, devices with the ability to tune across all available channels. Such devices do exist today; however, their capabilities are limited in terms of both tunability range and speed, and the ideal device, one that can tune across the useful optical spectrum in sub-microsecond times [11] remains elusive. We show, however, that careful network design can mask the effects of nonideal devices, making it possible to build single-hop WDM networks using *currently available* tunable optical transceivers.

Manuscript received November 1, 1995; revised April 2, 1997; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor R. Ramaswami. This work was supported in part by a grant from the Center for Advanced Computing and Communication, North Carolina State University. This paper was presented in part at the IEEE INFOCOM '96 conference, San Francisco, CA, March 26-28, 1996.

G. N. Rouskas is with the Department of Computer Science, North Carolina State University, Raleigh, NC 27695-8206 USA.

V. Sivaraman was with the Department of Computer Science, North Carolina State University, Raleigh, NC 27695-8206 USA. He is now with the Department of Computer Science, University of California at Los Angeles, Los Angeles, CA 90095 USA.

Publisher Item Identifier S 1063-6692(97)04911-X.

In a typical WDM environment, one with a large number of users and a limited number of wavelengths, it seems possible to mask the tuning times by letting a transceiver that was previously idle take the place of one that is taken off-line for retuning. Our work provides insight into this potentially difficult problem of coordinating the packet transmissions and retunings among the various nodes in the network. We provide novel heuristics and optimal (under certain conditions) scheduling algorithms, and we show that, if the network operates within a certain region, even large values of the tuning latency have no significant effect on the length of the schedule (which provides a measure of the delay and throughput performance).

Section II contains some background information, and Section III describes the system and traffic model. In Section IV we formulate the scheduling problem, and we derive lower bounds on the schedule length. We introduce a special class of schedules in Section V, and we develop scheduling algorithms which, under certain conditions, construct optimal schedules within this class. Scheduling heuristics are developed in Section VI, and in Section VII we present some numerical results. We conclude the paper in Section VIII.

II. Background

Let δ denote the *normalized tuning latency*, expressed in units of packet transmission time. The value of δ depends on the data rate, the packet size, and the transceiver tuning time, and can be less than, equal to, or greater than one. Underlying the design of a broad class of architectures is the assumption that $\delta \ll 1$, i.e., that transceiver tuning times are negligible compared to the duration of a packet transmission. Accordingly, a *padding* equal to δ time units can be included within each slot to allow the transceivers sufficient time to switch between wavelengths, with minimal effects on the overall performance. This is reflected in the design of network architectures and protocols for such environments [3], [12], [16], [17] which has been geared toward improving the delay and throughput characteristics of the network under various traffic assumptions, completely ignoring transceiver tuning times.

Emerging communication environments, however, are such that the tuning times of even the fastest available devices dominate over packet transmission times, making δ comparable to, and even greater than, one. Including a padding equal to δ within each slot would be highly inefficient in this case; instead, it is highly desirable to have the slot time equal to the packet transmission time alone. Let us now define parameter $\Delta = \lceil \delta \rceil$. In a slotted system with a slot time equal to the packet transmission time, a transceiver instructed to switch to a new channel will be unavailable for a number of slots equal to Δ.

Minimizing the effects of transceiver tuning times on network performance is possible only through specially designed protocols. In [8], for instance, the time-division multiple access (TDMA) scheme considered is such that the frame is divided into transmitting and tuning periods in a fashion reminiscent of the time slot assignment in TDM switching systems [5], [10], [13]. Each transceiver operates on a fixed channel during a transmitting period; no transmissions take place during the tuning periods, which are reserved to retune transceivers. The objective is to minimize the number of tuning periods within the frame.

When the number N of stations is greater than the number C of wavelengths, at most C stations may be transmitting at any given slot. Other stations may use that slot for retuning to a new channel, so that they will be ready to access that channel at a later slot. Thus, transceiver tuning times may be overlapped with transmissions by other stations. The objective, then, is to design schedules of minimum length, given a traffic demand matrix. Various versions of this problem have been formulated and studied previously in [1], [2], [6], [15]. In [1], [6], [15] uniform traffic demands are considered, and lower and upper bounds on the length of an optimal schedule are derived. The work in [2] considers a traffic demand matrix of 1's and 0's (representing the existence or not, respectively, of a head-of-line packet at the various queues), and values of $\delta \leq 1$. Our work is a generalization of the work in [1], [2], [6], [15], as it considers arbitrary traffic demands and arbitrary values of δ. Furthermore, our conclusions extend and provide further support to the results of earlier work.

The problem of scheduling nonuniform traffic with arbitrary tuning latencies has been previously studied in [4]. In contrast to [4] where one heuristic is used throughout, we make the fundamental observation that, depending on the traffic matrix and the system parameters, the network can be operating in one of two distinct regions. As a result, we develop different optimal algorithms and heuristics for each region.

III. System Model

We consider packet transmissions in an all-optical, single-hop WDM network with a passive star physical topology. Each of the N nodes in the network employs one transmitter and one receiver. The passive star supports C wavelengths, or channels; in general, $C \leq N$. We consider tunable-transmitter, fixed-receiver networks, but all our results can be easily adapted to fixed-transmitter, tunable-receiver systems. Each tunable transmitter can be tuned to any and all wavelengths $\lambda_c, c = 1, \cdots, C$. The fixed receiver at station j is assigned one of the C wavelengths, and we let $\mathcal{R}_c$ denote the set of receivers sharing wavelength λ_c.

Under the packet transmission scenario we are considering, there is an $N \times N$ traffic demand matrix $\mathbf{D} = [d_{ij}]$, with d_{ij} representing the number of slots to be allocated for transmissions from source i to destination j. Since a transmission on wavelength λ_c is heard by all receivers listening on λ_c, given a partition of the receiver set into sets $\mathcal{R}_c$, we obtain the *collapsed* [2] $N \times C$ traffic matrix $\mathbf{A} = [a_{ic}]$. Element a_{ic} of the collapsed matrix represents the number of slots to be assigned to source i for transmissions on channel λ_c:

$$a_{ic} = \sum_{j \in \mathcal{R}_c} d_{ij}, \qquad i = 1, \cdots, N, \quad c = 1, \cdots, C. \quad (1)$$

In this paper, we assume that $a_{ic} > 0 \forall i, c$, that is, each source i has to be allocated at least one slot on each channel. This assumption is reasonable when the number of nodes, N, is significantly greater than the number of available channels (a likely scenario in WDM environments), as each channel will be shared by many receivers. We also let D denote the total traffic demand $D = \sum_{i,j} d_{ij}$.

There are several situations in which such a scenario arises. Under a gated service discipline, d_{ij} may represent the number of packets with destination j in the queue of station i at the moment the "gate" is closed. A reservation protocol that can be used to communicate this queue-length information is described in [19]. Alternatively, d_{ij} may represent the number of slots to be allocated to the (i, j) source–destination pair to meet certain quality of service criteria; then, d_{ij} may be derived based on assumptions regarding the arrival process at the source. All our results are applicable to both scenarios, as they depend only on the matrices $\mathbf{D}$ and $\mathbf{A}$, not on how the elements of these matrices were obtained.

A. Transmission Schedules

A simultaneous transmission by two stations on the same channel results in a *collision*. To avoid packet loss due to collisions, some form of coordination among transmitters is necessary. A *transmission schedule* is an assignment of slots to source-channel pairs that provides this coordination: if slot τ is assigned to pair (i, λ_c), then in slot τ, source i may transmit a packet to any of the receivers listening on λ_c. Exactly a_{ic} slots must be assigned to the source-channel pair (i, λ_c), as specified by the collapsed matrix $\mathbf{A}$. This assignment is complicated by the fact that transmitters need time to tune from one wavelength to another.

