On the Design of Optimal TDM Schedules for Broadcast WDM Networks with Arbitrary Transceiver Tuning Latencies *

George N. Rouskas *Vijay Sivaraman*
Department of Computer Science, North Carolina State University, Raleigh, NC 27695-8206

Abstract

We consider the problem of scheduling packet transmissions in single-hop WDM networks, with tunability provided only at one end. Our objective is to design schedules of minimum length for a given traffic demand matrix. The contribution of our work is twofold. First we define a special class of schedules which permit an intuitive formulation of the scheduling problem. We then present algorithms which construct schedules of length equal to the lower bound provided that certain optimality conditions are satisfied. 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. In the first region the schedule length is determined by the tuning requirements, while in the second it is determined by the traffic demands. The point at which the network switches between the two regions is identified in terms of the number of nodes and channels, and the tuning latency. Accordingly, we show that it is possible to appropriately dimension the network to offset the effects of even large values of tuning latency.

1 Introduction

Wavelength division multiplexing (WDM) is the most promising approach to exploiting the vast information-carrying capacity of single-mode fiber. By dividing the bandwidth of the optical medium into narrower channels, WDM makes it possible to implement communication networks with a large number of users, and an aggregate throughput that can be in the order of Terabits per second. Our focus in this paper is on a WDM network architecture known as the *single-hop* architecture [1], which is *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* devices with the ability to access the various channels. Such devices do exist today; however, their capabilities are limited in terms of both tunability range and speed. Furthermore, ideal devices that can tune across the useful optical spectrum in sub-microsecond times [2] are not expected in the foreseeable future. As a result, for emerging communication environments characterized by very high data rates rates (Gigabits per second) and small packet sizes (e.g., 53-byte ATM cells), the latency of even the fastest available tunable devices dominate over packet transmission times. An important design goal in these environments is to minimize the impact of tuning latency on network performance.

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. This scheduling problem has been studied in various contexts [3, 4, 5, 6]. Our work is more general, as it considers arbitrary traffic demands and arbitrary values of tuning latency, and presents sufficient conditions for the existence of optimal schedules. We also make the fundamental observation that, depending on the traffic matrix and various system parameters, the network can be operating in one of two distinct regions. We then develop two scheduling algorithms, and demonstrate that an algorithm optimal for one region performs sub-optimally when applied to a network operating in the other region. We also present new heuristics (again one for each region) which are based on the intuition provided by an appropriate formulation of the scheduling problem.

In Section 2 we describe our system traffic model, and in Section 3 we show that the scheduling problem is $\mathcal{NP}$-complete; we also derive lower bounds, and discuss the effect of the dominant bound on the network operation. We introduce a special class of schedules in Section 4, and develop scheduling algorithms which, under certain conditions, construct optimal schedules within this class. Scheduling heuristics are developed in Section 5, and in Section 6 we present some numerical results. We conclude the paper in Section 7.

2 System Model

We consider packet transmissions in a single-hop WDM network with a passive star topology. Each of the N nodes in the network employs one transmitter and one receiver. The passive star supports C wavelengths; in general, $C \leq N$. Without loss of generality, we only consider tunable-transmitter, fixed-receiver networks. Each tunable transmitter can be tuned to any and all wavelengths $\lambda_c, c = 1, \ldots, C$. The fixed receiver at station j, on the other hand, is assigned wavelength $\lambda(j) \in \{\lambda_1, \ldots, \lambda_C\}$, and we define $\mathcal{R}_c = \{j \mid \lambda(j) =$

*This work was supported in part by a grant from the Center for Advanced Computing and Communication, NC State University.

$\lambda_c\}, c = 1, \ldots, C$, as the set of receivers sharing channel λ_c. We also let δ denote the *normalized transmitter tuning latency*, expressed in units of packet transmission time; then $\Delta = \lceil \delta \rceil$ is the number of transmitter *tuning slots*.

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. Given a partition of the receiver set into sets $\mathcal{R}_c$, we obtain the *collapsed* $N \times C$ traffic matrix $\mathbf{A} = [a_{ic}]$. Element $a_{ic} = \sum_{j \in \mathcal{R}_c} d_{ij}$ represents the number of slots to be assigned to source i for transmissions on channel λ_c. Without loss of generality, 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 [1]. We also let $D = \sum_{i=1}^{N} \sum_{j=1}^{N} d_{ij}$ denote the total traffic demand.

