ABSTRACT

SIVARAMAN, VIJAY. TDM Schedules for Broadcast WDM Networks with Arbitrary Transceiver Tuning Latencies. (Under the direction of Professor George Rouskas.)

We consider the problem of scheduling packet transmissions in a broadcast, single-hop WDM network. Tunability is provided only at one end, namely, at the transmitters. 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 to 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 offset the effects of even large values of the tuning latency.

TDM SCHEDULES FOR BROADCAST WDM NETWORKS WITH ARBITRARY TRANSCEIVER TUNING LATENCIES

by

VIJAY SIVARAMAN

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Master of Science

COMPUTER SCIENCE

Raleigh

1995

APPROVED BY:

Chair of Advisory Committee

BIOGRAPHY

Vijay Sivaraman was born in India on October 17, 1972. After finishing his schooling at the Delhi Public School, he joined the Indian Institute of Technology at Delhi in 1990 where he received the Bachelor of Technology degree in Computer Science and Engineering. He joined the Computer Science department at North Carolina State University in August 1994.

ACKNOWLEDGEMENTS

I would like to express my most sincere gratitude to Professor George Rouskas for having introduced me to the exciting world of Optical Networking and for having guided me at every step of this intellectual adventure. Many thanks to his limitless energy and constant enthusiasm, without which this thesis would not have been possible.

I would also like to thank Professors Harry Perros, Ioannis Viniotis and Carla Savage for their constant support throughout my work, and their timely guidance and advice.

To my parents and my brother, and all my friends who have been a source of constant encouragement, a deep-hearted thank you.

Contents

List of Figures

Chapter 1

Introduction

1.1 Optical Networks

Recent advances in lightwave technology have led to the design of third generation all-optical networks that exploit the unique properties of single-mode fiber, especially its enormous information carrying capacity. However, in order to effectively tap into the huge bandwidth of the optical medium, measured in the order of tens of THz over the low-loss windows at $1.3\mu m$ and $1.5\mu m$, the network architecture must overcome the so-called electro-optic bottleneck [13]. Today's electronics typically operate at rates of a few Gigabits per second, and can drastically limit the throughput available to the network users, unless the network architecture supports some form of concurrency.

Wave Division Multiplexing (WDM) divides the wavelength spectrum of the fiber into a number of independent, non-overlapping channels operating at a data rate accessible by the attached stations. Thus, WDM architectures have the ability to support multiple simultaneous communication paths over a single fiber, each on a different wavelength. As a result, WDM networks can deliver an aggregate throughput that grows with the number of wavelengths deployed, and can be in the order of Terabits per second.

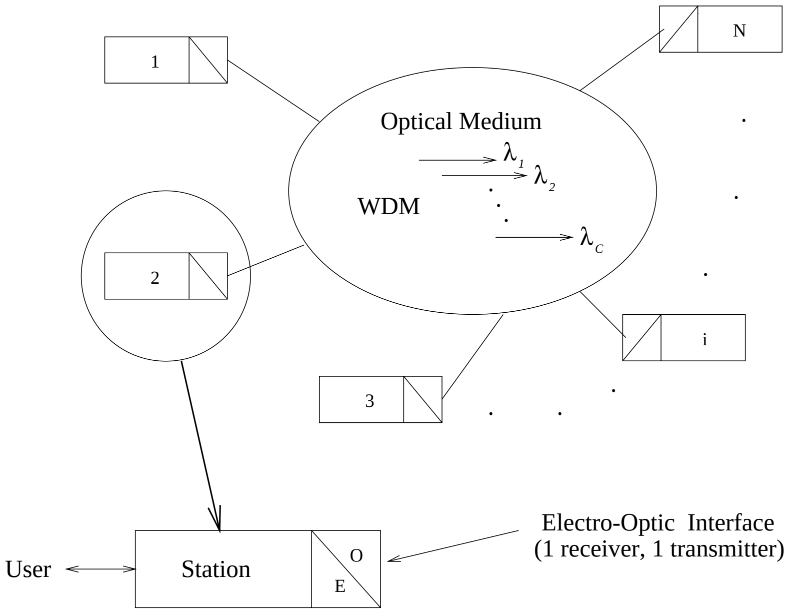


Figure 1.1: A Single-hop WDM Lightwave Network

1.2 Single Hop Networks

Our focus in this thesis is on a WDM network architecture known as the *single-hop* architecture [16]. It consists of N stations interconnected over a passive broadcast optical medium that can support C wavelengths $\lambda_1, \ldots, \lambda_C$, as in Figure 1.1.

In general, $C \leq N$. Each station taps into the optical medium through an electro-optic interface consisting of one transmitter and one receiver. Each wavelength can be considered as a channel operating at data rates accessible by the electronic interfaces at each station. Since transmissions on different wavelengths do not interfere with each other, the multiplexing of several channels provides the concurrency necessary to exploit the vast bandwidth of the optical medium.

The network operates in a broadcast-and-select mode; packets transmitted on wavelength λ_c, $c = 1 \ldots C$ are broadcast over the medium, but are only received by stations with a receiver listening on wavelength λ_c. This can be achieved by using a passive broadcast physical topology, such as a star, a unidirectional bus, or a tree. If necessary, optical amplifiers may be deployed to maintain adequate signal levels.

1.3 Fixed and Tunable Transceivers

Single-hop networks are *all-optical* in nature. In other words, they provide complete optical paths between any source-destination pair, and no conversion between electronics and photonics takes place within the network. For a successful packet transmission, the transmitter of the source and the receiver of the destination must operate on the same wavelength. Thus, tunable transmitters and/or receivers are required to provide full connectivity among the stations. A *fixed* transmitter is a laser that can only transmit on a certain wavelength. A *tunable* transmitter is one that can tune to, and transmit on several wavelengths, one at a time. Fixed and tunable receivers are distinguished in a similar way. Depending on the transceiver tunability characteristics, a single-hop network is classified as one with tunable transmitters and fixed receivers (TT-FR), or with fixed transmitters and tunable receivers (FT-TR), or with tunable transmitters and tunable receivers (TT-TR).

Tunable transceivers (lasers and/or optical filters) with the ability to tune fast across all available channels are crucial to the design of single-hop networks. Such devices do exist today; however, their capabilities are limited in terms of both tunability range and speed. Work in improving the performance characteristics of tunable devices proceeds at a fast pace; but the ideal device, one that can tune across the useful optical spectrum in sub-microsecond times [12] remains elusive, and, barring a technological breakthrough, will remain so at least for the foreseeable future. We show, however, that careful network design can mask the effects of non-ideal devices, making it possible to build single-hop WDM networks using *currently available* tunable optical transceivers.

1.4 Thesis Organization

The thesis is organized as follows. Chapter 2 contains some background, including discussion of related work on scheduling packet transmissions in WDM/TDM single-hop networks with tuning latency. In Chapter 3 we describe our system and traffic model, and formally introduce the concept of a schedule. In Chapter 4 we formulate the problem of finding schedules of minimum length, and show that it is $\mathcal{NP}$-complete; we also derive lower bounds on the schedule length, and discuss the effect of the dominant bound on the network operation. We introduce a special class of schedules in Chapter 5, and proceed to develop scheduling algorithms which, under certain conditions, construct optimal schedules within this class. Scheduling heuristics are developed in Chapter 6, and in Chapter 7 we present some numerical results. We then summarize our work, and point out directions for future research in Chapter 8.

Chapter 2

Background and Related Work

In the recent past, significant effort has been devoted to the design and study of protocols for single-hop WDM networks. In general, the various design approaches take one of two directions, depending on their assumptions regarding the relative values of packet transmission time and transceiver tuning times. In order to characterize these approaches, 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. This assumption is reasonable for communication environments with data rates in the order of a few hundreds Megabits per second, and relatively large packet sizes. For instance, with a 155 Megabits per second rate, 6000 bit packets, and 1 μs tuning time, the normalized tuning latency $\delta \approx 0.026 \ll 1$. 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 [18, 19, 3, 5, 6, 14] which has been geared towards improving the delay and throughput characteristics of the network under various traffic assumptions, completely ignoring transceiver tuning times.

With the current trend, however, being towards ever increasing data rates (Gigabits per second and beyond) and diminishing packet sizes (e.g., 53-byte ATM cells), emerging communication environments are such that the tuning times of even the fastest available tunable optical devices dominate over packet transmission times, making δ comparable to, and even greater than, 1. 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 Δ as

$$\Delta \quad = \quad \lceil \delta \rceil \tag{2.1}$$

Observe that, 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. We will use the term *tuning slots* in future references to parameter Δ.

Let $\Delta \geq 1$ be the number of tuning slots of the system under consideration. A straightforward approach to make the tuning latency transparent to higher level protocols, would be to equip each node with $\Delta + 1$ transceivers. A node would then use transceiver $t, t = 0, \ldots, \Delta$, in slots $t + m(\Delta + 1), m = 0, 1, 2, \ldots$, only. This configuration would, in effect, appear to higher level protocols as a single transceiver that can tune infinitely fast between channels. Its obvious disadvantages, however, including the cost of hardware and the complexity of managing and coordinating packet transmissions/receptions from multiple transceivers, especially as $\Delta > 1$, make this an unattractive solution. A design similar in concept, but oriented towards circuit switched traffic, can be found in [15].

If no extra hardware is used, minimizing the effects of transceiver tuning times on network performance is possible only through specially designed protocols. In [8], for instance, the TDMA scheme considered is such that the frame is divided into transmitting and tuning periods. 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 to be ready for the next transmitting period. The objective is to minimize the number of tuning periods within the frame. The MaTPi protocol [20], on the other hand, is a reservation based protocol that can be used by stations to reserve, in real time, a time slot that is Δ slots in the future.

Another design approach, which we pursue in this work, is based on the observation that, when the number of stations, N, in the network is greater than the number of available wavelengths, C, at most C stations may be transmitting at any given slot. The remaining 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 (at least partially) be overlapped with transmissions by other stations, keeping channel utilization at high levels. The objective, then, is to design schedules of minimum length, given a traffic demand matrix. In [17, 1] 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 $\delta \leq 1$. The main contribution of [2] was to identify, in terms of system parameters N, C, and $\delta \leq 1$, a region of operation for the network such that the inefficiency due to the tuning latency can be completely eliminated through a simple scheduling algorithm. Our work is more general, as it considers arbitrary traffic demands and arbitrary values of δ; furthermore, we develop scheduling algorithms which guarantee that, within certain regions of operation, the tuning latency has no effect on network performance, thus extending the results of [2] to values of $\delta > 1$.

The problem of scheduling non-uniform traffic under arbitrary tuning latencies has been previously studied in [4], where a scheduling heuristic was presented and shown to produce good results. There are significant differences between the work in [4] and ours, however. In contrast to [4] where one heuristic is used throughout, we make the fundamental observation that, depending on the traffic matrix and system parameters N, C, and δ, the network can be operating in one of two distinct regions. We then develop two scheduling algorithms, one for each region, which we prove to be optimal under certain conditions; further, we demonstrate that an algorithm optimal for one region performs sub-optimally when applied to a network operating in the other region. We also present heuristics (again one for each region) that are quite different from the one in [4], and which are based on the intuition provided by an appropriate formulation of the scheduling problem.

The next chapter discusses the system and traffic model, and defines formally the notion of a transmission schedule. It lays the groundwork for a formal definition of the scheduling problem, and defines various terms which will be used throughout the thesis.

Chapter 3

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 [1], $\lambda_1, \ldots, \lambda_C$. In general, $C \leq N$. Without loss of generality, we only consider tunable-transmitter, fixed-receiver (TT-FR) networks; all of our results can be easily adapted to fixed-transmitter, tunable-receiver systems.

Each tunable transmitter can be tuned to, and transmit on, any and all wavelengths $\lambda_c, c = 1, \ldots, C$. The fixed receiver at station j, on the other hand, is assigned wavelength $\lambda(j) \in \{\lambda_1, \ldots, \lambda_C\}$. If the number of channels, C, is equal to the number of nodes, N, then each receiver is assigned a unique wavelength. When $C < N$, however, a single wavelength may be assigned to a number of receivers. We define $\mathcal{R}_c$ as the set of receivers sharing wavelength λ_c:

$$\mathcal{R}_c = \{j \mid \lambda(j) = \lambda_c\}, \qquad c = 1, \ldots, C \tag{3.1}$$

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* $N \times C$ traffic matrix $\mathbf{A} = [a_{ic}]$. Element a_{ic}

[1] The terms "wavelength" and "channel" will be used interchangeably throughout this thesis.

of the collapsed matrix represents the number of slots to be assigned to source i for transmissions on channel λ_c:

$$a_{ic} = \sum_{j \in \mathcal{R}_c} d_{ij}, \quad i = 1, \ldots, N, \quad c = 1, \ldots, C \tag{3.2}$$

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 [2]. We also let D denote the total traffic demand, across all source-destination pairs:

$$D = \sum_{i=1}^{N} \sum_{j=1}^{N} d_{ij} = \sum_{i=1}^{N} \sum_{c=1}^{C} a_{ic} \tag{3.3}$$

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, therefore, it will be left unspecified.

Finally, observe that, 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 the next chapter.

3.1 Transmission Schedules

In the WDM environment we are considering, a simultaneous transmission by two or more stations on the same channel results in a *collision.* To avoid packet loss due to collisions, some form of coordination among transmitting sources is necessary

[2] This assumption is reasonable, especially 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.

[12]. 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 wavelength λ_c. Exactly a_{ic} slots must be assigned to the source-channel pair (i, λ_c), as specified by the collapsed matrix **A**. However, 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 *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, **A**, we only consider non-preemptive schedules.