If the a_{ic} slots are contiguously allocated for all pairs (i, λ_c), the schedule is said to be *nonpreemptive*; otherwise we have a *preemptive* schedule. Under a nonpreemptive schedule, each transmitter will tune to each channel exactly once, minimizing the overall time spent for tuning. Since our objective is to minimize the time needed to satisfy the traffic demands specified by the collapsed traffic matrix $\mathbf{A}$, we only consider nonpreemptive schedules. A nonpreemptive schedule is defined as a set $\mathcal{S} = \{\tau_{ic}\}$, with τ_{ic} the first of a block of a_{ic} contiguous slots assigned to the source-channel pair (i, λ_c). Since each source

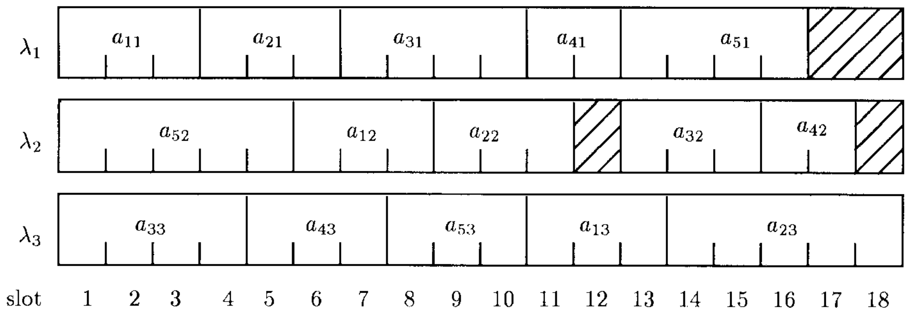

Fig. 1. Optimum length schedule for a network with $N = 5$, $C = 3$, and $\Delta = 2$.

has exactly one laser which needs Δ slots to tune between channels, all time intervals $[\tau_{ic} - 1, \tau_{ic} + a_{ic} + \Delta - 1)$ must be disjoint,[1] yielding a set of *hardware constraints* on schedule $\mathcal{S}$:

$$[\tau_{ic} - 1, \tau_{ic} + a_{ic} + \Delta - 1) \cap [\tau_{ic'} - 1, \tau_{ic'} + a_{ic'} + \Delta - 1) \\ = \phi \qquad \forall\, c \neq c'; \quad i = 1, \cdots, N. \tag{2}$$

In addition, to avoid collisions, at most one transmitter should be allowed to transmit on a given channel in any given slot, resulting in a set of *no-collision constraints*:

$$[\tau_{ic} - 1, \tau_{ic} + a_{ic} - 1) \cap [\tau_{i'c} - 1, \tau_{i'c} + a_{i'c} - 1) \\ = \phi \qquad \forall\, i \neq i'; \quad c = 1, \cdots, C. \tag{3}$$

A nonpreemptive schedule $\mathcal{S}$ is *admissible* if and only if $\mathcal{S}$ satisfies both the hardware and the no-collision constraints. Unless otherwise specified, from now on the term "schedule" will be used as an abbreviation for "admissible nonpreemptive schedule."

The *length*, M, of a schedule $\mathcal{S}$ for the collapsed traffic matrix **A** is the number of slots required to satisfy all traffic demands a_{ic} under $\mathcal{S}$. Fig. 1 shows an optimum length schedule for a network with $N = 5$ nodes, $C = 3$ channels, and $\Delta = 2$; matrix **A** can be easily deduced from the figure. Note that all hardware and no-collision constraints are satisfied. In particular, the first slot assigned to station 3 on channel λ_2 is slot 13, rather than slot 12, as its laser needs two slots to tune from λ_1 to λ_2.

In the following, we make the assumption that the schedule repeats over time; in other words, if τ_{ic} is the start slot of transmitter i on channel λ_c under schedule $\mathcal{S}$ of length M, then so are slots $\tau_{ic} + kM, k = 1, 2, 3, \cdots$, where k denotes the kth identical copy of the schedule. If parameters d_{ij} are derived based on the behavior and required quality of service of longer term connections, we expect the schedule to repeat until there is a change in the traffic demands. Under a gated service discipline scenario, however, a new schedule has to be computed after all transmissions under the current schedule have been completed. We now argue that the schedules we derive are applicable even under the latter scenario.

If the schedule is used only once, no transmissions are possible during an initial period of Δ slots when transmitters tune to their initial channels. On the other hand, if the schedule repeats over time, it may be possible to overlap this tuning period with transmissions in the *previous* frame, possibly resulting in a smaller schedule length.[2] Referring to Fig. 1, we see that in a repeating schedule, the tuning of transmitter 5 from λ_1 to λ_2 (to start the next frame) is overlapped with the packets sent by transmitter 2 on λ_3 during the current frame. In any case, the length of a schedule derived under the assumption that transmissions repeat over time will be at most Δ slots smaller than if this assumption is not made. We can then use the schedules derived here in situations where a schedule is used only once, after adding an initial period of at most Δ slots. Even though our assumption does affect the schedule length somewhat, it does not affect our conclusions about the network's regions of operation, to be discussed shortly.

IV. Schedule Optimization and Lower Bounds

Our objective is to determine an optimum length schedule for matrix **D**, as such a schedule would both minimize the delay and maximize throughput. This problem, which we will call the *Packet Scheduling with Tuning Latencies (PSTL)* problem, can be stated as follows.

Problem 1 [PSTL]: Given the number N of nodes, the number C of channels, the traffic matrix $\mathbf{D} = [d_{ij}]$, and the tuning slots Δ, find a schedule of minimum length for **D**.

Problem *PSTL* can be logically decomposed into two subproblems: a) sets of receivers, $\mathcal{R}_c$, sharing wavelength $\lambda_c, c = 1, \cdots, C$, must be obtained, and from them the collapsed traffic matrix, **A**$= [a_{ic}]$, must be constructed and b) for all i and c, a way of placing the a_{ic} slots to minimize the length of the schedule must be determined. The first subproblem can be solved by using an approximation algorithm [7] to assign receive wavelengths so that the traffic load, given by matrix **D**, is balanced across the C channels. Load balancing is a well-known $\mathcal{NP}$-complete problem, and will not be considered further. Let us now turn our attention to the second subproblem; for reasons that will become apparent shortly, we will refer to this as the *Open-Shop Scheduling with Tuning Latencies (OSTL)* problem. It can be expressed formally as a decision problem.

Problem 2 [OSTL]: Given the number N of nodes, the number C of channels, the matrix $\mathbf{A} = [a_{ic}]$, the tuning slots $\Delta \geq 0$, and a deadline $M > 0$, is there a schedule that meets the deadline?

[1] We make the assumption that slot τ starts at time $\tau - 1$ and occupies the time interval $[\tau - 1, \tau)$.

[2] Actually, a tuning period of Δ slots is still needed the very first time the schedule is used, but it can be ignored, especially if the schedule repeats for a relatively large number of times.

OSTL is a generalization of the nonpreemptive open-shop scheduling (*OS*) problem in [9]; it reduces to the latter when we let $\Delta = 0$. Problem OS is $\mathcal{NP}$-complete when the number of wavelengths $C \geq 3$ [9]. But for $C = 2$, *OS* admits a polynomial-time solution [9]. The following theorem confirms our intuition that *OSTL* is in a sense more difficult than *OS*. The proof of the theorem is omitted, but can be found in [18].

Theorem 1: OSTL is $\mathcal{NP}$-complete for any fixed $C \geq 2$.

A. Lower Bounds for PSTL and OSTL

The length of any schedule cannot be smaller than the number of slots required to satisfy all transmissions on any given channel, yielding the *bandwidth bound* (see also [2], [4], [9])

$$M_{bw}^{(l)} = \max_{1 \leq c \leq C} \left\{ \sum_{i=1}^{N} a_{ic} \right\} \geq \frac{D}{C}. \tag{4}$$

The term in the brackets depends on the assignment of receive wavelengths to the nodes, but the rightmost term depends only on the *total* traffic demand, D, and is a lower bound on *PSTL* independently of the elements d_{ij} of matrix **D**. Expression (4) implies that, given the number of wavelengths (which determines the amount of bandwidth available), the bandwidth bound is minimized when the traffic load is perfectly balanced across the C channels.

We obtain a different bound by adopting a transmitter's point of view. Each transmitter i needs a number of slots equal to the number of packets it has to transmit plus the number of slots required to tune to each of C channels. We call this the *tuning bound* [2], [4], [9]:

$$\begin{aligned} M_t^{(l)} &= \max_{1 \leq i \leq N} \left\{ \sum_{c=1}^{C} a_{ic} \right\} + C\Delta \\ &= \max_{1 \leq i \leq N} \left\{ \sum_{j=1}^{N} d_{ij} \right\} + C\Delta \geq \frac{D}{N} + C\Delta. \end{aligned} \tag{5}$$

The tuning bound is independent of the assignment of wavelengths to receivers, and only depends on parameters N, C, Δ, and the total demand D; it is minimized when each source contributes equally to the total traffic demand. We now obtain the overall lower bound as $M^{(l)} = \max\{M_{bw}^{(l)}, M_t^{(l)}\}$. The latter is minimized when

$$\frac{D}{C} = \frac{D}{N} + C\Delta \Leftrightarrow \frac{D}{C} = \frac{NC\Delta}{N - C}. \tag{6}$$

Quantity $\frac{NC\Delta}{N-C}$, which we will call the *critical length*, is independent of the demand matrix, and characterizes the network under consideration. Relationship (6) between the minimum bandwidth bound, D/C, and the critical length is a fundamental one, and represents the point at which wavelength concurrency balances the tuning latency. If a schedule has length equal to the critical length, it is such that exactly C (respectively, $N - C$) nodes are in the transmitting (respectively, tuning) state within each slot. Consequently, all $NC\Delta$ tuning slots are overlapped with packet transmissions, and vice versa. Such a schedule is highly desirable, as it has three important properties: a) it completely masks the tuning latency, b) it is the shortest schedule for transmitting a total demand of D packets, and c) it achieves 100% utilization of the available bandwidth, as no channel is ever idle. The significance of the actual schedule length relative to the critical length is explored next.