There are several situations in which such a transmission scenario arises. For instance, under a gated service discipline, quantity d_{ij} may represent the number of packets with destination j in the queue of station i at the moment the "gate" is closed. Alternatively, it may represent the number of slots to be allocated to the (i, j) source-destination pair to meet certain quality of service (QOS) criteria; in the latter case d_{ij} may not directly depend on actual queue lengths, but may be derived based on assumptions regarding the arrival process at the source. The exact nature of d_{ij} is not important in this work and does not affect our conclusions.

While the traffic matrix, $\mathbf{D}$, is given, the collapsed matrix, $\mathbf{A}$, is not uniquely specified, but depends on the assignment of receivers to wavelengths. For the moment, we will assume that the receiver sets $\mathcal{R}_c$ are known; how to construct these sets will be discussed in Section 3.1.

2.1 Transmission Schedules

A *transmission schedule* is an assignment of slots to source-channel pairs such that 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 $\mathbf{A}$. If the a_{ic} slots are contiguously allocated for all pairs (i, λ_c), the schedule is said to be *non-preemptive*; otherwise we have a *preemptive* schedule. Under a non-preemptive schedule, each transmitter will tune to each channel exactly once, minimizing the overall time spent for tuning. Since our objective is to assign slots so as to minimize the time needed to satisfy the traffic demands specified by the collapsed traffic matrix, $\mathbf{A}$, we only consider non-preemptive schedules.

A non-preemptive 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 transmitter needs Δ slots to tune between channels, all time intervals $[\tau_{ic} - 1, \tau_{ic} + a_{ic} + \Delta - 1)$ must be disjoint [2], yielding a set of

[1] This assumption is reasonable, especially when the number of nodes, N, is significantly greater than the number of channels.

[2] We make the assumption that slot τ occupies the interval $[\tau - 1, \tau)$.

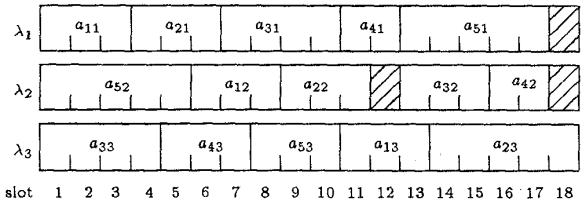

Figure 1: Schedule for a network with $N = 5$, $C = 3$, $\Delta = 2$.

hardware constraints on schedule $\mathcal{S}$, $\forall \; c \neq c', \; i = 1, \ldots, N$:

$$[\tau_{ic} - 1, \tau_{ic} + a_{ic} + \Delta - 1) \bigcap [\tau_{ic'} - 1, \tau_{ic'} + a_{ic'} + \Delta - 1) = \phi \quad (1)$$

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*, $\forall \; i \neq i', \; c = 1, \ldots, C$:

$$[\tau_{ic} - 1, \tau_{ic} + a_{ic} - 1) \bigcap [\tau_{i'c} - 1, \tau_{i'c} + a_{i'c} - 1) = \phi \quad (2)$$

A non-preemptive schedule $\mathcal{S}$ is *admissible* if and only if $\mathcal{S}$ satisfies both the hardware and the no-collision constraints. The *length*, M, of a schedule $\mathcal{S}$ for the collapsed traffic matrix $\mathbf{A}$ is the number of slots required to satisfy all traffic demands a_{ic} under $\mathcal{S}$. An *optimum length schedule* for $\mathbf{A}$ is one with the least length among all schedules. Figure 1 shows an optimum length non-preemptive schedule for a network with $N = 5$ nodes, $C = 3$ channels, and $\Delta = 2$; the collapsed traffic matrix $\mathbf{A}$ can be easily deduced from the figure.

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 $\tau_{ic} + kM, k = 1, 2, 3, \ldots$, where k denotes the k-th identical copy of the schedule as it repeats in time. Also, the term "schedule" will be used as an abbreviation for "admissible non-preemptive schedule".