Formally, 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 source has exactly one laser which needs Δ slots to tune between channels, all time intervals $[\tau_{ic} \Leftrightarrow 1, \tau_{ic} + a_{ic} + \Delta \Leftrightarrow 1)$ must be disjoint [3], yielding a set of *hardware constraints* on schedule $\mathcal{S}$:

$$[\tau_{ic} \Leftrightarrow 1, \tau_{ic} + a_{ic} + \Delta \Leftrightarrow 1) \bigcap [\tau_{ic'} \Leftrightarrow 1, \tau_{ic'} + a_{ic'} + \Delta \Leftrightarrow 1) = \phi \quad \forall\, c \neq c', \quad i = 1, \ldots, N \tag{3.4}$$

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*:

$$[\tau_{ic} \Leftrightarrow 1, \tau_{ic} + a_{ic} \Leftrightarrow 1) \bigcap [\tau_{i'c} \Leftrightarrow 1, \tau_{i'c} + a_{i'c} \Leftrightarrow 1) = \phi \quad \forall\, i \neq i', \quad c = 1, \ldots, C \tag{3.5}$$

A non-preemptive schedule $\mathcal{S}$ is *admissible* if and only if $\mathcal{S}$ satisfies both the hardware and the no-collision constraints.

Consider now transmitter i and an admissible schedule $\mathcal{S} = \{\tau_{ic}\}$. Based on the above discussion, transmitter i can be in one of three states during a slot τ.

1. *Transmitting state*, if, according to the schedule, i is assigned to transmit on some channel λ_c. Transmitter i is in the transmitting state in slots τ_{ic} through

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

$\tau_{ic} + a_{ic} \Leftrightarrow 1$, $c = 1, \ldots, C$.

2. *Tuning state.* Immediately after completing its transmission on channel λ_c, i instructs its laser to tune to the next channel, say, channel $\lambda_{c'}$, and will be in the tuning state for exactly Δ slots.

3. *Idle state.* The laser at station i will be ready to transmit on channel $\lambda_{c'}$ at the beginning of slot $\tau = \tau_{ic} + a_{ic} + \Delta$. If, however, $\tau_{ic'} > \tau$, i will simply wait for slot $\tau_{ic'}$ before it starts transmitting on channel $\lambda_{c'}$. We say that i is idle in these $\tau_{ic'} \Leftrightarrow \tau$ slots.

Similarly, we say that channel λ_c is *busy* in slot τ if some station has been assigned to transmit on λ_c in that slot (because of the no-collision constraint, there will be exactly one such station), and *idle*, otherwise. Channel idling results in wasted bandwidth; one of the contributions of this work is to show that it is possible to properly dimension the network to minimize channel idling.

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. Note that an optimum length schedule does not preclude the existence of slots with idle channels (see also Figure 3.1), but a schedule in which no channel is ever idle is necessarily an optimum length schedule.

Figure 3.1 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. Observe 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. Also, the fact that channel λ_2 is idle in slots 12 and 18, and channel λ_1 is idle in slot 18, does not affect the overall length of the schedule.

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

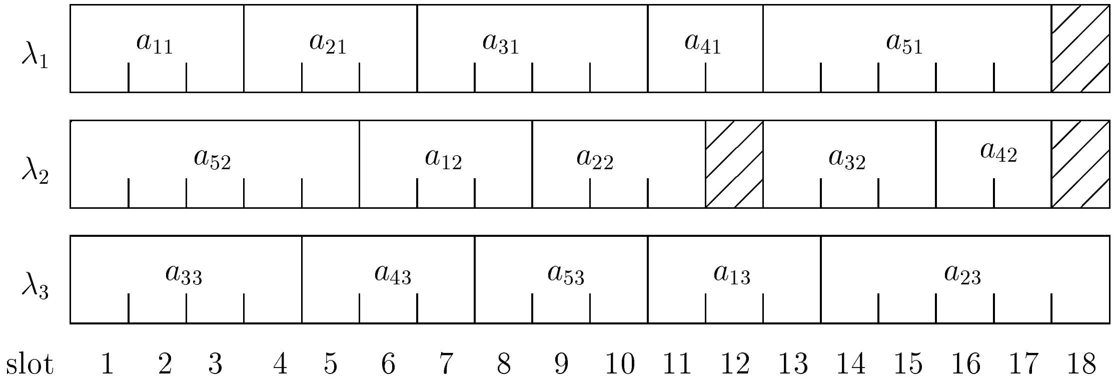


Figure 3.1: An optimum length schedule for a network with $N = 5$, $C = 3$, and $\Delta = 2$.

identical copy of the schedule as it repeats in time. If the traffic parameters d_{ij} are derived based on the behavior and required quality of service of longer term (relative to a packet transmission time) connections between the various source-destination pairs, we expect the schedule to repeat until a change in traffic demands triggers an update of the demand matrix. Under the gated service discipline scenario discussed above, 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, then a period of Δ tuning slots is necessary to allow transmitters to tune to their initial channels; no transmissions are possible during this tuning period. On the other hand, if the schedule repeats over time, this tuning period can be overlapped with transmissions in the *previous* frame of the schedule, possibly resulting in a smaller overall schedule length [4]. 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 Δ slots. Furthermore, 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.

[4] 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.

Unless otherwise specified, from now on the term “schedule” will be used as an abbreviation for “admissible non-preemptive schedule”.

Now that the notion of a schedule has been formalised, the next chapter goes on to define formally the scheduling problem, and to prove that it is $\mathcal{NP}$-complete. It also derives lower bounds for the schedule length, and identifies two distinct regions of network operation based on which of these lower bounds is dominant.

Chapter 4

Schedule Optimization and Lower Bounds

The length, M, of a schedule for a traffic matrix $\mathbf{D}$, is a measure of both the packet delay incurred while transmitting $\mathbf{D}$, and the system-wide throughput (the average number of packets transmitted per slot, $\frac{D}{M}$). Our objective then, is to determine an optimum length schedule to transmit the demand matrix $\mathbf{D}$, 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 concisely as:

Problem 4.1 (PSTL) *Given the number of nodes, N, the number of available wavelengths, C, the traffic demand matrix, $\mathbf{D} = [d_{ij}]$, and the tuning slots, Δ, find a schedule of minimum length for matrix $\mathbf{D}$.*

Problem *PSTL* can be logically decomposed into two subproblems:

- the 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
- 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 4.2 (OSTL) *Given the number of nodes, N, the number of available wavelengths, C, the collapsed traffic demand matrix, $\mathbf{A} = [a_{ic}]$, the tuning slots, $\Delta \geq 0$, and an overall deadline, $M > 0$, is there a schedule $\mathcal{S} = \{\tau_{ic}\}$ that meets the deadline, in other words, is there a schedule of length at most M, satisfying constraints (3.4) and (3.5)?*

As stated, *OSTL* is a generalization of the non-preemptive open-shop scheduling (*OS*) problem studied in [11] [1]; it reduces to the latter when we let $\Delta = 0$. It was shown in [11] that problem *OS* is $\mathcal{NP}$-complete when the number of wavelengths is $C \geq 3$. But for $C = 2$, problem *OS* admits a polynomial-time solution, and algorithm *OPEN_SHOP* was developed in [11] that constructs an optimum length *OS* schedule in time linear in the number of nodes, N.

Drawing upon the results of [11], we now prove the following theorem, which confirms our intuition that *OSTL* is in a sense more difficult than *OS*. Furthermore, it implies that a polynomial-time algorithm for *OSTL*, and consequently for *PSTL*, is unlikely to be found.

Theorem 4.1 *OSTL is $\mathcal{NP}$-complete for any fixed $C \geq 2$.*

Proof. See Appendix A. □

We now derive lower bounds for problems *PSTL* and *OSTL*, and discuss their implications.

4.1 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*

[1] In the terminology of [11], C is the number of processors and N is the number of jobs. Each job consists of C tasks; the c-th task, $c = 1, \ldots, C$, of job i requires a_{ic} processing time, and is to be processed by processor c.

bound:

$$M_{bw}^{(l)} \;=\; \max_{1 \le c \le C} \left\{ \sum_{i=1}^{N} a_{ic} \right\} \;\ge\; \frac{D}{C} \tag{4.1}$$

Note that the term in the brackets depends on the assignment of receive wavelengths to the nodes (i.e., the sets $\mathcal{R}_c$); the rightmost term, however, depends only on the *total* traffic demand, D, and is a lower bound on $PSTL$ independently of the actual elements d_{ij} of the demand matrix $\mathbf{D}$. Expression (4.1) 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 can obtain a different lower 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 wavelengths [2]. We call this the *tuning bound*:

$$M_{t}^{(l)} \;=\; \max_{1 \le i \le N} \left\{ \sum_{c=1}^{C} a_{ic} \right\} + C\,\Delta \;=\; \max_{1 \le i \le N} \left\{ \sum_{j=1}^{N} d_{ij} \right\} + C\,\Delta \;\ge\; \frac{D}{N} + C\,\Delta \tag{4.2}$$

The tuning bound is independent of the assignment of receive wavelengths to the nodes, and only depends on the system 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 now obtain the overall lower bound as

$$M^{(l)} \;=\; \max\left\{ M_{bw}^{(l)}, M_{t}^{(l)} \right\} \tag{4.3}$$

This overall bound is minimized when

$$\frac{D}{C} \;=\; \frac{D}{N} + C\,\Delta \quad \Leftrightarrow \quad \frac{D}{C} \;=\; \frac{N\,C\,\Delta}{N \Leftrightarrow C} \tag{4.4}$$

It is interesting to note that the quantity $\frac{NC\Delta}{N-C}$ is independent of the demand matrix, and as such it characterizes the network under consideration. We will call this quantity the *critical length*. Now, the relationship (4.4) between the minimum bandwidth bound, $\frac{D}{C}$, and the critical length is a fundamental one, and represents the point at which wavelength concurrency balances the tuning latency. Indeed, if

[2] Recall that we have assumed that $a_{ic} > 0$ for all i and c.

a schedule has length equal to the critical length, because of (4.4) it is such that exactly C (respectively, $N \Leftrightarrow C$) 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 in the following section.

4.2 Bandwidth Limited vs. Tuning Limited Networks

To get further insight on (4.4), let us consider the case of uniform traffic, whereby each source has $\beta \geq 1$ packets for each possible destination:

$$d_{ij} \;=\; \beta \;\geq 1 \quad \forall\, i,j \quad \Rightarrow \quad D \;=\; \beta N^2 \qquad \text{(integer } \beta\text{)} \tag{4.5}$$

This is a generalization of the *all-to-all* schedules studied in [17, 2], where the value of β was taken equal to 1. Substituting this value of D into (4.4) we get

$$\frac{\beta\, N^2}{C} \;=\; \beta\, N \,+\, C\,\Delta \;\;\Leftrightarrow\;\; \frac{\beta\, N^2}{C} \;=\; \frac{N\; C\;\Delta}{N \Leftrightarrow C} \tag{4.6}$$

In [17, 2] the quadratic equation (4.6) 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; one could then solve (4.6) to obtain an optimal value for β, which we will denote with $\beta^\star$; in general, $\beta^\star$ may not be an integer.

$$\beta^\star \;=\; \frac{C^2}{N(N \Leftrightarrow C)}\,\Delta \tag{4.7}$$

Suppose now that we choose $\beta < \beta^\star$ in (4.5); for simplicity, also let $N = kC$, so that the traffic demand can be perfectly balanced across the channels. In this case, 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 [3]. Since the total traffic demand is βN^2 and $\beta < \beta^\star$, the throughput achievable under such a schedule is

$$\frac{\beta N^2}{\beta N + C\Delta} \quad < \quad C \tag{4.8}$$

As we can see, the larger the value of β the higher the throughput; once the value of β has increased beyond $\beta^\star$, 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, either through the tuning bound $\beta N + C\Delta$, or through the bandwidth bound $\frac{\beta N^2}{C}$. But this length is a measure of packet delay, and cannot be increased beyond a certain level perceived as acceptable by the various higher layer applications. Within the family of matrices described in (4.5) therefore, 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. One might have to settle for less than 100% utilization, however, if satisfying the delay requirements would mean choosing $\beta < \lceil \beta^\star \rceil$. It is in these situations that advances in optical device technology would really make a difference [4]. From (4.7) we see 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 (see also (4.8)). Alternatively, according to (4.7), the same effect could be achieved by employing fewer wavelengths, a larger number of nodes, or a combination of the two.

The above observations are of general nature, applying to non-uniform demand matrices as well. In general, 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

[3] As we shall see in the following chapter, when C is a divisor of N, it is always possible to construct an optimal schedule (in this case, a schedule of length equal to the tuning bound) for uniform traffic demands.

[4] In contrast, the conclusion in [2] was that further advances in device technology would have negligible impact. This conclusion however, was due to the fact that only the case $\Delta = 1$ was considered there.

- *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 of the schedule is determined by the tuning bound in (4.2), 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 we saw that this can be accomplished by selecting $\beta = \lceil \beta^\star \rceil$. But the effect of choosing such a value for β is to make the bandwidth bound greater than the critical length in (4.6). In the general case (non-uniform traffic matrix $\mathbf{D}$) we would like to make the bandwidth bound in (4.4) greater than the critical length:

$$\frac{D}{C} \quad > \quad \frac{NC\Delta}{N \Leftrightarrow C} \tag{4.9}$$

Given a value for Δ, and some information about the delay requirements of higher layer applications, expression (4.9) 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 (4.9) for a given system, in the following we develop scheduling algorithms and heuristics for both regions of network operation.