B. Bandwidth Limited Versus Tuning Limited Networks

To get further insight on (6), let us consider the case of uniform traffic, whereby each source has $\beta \geq 1$ packets for each possible destination: $d_{ij} = \beta \geq 1 \; \forall i, j$. Then, $D = \beta N^2$, and substituting this value into (6) we get

$$\frac{\beta N^2}{C} = \beta N + C\Delta \Leftrightarrow \frac{\beta N^2}{C} = \frac{NC\Delta}{N - C}. \tag{7}$$

In [2] and [15], (7) was solved (with $\beta = 1$) to obtain the value of C that minimizes the lower bound for all-to-all schedules. Typically, however, $C, N,$ and Δ are given parameters. We can then solve (7) to obtain the optimal value for β, $\beta^\star = \frac{C^2}{N(N-C)}\Delta$.

Suppose now that we choose $\beta < \beta^\star$; for simplicity, also let $N = kC$, so that the traffic demand can be perfectly balanced across the channels. Then, the tuning bound $\beta N + C\Delta$ becomes greater than the bandwidth bound $\frac{\beta N^2}{C}$, and the length of the schedule is determined by the transmitter tuning requirements. Since the total traffic demand is βN^2 and $\beta < \beta^\star$, the throughput achievable under such a schedule is $\frac{\beta N^2}{\beta N + C\Delta} < C$. The throughput increases with β; once β becomes greater than $\beta^\star$, the bandwidth bound becomes dominant and the throughput becomes equal to its maximum value, C.

Increasing the value of β, however, has the effect of increasing the length of the schedule. But this length is a measure of packet delay, and it cannot be increased beyond a certain level perceived as acceptable by the various higher layer applications. The demand matrix corresponding to the value $\beta = \lceil \beta^\star \rceil$ achieves a perfect balance between delay and throughput, as it provides for the smallest schedule length that results in a 100% channel utilization. Satisfying the delay requirements, however, might mean choosing $\beta < \lceil \beta^\star \rceil$. It is in these situations that advances in optical device technology would make a difference. Note that the value of $\beta^\star$, and consequently, the value of the critical length, is proportional to Δ. Employing faster tunable transceivers would then bring $\beta^\star$ closer to the acceptable (in terms of delay) operating value of β, and improve the throughput.

The above observations are of general nature, applying to nonuniform demand matrices as well. Similar to [2], we will say that a network is

- *tuning limited*, if the tuning bound dominates, i.e., $M^{(l)} = M_t^{(l)} > M_{bw}^{(l)}$, or
- *bandwidth limited*, if the bandwidth bound is dominant; then, $M^{(l)} = M_{bw}^{(l)} > M_t^{(l)}$.

To see why this distinction is important, note that any near-optimal scheduling algorithm, including the ones to be presented shortly, will construct schedules of length very close to the lower bound. If the network is tuning limited, the length

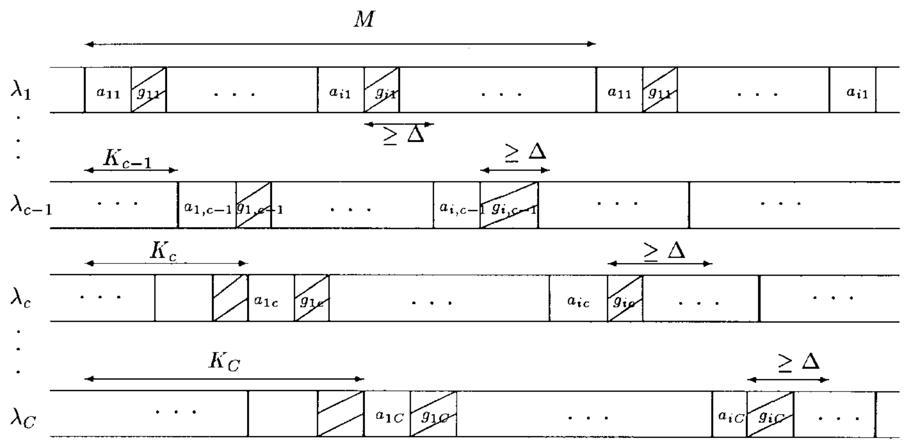

Fig. 2. Schedule for a bandwidth limited network.

of the schedule is determined by the tuning bound in (5), which in turn is directly affected by the tuning latency. The schedule length of a bandwidth limited network, on the other hand, depends only on the traffic requirements of the dominant channel, i.e., the channel λ_c such that $\sum_{i=1}^{N} a_{ic} = M_{bw}^{(l)}$.

Based on this discussion, it is desirable to operate the network at the bandwidth limited region, as doing so would eliminate the effects of tuning latency. For uniform traffic this can be accomplished by selecting $\beta = \lceil \beta^\star \rceil$ since that would make the bandwidth bound greater than the critical length in (7). In the general case (nonuniform matrix **D**) we would like to make the bandwidth bound in (6) greater than the critical length, $\frac{D}{C} > \frac{NC\Delta}{N-C}$. Given a value for Δ, and some information about the delay requirements of higher layer applications, this expression may be satisfied by carefully dimensioning the network (i.e., initially choosing appropriate values for N and C) so that it operates in the bandwidth limited region. Since, however, delay constraints and/or constraints on the values of N and C may make it impossible to satisfy this expression for a given system, we have developed scheduling algorithms and heuristics for both regions of network operation.

V. A Class of Schedules for *OSTL*

Let **A** be a collapsed traffic matrix, and $\mathcal{S}$ a schedule of length M satisfying the hardware and no-collision constraints (2) and (3), respectively. Consider now the order in which the various transmitters are assigned slots within, say, channel λ_1, starting with some transmitter π_1. We will say that $s_1 = (\pi_1, \pi_2, \cdots, \pi_N)$ is the *transmitter sequence* on channel λ_1 if π_2 is the first node after π_1 to transmit on λ_1, π_3 is the second such node, and so on. Since we have assumed that schedule $\mathcal{S}$ repeats over time, after node π_N has transmitted its packets on λ_1, the sequence of transmissions implied by s_1 starts anew. Similarly, we will say that $v_1 = (\lambda_{\pi_1}, \lambda_{\pi_2}, \cdots, \lambda_{\pi_C})$ is the *channel sequence* for node 1, if this is the order in which node 1 is assigned to transmit on the various channels, starting with channel λ_{π_1}. Given $\mathcal{S}$, the transmitter sequences with π_1 as the first node, are completely specified for all channels λ_c. In general, these sequences can be different for the various channels. However, in what follows we concentrate on a class of schedules such that the transmitter sequences (with π_1 as the first node) are the same for all channels:

$$s_c = (\pi_1, \pi_2, \cdots, \pi_N), \qquad c = 1, \cdots, C. \tag{8}$$

The class of schedules defined in (8) is equivalent to the class of schedules such that the channel sequences (with λ_{π_1} as the first channel) are the same for all nodes:

$$v_i = (\lambda_{\pi_1}, \lambda_{\pi_2}, \cdots, \lambda_{\pi_C}), \qquad i = 1, \cdots, N. \tag{9}$$

This class of schedules greatly simplifies the analysis, allowing us to formulate the *OSTL* problem in a way that provides insight into the properties of good scheduling algorithms.

We now proceed to derive sufficient conditions for optimality and optimal algorithms for the class of schedules defined in (8) and (9). Since we have found [18] that bandwidth limited and tuning limited networks are dual of each other, in this paper we only present results for bandwidth limited networks. Very similar conditions for optimality and algorithms have been derived for tuning limited networks, and can be found in [18].