3 Optimization and Lower Bounds

Our objective is to determine an optimum length schedule for a traffic matrix $\mathbf{D}$. This problem, which we will call the *Packet Scheduling with Tuning Latencies (PSTL)* problem, can be stated concisely as:

Problem 3.1 (PSTL) *Given the number N of nodes, the number C of wavelengths, the traffic matrix $\mathbf{D} = [d_{ij}]$, and the tuning slots Δ, find a schedule of minimum length.*

Problem *PSTL* can be logically decomposed into two subproblems: (a) sets of receivers, $\mathcal{R}_c$, sharing wavelength $\lambda_c, c = 1, \ldots, C$, must be obtained, and from them the collapsed traffic matrix, $\mathbf{A} = [a_{ic}]$, 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. 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 3.2 (OSTL) *Given the number N of nodes, the number C of wavelengths, the collapsed traffic matrix, $\mathbf{A}$, the tuning slots $\Delta \geq 0$, and a deadline, $M > 0$, is there a schedule $\mathcal{S} = \{\tau_{ic}\}$ that meets the deadline?*

OSTL reduces to the non-preemptive open-shop scheduling (*OS*) problem in [7] when we let $\Delta = 0$. Problem *OS* is $\mathcal{NP}$-complete for $C \geq 3$; but for $C = 2$, *OS* admits a polynomial-time solution [7]. The following theorem confirms our intuition that *OSTL* is in a sense more difficult than *OS*; its proof can be found in [8]. We next derive lower bounds for problems *PSTL* and *OSTL*, and discuss their implications.

Theorem 3.1 *OSTL is $\mathcal{NP}$-complete for any fixed $C \geq 2$.*

3.1 Lower Bounds for *PSTL* and *OSTL*

First, observe that 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*:

$$M_{bw}^{(l)} = \max_{1 \leq c \leq C} \left\{ \sum_{i=1}^{N} a_{ic} \right\} \geq \frac{D}{C} \qquad (3)$$

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 $\mathbf{D}$. Expression (3) implies that the bandwidth bound is minimized when the traffic load is perfectly balanced across the C channels.

Alternatively, 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 wavelengths. We call this the *tuning bound*:

$$M_{t}^{(l)} = \max_{1 \leq i \leq N} \left\{ \sum_{j=1}^{N} d_{ij} \right\} + C\Delta \geq \frac{D}{N} + C\Delta \qquad (4)$$

The tuning bound is independent of the assignment of receive wavelengths to the nodes, and only depends on parameters N, C, and Δ, and the total traffic demand D; it is minimized when each source contributes equally to the total traffic demand. We obtain the overall lower bound as

$$M^{(l)} = \max\left\{ M_{bw}^{(l)}, M_{t}^{(l)} \right\} \qquad (5)$$

This overall bound is minimized when

$$\frac{D}{C} = \frac{D}{N} + C\,\Delta \quad \Leftrightarrow \quad \frac{D}{C} = \frac{N\,C\,\Delta}{N - C} \qquad (6)$$

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, $\frac{D}{C}$, and the critical length 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.

In general, we will say that a network is *tuning limited*, if the tuning bound dominates, ($M^{(l)} = M_{t}^{(l)} > M_{bw}^{(l)}$), or *bandwidth limited*, if the bandwidth bound is dominant ($M^{(l)} = M_{bw}^{(l)} > M_{t}^{(l)}$). 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 of the schedule is determined by the tuning bound in (4), 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 $\sum_{i=1}^{N} a_{ic} = M_{bw}^{(l)}$. It is then desirable to operate the network at the bandwidth limited region, as doing so would eliminate the effects of tuning latency. Consequently, we would like to make the bandwidth bound in (6) greater than the critical length:

$$\frac{D}{C} > \frac{NC\Delta}{N - C} \qquad (7)$$

Given a value for Δ, the above 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.

Let us now suppose that expression (7) is satisfied, i.e., that the network operates in the bandwidth limited region with the bandwidth bound $M_{bw}^{(l)}$ the dominant one. Recall that $M_{bw}^{(l)}$ represents the total slot requirements for some channel, hence, under the non-uniform traffic scenario we are considering, it is possible for $M_{bw}^{(l)}$ to be significantly greater than $\frac{D}{C}$. Since, assuming that a near-optimal algorithm is available, the length of the final schedule will depend on $M_{bw}^{(l)}$, it is important that the receiver sets $\mathcal{R}_c$ be constructed so that the offered traffic is well balanced across all channels. This load balancing problem [9, 10] is a well-known and widely-studied $\mathcal{NP}$-complete problem. We will not consider this problem any further, but we will once more emphasize the importance of using some approximation scheme to effectively balance the traffic across the channels.