Let us now suppose that expression (4.9) 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 $M^{(l)}$ represents the total slot requirements for some channel, hence, under the non-uniform traffic scenario we are considering, it is possible for $M^{(l)}$ 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 $M^{(l)}$, it is extremely important that the receiver sets $\mathcal{R}_c$ be constructed so that the offered traffic is well balanced across all channels [5]. This load balancing problem [7] is a well-known and widely-studied $\mathcal{NP}$-complete problem (refer also to the *PARTITION* problem in Appendix A), and several heuristics (such as the one in [9] which guarantees a performance of at most 1.22 times away from the optimal) as well as polynomial approximation algorithms have been derived for it. As such, 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, in addition to the heuristics presented here for the *OSTL* problem.

Now that we have identified the two operating regions of the network, we go on to define a special class of schedules in the next chapter. We derive algorithms (one for each of the two regions) which under certain conditions give optimal schedules within this class. We also show that optimal schedules are very difficult to obtain at the boundary between the bandwidth limited and the tuning limited region.

[5] Recall that constructing the sets $\mathcal{R}_c$ was the first of two subproblems into which problem *PSTL* was decomposed; the second being, of course, problem *OSTL*.

Chapter 5

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 (3.4) and (3.5), 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 $s_1 = (\pi_1, \pi_2, \ldots, \pi_N)$ 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 [1]. Similarly, we will say that $v_1 = (\lambda_{\pi_1}, \lambda_{\pi_2}, \ldots, \lambda_{\pi_C})$ 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:

$$s_c \quad = \quad (\pi_1, \pi_2, \ldots, \pi_N) \qquad c = 1, \ldots, C \tag{5.1}$$

It is easy to see that the class of schedules defined in (5.1) is equivalent to the class

[1] Note that, since the schedule repeats over time, *any* contiguous chunk of M slots constitutes a frame. Furthermore, frames on the various channels starting with the transmissions of, say, node π_1, will not be aligned in time (refer also to Figure 5.1).

of schedules such that the channel sequences (with λ_{π_1} as the first channel) are the same for all nodes [2]:

$$v_i \quad = \quad (\lambda_{\pi_1}, \lambda_{\pi_2}, \ldots, \lambda_{\pi_C}) \qquad i = 1, \ldots, N \tag{5.2}$$

Our examination of this class of schedules is motivated by several factors. First, the *OPEN_SHOP* algorithm for the *OS* problem with $C = 2$ channels [11] produces optimal schedules within this class. Secondly, for a uniform collapsed traffic matrix (i.e., $a_{ic} = a \; \forall \; i, c$), optimal schedules within this class do exist for the $OSTL$ problem. More importantly, 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. As a result, for schedules in this class, we have been able to prove certain optimality properties and derive scheduling algorithms, and have obtained optimal or near-optimal schedules for a wide range of the system parameters N, C, and Δ.

We now proceed to derive sufficient conditions for optimality, as well as algorithms for constructing optimal schedules within the class of schedules defined in (5.1) and (5.2). In our study, we distinguish between bandwidth limited and tuning limited networks. As we shall shortly show, different conditions of optimality apply to each of the two cases; thus, scheduling algorithms specially designed for bandwidth limited networks perform sub-optimally on tuning limited networks, and vice versa.

5.1 Bandwidth Limited Networks

We start by presenting an alternative formulation of problem $OSTL$, applicable to bandwidth limited schedules within the class (5.1). This new formulation will provide insight into the design of good scheduling algorithms.

Let $\mathcal{S}$ be a schedule of length M for a bandwidth limited 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

[2] By "equivalent" we mean that if a schedule is such that the transmitter sequence is the same for all channels, then the channel sequence is the same for all transmitters, and vice versa.

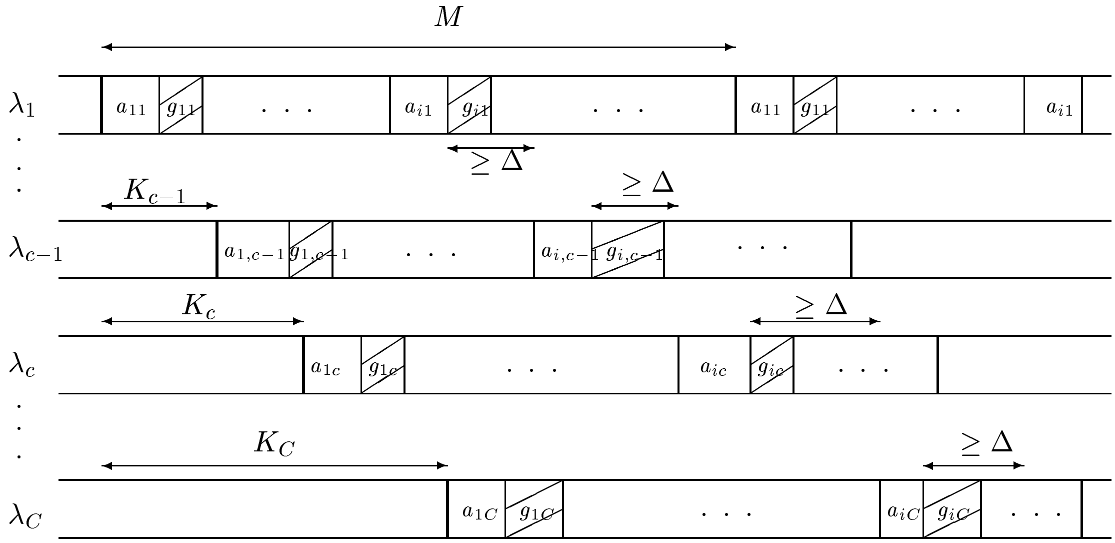


Figure 5.1: Schedule for a bandwidth limited network

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; this is illustrated in Figure 5.1. Note also that $K_1 = 0$.

Consider now the transmissions on, say, channel λ_c, within a frame of M slots. Following the a_{1c} slots assigned to transmitter 1, the next a_{2c} slots are assigned to transmitter 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 will 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.

Based on the above discussion, the problem of finding an optimum length schedule such that (a) the schedule is within the class defined in (5.1) and (b) the transmitter sequence is $(1, 2, \ldots, N)$, can be formulated as an integer programming problem, to be referred to as *bandwidth limited OSTL (BW-OSTL)*, as follows.

$$BW \Leftrightarrow OSTL: \qquad \min_{g_{ic}, K_c} \; M \;\; = \;\; \max_{c} \left\{ \sum_{i=1}^{N} (a_{ic} + g_{ic}) \right\} \tag{5.3}$$

subject to:

$$K_c \; + \; \sum_{j=1}^{i-1} (a_{jc} + g_{jc}) \;\; \geq \;\; K_{c-1} \; + \; \sum_{j=1}^{i-1} (a_{j,c-1} + g_{j,c-1}) \; + \; a_{i,c-1} \; + \; \Delta$$
$$c = 2, \ldots, C, \;\; i = 1, \ldots, N \tag{5.4}$$

$$M \; + \; \sum_{j=1}^{i-1} (a_{j1} + g_{j1}) \;\; \geq \;\; K_C \; + \; \sum_{j=1}^{i-1} (a_{jC} + g_{jC}) \; + \; a_{iC} \; + \; \Delta \qquad i = 1, \ldots, N \tag{5.5}$$

$$g_{ic}, K_c, M : \text{integers}; \; g_{ic} \geq 0 \; \forall \, i, c; \; K_1 = 0; \; K_c > K_{c-1} \; c = 2, \ldots, C \; ; \; M > K_C \tag{5.6}$$

Constraint (5.4) 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 (5.5) is essentially the same as the previous one – it 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 (3.4). The no-collision constraints (3.5) 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. Since constraint (5.4) and (5.5) are essentially the same, we combine them and rewrite the above constraints as :

$$K_{c+1} \; + \; \sum_{j=1}^{i-1} (a_{j,c+1} + g_{j,c+1}) \;\; \geq \;\; K_c \; + \; \sum_{j=1}^{i-1} (a_{j,c} + g_{j,c}) \; + \; a_{i,c} \; + \; \Delta$$
$$c = 1, \ldots, C, \;\; i = 1, \ldots, N \tag{5.7}$$

$$g_{ic}, K_c, M : \text{integers}; \;\; g_{ic} \geq 0 \; \forall \; i, c; \;\; K_1 = 0; \;\; K_{c+1} > K_c \; c = 1, \ldots, C \; ; \;\; M \geq K_{C+1} \tag{5.8}$$

where all terms with references to channel λ_{C+1} signify the next frame on channel λ_1.

Note that, finding an optimal schedule within the class (5.1) for problem *OSTL* involves solving $N!$ *BW-OSTL* problems, one for each possible transmitter sequence, and choosing the schedule of 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, and the size of its state space becomes unmanageable for anything but trivial values of N and C.

Recall, however, that we are considering bandwidth limited networks. For such networks, the bandwidth bound (4.1) dominates, therefore, the lower bound on the schedule length (see (4.3)) is such that

$$M^{(l)} \;=\; M_{bw}^{(l)} \;>\; M_t^{(l)} \tag{5.9}$$

In other words, there can be no schedule of length less than $M^{(l)}$, as there exists at least one channel λ_c such that $\sum_{i=1}^{N} a_{ic} = M^{(l)}$. 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, let us attempt to answer a fundamental question related to the existence of schedules of length $M^{(l)}$ within the class (5.1).

5.1.1 A Sufficient Condition for Optimality

Let $\mathbf{A}$ be the collapsed traffic matrix of a bandwidth limited network, and let $M^{(l)}$ be the lower (bandwidth) bound on any schedule for $\mathbf{A}$. We now define the *average slot requirement* for a source-destination pair as $a = \frac{M^{(l)}}{N}$. Our first observation is that if $a_{ic} = a \; \forall \; i, c$, then an optimum length schedule is easy to construct; just let

$$K_{c+1} \;=\; c\,(a+\Delta) \;\forall\; c; \;\; g_{ic} \;=\; 0 \;\forall\; i, c; \quad M \;=\; M^{(l)} \;=\; Na \tag{5.10}$$

and all of (5.7) – (5.8) will be satisfied. 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.

Lemma 5.1 *Let* $\mathbf{A}$ *be a collapsed traffic matrix such that the lower bound in (4.3)* $M^{(l)} = M_{bw}^{(l)} > M_t^{(l)}$ *(bandwidth limited network). Then, a schedule of length equal to the lower bound,* $M^{(l)}$*, exists within the class (5.1) for any transmitter sequence, if* *the elements of* **A** *satisfy the following condition:*

$$\left| \, a_{ic} \Leftrightarrow \frac{M^{(l)}}{N} \, \right| \quad \leq \quad \epsilon \qquad \forall \; i, c \tag{5.11}$$

with ϵ *given by:*

$$\epsilon \quad = \quad \frac{M^{(l)}}{N+1} \left(\frac{1}{C} \Leftrightarrow \frac{1}{N} \Leftrightarrow \frac{\Delta}{M^{(l)}} \right) \tag{5.12}$$

In proving Lemma 5.1 we will make use of the following result.

Lemma 5.2 *If constraints (5.11) on the elements of* **A** *hold, then for all* $\mathcal{P} \subseteq \{1, \ldots, N\}$ *with* $\mid \mathcal{P} \mid = n$, *and any two channels* λ_{c_1} *and* λ_{c_2}*:*

$$\left| \, \sum_{i \in \mathcal{P}} a_{i,c_1} \Leftrightarrow \sum_{i \in \mathcal{P}} a_{i,c_2} \, \right| \quad \leq \quad N \, \epsilon \tag{5.13}$$

Proof (of Lemma 5.2). Because of (5.11), for any $n \in \{1, \ldots, N\}$, and any channel λ_c we get:

$$n \left(\frac{M^{(l)}}{N} \Leftrightarrow \epsilon \right) \quad \leq \quad \sum_{i \in \mathcal{P}} a_{ic} \quad \leq \quad M^{(l)} \Leftrightarrow (N \Leftrightarrow n) \left(\frac{M^{(l)}}{N} \Leftrightarrow \epsilon \right) \tag{5.14}$$

Given the above, the result in (5.13) can be easily derived. □

We are now ready to prove Lemma 5.1. Note that, although the proof refers to the problem formulation in (5.3) – (5.6), 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 (5.3) – (5.6).

Proof (of Lemma 5.1). 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:

$$\sum_{i=1}^{N} a_{ic} \quad = \quad M^{(l)} \qquad \forall \; c \tag{5.15}$$

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 (5.3)). We can rewrite constraint (5.7) as

$$K_{c+1} \Leftrightarrow K_c \;\; \geq \;\; \left(\sum_{j=1}^{i-1} a_{j,c} \Leftrightarrow \sum_{j=1}^{i-1} a_{j,c+1} \right) \; + \; a_{i,c} \; + \; \Delta \qquad c = 1, \ldots, C, \; i = 1, \ldots, N \tag{5.16}$$

Hence, Lemma 5.2 guarantees that choosing $K_{c+1} \Leftrightarrow K_c = N\epsilon + \frac{M^{(l)}}{N} + \epsilon + \Delta, c = 1, \ldots, C$, satisfies constraint (5.16). Noting that $K_1 = 0$, we can set:

$$K_{c+1} \;\; = \;\; c \left((N+1)\epsilon \; + \; \frac{M^{(l)}}{N} \; + \; \Delta \right) \qquad c = 1, \ldots, C \tag{5.17}$$

Finally, it is easy to check that letting $M = M^{(l)}$ ensures that (5.8) is also satisfied. □

Lemma 5.1 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. To get a feeling of how restrictive this bound is, let us rewrite expression (5.12) as

$$\frac{\epsilon}{M^{(l)}/N} \;\; = \;\; \frac{N}{N+1} \left(\frac{1}{C} \Leftrightarrow \frac{1}{N} \Leftrightarrow \frac{\Delta}{M^{(l)}} \right) \tag{5.18}$$

For $N = 100$, $C = 10$, and ignoring the term $\frac{\Delta}{M^{(l)}}$ [3], we get $\frac{\epsilon}{M^{(l)}/N} \approx .089$. Thus, the variation of elements a_{ic} around $\frac{M^{(l)}}{N}$ can be up to 8.9% to guarantee a schedule of length $M^{(l)}$. Note, however, that the analysis presented here is not tight; in Appendix B we present an alternate analysis which relaxes the upper bound on the degree of non-uniformity of the traffic matrix by a factor of 2. Also, the proofs are based on a worst case scenario; in general, we expect such a schedule to exist for significantly higher degrees of variation.