A. Bandwidth Limited Networks

We present a formulation of problem *OSTL* applicable to bandwidth limited schedules within the class (8). Let $\mathcal{S}$ be a schedule of length M for such a network, and let $(1, 2, \cdots, N)$ be the transmitter sequence on all channels. For each channel, consider the frame which begins with the first slot assigned to transmitter 1. Let the start of the frame on channel λ_1 be our reference point, and let K_c denote the distance between the start of a frame on channel λ_c and the start of the frame on the first channel, as in Fig. 2. Obviously, $K_1 = 0$. We also let g_{ic} denote the number of slots that channel λ_c remains idle between the end of transmissions by node i and the start of transmissions by node $i+1$; we will refer to quantities g_{ic} as the *gaps* within the channels.

The problem of finding an optimum schedule such that a) the schedule is in the class defined in (8) and b) the transmitter sequence is $(1, 2, \cdots, N)$, can be formulated as an integer programming problem, to be referred to as *bandwidth limited*

OSTL (BW-OSTL):

$$BW-OSTL: \min_{g_{ic},K_c} M = \max_c \left\{ \sum_{i=1}^{N} (a_{ic} + g_{ic}) \right\} \quad (10)$$

subject to

$$K_c + \sum_{j=1}^{i-1}(a_{jc}+g_{jc}) \geq K_{c-1} + \sum_{j=1}^{i-1}(a_{j,c-1}+g_{j,c-1}) + a_{i,c-1} + \Delta, \qquad c = 2,\cdots,C; \;\; i=1,\cdots,N \quad (11)$$

$$M + \sum_{j=1}^{i-1}(a_{j1}+g_{j1}) \geq K_C + \sum_{j=1}^{i-1}(a_{jC}+g_{jC}) + a_{iC} + \Delta, \quad i=1,\cdots,N \quad (12)$$

$$g_{ic}, K_c, M: \text{ integers}; \; g_{ic} \geq 0 \quad \forall i,c; K_1 = 0; \quad K_c > K_{c-1}; \quad c=2,\cdots,C; \quad M > K_C. \quad (13)$$

Constraint (11) ensures that following its packet transmissions on channel λ_{c-1}, the laser at node i has enough time to switch to wavelength λ_c. Constraint (12) ensures that transmitter i has enough time to tune from channel λ_C (the last channel) to channel λ_1 to transmit in the next frame. These two constraints correspond to the hardware constraints (2). The no-collision constraints (3) are accounted for in the above description by the constraint $g_{ic} \geq 0 \; \forall i,c$; by definition of g_{ic}, this guarantees that the slots assigned to node $i+1$ on channel λ_c will be scheduled after the slots assigned to node i in the same channel.

Finding an optimal schedule within the class (8) for problem *OSTL*, involves solving $N!$ *BW-OSTL* problems, one for each possible transmitter sequence, and choosing the schedule of smallest frame size. Solving problem *BW-OSTL* is itself a hard task since it is an integer programming problem. Recall, however, that we are considering bandwidth limited networks, and the bandwidth bound (4) dominates. In other words, there exists at least one channel λ_c such that $\sum_{i=1}^{N} a_{ic} = M^{(l)}$. If a schedule of length $M^{(l)}$ exists, then at least one channel, say, channel λ_c, will never be idle; this schedule is such that $g_{ic} = 0 \forall i$. We will show that fixing the values of g_{ic} for one channel makes it possible to solve problem *BW-OSTL* in polynomial time. But first, we answer a fundamental question related to the existence of schedules of length $M^{(l)}$ within the class (8).

1) A Sufficient Condition for Optimality: Let $\mathbf{A}$ be the matrix of a bandwidth limited network, $M^{(l)}$ be the lower bound, and define the *average slot requirement* as $a = M^{(l)}/N$. If $a_{ic} = a \forall i,c$, then an optimum length schedule is easy to construct; all of (11)–(13) will be satisfied by letting $K_c = (c-1)(a+\Delta)\forall c$; $g_{ic} = 0 \quad \forall i,c$; and $M = M^{(l)} = Na$. The question that naturally arises then, is whether we can guarantee a schedule of $M^{(l)}$ slots when we allow nonuniform traffic. The answer is provided by the following lemma whose proof can be found in Appendix A. Note that ϵ in the lemma is greater than zero only when $M^{(l)} > \frac{NC\Delta}{N-C}$; this is consistent with our hypothesis of a bandwidth limited network.

Lemma 1: Let $\mathbf{A}$ be a collapsed traffic matrix such that the lower bound $M^{(l)} = M_{bw}^{(l)} > M_t^{(l)}$ (bandwidth limited network). Then, a schedule of length equal to $M^{(l)}$ exists within the class (8) for any transmitter sequence, if the elements of $\mathbf{A}$ satisfy the following condition:

$$\left| a_{ic} - \frac{M^{(l)}}{N} \right| \leq \epsilon \qquad \forall i,c \quad (14)$$

with ϵ given by

$$\epsilon = \frac{2M^{(l)}}{N+2}\left(\frac{1}{C} - \frac{1}{N} - \frac{\Delta}{M^{(l)}}\right). \quad (15)$$

Lemma 1 provides an upper bound on the "degree of nonuniformity" of matrix $\mathbf{A}$ in order to guarantee a schedule of length equal to the lower bound. Its proof, however, is based on a worst case scenario; in general, we expect such an optimal schedule to exist for significantly higher degrees of nonuniformity.

2) Scheduling Algorithm: We develop an algorithm which, under the conditions of Lemma 1, produces schedules of length $M^{(l)}$. In fact, the algorithm is optimal under looser conditions that do not impose any bound on the variation of a_{ic} around $\frac{M^{(l)}}{N}$. The key idea is to schedule transmissions on λ_1 so that this channel is always busy, except, maybe after all nodes have been given a chance to transmit; we expect this strategy to work well when channel λ_1 is dominant, that is, $\sum_{i=1}^{N} a_{i1} = M^{(l)}$.

Algorithm *Make_Bandwidth_Limited_Schedule (MBLS)*, described in detail in Fig. 3, operates as follows. All gaps in channel λ_1 are initialized to zero; then, during Pass 1, transmissions in channels λ_2 through λ_C are scheduled at the earliest possible time that satisfies constraints (11). Doing so, however, may introduce large gaps into these channels, resulting in a sub-optimal schedule (refer to (10)). During the second pass, the algorithm attempts to compact the gaps within each channel by shifting the slots to the right or left, but only as far as constraints (11) and (12) allow. That algorithm *MBLS* is correct follows from the fact that it constructs a schedule which satisfies constraints (11)–(13). It is also easy to verify that its running-time complexity is $\mathcal{O}(CN^2)$. We now state the optimality properties of algorithm *MBLS*; the proof of Theorem 2 can be found in Appendix B.

Theorem 2: Algorithm $MBLS$ constructs a schedule of minimum length among the schedules that: a) are within the class (8) and the sequence of transmitters is $(1,2,\cdots,N)$; b) channel λ_1 is a dominant channel; and c) channel λ_1 is never idle, except, possibly, at the very end of the frame (i.e., $g_{i1} = 0, i = 1,\cdots,N-1$).

Corollary 1 [Optimality of Algorithm MBLS]: Let λ_1 be a channel such that $\sum_{i=1}^{N} a_{i1} = M^{(l)}$, and arbitrarily label the transmitters 1 through N. Then, under the conditions of Lemma 1, algorithm MBLS constructs an optimum length schedule.