4 A Class of Schedules for *OSTL*

Let $\mathbf{A}$ be a collapsed traffic matrix, and $\mathcal{S}$ a schedule of length M satisfying the hardware and no-collision constraints (1) and (2), respectively. Consider now the order in which

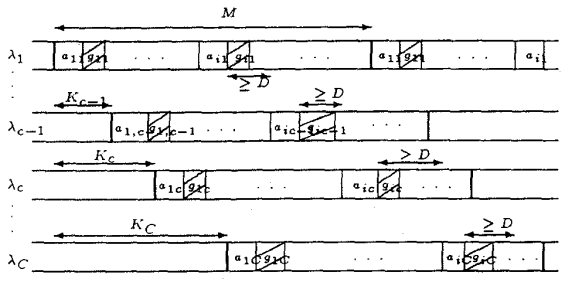

Figure 2: Schedule for a bandwidth limited network

the various transmitters are assigned slots within, say, channel λ_1, starting with some transmitter π_1. We will say that $s_1 = (\pi_1, \pi_2, \ldots, \pi_N)$ 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 above starts anew. 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:

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

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 algorithms for the class of schedules defined in (8). At this point, it is important that we distinguish between bandwidth and tuning limited networks, as different conditions of optimality apply to each case [8]. However, we have found that the two cases are in a sense dual of each other (see [8] for details), so we only discuss bandwidth limited networks here.

4.1 Bandwidth Limited Networks

We start by presenting an alternative 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 $(1, 2, \ldots, N)$ 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, in slots, between the start of a frame on channel λ_c and the start of the frame on the first channel, as in Figure 2. Note also that $K_1 = 0$.

Consider the transmissions on, say, channel λ_c, within a frame of M slots. Following the a_{1c} slots assigned to node 1, the next a_{2c} slots are assigned to node 2, unless this assignment does not allow the laser of 2 enough time to tune from λ_{c-1} to λ_c. In the latter case, channel λ_c has to remain idle for a number of slots before node 2 starts transmitting. In general, we 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 $(1, 2, \ldots, N)$, can now be formulated as an integer programming problem, to be referred to as *bandwidth limited OSTL (BW-OSTL)*. Note that constraints (10) and (11) in the formulation below correspond to the hardware constraints (1). The no-collision constraints (2) 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.

$$BW-OSTL: \quad \min_{g_{ic}, K_c} M = \max_c \left\{ \sum_{i=1}^{N} (a_{ic} + g_{ic}) \right\} \tag{9}$$

subject to:

$$K_c + \sum_{j=1}^{i-1} (a_{jc} + g_{jc}) \geq K_{c-1} + \sum_{j=1}^{i-1} (a_{j,c-1} + g_{j,c-1}) + a_{i,c-1} + \Delta \qquad c = 2, \ldots, C, \ i = 1, \ldots, N \tag{10}$$

$$M + \sum_{j=1}^{i-1} (a_{j1} + g_{j1}) \geq K_C + \sum_{j=1}^{i-1} (a_{jC} + g_{jC}) + a_{iC} + \Delta \qquad i = 1, \ldots, N \tag{11}$$

$$g_{ic}, K_c, M : \text{integers}; \ g_{ic} \geq 0 \ \forall \ i, c; \ K_1 = 0; \ K_c > K_{c-1} \ c = 2, \ldots, C \ ; \ M > K_C \tag{12}$$

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 sequence resulting in the smallest frame size. Furthermore, solving problem *BW-OSTL* is itself a hard task, as it is an integer programming problem with a non-linear objective function. Recall, however, that we are considering bandwidth limited networks. For these networks, the bandwidth bound (3) dominates, therefore, the lower bound on the schedule length is such that $M^{(l)} = M_{bw}^{(l)} > M_t^{(l)}$. The key observation which we will exploit in the following analysis is that, if a schedule of length $M^{(l)}$ exists, then at least one channel, say, channel λ_c, will never be idle; in terms of the above problem formulation, this schedule will be such that $g_{ic} = 0 \ \forall \ i$. It will be shown shortly 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 class (8).