As a final observation, ϵ is greater than zero only when $M^{(l)} > \frac{NC\Delta}{N-C}$. This is consistent with our hypothesis of a bandwidth limited network.

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

5.1.2 Scheduling Algorithm

Lemma 5.1 provides a sufficient condition for the existence of an optimum length schedule, but does not state how to construct one. We now develop a polynomial time scheduling algorithm which, under the conditions of Lemma 5.1, 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 behind the algorithm is to schedule the transmissions on the first channel 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 this first channel is the dominant one, that is $\sum_{i=1}^{N} a_{i1} = M^{(l)}$.

Algorithm *Make_Bandwidth_Limited_Schedule (MBLS)* is described in detail in Figure 5.2, and operates as follows. All the 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 (5.4). Doing so, however, may introduce large gaps into the channels, resulting in a sub-optimal schedule length (refer to (5.3)). During the second pass then, 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 (5.4) and (5.5) allow.

That algorithm *MBLS* is correct follows from the fact that it constructs a schedule which satisfies the constraints (5.4) – (5.6), and hence gives an admissible schedule. We now state and prove its optimality properties.

Theorem 5.1 *Algorithm MBLS constructs a schedule of minimum length among the schedules that (a) are within the class (5.1) 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.,* $g_{i1} = 0, i = 1, \ldots, N \Leftrightarrow 1$*).*