Proof: According to Lemma 1, there exists a schedule of length $M^{(l)}$ within the class defined by (8), such that the transmitter sequence is $(1,2,\cdots,N)$. Since λ_1 is the dominant channel, any schedule of length $M^{(l)}$ is such that channel λ_1 is

Algorithm *Make_Bandwidth_Limited_Schedule (MBLS)*

The algorithm assumes that channel λ_1 is dominant. Also, references to channel λ_{c+1} when $c = C$ denote the next frame on channel λ_1.

```
begin
   Set M = Σ_{i=1}^{N} a_{i1}
   Set K_1 and all gaps g_{i1} on λ_1 equal to 0
   // Begin Pass 1
   for c = 2 to C do
      for i = 1 to N do
         Schedule the a_{ic} slots at the earliest possible time
         such that constraint (11) is satisfied between channels λ_c and λ_{c-1}
   // end of for c loop
   // End of Pass 1 - initial values to all g_{ic} have now been determined
   Let M' be the smallest integer satisfying constraint (12)
   Set M = max{M, M'}
   // Begin Pass 2
   for c = C downto 2 do
      for i = N downto 1 do
         Shift the a_{ic} slots as much right as possible while
         maintaining constraint (11) between channels λ_c and λ_{c+1}
         for j = i + 1 to N do
            Shift the a_{jc} slots as much left as possible while
            maintaining constraint (11) between channels λ_c and λ_{c-1}
      // end of for i loop - the final values of gaps for this channel have now been determined
      Let M_c = Σ_{i=1}^{N} (a_{ic} + g_{ic})
      M = max(M, M_c)
   // end of for c loop - M is now the final length of the schedule
// end of algorithm
```

Fig. 3. Scheduling algorithm for bandwidth limited networks.

never idle. Therefore, because of Theorem 2, algorithm MBLS will construct such a schedule. □

B. Tuning and Bandwidth Balanced Networks

We now study the operation of the network when the tuning and bandwidth bounds are equal, $M^{(l)} = M_t^{(l)} = M_{bw}^{(l)}$. Theorem 3 states that, in this case, even arbitrarily small nonuniformities in the traffic pattern may result in *every* admissible schedule having length *greater* than the lower bound. Note that neither the theorem nor its proof refer to the class of schedules defined by (8) and (9), therefore, this result holds for arbitrary schedules.

Theorem 3: Let **A** be a matrix such that $M^{(l)} = M_{bw}^{(l)} = M_t^{(l)}$, and such that each transmitter and each channel are *tight* (i.e., the slot requirement on each channel and the slot-plus-tuning requirement of each transmitter are equal to $M^{(l)}$). Then, the optimal schedule has length strictly greater than $M^{(l)}$, even for any arbitrarily small nonuniformity among the elements of **A**.

Proof: Consider a system of N nodes, C channels, and Δ tuning slots, and suppose that $a = \frac{C\Delta}{N-C} > 1$ is an integer. Letting $a_{ic} = a \forall i, c$, we obtain a uniform matrix satisfying the conditions of the theorem, i.e., such that $M_{bw}^{(l)} = Na = C(a+\Delta) = M_t^{(l)}$, and such that all transmitters and channels are tight. Let us now modify some of the elements of this matrix to construct a nonuniform matrix **A** that continues to satisfy the conditions of the theorem. At least four elements have to be modified to achieve this result. For if for some i, c_1, we let $a_{i,c_1} < a$, we have to increase $a_{i,c_2}, c_2 \neq c_1$, to make transmitter i tight again. And to make channels λ_{c_1} and λ_{c_2} tight again, the least number of elements that need to be changed are the two elements a_{j,c_1} and a_{j,c_2}, for some transmitter $j \neq i$. We now let the matrix **A** be such that $a_{11} = a_{22} = a-1$, $a_{21} = a_{12} = a+1$, and $a_{ic} = a$ for all other i, c. It is easy to verify that $M_t^{(l)} = C(a+\Delta) = Na = M_{bw}^{(l)}$, and that all channels and transmitters are tight. This traffic matrix is as close to the uniform matrix as possible, while still satisfying the conditions of the theorem. Suppose now that a schedule of length $Na = C(a+\Delta)$ exists. Then neither any channel nor any transmitter can be idle at any time in such a schedule. Assuming that the schedule starts at time 0, all transmissions on λ_3 through λ_C begin and end at times which are multiples of a; similarly for transmissions by stations 3 through N.

Bandwidth Limited Scheduling Heuristic (BLSH)

1. Relabel the channels such that:

$$M^{(l)} = \sum_{i=1}^{N} a_{i1} \geq \sum_{i=1}^{N} a_{i2} \geq \cdots \geq \sum_{i=1}^{N} a_{iC}$$

   Label the transmitters as $1, \cdots, N$, and let $s^{(1)} = (1)$. Repeat Step 2 for $i = 2, \cdots, N$.

2. Let $s^{(i-1)} = (\pi_1, \cdots, \pi_{i-1})$ be the permutation produced by the previous iteration on a network with only the first $i-1$ transmitters of the original network. Consider transmitter i. Run algorithm *MBLS* on each of the i permutations

$$(i, \pi_1, \cdots, \pi_{i-1}), (\pi_1, i, \pi_2, \cdots, \pi_{i-1}), \cdots, (\pi_1, \cdots, \pi_j, i, \pi_{j+1}, \cdots, \pi_{i-1}), \cdots, (\pi_1, \cdots, \pi_{i-1}, i)$$

   Let $s^{(i)}$ be the permutation that results in the least length schedule.

Fig. 4. Scheduling heuristic.

Without loss of generality, assume that node 1 is before node 2 in the transmitter sequence of channel λ_1 in this schedule, and let $i > 2$ be the transmitter immediately before node 1 in this sequence. Let t be the time i's transmission on λ_1 ends; then t must be a multiple of a. Since channel λ_1 is never idle, the transmission by node 1 on λ_1 starts at time t, and ends at time $t+a-1$. At that (nonmultiple of a) time, node 2 is the only candidate for immediate transmission on λ_1, so node 2 must start transmitting on λ_1 at time $t+a-1$. Node 1, on the other hand, after a tuning period of Δ slots, is ready for its next transmission at time $t+a-1+\Delta$; since this is not a multiple of a, and since node 1 can never be idle, it can only start transmission on channel λ_2. Using similar arguments, node 2's transmission must have just ended on channel λ_2. We have established that, under this schedule, on channel λ_1 node 1 transmits from time t to time $t+a-1$, and node 2 from time $t+a-1$ to time $t+2a$, and on channel λ_2 node 2 transmits from time $t+\Delta$ to time $t+\Delta+a-1$, and node 1 from time $t+\Delta+a-1$ to time $t+\Delta+2a$. But, regardless of the values of a and Δ, this sequence of transmissions is impossible,[3] contradicting our hypothesis that an admissible schedule of length equal to the lower bound $Na = C(a+\Delta)$ exists. □

VI. Optimization Heuristics

We now develop a scheduling heuristic for bandwidth limited networks that performs well when applied to arbitrary instances of *OSTL* that may not satisfy the optimality conditions of the previous section. Using a very similar reasoning, it is relatively straightforward to obtain a heuristic for tuning limited networks.

Recall that for bandwidth limited networks, finding a schedule within the class (8) that solves the *OSTL* problem involves solving $N!$ *BW-OSTL* problems, one for each possible transmitter sequence. On the other hand, we have no efficient algorithm for solving the most general version of *BW-OSTL*, but we have developed *MBLS*, to solve *BW-OSTL* for a given transmitter sequence under the additional constraint that any idling of the first channel occurs after all nodes have transmitted on that channel. Thus, our approach to obtaining near-optimal schedules for *OSTL* is based on making two compromises.

Suppose that an optimal transmitter sequence for a network of n nodes has been determined, and that a new node is added to the network. Instead of checking all possible $(n+1)!$ transmitter sequences, our first approximation is to assume that, in the optimal sequence for the $(n+1)$-node network, the relative positions of nodes 1 through n are the same as in the sequence for the n-node network; thus, we only need to determine where in the latter sequence node $n+1$ has to be inserted. This can be accomplished by solving $n+1$ *BW-OSTL* problems on a $(n+1)$-node network, one for each possible placement of node $n+1$ within the sequence of n nodes. Now let λ_1 be the dominant channel. Our second compromise is to use algorithm *MBLS* to solve the version of *BW-OSTL* which requires that λ_1 is never idle except at the end of the frame. From Theorem 2, if a schedule of length equal to $M^{(l)}$ exists for the given transmitter sequence, *MBLS* will find such a schedule. But if the optimal schedule has length greater than $M^{(l)}$, *MBLS* may fail to produce an optimal solution as the idling in the first channel may be anywhere within the frame, not necessarily at the end.

For bandwidth limited networks, our heuristic is described in Fig. 4. Regarding the complexity of the heuristic, note that Step 2 will dominate. During the ith iteration of Step 2, algorithm $MBLS$ is called i times on a network of i nodes. Since the complexity of $MBLS$ on a network of i nodes is $\mathcal{O}(Ci^2)$, the overall complexity of the heuristic is $\mathcal{O}(CN^4)$.

VII. Numerical Results

We consider four algorithms for the *OSTL* problem: 1) algorithm *MBLS*, described in Fig. 3; the algorithm is applied after the channels have been labeled λ_1 through λ_C in decreasing order of $\sum_{i=1}^{N} a_{ic}$, and the transmitters have

[3] For instance, if $\Delta < a-1$, we have $t+\Delta < t+a-1 < t+\Delta+a-1 < t+2a$, and the transmissions of node 2 on channels λ_1 and λ_2 overlap; similarly for $\Delta > a-1$.