4.1.1 A Sufficient Condition for Optimality

Let $\mathbf{A}$ be the collapsed traffic matrix of a bandwidth limited network, $M^{(l)}$ be the lower bound on any schedule for $\mathbf{A}$,

and define the *average slot requirement* as $a = \frac{M^{(l)}}{N}$. If $a_{ic} = a \; \forall \, i, c$, then an optimum length schedule is easy to construct; all of (10) – (12) will be satisfied by letting

$$K_c = (c-1)(a+\Delta) \; \forall c; g_{ic} = 0 \; \forall i, c; M = M^{(l)} = Na \quad (13)$$

The question that naturally arises then, is whether we can guarantee a schedule of $M^{(l)}$ slots when we allow non-uniform traffic. The answer is provided by the following lemma. 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 4.1 *Let* **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 the lower bound exists within the class (8) for any transmitter sequence, if the elements of* **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:*

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

Proof. See Appendix A. □

Lemma 4.1 provides an upper bound on the "degree of non-uniformity" of matrix **A** in order to guarantee a schedule of length equal to the lower bound. For $N = 100$, $C = 10$, and ignoring the term $\frac{\Delta}{M^{(l)}}$ [3], we get $\frac{\epsilon}{M^{(l)}/N} \approx .89$. Thus, the variation of elements a_{ic} around $\frac{M^{(l)}}{N}$ can be up to 8.9% to guarantee a schedule of length $M^{(l)}$. Our proof, however, is based on a worst case scenario; in general, we expect such an optimal schedule to exist for higher degrees of variation.

4.1.2 Scheduling Algorithm

We now develop an algorithm which, under the conditions of Lemma 4.1, produces schedules of length $M^{(l)}$. In fact, we shall shortly prove that 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 the transmissions on channel 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 the dominant one, that is $\sum_{i=1}^{N} a_{i1} = M^{(l)}$.

Algorithm *Make_Bandwidth_Limited_Schedule (MBLS)*, described in detail in Figure 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 (10). Doing so, however, may introduce large gaps into these channels, resulting in a sub-optimal schedule (refer to (9)). 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 (10) and (11) allow.

[3] In general, we expect the frame length to be much greater than Δ.

Algorithm *Make_Bandwidth_Limited_Schedule (MBLS)*
Channel λ_1 is assumed to be dominant. Also, references to channel λ_{c+1} when $c = C$ denote the next frame on λ_1.

```
1.  begin
2.      Set M = Σ_{i=1}^{N} a_{i1}
3.      Set K_1 and all gaps g_{i1} on λ_1 equal to 0
        // Begin Pass 1
4.      for c = 2 to C do
5.          for i = 1 to N do
6.              Schedule the a_{ic} slots at the earliest time
                such that (10) is satisfied between λ_c and λ_{c-1}
7.      // end of for c loop
        // End of Pass 1 – initial values to all g_{ic} have now
        been determined
8.      Let M' be the smallest integer satisfying (11)
9.      Set M = max{M, M'}
        // Begin Pass 2
10.     for c = C downto 2 do
11.         for i = N downto 1 do
12.             Shift the a_{ic} slots as much right as possible
                while maintaining (10) between λ_c and λ_{c+1}
13.             for j = i + 1 to N do
14.                 Shift the a_{jc} slots as much left as possible
                    while maintaining (10) between λ_c and λ_{c-1}
15.         // end of for i loop – the final values of gaps for
            this channel have now been determined
16.         Let M_c = Σ_{i=1}^{N} (a_{ic} + g_{ic})
17.         M = max(M, M_c)
18.     // end of for c loop – M is the final schedule length
19. // end of algorithm
```

Figure 3: Scheduling algorithm

That algorithm *MBLS* is correct follows from the fact that it constructs a schedule satisfying constraints (10) – (12). It is easy to verify that its running-time complexity is $\mathcal{O}(CN^2)$. We now state and prove its optimality properties.

Theorem 4.1 *Algorithm MBLS constructs a schedule of minimum length among the schedules that (a) are within the class (8) and the sequence of transmitters is* $(1, 2, \ldots, N)$, *(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.,* $g_{i1} = 0, i = 1, \ldots, N-1$*).*

Proof. See Appendix B. □

Corollary 4.1 (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. Under the conditions of Lemma 4.1, MBLS constructs an optimum length schedule.*

Proof. According to Lemma 4.1, there exists a schedule of length $M^{(l)}$ within the class defined by (8), such that the transmitter sequence is $(1, 2, \ldots, N)$. Since λ_1 is the dominant channel, any schedule of length $M^{(l)}$ is such that channel λ_1 is never idle. Therefore, because of Theorem 4.1, algorithm MBLS will construct such a schedule. □

5 Optimization Heuristic

We now develop a heuristic to obtain near-optimal schedules for arbitrary instances of *OSTL* and bandwidth limited networks. Recall that solving the *OSTL* problem involves solving $N!$ *BW-OSTL* problems, one for each possible transmitter sequence, and that we have no efficient algorithm for solving the most general version of *BW-OSTL*. Our approach then 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 (a new row is added to the collapsed traffic matrix $\mathbf{A}$). 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 (before the first node, between the first and second nodes, etc.). 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. Our second compromise has to do with the fact that we have no efficient algorithm for *BW-OSTL*. Thus, we let λ_1 be the dominant channel, and 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 4.1, we know that if a schedule of length equal to the lower bound exists for the given transmitter sequence, *MBLS* will find such a schedule. But if the optimal schedule has length greater than the lower bound, *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.