Proof. See Appendix C. □

Corollary 5.1 (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*

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 $M = \sum_{i=1}^{N} a_{i1}$
    Set $K_1$ and all gaps $g_{i1}$ on $\lambda_1$ 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 (5.4) is satisfied between channels $\lambda_c$ and $\lambda_{c-1}$
    // 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 (5.5)
    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 (5.4) between channels $\lambda_c$ and $\lambda_{c+1}$
            for $j = i + 1$ to $N$ do
                Shift the $a_{jc}$ slots as much left as possible while
                maintaining constraint (5.4) between channels $\lambda_c$ and $\lambda_{c-1}$
        // end of for $i$ loop – the final values of gaps for this channel determined
        Let $M_c = \sum_{i=1}^{N} (a_{ic} + g_{ic})$
        $M = \max(M, M_c)$
    // end of for $c$ loop – $M$ is now the final length of the schedule
// end of algorithm
```

Figure 5.2: Scheduling algorithm for bandwidth limited networks

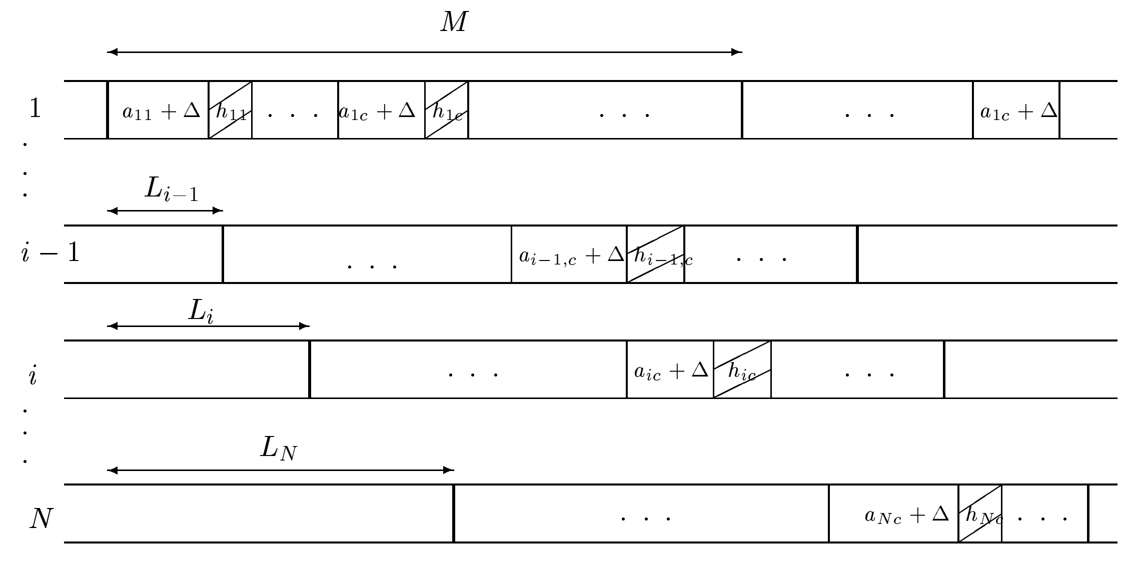


Figure 5.3: Schedule for a tuning limited network

conditions of Lemma 5.1, algorithm MBLS constructs an optimum length schedule, i.e., a schedule of length $M^{(l)}$.

Proof. According to Lemma 5.1, there exists a schedule of length $M^{(l)}$ within the class defined by (5.1), 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, by Theorem 5.1, algorithm MBLS will construct such a schedule. $\Box$

Regarding the complexity of *MBLS*, it is easy to verify that the algorithm takes time $\mathcal{O}(CN^2)$, regardless of the actual values of the traffic elements a_{ic}.

5.2 Tuning Limited Networks

The analysis and scheduling algorithm presented in the previous section pertain specifically to bandwidth limited networks. We now turn our attention to tuning limited networks. Since our discussion above focused on how transmissions are scheduled within each channel, it is only natural that we now adopt a transmitter's point of view, and concentrate on how its transmissions are scheduled across the various channels [4].

Let $\mathcal{S}$ be a schedule of length M for a tuning limited network, and let $(\lambda_1, \ldots, \lambda_C)$ be the channel sequence for all nodes. For each transmitter we consider the frame that begins with the first slot assigned on channel λ_1. Let the start of the frame for node 1 be our reference point, and let L_i denote the distance, in slots, between the start of the frame for node i, and the reference point (see Figure 5.3).

Consider now the transmissions of, say, node i, within a frame of M slots. Following its a_{i1} slots on channel λ_1, and a tuning period of Δ slots in which it tunes its laser to λ_2, node i is ready to transmit on that channel. However, node $i \Leftrightarrow 1$ may still be transmitting on λ_2, in which case node i will remain idle for several slots before it starts transmitting on λ_2. In general, we let h_{ic} denote the number of slots transmitter i remains idle between the time it has tuned its laser to channel λ_{c+1} and the start of its transmissions on the same channel.

Based on these observations, the problem of finding an optimum length schedule such that (a) the schedule is within the class (5.2) [5], and (b) the channel sequence is $(\lambda_1, \ldots, \lambda_C)$ can be formulated as an integer programming problem, referred to as *tuning limited OSTL (T-OSTL)*:

$$T \Leftrightarrow OSTL: \qquad \min_{h_{ic}, L_i} \; M \;=\; \max_i \left\{ \sum_{c=1}^{C} (a_{ic} + \Delta + h_{ic}) \right\} \tag{5.19}$$

subject to:

$$L_i \;+\; \sum_{l=1}^{c-1}(a_{il} + \Delta + h_{il}) \;\geq\; L_{i-1} \;+\; \sum_{l=1}^{c-1}(a_{i-1,l} + \Delta + h_{i-1,l}) \;+\; a_{i-1,c}$$
$$i = 2, \ldots, N, \;\; c = 1, \ldots, C \tag{5.20}$$

$$M \;+\; \sum_{l=1}^{c-1}(a_{1l} + \Delta + h_{1l}) \;\geq\; L_N \;+\; \sum_{l=1}^{c-1}(a_{Nl} + \Delta + h_{Nl}) \;+\; a_{Nc} \qquad i = 1, \ldots, N \tag{5.21}$$

$$h_{ic}, L_i, M : \text{integers}; \; h_{ic} \geq 0 \; \forall \, i, c; \; L_1 = 0; \; L_i > L_{i-1} \; i = 2, \ldots, N \;; \; M > L_N \tag{5.22}$$

Constraints (5.20) and (5.21) are essentially the no-collision constraints (3.5); they ensure that transmissions by some node i on some channel λ_c start after the end of

[4] As the reader will soon notice, the following discussion and results mirror those in the previous section. This confirms our intuition that bandwidth limited and tuning limited networks are in a sense dual of each other.

[5] Recall that classes (5.2) and (5.1) are equivalent.

transmissions by node $i \Leftrightarrow 1$ on the same channel. The hardware constraints (3.4) correspond to constraints $h_{ic} \geq 0 \; \forall \; i, c$.

Note that the above formulation is very similar to the one in (5.3) – (5.6), and that problem *T-OSTL* is equally as hard to solve as *BW-OSTL*. However, in the case we are considering, the tuning bound dominates, thus

$$M^{(l)} \;=\; M_t^{(l)} \;>\; M_{bw}^{(l)} \tag{5.23}$$

In other words, there exists a transmitter i such that $\sum_{c=1}^{C}(a_{ic} + \Delta) = M^{(l)}$. Using this observation we were able to prove optimality properties and derive a scheduling algorithm for tuning limited networks. In the following, however, the various proofs are omitted, as they are very similar to the corresponding proofs for the bandwidth limited case presented earlier.

5.2.1 A Sufficient Condition for Optimality

For tuning limited networks we define the average slot requirement (which now includes both transmission and tuning slots) as $a' = \frac{M^{(l)}}{C}$. Then, the following lemma, analogous to Lemma 5.1, provides a sufficient condition for the existence of a schedule of length $M^{(l)}$.

Lemma 5.3 *Let* **A** *be a collapsed traffic matrix such that the lower bound in (4.3) $M^{(l)} = M_t^{(l)} > M_{bw}^{(l)}$ (tuning limited network). Then, a schedule of length equal to the lower bound, $M^{(l)}$, exists within the class (5.2) for any channel sequence, if the elements of* **A** *satisfy the following condition:*

$$\left| (a_{ic} \;+\; \Delta) \Leftrightarrow \frac{M^{(l)}}{C} \right| \;\leq\; \epsilon' \qquad \forall \; i, c \tag{5.24}$$

with ϵ' given by:

$$\epsilon' \;=\; \frac{M^{(l)}}{C+1}\left(\frac{\Delta}{M^{(l)}} + \frac{1}{N} \Leftrightarrow \frac{1}{C}\right) \tag{5.25}$$

The proof of Lemma 5.3, uses the following result, similar to Lemma 5.2. Both proofs are omitted. Observe also that ϵ' is greater than zero only when $M^{(l)} < \frac{NC\Delta}{N-C}$.

Lemma 5.4 *If constraints (5.24) on the elements of* $\mathbf{A}$ *hold, then for all* $\mathcal{Q} \subseteq \{1, \ldots, C\}$ *with* $\mid \mathcal{Q} \mid = k$*, and any two transmitters, i and j:*

$$\left| \sum_{c \in \mathcal{Q}} (a_{ic} + \Delta) \Leftrightarrow \sum_{c \in \mathcal{Q}} (a_{jc} + \Delta) \right| \quad \leq \quad C \, \epsilon' \tag{5.26}$$

Note that as in the bandwidth limited case, the upper bound in Lemma 5.3 on the degree of non-uniformity of the traffic matrix in order to guarantee a schedule of length equal to the lower bound can be improved by a factor of 2 by a tighter analysis analogous to the one presented in Appendix B.

5.2.2 Scheduling Algorithm

As in the case of bandwidth limited schedules, we have developed a scheduling algorithm for tuning limited networks which is optimal under the conditions of Lemma 5.3. This algorithm, which we will call *Make_Tuning_Limited_Schedule (MTLS)*, is very similar to *MBLS*, and is omitted. The key idea is to schedule the slots of a certain node so that its transmitter is never idle, except possibly at the end of a frame. The optimality properties of the *MTLS* algorithm are now stated without proof.

Theorem 5.2 *Algorithm* $MTLS$ *constructs a schedule of minimum length among the schedules that (a) are within the class (5.2) and the sequence of channels is* $(\lambda_1, \ldots, \lambda_C)$*, (b) transmitter 1 is a dominant transmitter, and (c) transmitter 1 is never idle, except, possibly, at the very end of the frame (i.e.,* $h_{1c} = 0, c = 1, \ldots, C \Leftrightarrow 1$*).*

Corollary 5.2 (Optimality of Algorithm MTLS) *Let transmitter 1 be such that* $\sum_{c=1}^{C} (a_{1c} + \Delta) = M^{(l)}$*, and arbitrarily label the channels* λ_1 *through* λ_C*. Then, under the conditions of Lemma 5.3, algorithm MBLS constructs an optimum length schedule, i.e., a schedule of length* $M^{(l)}$*.*

5.3 Tuning and Bandwidth Balanced Networks

The last two sections considered the operation of the network in two distinct regions, the bandwidth and tuning limited regions, respectively. We now study the effect of operating the network at the boundary of the two regions, namely, when the tuning and bandwidth bounds are equal:

$$M^{(l)} \;=\; M_t^{(l)} \;=\; M_{bw}^{(l)} \tag{5.27}$$

In particular, we show that if (5.27) holds, even arbitrarily small non-uniformities in the traffic pattern may result in *every* admissible schedule having length *greater* than the lower bound $M^{(l)}$. Our claim is stated in the following theorem; note that neither the theorem nor its proof refer to the class of schedules defined by (5.1) and (5.2), therefore, this result holds for arbitrary schedules, not only the ones implied by (5.1) or (5.2).

Theorem 5.3 *Let* **A** *be a collapsed traffic matrix such that the bandwidth and tuning bounds are equal, and such that each transmitter and each channel are* tight *(i.e., the slot requirement on each channel is equal to the lower bound, and the slot-plus-tuning requirement of each transmitter is also equal to the lower bound). Then, the optimal schedule has length strictly greater than the lower bound, even for any arbitrarily small non-uniformity among the elements of matrix* **A***.*

Proof. Consider a system of $N \geq 2$ nodes, $C \geq 2$ channels, and Δ tuning slots. Further suppose that

$$a \;=\; \frac{C}{N \Leftrightarrow C}\,\Delta \tag{5.28}$$

is an integer. Letting $a_{ic} = a \;\forall\; i, c$, we then obtain a uniform collapsed traffic matrix satisfying the conditions of the theorem, i.e., such that the bandwidth bound $M_{bw}^{(l)} = Na$ is equal to the tuning bound $M_t^{(l)} = C(a+\Delta)$, and such that all transmitters and channels are tight.

Let us now modify some of the elements of this uniform matrix to construct a non-uniform matrix **A** that continues to satisfy the conditions of the theorem. Observe, then, that at least four elements have to be appropriately modified to achieve this result. For if for some i, c_1, we let $a_{i,c_1} < a$, we have to appropriately 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 $\mathbf{A}$ be such that