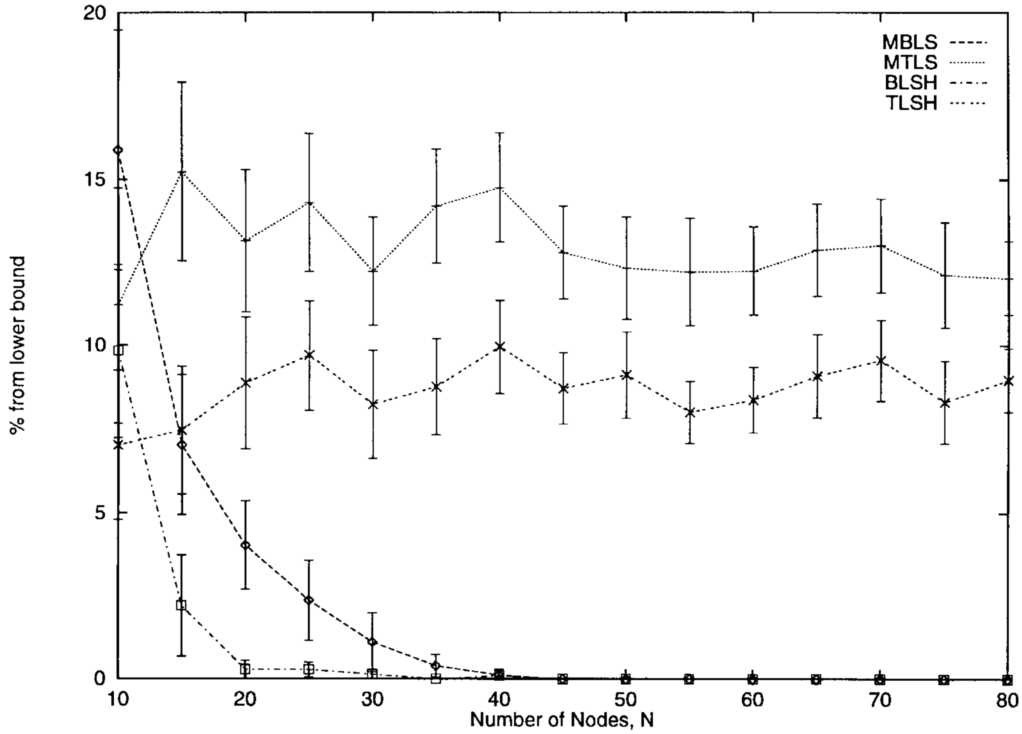

Fig. 5. Algorithm comparison for $C = 10$ channels and $\Delta = 1$ tuning slots.

been labeled 1 through N in decreasing order of $\sum_{c=1}^{C} a_{ic}$; 2) algorithm *MTLS*, with the same labeling of channels and transmitters; *MTLS* has not been described, but is very similar to *MBLS*, only targeted to tuning limited networks; 3) heuristic *BLSH*, described in Fig. 4; 4) heuristic *TLSH* for tuning limited networks; this heuristic has not been described, but is very similar to *BLSH*.

Let M be the actual length of a schedule for matrix **A** produced by some scheduling algorithm. Figs. 5–7 plot quantity $\frac{M-M^{(l)}}{M^{(l)}}100\%$ against the number of nodes, N, for the four algorithms described above; 95% confidence intervals are also shown in the figures. The elements of each matrix **A** were chosen, with equal probability, among the integers 1 through 20. We show results only for $C = 10$; very similar results for other values of C, as well as for matrices **A** generated using other distributions, can be found in [18]. We have used three different values for Δ, namely $\Delta = 1, 4, 16$, while N varies from 10 to 80.

Our first observation from Figs. 5–7 is that the two heuristics, *BLSH* and *TLSH*, always perform as good as, or better than the corresponding algorithms, *MBLS* and *MTLS*, respectively, as expected. The results also confirm our intuition regarding the two regions of network operation, and justify the need for algorithms specially designed for each region. Let us, for the moment, refer to Fig. 7 which shows results for $C = 10, \Delta = 16$. For these values of C and Δ, and the way the traffic matrices are constructed, a network is in the bandwidth limited region if $N > 25$, and in the tuning limited region, otherwise. It is not surprising then that algorithms *MBLS* and *BLSH* outperform their counterparts, *MTLS* and *TLSH*, respectively, when $N > 25$, while the opposite is true for $N < 25$.

Consider the performance of *MBLS* and *BLSH* in the *bandwidth limited* region; similar conclusions can be drawn for the performance of *MTLS* and *TLSH* in the *tuning limited* region. In general, the length of schedules produced by *MBLS* and *BLSH* are very close to the lower bound, and, for networks well within the bandwidth limited region (i.e., for sufficiently large N), *BLSH* and *MBLS* construct schedules of length *equal* to the lower bound. This is an important result, as it establishes that the lower bound accurately characterizes the scheduling efficiency. Since the lower bound is independent of the tuning latency in this region, this result also implies that it is possible to appropriately dimension the network to minimize the effects of even large values of Δ. In the boundary of the tuning and bandwidth limited regions where the tuning and bandwidth bounds are close to each other, the algorithms do not perform as close to the lower bound (although they are never more than 15% away from it). When several channels and nodes have similar slot requirements, the algorithms have less flexibility in placing the slots to obtain schedules of length close to the lower bound. Theorem 3, however, suggests that this behavior is not due to inefficiency inherent to the algorithms, but is rather due to the fact that the optimal schedules in this region have length greater than the lower bound.

We conclude that *BLSH* and *TLSH* achieve the best performance within the bandwidth and tuning limited regions, respectively. Algorithms *MBLS* and *MTLS* can achieve almost similar performance, but they are more efficient in terms of running time.

VIII. Concluding Remarks

We considered the problem of designing TDM schedules for broadcast optical networks. Based on a new formulation of the

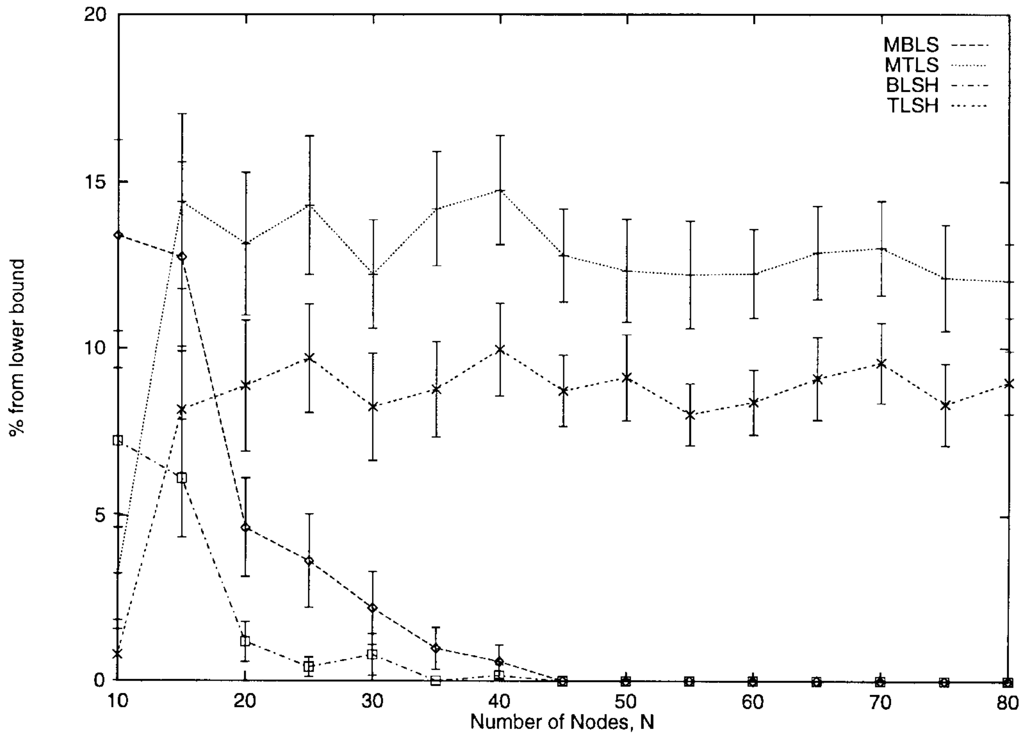

Fig. 6. Algorithm comparison for $C = 10$ channels and $\Delta = 4$ tuning slots.