Our heuristic is described in Figure 4. Regarding its complexity, note that Step 2 will dominate. During the i-th 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 $\mathcal{O}(CN^4)$.

Bandwidth Limited Scheduling Heuristic (BLSH)

1. Relabel the channels such that:

$$M^{(l)} = \sum_{i=1}^{N} a_{i1} \geq \sum_{i=1}^{N} a_{i2} \geq \ldots \geq \sum_{i=1}^{N} a_{iC} \qquad (16)$$

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

2. Let $s^{(i-1)} = (\pi_1, \ldots, \pi_{i-1})$ 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 *MBLS* on each of the i permutations

$$(i, \pi_1, \ldots, \pi_{i-1}), (\pi_1, i, \pi_2, \ldots, \pi_{i-1}), \ldots,$$
$$(\pi_1, \ldots, \pi_j, i, \pi_{j+1}, \ldots, \pi_{i-1}), \ldots, (\pi_1, \ldots, \pi_{i-1}, i) \qquad (17)$$

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

Figure 4: Scheduling Heuristic

6 Numerical Results

We now consider four different algorithms for the *OSTL* problem and compare their performance: (1) algorithm *MBLS*, described in Figure 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 been labeled 1 through N in decreasing order of $\sum_{c=1}^{C} a_{ic}$; (2) algorithm *MTLS*, with the same labeling of both channels and transmitters; *MTLS* has not been described, but is very similar to *MBLS*, only targeted to tuning limited networks; (3) scheduling heuristic *BLSH*, described in Figure 4; (4) scheduling heuristic *TLSH* for tuning limited networks; this heuristic has not been described, but is very similar to *BLSH*.

Given a matrix $\mathbf{A}$, the lower bound $M^{(l)}$ on the schedule length can be obtained from (5). Let M be the actual length of a schedule for $\mathbf{A}$ produced by some scheduling algorithm. Quantity $\frac{M-M^{(l)}}{M^{(l)}}$ 100% then represents how far the length M of the schedule produced by the algorithm is from the lower bound. All figures in this section plot the above quantity against the number of nodes, N, for the four algorithms described here. Each point plotted represents the average of twenty randomly generated matrices $\mathbf{A}$ for the stated values of N, C, and Δ. The elements of each matrix $\mathbf{A}$ were chosen, with equal probability, among the integers 1 through 20.

In Figures 5 – 7 we show results for two values of the number of channels, namely $C = 5$ and $C = 20$ (additional results can be found in [8]). The number N of nodes within each figure takes values from C to 80. We also use three different values for Δ, $\Delta = 1, 4, 16$. For data rates of 1 Gigabits per second, and ATM cell sizes, these values of Δ correspond to transceiver tuning times of $424ns$, $1.7\mu s$, and $6.8\mu s$, respectively; the last two values are representative of current state of the art in optical transceiver technology [2].

We first observe that the two heuristics, *BLSH* and *TLSH*, always perform as good as, or better than the corresponding algorithms, *MBLS* and *MTLS*, respectively. However, this performance gain is achieved at the expense of higher computational complexity. The figures also confirm our intuition regarding the two regions of network operation, and justify the need for algorithms specially designed for each region. As we can see, *MBLS* and *BLSH* outperform their

counterparts within the bandwidth limited region, while the opposite is true within the bandwidth limited region. In addition, when the network operates well within the bandwidth limited region (i.e., for sufficiently large values of N), *BLSH*, and sometimes *MBLS*, construct schedules of length *equal* to the lower bound (similar observations can be drawn regarding the performance of *MTLS* and *TLSH* in the tuning limited region). This is an important result, as it establishes that the lower bound accurately characterizes the scheduling efficiency in this type of environment. 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 eliminate the effects of even large values of tuning latency. Finally, the fact that our algorithms deviate from the lower bound at the boundary between the tuning and bandwidth limited regions is not due to inefficiency inherent in the algorithms, rather, it is due to the fact that optimal schedules at the boundary of the two regions have length greater than the lower bound, as we proved in [8].