$$a_{11} = a_{22} = a \Leftrightarrow \epsilon, \qquad a_{21} = a_{12} = a + \epsilon, \qquad a_{ic} = a \text{ for all other } i, c \tag{5.29}$$

for some arbitrary $\epsilon < 1$. It is easy to verify that $M_t^{(l)} = C(a + \Delta) = Na = M_{bw}^{(l)}$, and that all channels and transmitters are tight. Also, based upon the above discussion, 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 instant in such a schedule. Also, assuming that the schedule starts at time 0, all transmissions on channels λ_3 through λ_C begin and end at integral time values; similarly for transmissions by stations 3 through N.

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 integral. Since channel λ_1 is never idle, the transmission by node 1 on λ_1 starts at time t, and ends at time $t + a \Leftrightarrow \epsilon$. At that (non-integral) time, node 2 is the only candidate for immediate transmission on λ_1, so node 2 must start transmitting on λ_1 at time $t + a \Leftrightarrow \epsilon$. Node 1, on the other hand, after a tuning period of Δ slots, is ready for its next transmission at time $t + a \Leftrightarrow \epsilon + \Delta$; since this is a non-integral time, 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 $t + a \Leftrightarrow \epsilon$, and node 2 from time $t + a \Leftrightarrow \epsilon$ to time $t + 2a$, and on channel λ_2 node 2 transmits from time $t + \Delta$ to time $t + \Delta + a \Leftrightarrow \epsilon$, and node 1 from time $t + \Delta + a \Leftrightarrow \epsilon$ to time $t + \Delta + 2a$. But, regardless of the values of a, Δ, and ϵ, this sequence of transmissions is impossible [6], contradicting our hypothesis that an admissible schedule of length equal to the lower bound $Na = C(a + \Delta)$ exists. □

In this chapter we have defined a class of schedules for which we have presented algorithms that construct optimal schedules under certain conditions. In the next chapter, we consider the performance of these algorithms when these optimality conditions do not hold, and develop heuristics which improve performance under these circumstances.

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

Chapter 6

Optimization Heuristics

In the previous chapter we presented scheduling algorithms which construct schedules of length equal to the lower bound as long as traffic parameters a_{ic} satisfy certain conditions. However, we have seen that $OSTL$ is $\mathcal{NP}$-complete, meaning that an algorithm that efficiently solves any arbitrary instance of $OSTL$ (i.e., one in which a_{ic} may not satisfy the optimality conditions) may not exist. Hence, the most we can hope for is to be able to devise a heuristic that can be expected to perform well for general traffic matrices. We now develop a scheduling heuristic for bandwidth limited networks; using a very similar reasoning, it is relatively straightforward to determine a heuristic for tuning limited networks.

Recall that for bandwidth limited networks, finding a schedule within the class (5.1) that solves the $OSTL$ problem involves solving $N!$ $BW\text{-}OSTL$ problems (see (5.3) – (5.6)), one for each possible transmitter sequence; obviously, no polynomial-time heuristic can consider that many sequences. On the other hand, we have no efficient algorithm for solving the most general version of $BW\text{-}OSTL$, but we have developed *MBLS*, a polynomial time algorithm that solves $BW\text{-}OSTL$ 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 (refer also to Theorem 5.1). With these considerations in mind, our approach to obtaining near-optimal schedules for $OSTL$ is based on making two compromises:

- Suppose that an optimal transceiver sequence for a network of n nodes has been determined, and that a new node is added to the network (that is, a new row is added to the collapsed traffic matrix $\mathbf{A}$, while all other elements remain as before). 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 (i.e., 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 initial 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 5.1, 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. Numerical results to be presented, however, do suggest that overall the performance of *MBLS* is very close to being optimal.

For bandwidth limited networks, our heuristic is described in Figure 6.1. About the complexity of the heuristic, 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 $\mathcal{O}(CN^4)$.

It is very easy to come up with a similar algorithm for the tuning limited case. Instead of adding one node at a time, the *Tuning Limited Scheduling Heuristic (TLSH)* would add one channel at a time. It can be verified that the complexity of *TLSH* would be $\mathcal{O}(C^3N^2)$

Bandwidth Limited Scheduling Heuristic (BLSH)

1. Relabel the channels such that:

$$M^{(l)} \quad = \quad \sum_{i=1}^{N} a_{i1} \quad \geq \quad \sum_{i=1}^{N} a_{i2} \quad \geq \quad \ldots \quad \geq \quad \sum_{i=1}^{N} a_{iC} \tag{6.1}$$

   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 $i \Leftrightarrow 1$ transmitters of the original network. Consider transmitter i. Run algorithm *MBLS* on each of the i permutations

$$(i, \pi_1, \ldots, \pi_{i-1}), \ldots, (\pi_1, \ldots, \pi_j, i, \pi_{j+1}, \ldots, \pi_{i-1}), \ldots, (\pi_1, \ldots, \pi_{i-1}, i) \tag{6.2}$$

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

Figure 6.1: Scheduling Heuristic

In the next chapter, we run the various algorithms we have developed so far on random traffic matrices and study their relative performance. Since the optimal schedule length is impossible to compute, we judge the performance of each algorithm by comparing its outcome to the lower bound developed in chapter 4.

Chapter 7

Numerical Results

We now consider four different algorithms for the *OSTL* problem and compare their performance. The four algorithms are:

1. algorithm *MBLS*, described in Figure 5.2; 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; the algorithm has not been described here, but is very similar to *MBLS*, only targeted to tuning limited networks;
3. scheduling heuristic *BLSH*, described in Figure 6.1;
4. scheduling heuristic *TLSH* for tuning limited networks; again, this heuristic has not been described, but is very similar to *BLSH*.

We have generated random instances of the *OSTL* problem, i.e., random matrices $\mathbf{A}$ for various values of N, C, and Δ, and fed them as input to the algorithms. Given $\mathbf{A}$, the lower bound $M^{(l)}$ on the schedule length can be obtained from (4.3), (4.2) and (4.1). Let M be the actual length of a schedule for $\mathbf{A}$ produced by some scheduling algorithm. Then the quantity

$$\frac{M}{M^{(l)}} 100\% \tag{7.1}$$

represents how far the length M of the schedule produced by the algorithm is from the lower bound. All figures in this chapter plot the quantity in (7.1) against the number of nodes, N, for the four algorithms described above. Each point plotted represents the average of twenty different matrices $\mathbf{A}$ for the stated values of N, C, and Δ.

For the results shown in Figures 7.1 – 7.12, the elements of each matrix $\mathbf{A}$ were chosen, with equal probability, among the integers 1 through 20 (this is the $uniform(1, 20)$ distribution). We show four sets of three figures each, corresponding to the four values of the number of channels $C = 5, 10, 15, 20$. Within each set with the same value of C we use three different values for Δ, namely $\Delta = 1, 4, 16$. The number N of nodes within each figure takes values from C to 80. Note that, for data rates of 1 Gigabits per second, and packet sizes equal to the ATM cell size (53 bytes), the packet transmission time (slot length) is $424ns$. Hence the three values of Δ considered here correspond to transceiver tuning times of $424ns$, $1.7\mu s$, and $6.8\mu s$, respectively; the last two values are representative of the current state of the art in optical transceiver technology [12].

Let us now concentrate on the relative performance of the four algorithms. Our first observation is that the two heuristics, *BLSH* and *TLSH*, always perform as good as, or better than the corresponding algorithms, *MBLS* and *MTLS*, respectively. This is in fact expected. As explained above, we applied algorithms *MBLS* and *MTLS* to a single transmitter and channel sequence; since the random traffic matrices do not necessarily satisfy the optimality conditions of Lemmas 5.1 and 5.3, these algorithms may fail to produce an optimal schedule. Heuristic *BLSH* (respectively, *TLSH*) on the other hand, calls *MBLS* (respectively, *MTLS*) on several transmitter (respectively, channel) sequences, and is more likely to construct schedules of length close to the lower bound.

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 Figure 7.6 which shows results for $C = 10, \Delta = 16$. As we can see, algorithms *MBLS* and *BLSH* outperform their counterparts, *MTLS* and *TLSH*, respectively, when $N > 25$, while the opposite is true for $N < 25$. Indeed, for these

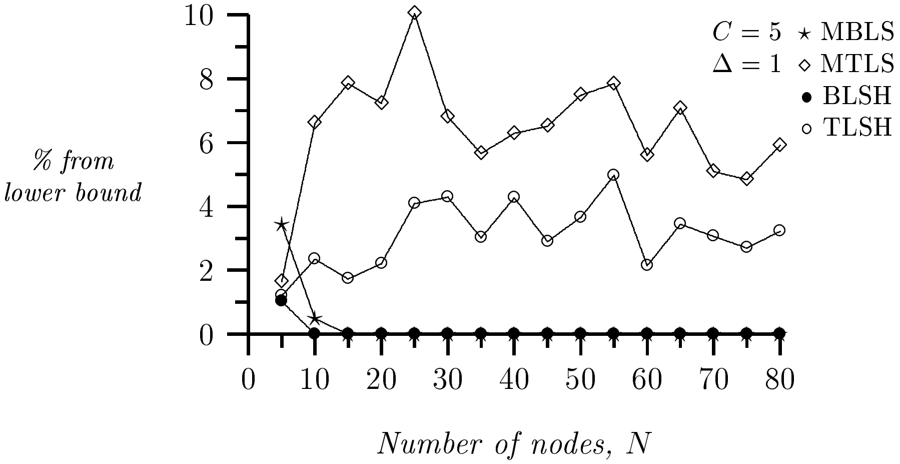


Figure 7.1: Algorithm comparison for $C = 5$ channels and $\Delta = 1$ tuning slots

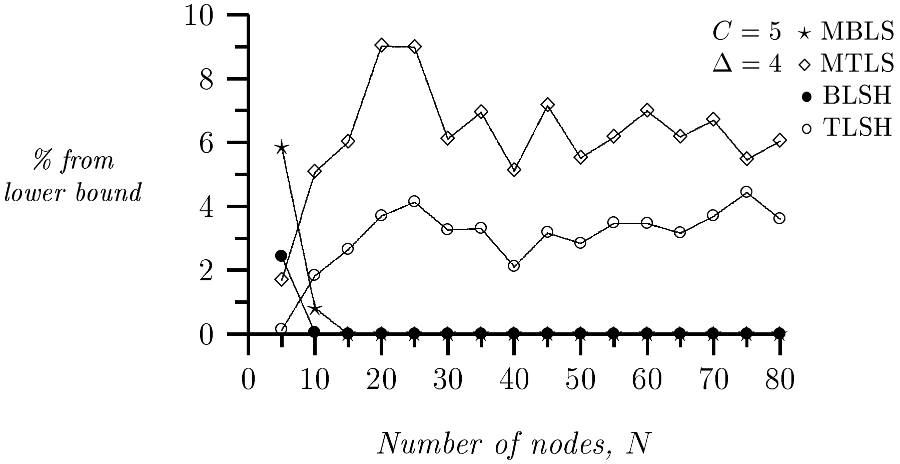


Figure 7.2: Algorithm comparison for $C = 5$ channels and $\Delta = 4$ tuning slots

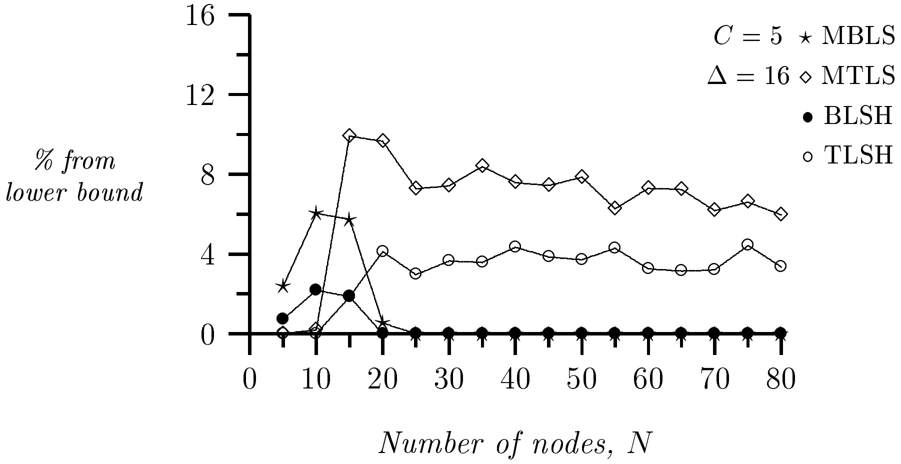


Figure 7.3: Algorithm comparison for $C = 5$ channels and $\Delta = 16$ tuning slots

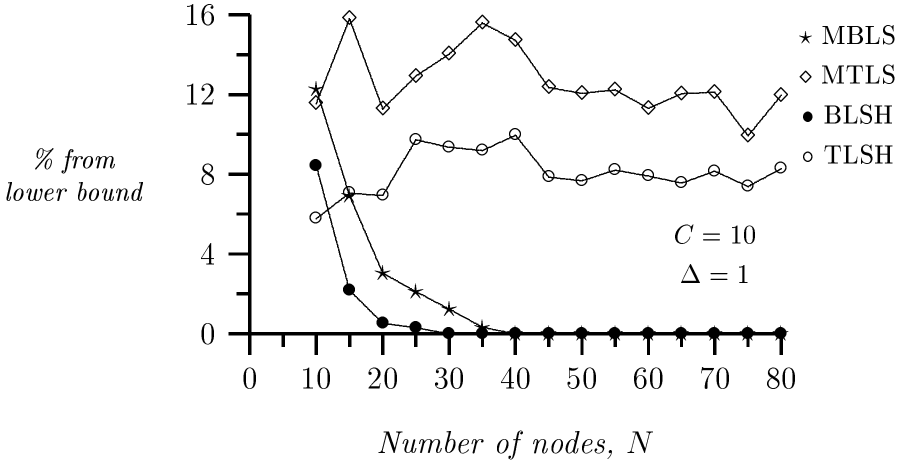


Figure 7.4: Algorithm comparison for $C = 10$ channels and $\Delta = 1$ tuning slots

values of C and Δ, and the way the traffic matrices are constructed, a network is in the bandwidth limited region if it has more than 25 nodes, and in the tuning limited region, otherwise (see 4.9). It should come as no surprise, then, that algorithms *MBLS* and *BLSH* (respectively, *MTLS* and *TLSH*), designed for bandwidth (respectively, tuning) limited networks, perform very close to the optimal within their intended region of operation, and sub-optimally within the other. Very similar observations can be made for all other figures presented here.

The figures can also explain how the point at which the network becomes tuning or bandwidth limited depends on the system parameters N, C, and Δ (the traffic parameters are the same for all figures, so they do not play a role at this point; we will consider their effect shortly). Consider Figures 7.4 – 7.6, corresponding to the same value of $C = 10$. As the value of Δ increases, a larger number of nodes N is needed if expression (4.9) is to be satisfied [1]. This is indeed reflected in the above figures, as the point at which algorithms *MBLS* and *BLSH* outperform *MTLS* and *TLSH*, respectively, moves to the right (i.e., towards greater values of N) as Δ increases from 1 to 16. Similarly, if we concentrate on how the number C of channels alone affects the region of operation, we can see from, say, Figures 7.2, 7.5, 7.8, and 7.11, all of which show results with $\Delta = 4$, that for larger C it takes more nodes to keep the network in the bandwidth limited region; this is in accordance to (4.9), as expected.