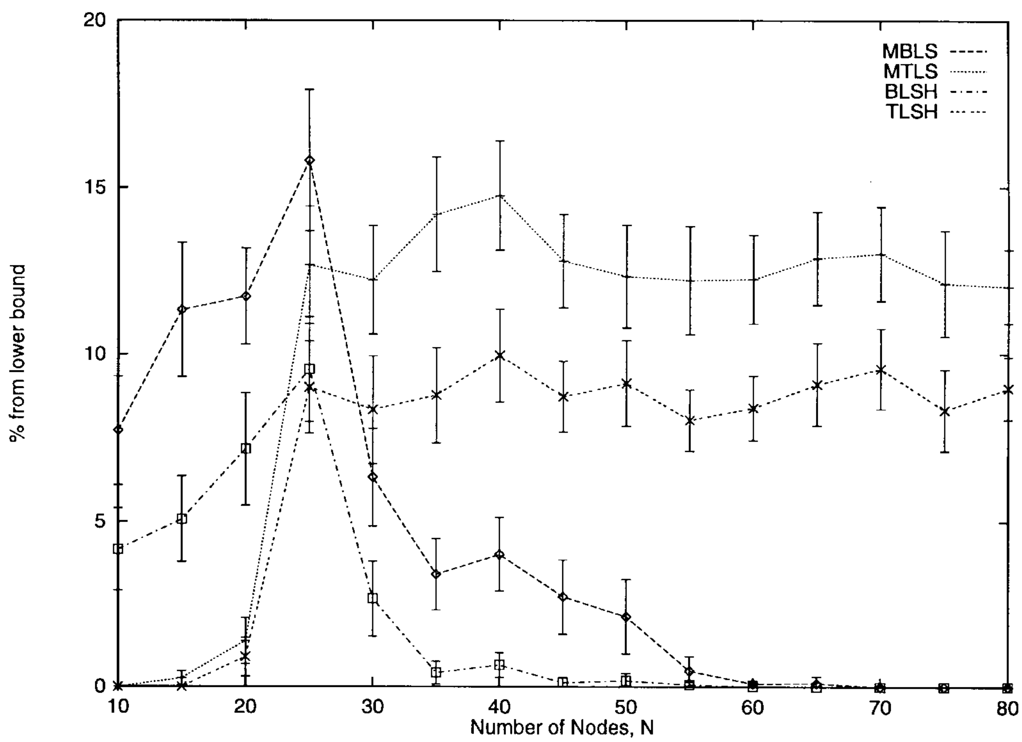

Fig. 7. Algorithm comparison for $C = 10$ channels and $\Delta = 16$ tuning slots.

scheduling problem, we presented algorithms which construct schedules of length very close to, or equal to the lower bound. We also established that, as long as the network operates within the bandwidth limited region, even large values of the tuning latency have no effect on the length of the schedule. The main conclusion of our work is that through careful design, it is possible to realize single-hop WDM networks operating at very high data rates, using *currently available* optical tunable devices.

APPENDIX

A. Proof of Lemma 1

In proving Lemma 1 we will make use of the following result.

Lemma 2: If constraints (14) on the elements of **A** hold, then for any subset $\mathcal{T}$ of transmitters from $\{1, \cdots, N\}$ such that $\mid \mathcal{T} \mid = n$, and any channel λ_c:

$$\frac{nM^{(l)}}{N} - \frac{N\epsilon}{2} \leq \sum_{i \in \mathcal{T}} a_{ic} \leq \frac{nM^{(l)}}{N} + \frac{N\epsilon}{2}. \qquad (16)$$

Proof: If $n \leq N/2$, (16) follows directly from (14). If $n > N/2$, let $\overline{\mathcal{T}} = \{1, 2, \cdots, N\} - \mathcal{T}$. The result in (13) follows from (14) and the fact that the size of $\overline{\mathcal{T}}$ is less than $N/2$. □

We are now ready to prove Lemma 1. Although the proof refers to the problem formulation in (10)–(13), it does not depend on the transmitter sequence. As a result, it holds for any transmitter sequence, not just the $(1, 2, \cdots, N)$ sequence implied in (10)–(13).

Proof of Lemma 1: Let us introduce a new variable $K_{C+1} = M$ in the formulation of the *BW-OSTL* problem in (10) - (13), Then, inequalities (11) and (12) can be rewritten as one inequality (references to channel λ_{c+1} when $c = C$ refer to the next frame on λ_1):

$$K_{c+1} - K_c \geq \sum_{j=1}^{i-1} (a_{jc} + g_{jc}) - \sum_{j=1}^{i-1} (a_{j,c+1} + g_{j,c+1}) + a_{ic} + \Delta, \qquad c = 1, \cdots, C; \quad i = 1, \cdots, N. \qquad (17)$$

For the proof we consider a worst case scenario, under which the total slot requirement on *each* channel is equal to the lower bound. A schedule of length $M^{(l)}$ under such a scenario will ensure a schedule of length $M^{(l)}$ when the slot requirement on some channel is less than $M^{(l)}$, as one can simply introduce slots in which this channel is idle.

Since we are trying to achieve a schedule of length $M^{(l)}$, we are seeking a solution to problem *BW-OSTL* such that $g_{ic} = 0 \quad \forall i, c$. We can then rewrite inequality (17) as

$$K_{c+1} - K_c \geq \left(\sum_{j=1}^{i-1} a_{jc} - \sum_{j=1}^{i-1} a_{j,c+1} \right) + a_{ic} + \Delta, \qquad c = 1, \cdots, C; \quad i = 1, \cdots, N. \qquad (18)$$

To satisfy (18) we need to set $K_{c+1} - K_c$ to the largest possible value of the right hand side in (18). If n_c is the value of i which maximizes the the right hand side of (18), we set:

$$K_{c+1} - K_c = \left(\sum_{j=1}^{n_c-1} a_{jc} - \sum_{j=1}^{n_c-1} a_{j,c+1} \right) + a_{n_c,c} + \Delta, \qquad c = 1, \cdots, C. \qquad (19)$$

Adding up the above equations for all channels, we get:

$$K_{C+1} = \left\{ \sum_{j=1}^{n_1-1} a_{j1} + \sum_{c=2}^{C} \left(\sum_{j=1}^{n_c-1} a_{jc} - \sum_{j=1}^{n_{c-1}-1} a_{jc} \right) - \sum_{j=1}^{n_C-1} a_{jC} \right\} + \sum_{c=1}^{C} a_{n_c,c} + C\Delta. \qquad (20)$$

Equation (20) can be expressed in a compact form by introducing constants $n_0 = n_{C+1} = 1$:

$$K_{C+1} = \sum_{c=1}^{C+1} \left(\sum_{j=1}^{n_c-1} a_{jc} - \sum_{j=1}^{n_{c-1}-1} a_{jc} \right) + \sum_{c=1}^{C} a_{n_c,c} + C\Delta. \qquad (21)$$

We now define the set S to be a subset of $\{1, 2, \cdots, C\}$ such that $c \in S$ if and only if $n_c \geq n_{c-1}$; we also let $\overline{S} = \{1, 2, \cdots, C\} - S$. Inequality (21) can then be written as:

$$K_{C+1} = \left\{ \sum_{c \in S} \left(\sum_{j=n_{c-1}}^{n_c-1} a_{jc} \right) - \sum_{c \in \overline{S}} \left(\sum_{j=n_c}^{n_{c-1}-1} a_{jc} \right) \right\} + \sum_{c=1}^{C} a_{n_c,c} + C\Delta. \qquad (22)$$

Let $|S| = k$. Then, from Lemma 2 we have that

$$\sum_{c \in S} \left(\sum_{j=n_{c-1}}^{n_c-1} a_{jc} \right) \leq \left[\sum_{c \in S} (n_c - n_{c-1}) \right] \frac{M^{(l)}}{N} + \frac{kN\epsilon}{2} \qquad (23)$$

$$\sum_{c \in \overline{S}} \left(\sum_{j=n_c}^{n_{c-1}-1} a_{jc} \right) \geq \left[\sum_{c \in \overline{S}} (n_{c-1} - n_c) \right] \frac{M^{(l)}}{N} - \frac{(C-k)N\epsilon}{2}. \qquad (24)$$