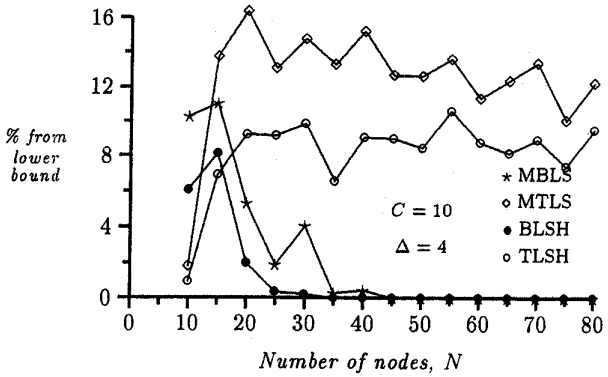

Figure 5: Algorithm comparison for $C = 10$ and $\Delta = 4$

7 Concluding Remarks

We have considered the problem of designing TDM schedules for arbitrary traffic demands in broadcast optical networks. Based on the insight provided by an appropriate new formulation of the 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.

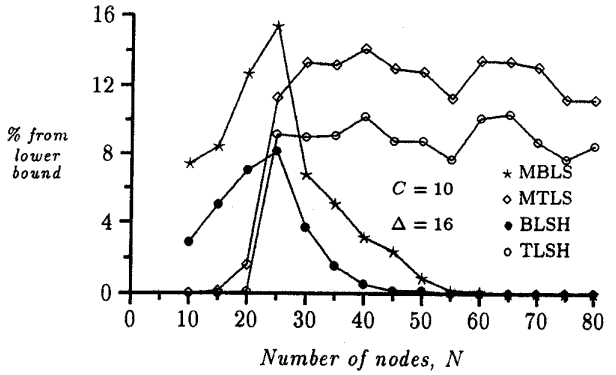

Figure 6: Algorithm comparison for $C = 10$ and $\Delta = 16$

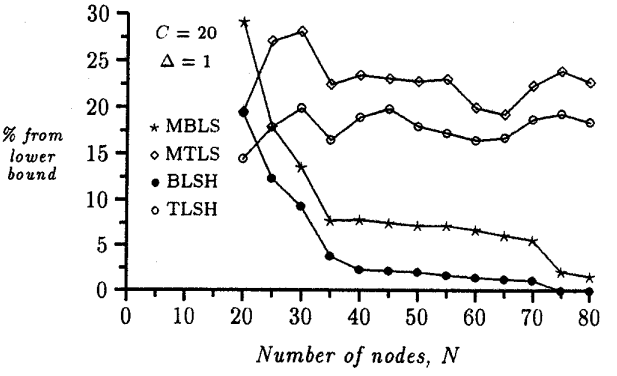

Figure 7: Algorithm comparison for $C = 20$ and $\Delta = 1$

References

[1] B. Mukherjee. WDM-Based local lightwave networks Part I: Single-hop systems. *IEEE Network Magazine*, pages 12–27, May 1992.

[2] P. E. Green. *Fiber Optic Networks*. Prentice-Hall, Englewood Cliffs, New Jersey, 1993.

[3] G. R. Pieris and G. H. Sasaki. Scheduling trabsmissions in WDM broadcast-and-select networks. *IEEE/ACM Transactions on Networking*, 2(2):105–110, April 1994.

[4] A. Aggarwal, A. Bar-Noy, D. Coppersmith, R. Ramaswami, and B. Schieber. Efficient routing and scheduling algorithms for optical networks. Technical Report RC 18967, IBM Research Report, 1994.

[5] M. Azizoglu, R. A. Barry, and A. Mokhtar. The effects of tuning time in banwidth-limited optical broadcast networks. In *IEEE INFOCOM '95*, pages 138–145.

[6] M. S. Borella and B. Mukherjee. Efficient scheduling of nonuniform packet traffic in a WDM/TDM local lightwave network with arbitrary transceiver tuning latencies. In *IEEE INFOCOM '95*, pages 129–136.

[7] T. Gonzalez and S. Sahni. Open shop scheduling to minimize finish time. *Journal of the Association for Computing Machinery*, 23(4):665–679, Oct 1976.

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

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

[10] M. R. Garey, R. L. Graham, and D. S. Johnson. Performance guarantees for scheduling algorithms. *Operations Research*, 26:3–21, Jan 1978.