Let us now consider the performance of *MBLS* and *BLSH* for *bandwidth limited* networks; very similar conclusions can be drawn regarding the performance of *MTLS* and *TLSH* in the *tuning limited* region. From the various figures we observe that, in general, the length of schedules produced by *MBLS* and *BLSH* are very close to the lower bound, and that, for networks well within the bandwidth limited region (i.e., for sufficiently large N), *BLSH*, and sometimes *MBLS*, construct schedules of length *equal* to the lower bound. This is a very important result, as it establishes that the lower bound accurately characterizes the scheduling efficiency in this type of environment.

[1] Note that the total traffic demand D in (4.9) can be expressed as $NC\alpha$, where α is the average transmitter-channel slot requirement. For the $uniform(1, 20)$ distribution we consider here, $\alpha = 10.5$.

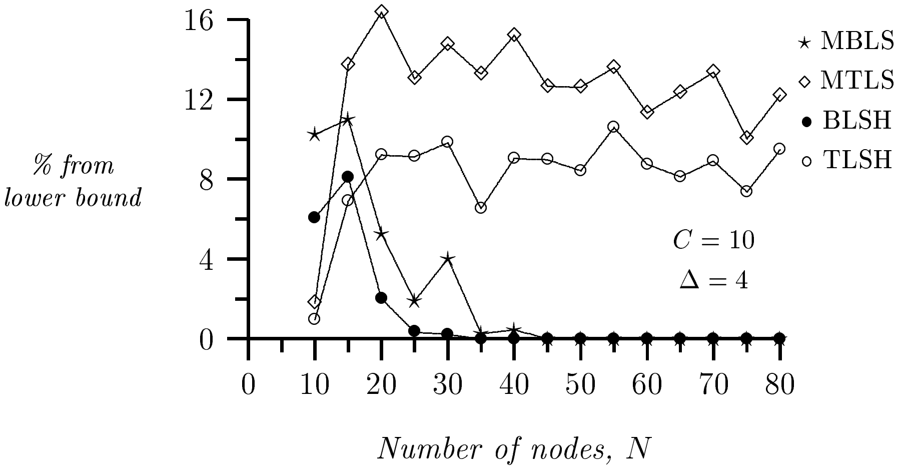


Figure 7.5: Algorithm comparison for $C = 10$ channels and $\Delta = 4$ tuning slots

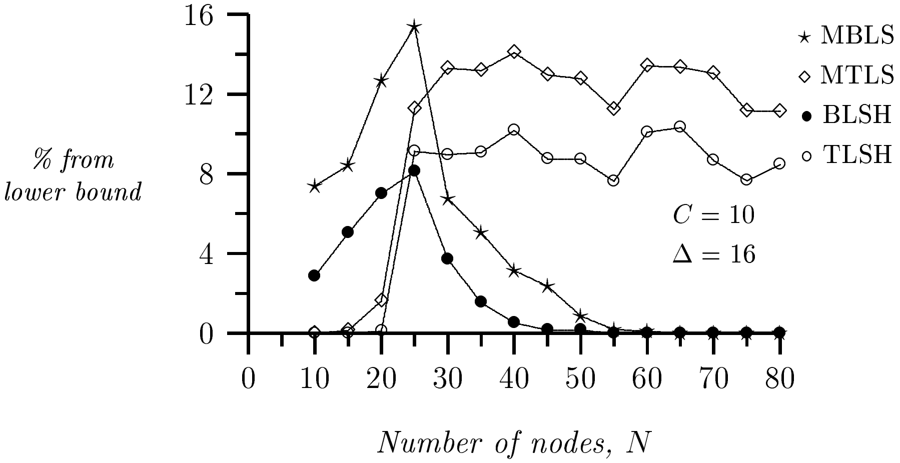


Figure 7.6: Algorithm comparison for $C = 10$ channels and $\Delta = 16$ tuning slots

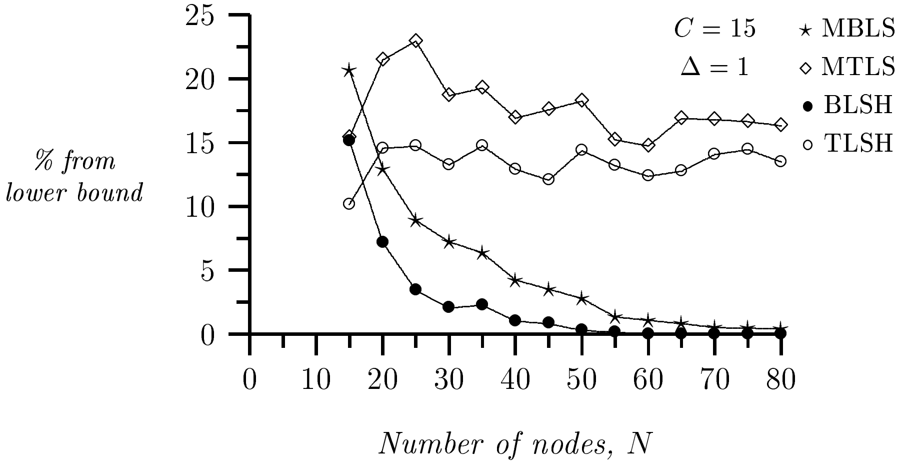


Figure 7.7: Algorithm comparison for $C = 15$ channels and $\Delta = 1$ tuning slots

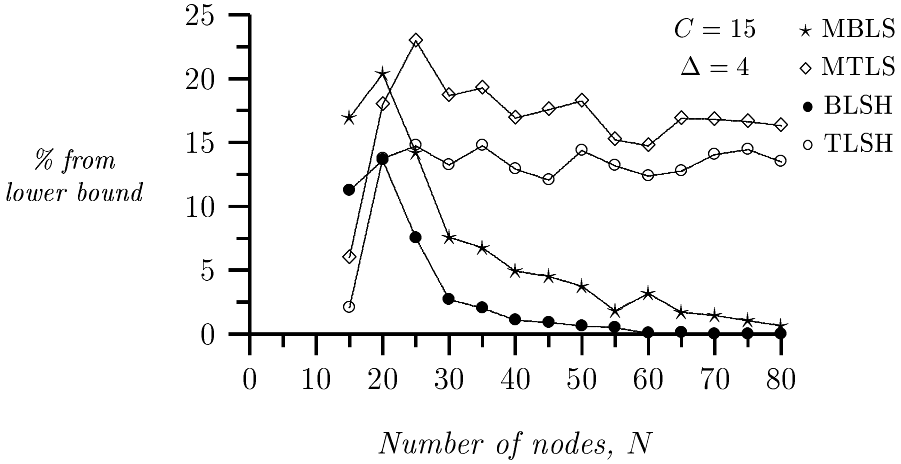


Figure 7.8: Algorithm comparison for $C = 15$ channels and $\Delta = 4$ tuning slots

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.

Another important observation is that, although the four algorithms perform very close to the lower bound within their respective regions of operation, they deviate from it for values of N in the boundary of the tuning and bandwidth limited regions (although they are never more than 30% away from the lower bound, and in many cases they are as close as 15%). This can be explained by noting that, for those values of N, the tuning and bandwidth bounds are close to each other. When there are several channels and nodes with similar slot requirements, the scheduling algorithms have less flexibility in placing the various slots to obtain schedules of length close to the lower bound. A similar behavior was observed in [2] for *SRA*, a scheduling algorithm that operates under a totally different strategy. Combining these results with Theorem 5.3 suggests that the behavior of our algorithms in the boundary between the tuning and bandwidth limited regions 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 now study the effect of the traffic demands on the operation of the network. In Figure 7.13 we plot the performance of the four algorithms for $C = 10$ channels and $\Delta = 16$ tuning slots. In this case, however, the elements of matrix $\mathbf{A}$ were selected from the $uniform(1, 40)$ distribution. Comparing to Figure 7.6 which plots results for the same values of C and Δ, but with the traffic parameters a_{ic} selected from the $uniform(1, 20)$ distribution, we see that the point at which the network becomes bandwidth limited has moved to the left (i.e., towards a smaller number of nodes); this is in accordance to (4.9). We also observe that the behavior of the four algorithms is very similar to the one observed before. Finally, in Figure 7.14 we plot results for $C = 10$ and $\Delta = 16$, but now elements a_{ic} have been selected according to a bimodal distribution as follows: with probability $\frac{1}{2}$ an element is chosen from the $uniform(1, 7)$ distribution, and with probability $\frac{1}{2}$ from the $uniform(14, 20)$ distribution (the mean, however, is the same as the $uniform(1, 20)$ distribution in Figure 7.6). As we can see the plots of Figure 7.14 are very similar to the ones in

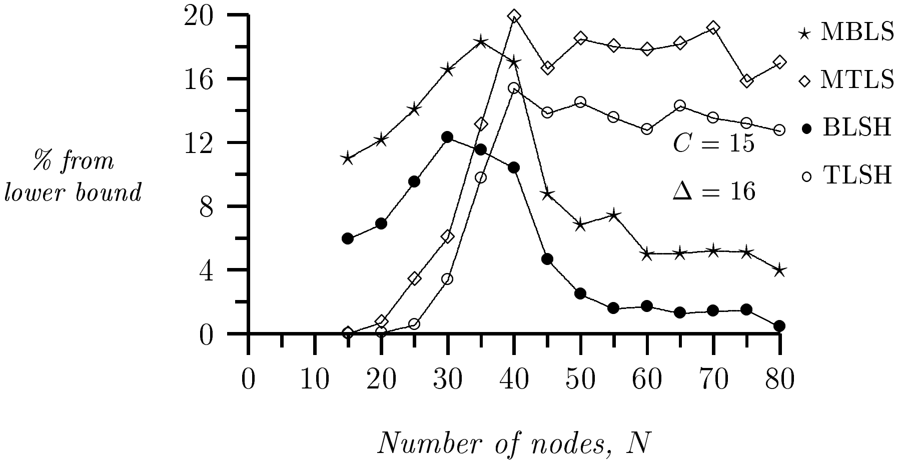
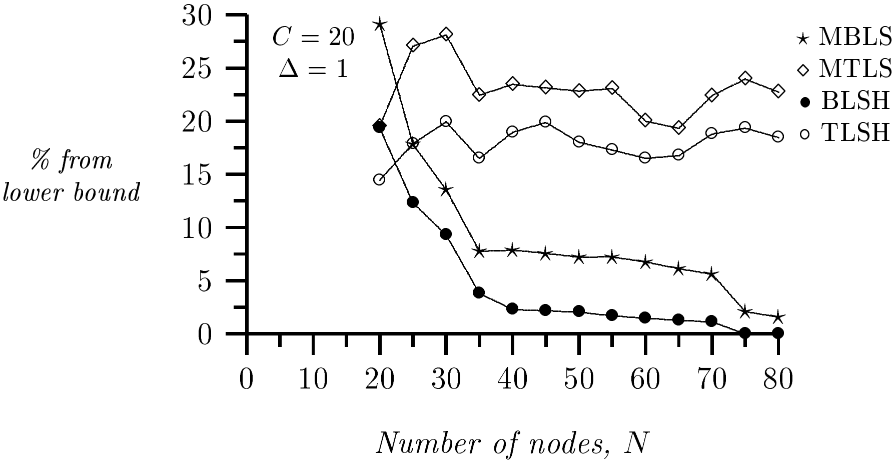


Figure 7.9: Algorithm comparison for $C = 15$ channels and $\Delta = 16$ tuning slots



g

Figure 7.10: Algorithm comparison for $C = 20$ channels and $\Delta = 1$ tuning slots

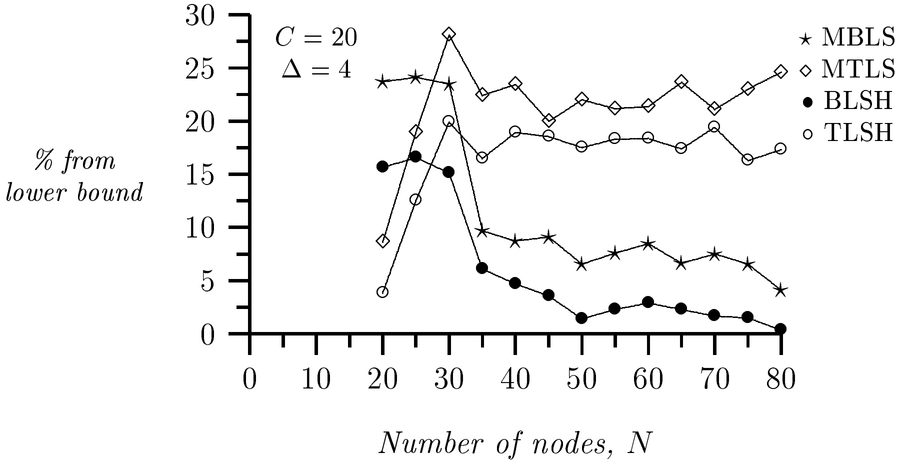


Figure 7.11: Algorithm comparison for $C = 20$ channels and $\Delta = 4$ tuning slots

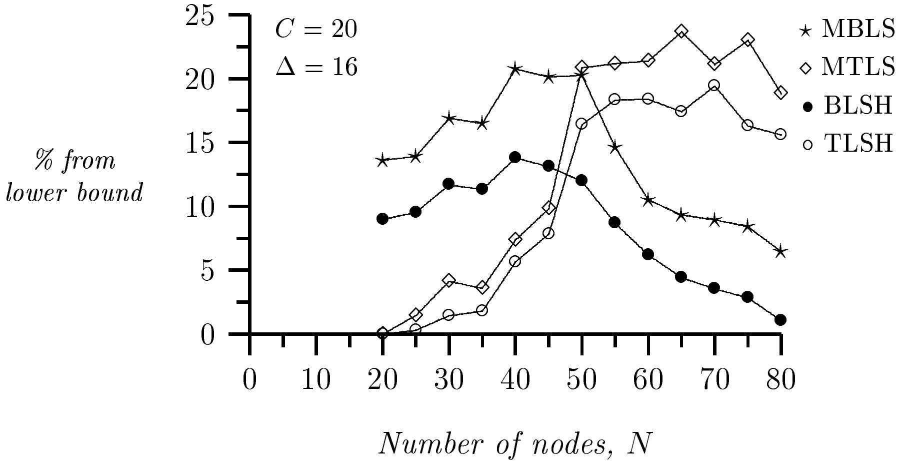


Figure 7.12: Algorithm comparison for $C = 20$ channels and $\Delta = 16$ tuning slots

Figure 7.6, especially within the bandwidth and tuning limited regions. In general, we have noticed that the quality of the schedules produced by the four algorithms is not significantly affected by the actual distribution of the traffic demands.

Based on the results presented here, 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 their running time requirements. The best algorithm for a given system will then depend not only on the region of operation, but also on the desired tradeoff between quality of the final schedule and speed.

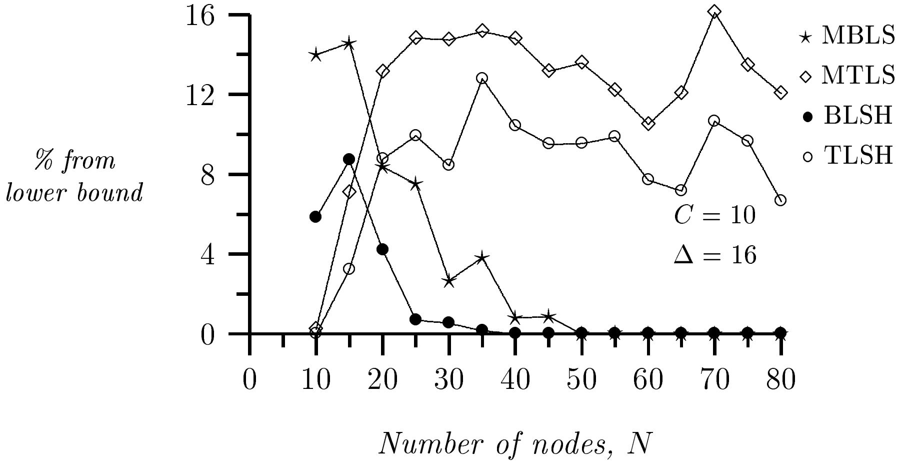


Figure 7.13: Algorithm comparison for $C = 10$ channels and $\Delta = 16$ tuning slots ($uniform(1, 40)$ distribution)

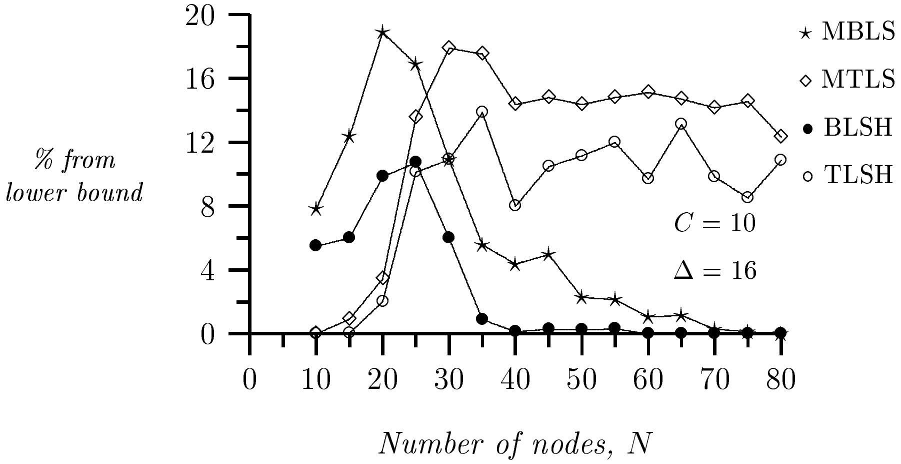


Figure 7.14: Algorithm comparison for $C = 10$ channels and $\Delta = 16$ tuning slots (bimodal distribution)

Chapter 8

Summary and Future Research

8.1 Summary

We have considered the problem of designing TDM schedules to accommodate arbitrary traffic demands in a broadcast optical network. Our objective was to investigate the effects of transceiver tuning latency on the length of the schedule, which is a measure of both delay and throughput. 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 (as determined by system parameters such as the number of nodes, the number of wavelengths, and the number of tuning slots), even large (relative to the packet transmission time) 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.

8.2 Future Research

Though we have outlined efficient algorithms for constructing optimal or near-optimal TDM schedules given the traffic matrix, there is ample scope for future research in developing distributed protocols which will implement these ideas in an actual network. In particular, issues related to how the global traffic matrix information is to be shared, whether the algorithm should be run at a central node or in a distriubuted fashion, and whether the signalling should be in-band or out-of-band need to be addressed. Currently we are working on a protocol that will reliably provide the global information required by the algorithms presented here.

Another direction for future research would be to mathematically analyze the throughput and delay performance of a network which uses the algorithms for TDM scheduling developed in this thesis and which has a specified arrival pattern. For a packet-switched network, it would be interesting to determine the load conditions under which the scheduling algorithm would result in the network queues being stable. For a virtual-circuit based network (an ATM network for example), it would be interesting to study the effect of the proposed scheduling algorithms on the Quality of Service (QoS) guarantees (like cell delay, cell loss and cell jitter). This is going the focus of our research in the near future.

Bibliography

[1] 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.

[2] M. Azizoglu, R. A. Barry, and A. Mokhtar. The effects of tuning time in banwidth-limited optical broadcast networks. In *Proceedings of INFOCOM '95*, pages 138–145. IEEE, April 1995.

[3] K. Bogineni, K. M. Sivalingam, and P. W. Dowd. Low-complexity multiple access protocols for wavelength-division multiplexed photonic networks. *IEEE Journal on Selected Areas in Communications*, 11(4):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. In *Proceedings of INFOCOM '95*, pages 129–136. IEEE, April 1995.

[5] Mon-Song Chen, N. R. Dono, and R. Ramaswami. A media-access protocol for packet-switched wavelength division multiaccess metropolitan area networks. *IEEE Journal on Selected Areas in Communications*, 8(6):1048–1057, August 1990.

[6] R. Chipalkatti, Z. Zhang, and A. S. Acampora. Protocols for optical star-coupler network using WDM: Performance and complexity study. *IEEE Journal on Selected Areas in Communications*, 11(4):579–589, May 1993.

[7] 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.

[8] A. Ganz and Y. Gao. Time-wavelength assignment algorithms for high performance WDM star based networks. *IEEE Transactions on Communications*, 42(4):1827–1836, April 1994.

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

[10] M. R. Garey and D. S. Johnson. *Computers and Intractability.* W. H. Freeman and Co., New York, 1979.

[11] 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.

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

[13] P. S. Henry. High-capacity lightwave local area networks. *IEEE Communication Magazine*, pages 20–26, October 1989.

[14] P. A. Humblet, R. Ramaswami, and K. N. Sivarajan. An efficient communication protocol for high-speed packet-switched multichannel networks. *IEEE Journal on Selected Areas in Communications*, 11(4):568–578, May 1993.

[15] M. Kovacevic, M. Gerla, and J. Bannister. Time and wavelength division multiaccess with acoustooptic tunable filters. *Journal of Fiber and Integrated Optics*, 12:113–132, August 1993.

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

[17] 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.

[18] G. N. Rouskas and M. H. Ammar. Analysis and optimization of transmission schedules for single-hop WDM networks. *IEEE/ACM Transactions on Networking*, 3(2):211–221, April 1995.

[19] G. N. Rouskas and M. H. Ammar. Minimizing delay and packet loss in single-hop lightwave WDM networks using TDM schedules. In *Proceedings of ICC '95*, pages 1267–1271, June 1995.

[20] S. Tridandapani, J. S. Meditch, and A. K. Somani. The MaTPi protocol: Masking tuning times through pipelining in WDM optical networks. In *Proceedings of INFOCOM '94*, pages 1528–1535. IEEE, June 1994.

Appendices

Appendix A

Proof that *OSTL* is $\mathcal{NP}$-complete for $C = 2$ Channels

We now show that *OSTL* is $\mathcal{NP}$-complete, even when the number of wavelengths $C = 2$. As in [11], the proof uses a transformation from *PARTITION*, a well-known $\mathcal{NP}$-complete problem [10] described below. However, our proof is substantially different, reflecting the fact that (a) the transformation is to an instance of *OSTL* with $C = 2$ channels, while the transformation in [11] is to an instance of *OS* with $C = 3$ channels, and (b) the tuning latency, Δ, is now an influencing parameter.