If we subtract (24) from (23) all but one term cancel out on the right hand side:

$$\sum_{c \in S} \left(\sum_{j=n_{c-1}}^{n_c-1} a_{jc} \right) - \sum_{c \in \overline{S}} \left(\sum_{j=n_c}^{n_{c-1}-1} a_{jc} \right) \leq \frac{CN\epsilon}{2}. \qquad (25)$$

Using this in (22) and using (14) to bound a_{jc}, we have an upper bound on K_{C+1}:

$$K_{C+1} \leq \frac{CN\epsilon}{2} + C\left(\frac{M^{(l)}}{N} + \epsilon \right) + C\Delta. \qquad (26)$$

We can then guarantee that $M^{(l)} \geq K_{C+1}$ if (15) is satisfied. □

B. Proof of Theorem 2

Proof: Let *Sched(c)* denote the frame of the schedule on λ_c starting with the first slot in which transmitter 1 transmits on λ_c. *Sched(C+1)* refers to the next frame on λ_1. Once the schedule length M and gaps $g_{ic}, i = 1, \cdots, N-1$, are known, gap g_{Nc} is uniquely determined, and will be ignored in the following. Let *OPT* denote the optimal length under the assumptions of Theorem 2. We prove that *OPT* $= M$ by tracing the algorithm and showing that *OPT* $\geq M$ at every step. That *OPT* $\geq M$ at the end of Step 2 is obvious, since the optimal can be no smaller than the lower bound. In Pass 1, all transmitters are assigned the earliest possible slots on each channel, and Step 9 makes sure that the schedule length is large enough so that each transmitter gets enough time to tune back to λ_1 after its transmission on λ_C. Therefore *OPT* $\geq M$ at the end of Pass 1.

In Pass 2, channels and transmitters are processed in reverse order, and the algorithm tries to compact the gaps g_{ic} as much as possible. We show that once the gaps on a channel λ_c have been compacted by Pass 2, it is not possible to compact them any further to reduce the length, thus proving that *OPT* $\geq M$. The proof is by a two-level induction—the first on c and the second on i within the same channel λ_c. The induction proceeds by assuming that *Sched(c+1)* is optimal (meaning that the gaps on λ_{c+1} cannot be compacted any further), and that transmitters $i+1, \cdots, N$ are optimally scheduled on λ_c (i.e., that the gaps $g_{i+1,c} \cdots g_{N-1,c}$ cannot be compacted any further), and then showing that the gap g_{ic} cannot be compacted any more than what Pass 2 does. There are only 2 ways gap g_{ic} can be compacted—either by moving the a_{ic} slots to the right, or by moving slots $a_{jc}, j = i+1, \cdots, N$, to the left. But the a_{ic} slots cannot be moved any more to the right (otherwise Step 12 would have done so), neither can slots a_{jc} be moved any more to the left (otherwise Step 14 would have done so). Hence gap g_{ic} is as compact as can be, and λ_c is optimal by induction. To complete the induction proof, note that the inductive hypothesis holds for $c = C$, since *Sched(C+1)* is the same as the schedule on channel λ_1, which is optimal by assumption, as we only consider schedules in which channel λ_1 is idle only at the end of the frame. □

REFERENCES

[1] A. Aggarwal, A. Bar-Noy, D. Coppersmith, R. Ramaswami, and B. Schieber, ''Efficient routing and scheduling algorithms for optical networks,'' IBM Res. Rep., Tech. Rep. RC 18967, 1994.

[2] M. Azizoglu, R. A. Barry, and A. Mokhtar, ''Impact of tuning delay on the performance of banwidth-limited optical broadcast networks with uniform traffic,'' *IEEE J. Select. Areas Commun.*, vol. 14, no. 5, pp. 935–944, June 1996.

[3] K. Bogineni, K. M. Sivalingam, and P. W. Dowd, ''Low-complexity multiple access protocols for wavelength-division multiplexed photonic networks,'' *IEEE J. Select. Areas Commun.*, vol. 11, no. 4, pp. 590–604, May 1993.

[4] M. S. Borella and B. Mukherjee, ''Efficient scheduling of nonuniform packet traffic in a WDM/TDM local lightwave network with arbitrary transceiver tuning latencies,'' *IEEE J. Select. Areas Commun.*, vol. 14, no. 5, pp. 923–934, June 1996.

[5] S. Chalasani and A. Varma, ''Efficient time-slot assignment algorithms for SS/TDMA systems with variable bandwidth beams,'' *IEEE Trans. Commun.*, vol. 42, no. 3, pp. 1359–1370, Mar. 1994.

[6] H. Choi, H.-A. Chois, and M. Azizoglu, ''Optimum transmission scheduling in optical broadcast networks,'' in *Proc. ICC '95*, pp. 266–270. IEEE, 1995.

[7] E. Coffman, M. R. Garey, and D. S. Johnson, ''An application of bin-packing to multiprocessor scheduling,'' *SIAM J. Computing*, vol. 7, pp. 1–17, Feb. 1978.

[8] A. Ganz and Y. Gao, ''Time-wavelength assignment algorithms for high performance WDM star based networks,'' *IEEE Trans. Commun.*, vol. 42, no. 4, pp. 1827–1836, Apr. 1994.

[9] T. Gonzalez and S. Sahni, ''Open shop scheduling to minimize finish time,'' *J. Assoc. Comput. Mach.*, vol. 23, no. 4, pp. 665–679, Oct. 1976.

[10] I. Gopal and C. Wong, ''Minimizing the number of switchings in an SS/TDMA system,'' *IEEE Trans. Commun.*, vol. 33, no. 6, pp. 497–501, June 1985.

[11] P. E. Green, *Fiber Optic Networks*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[12] P. A. Humblet, R. Ramaswami, and K. N. Sivarajan, ''An efficient communication protocol for high-speed packet-switched multichannel networks,'' *IEEE J. Select. Areas Commun.*, vol. 11, no. 4, pp. 568–578, May 1993.

[13] T. Inukai, ''An efficient SS/TDMA time slot assignment algorithm,'' *IEEE Trans. Commun.*, vol. 27, no. 10, pp. 1449–1455, Oct. 1979.

[14] B. Mukherjee, ''WDM-Based local lightwave networks—Part I: Single-hop systems,'' *IEEE Network Mag.*, pp. 12–27, May 1992.

[15] G. R. Pieris and G. H. Sasaki, ''Scheduling transmissions in WDM broadcast-and-select networks,'' *IEEE/ACM Trans. Networking*, vol. 2, no. 2, pp. 105–110, Apr. 1994.

[16] G. N. Rouskas and M. H. Ammar, ''Analysis and optimization of transmission schedules for single-hop WDM networks,'' *IEEE/ACM Trans. Networking*, vol. 3, no. 2, pp. 211–221, Apr. 1995.

[17] G. N. Rouskas and M. H. Ammar, ''Minimizing delay and packet loss in single-hop lightwave WDM networks using TDM schedules,'' in *IEEE Proc. ICC '95*, pp. 1267–1271, June 1995.

[18] G. N. Rouskas and V. Sivaraman, ''On the design of optimal TDM schedules for broadcast WDM networks with arbitrary transceiver tuning latencies,'' North Carolina State University, Raleigh, NC, Tech. Rep. TR-95-07, 1995.

[19] V. Sivaraman and G. N. Rouskas, ''HiPeR-ℓ: A $\underline{H}i$gh $\underline{P}e$rformance $\underline{R}$eservation protocol with ℓook-ahead for broadcast WDM networks,'' in *IEEE Proc. INFOCOM '97*, Apr. 1997.

George N. Rouskas (M'95) received the Diploma in electrical engineering from the National Technical University of Athens (NTUA), Athens, Greece, in 1989, and the M. S. and Ph. D. degrees in computer science from the College of Computing, Georgia Institute of Technology, Atlanta, in 1991 and 1994, respectively.

He joined the Department of Computer Science, North Carolina State University, in August 1994, as an Assistant Professor. His research interests include lightwave network architectures, multicast communication, high-speed networks, and performance evaluation.

Dr. Rouskas received the 1995 Outstanding New Teacher Award from the Department of Computer Science, North Carolina State University. He was also the recipient of the 1994 College of Computing Graduate Research Assistant Award. He is a member of the ACM and of the Technical Chamber of Greece.

Vijay Sivaraman received the B.S. degree in computer science and engineering from the Indian Institute of Technology, Delhi, India, in 1990 and the M.S. degree in computer science from North Carolina State University, Raleigh, in December 1995. He is currently pursuing the P.hD. degree from the Department of Computer Science, University of California, Los Angeles.

His research interests include protocols for optical networks and issues in wireless and mobile networks.