A Proof of Lemma 4.1

In proving Lemma 4.1 we will make use of the following result whose proof is straightforward and is omitted:

Lemma A.1 *If constraints (14) hold, then for all* $\mathcal{P} \subseteq \{1,\ldots,N\}$ *with* $|\mathcal{P}| = n$, *and any two channels* λ_{c_1}, λ_{c_2}:

$$\left| \sum_{i\in\mathcal{P}} a_{i,c_1} - \sum_{i\in\mathcal{P}} a_{i,c_2} \right| \leq N\epsilon \qquad (18)$$

We are now ready to prove Lemma 4.1. Although the proof refers to the problem formulation in (9) – (12), it does not depend on the actual transmitter sequence. As a result, it holds for any transmitter sequence, not just the $(1,2,\ldots,N)$ sequence implied in (9) – (12).

Proof (of Lemma 4.1). By our hypothesis, we have that $\sum_{i=1}^{N} a_{ic} \leq M^{(l)}\ \forall\ c$. For the proof we consider a worst case scenario, under which the total slot requirement on *each* channel is equal to the lower bound: $\sum_{i=1}^{N} a_{ic} = M^{(l)}\ \forall\ c$. A schedule of length $M^{(l)}$ under such a scenario would ensure a schedule of length $M^{(l)}$ for the case 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 $M^{(l)}$, and because of the above worst case assumption, we are seeking a solution to problem *BW-OSTL* such that $g_{ic} = 0\ \forall\ i,c$ (refer also to the objective function (9)). We can then rewrite constraints (10) and (11), respectively, as

$$K_c - K_{c-1} \geq \left(\sum_{j=1}^{i-1} a_{j,c-1} - \sum_{j=1}^{i-1} a_{jc}\right) + a_{i,c-1} + \Delta \qquad c = 2,\ldots,C,\ i = 1,\ldots,N \qquad (19)$$

$$M - K_C \geq \left(\sum_{j=1}^{i-1} a_{jC} - \sum_{j=1}^{i-1} a_{j1}\right) + a_{iC} + \Delta \qquad (20)$$

Hence, Lemma A.1 guarantees that choosing $K_c - K_{c-1} = N\epsilon + \frac{M^{(l)}}{N} + \epsilon + \Delta, c = 2,\ldots,C$, satisfies constraints (19) and (12). Noting that $K_1 = 0$, we can set:

$$K_c = (c-1)\left((N+1)\epsilon + \frac{M^{(l)}}{N} + \Delta\right) \qquad c = 1,\ldots,C \qquad (21)$$

Finally, it is easy to check that letting $M = M^{(l)}$ ensures that (20) is also satisfied. □

B Proof of Theorem 4.1

Proof (of Theorem 4.1). Let *Sched(c)* denote the frame of the schedule on channel λ_c starting with the first slot in which transmitter 1 transmits on channel λ_c. *Sched(C+1)* refers to the next frame on channel λ_1. Once the schedule length M and gaps $g_{ic}, i = 1,\ldots,N-1$, are known, gap g_{Nc} after the last transmitter is uniquely determined. Therefore, any reference to "gaps" in what follows does not include this last gap on each channel. Let OPT denote the optimal schedule length under the assumptions of Theorem 4.1. We will prove that $OPT \geq M$, hence proving that $OPT = M$. To do so, we trace through the algorithm as it computes M and show that $OPT \geq M$ at every step of the algorithm.

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 channel λ_1 after its transmission on channel λ_C (in fact this is exactly what constraint (11) tries to capture). Therefore $OPT \geq M$ at the end of Pass 1.

In Pass 2, channels as well as transmitters are processed in reverse order, and the algorithm tries to compact the gaps $g_{ic}, i = 1,\ldots,N-1, c = 2,\ldots C$, as much as possible. We show that once the gaps on a channel λ_c have been compacted by Pass 2 of the algorithm above, it is not possible to compact them any further to reduce the schedule 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 channel λ_{c+1} cannot be compacted any further), and that transmitters $i+1,\ldots,N$ are optimally scheduled on channel λ_c (i.e., that the gaps $g_{i+1,c}\cdots g_{N-1,c}$ cannot be compacted any further; note that gap g_{Nc} is not considered), 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 $a_{jc}, j = i+1,\ldots,N$, 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 hence channel 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 (this will happen if at the end of the algorithm $M > \sum_{i=1}^{N} a_{i1}$). □