Problem A.1 (PARTITION) *Given a set* $\mathcal{V} = \{1, 2, \ldots, n\}$ *with* w_i *the weight of element i, and* $W = \sum_{i=1}^{n} w_i$*, is there a partition of* $\mathcal{V}$ *into two sets,* $\mathcal{V}_1$ *and* $\mathcal{V}_2$*, such that* $\sum_{i \in \mathcal{V}_1} w_i = \sum_{j \in \mathcal{V}_2} w_j = \frac{W}{2}$*? (The* w_i*'s may be assumed integer.)*

Proof (of Theorem 4.1). It is easy to see that *OSTL* is in the class $\mathcal{NP}$, since a nondeterministic algorithm need only (a) guess the optimal set of start slots $\{\tau_{ic}\}$ satisfying constraints (3.4) and (3.5), and (b) verify that the length of the schedule is less than or equal to M.

We now transform *PARTITION* to *OSTL*; note that it is sufficient to find a transformation for the case $C = 2$. Let $\mathcal{V} = \{1, 2, \ldots, n\}$ be the set of elements of weights $w_i, i = 1, \ldots, n$, making up an arbitrary instance of *PARTITION*, and let $W = \sum_{i=1}^{n} w_i$. We construct an instance of *OSTL* with $N = n + 1$ stations, $C = 2$

wavelengths, $\Delta = \frac{W^2}{2}$, $M = \frac{3W^2}{2} + W$, and a collapsed traffic matrix, $\mathbf{A}$, such that:

$$a_{i1} \; = \; w_i, \qquad a_{i2} \; = \; (W+1)\; w_i, \qquad i = 1, \ldots, n \tag{A.1}$$

$$a_{n+1,1} \; = W^2, \qquad a_{n+1,2} \; = \; 0 \tag{A.2}$$

It is obvious that this transformation can be performed in polynomial time. We now show that a schedule of length $\frac{3W^2}{2} + W$ exists for the above instance of *OSTL* if and only if $\mathcal{V}$ has a partition. If $\mathcal{V}$ has a partition, $\mathcal{V}_1, \mathcal{V}_2$, then there is a schedule of length equal to $\frac{3W^2}{2} + W$; one such schedule is shown in Figure A.1 (the initial tuning period of Δ slots is not shown there). Conversely, if an *OSTL* schedule of length $\frac{3W^2}{2} + W$ exists, then $\mathcal{V}$ has a partition. This is shown in the following.

Let $\mathcal{S} = \{\tau_{ic}\}, i = 1, \ldots, n+1, c = 1, 2$, be a schedule of length $\frac{3W^2}{2} + W$. The first $\Delta = \frac{W^2}{2}$ slots are used for tuning of the transmitters to the first channel in their respective channel sequence, so let us consider the remaining $W^2 + W$ slots. Also let our reference point (i.e., slot 1) be the first slot following these initial Δ slots. Since $\sum_{i=1}^{n} a_{i2} = W^2 + W$, all these slots of the schedule will contain a transmission by some source on channel λ_2. Now, the start slot, $\tau_{n+1,1}$, of station $n+1$ on channel λ_1, must be > 1. Otherwise, stations 1 through n are all assigned to transmit on channel λ_1 in slots $W^2 + 1$ through $W^2 + W$. But then, since $\Delta = \frac{W^2}{2}$, the hardware constraint (3.4) would not be satisfied for the station, say, i, assigned to transmit on channel λ_2 in slot $W^2 + \frac{W}{2}$, contradicting the hypothesis that $\mathcal{S}$ is an admissible schedule. Similarly, it can be shown that $\tau_{n+1,1} \leq W$; in other words, station $n+1$ is not assigned to transmit in slots $W+1$ through $W^2 + W$. As a result, stations 1 through n are divided into two sets, $\mathcal{V}_1$ and $\mathcal{V}_2$, such that all stations in $\mathcal{V}_1$ (respectively, $\mathcal{V}_2$) have start slots on channel λ_1 less than $\tau_{n+1,1}$ (respectively, greater than or equal to $\tau_{n+1,1} + a_{n+1,1}$).

Let $W_1 = \sum_{i \in \mathcal{V}_1} w_i$ and $W_2 = \sum_{j \in \mathcal{V}_2} w_j$. Without loss of generality, let $W_1 \geq W_2$; the case $W_1 \leq W_2$ is handled similarly. Then, $W_1 = \frac{W}{2} + e$, where $e \geq 0$. Since $\mathcal{S}$ satisfies the hardware constraints (3.4), we have that, for all $i \in \mathcal{V}_1$: $\tau_{i1} + a_{i1} + \Delta \leq \tau_{i2}$. But $\tau_{i1} \geq 1$ and $a_{i1} \geq 1$ for all stations 1 through n, therefore $\tau_{i2} \geq \Delta + 2 = \frac{W^2}{2} + 2$, for all $i \in \mathcal{V}_1$, leaving a total of at most $\frac{W^2+2W}{2} \Leftrightarrow 1$ slots for transmissions by stations in

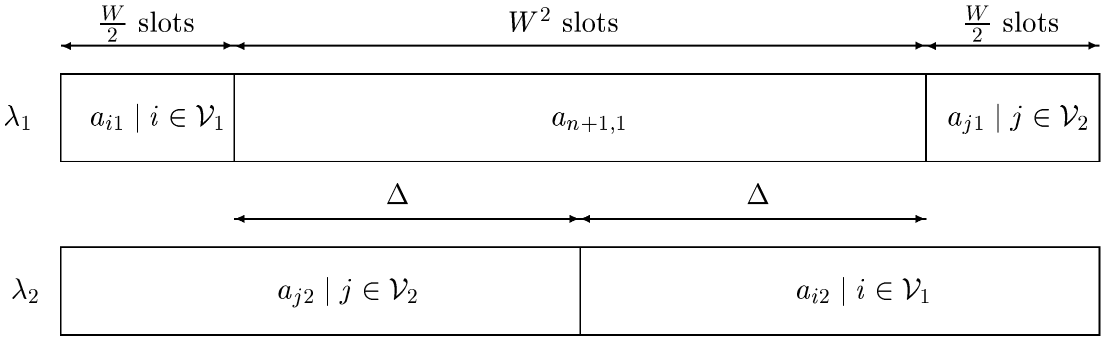

$\frac{W}{2}$ slots
W^2 slots
$\frac{W}{2}$ slots
λ_1
$a_{i1} \mid i \in \mathcal{V}_1$
$a_{n+1,1}$
$a_{j1} \mid j \in \mathcal{V}_2$
Δ
Δ
λ_2
$a_{j2} \mid j \in \mathcal{V}_2$
$a_{i2} \mid i \in \mathcal{V}_1$

Figure A.1: Optimum length schedule when $\mathcal{V}$ has a partition $\mathcal{V}_1, \mathcal{V}_2$ (the initial tuning period of $\Delta = \frac{W^2}{2}$ slots is not shown)

$\mathcal{V}_1$ on λ_2. Also, the total number of slots allocated to stations in $\mathcal{V}_1$ for transmissions on channel λ_2 must be equal to $\sum_{i \in \mathcal{V}_1} a_{i2} = (W+1)\left(\frac{W}{2}+e\right)$. Then,

$$\frac{W^2+2W}{2} \Leftrightarrow 1 \quad \geq \quad (W+1)\left(\frac{W}{2}+e\right) \quad \Rightarrow \quad e \quad \leq \quad \frac{W \Leftrightarrow 2}{2W+2} \quad < \quad \frac{1}{2} \;\; \forall\, W > 0 \quad (A.3)$$

Since $W_1 = \frac{W}{2} + e$ must also be an integer, e must be zero. In other words, $W_1 = W_2 = \frac{W}{2}$, and $\mathcal{V}_1, \mathcal{V}_2$ constitute a partition of $\mathcal{V}$. □

Appendix B

Modified Sufficiency Conditions for existence of an Optimal Schedule in a Bandwidth-Limited Network

The upper bound on the degree of non-uniformity ϵ of matrix $\mathbf{A}$ in order to guarantee a schedule of length equal to the lower bound in Lemma 5.1 (and as rewritten in equation (5.18)) is not tight, and can be improved upon. This is exactly what we intend to show now. We first prove the following Lemma :

Lemma B.1 *If constraints (5.11) on the elements of* $\mathbf{A}$ *hold, then for any subset* $\mathcal{T}$ *of transmitters from* $\{1, \ldots, N\}$ *such that* $\mid \mathcal{T} \mid = n$*, and any channel* λ_c*:*

$$\frac{nM^{(l)}}{N} \Leftrightarrow \frac{N\ \epsilon}{2} \quad \leq \quad \sum_{i \in \mathcal{T}} a_{i,c} \quad \leq \quad \frac{nM^{(l)}}{N} \ + \ \frac{N\ \epsilon}{2} \tag{B.1}$$

Proof (of Lemma B.1). Follows from equation (5.11) by using an argument very similar to the one used in the proof of Lemma 5.2. □

In the proof of Lemma 5.1 we had, in order to satisfy constraint (5.16), set $K_{c+1} \Leftrightarrow K_c = N.\epsilon + (M^{(l)}/N + \epsilon) + \Delta$, thereby ensuring a valid schedule, but this was an overkill in the sense that we are unnecesarily choosing such a large value for $K_{c+1} \Leftrightarrow K_c$ though the right side in equation (5.16) might be much smaller. So the logical thing to do would be to set the smallest possible value for $K_{c+1} \Leftrightarrow K_c$, so that we not only have a feasible schedule but also a schedule which is as small as possible. So we set $K_{c+1} \Leftrightarrow K_c$ to be the largest possible value of the right side in equation (5.16). Thus, if n_c is the value of i which maximizes the expression on the right side in the equation, we can set :

$$K_{c+1} \Leftrightarrow K_c = (\sum_{j=1}^{n_c-1} a_{j,c} \Leftrightarrow \sum_{j=1}^{n_c-1} a_{j,c+1}) + a_{n_c,c} + \Delta \qquad c = 1, \ldots, C \tag{B.2}$$

Writing out the above equation for all the channels (i. e. from $c = 1$ through $c = C$), and adding up the equations, we get :

$$K_{C+1} = \{ \sum_{j=1}^{n_1-1} a_{j,1} + \sum_{c=2}^{C} (\sum_{j=1}^{n_c-1} a_{j,c} \Leftrightarrow \sum_{j=1}^{n_{c-1}-1} a_{j,c}) \Leftrightarrow \sum_{j=1}^{n_C-1} a_{j,C} \} + \sum_{c=1}^{C} a_{n_c,c} + C\Delta \tag{B.3}$$

where we have combined the same-channel-terms, and used the fact that K_1 equals zero. The above equation can be expressed in more compact notation by introducing numbers n_0 and n_{C+1}, both equal to 1, as follows :

$$K_{C+1} = \sum_{c=1}^{C+1} (\sum_{j=1}^{n_c-1} a_{j,c} \Leftrightarrow \sum_{j=1}^{n_{c-1}-1} a_{j,c}) + \sum_{c=1}^{C} a_{n_c,c} + C\Delta \tag{B.4}$$

where $n_0 = n_{C+1} = 1$. Now define the set S to be a subset of $\{1, 2, \ldots, C\}$ such that $c \in S$ if and only if $n_c \geq n_{c-1}$. Then equation (B.4) can be written as :

$$K_{C+1} = \{ \sum_{c \in S} (\sum_{j=n_{c-1}}^{n_c-1} a_{j,c}) \Leftrightarrow \sum_{c \in \overline{S}} (\sum_{j=n_c}^{n_{c-1}-1} a_{j,c}) \} + \sum_{c=1}^{C} a_{n_c,c} + C\Delta \tag{B.5}$$

Let $|S| = k$. Then, from Lemma B.1 we have :

$$\sum_{c \in S} (\sum_{j=n_{c-1}}^{n_c-1} a_{j,c}) \leq [\sum_{c \in S} (n_c \Leftrightarrow n_{c-1})] \frac{M^{(l)}}{N} + \frac{kN\epsilon}{2} \tag{B.6}$$

and also :

$$\sum_{c \in \overline{S}} (\sum_{j=n_c}^{n_{c-1}-1} a_{j,c}) \geq [\sum_{c \in \overline{S}} (n_{c-1} \Leftrightarrow n_c)] \frac{M^{(l)}}{N} \Leftrightarrow \frac{(C \Leftrightarrow k)N\epsilon}{2} \tag{B.7}$$

Subtracting (B.7) from (B.6), and noting that all but one term cancel out on the right hand side, we have :

$$\sum_{c \in S} (\sum_{j=n_{c-1}}^{n_c-1} a_{j,c}) \Leftrightarrow \sum_{c \in \overline{S}} (\sum_{j=n_c}^{n_{c-1}-1} a_{j,c}) \leq \frac{CN\epsilon}{2} \tag{B.8}$$

Using this in equation (B.5) and using equation (5.11) to bound the value of $a_{i,c}$, we have an upper bound on K_{C+1} :

$$K_{C+1} \leq \frac{CN\epsilon}{2} + C(\frac{M^{(l)}}{N} + \epsilon) + C\Delta \tag{B.9}$$

Since we require that $M^{(l)} \geq K_{C+1}$, we can guarantee that if :

$$M^{(l)} \geq \frac{CN\epsilon}{2} + C(\frac{M^{(l)}}{N} + \epsilon) + C\Delta \tag{B.10}$$

or, equivalently, if :

$$\frac{\epsilon}{M^{(l)}/N} \leq \frac{2}{1 + 2/N}(\frac{1}{C} \Leftrightarrow \frac{1}{N} \Leftrightarrow \frac{\Delta}{M^{(l)}}) \tag{B.11}$$

Compare this to equation (5.18) and note that equation (B.11) almost gives a factor 2 improvement on the upper bound of the traffic non-uniformity. For the example of a system with $N = 100$, $C = 10$, and ignoring the term $\frac{\Delta}{M^{(l)}}$ (as in chapter 5), we get $\frac{\epsilon}{M^{(l)}/N} \approx .176$, and thus the variation of elements a_{ic} around $\frac{M^{(l)}}{N}$ can be up to 17.6% to guarantee a schedule of length $M^{(l)}$.

Appendix C

Proof of Optimality of *MBLS*

We now prove Theorem 5.1.

Proof (of Theorem 5.1). 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. Note that once the schedule length M and the gaps $g_{ic}, i = 1, \ldots, N \Leftrightarrow 1$, are known, the gap g_{Nc} after the last transmitter is uniquely determined. Therefore we are not interested in the gaps that follow the last transmitter on each channel, and 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 5.1. 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 (5.5) 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 $g_{ic}, i = 1, \ldots, N \Leftrightarrow 1, c = 2, \ldots C$, 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 $g_{i+1,c} \ldots g_{N-1,c}$ 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 $M > \sum_{i=1}^{N} a_{i1}$).