

# Minimizing Delay and Packet Loss in Single-Hop Lightwave WDM Networks using TDM Schedules

G. N. Rouskas  
Mostafa H. Ammar

Center for Communications and Signal Processing  
Department of Electrical and Computer Engineering  
North Carolina State University

TR-94/18  
September 1994

# Minimizing Delay and Packet Loss in Single-Hop Lightwave WDM Networks Using TDM Schedules

*George N. Rouskas*

*Mostafa H. Ammar*

Computer Science Department

College of Computing

North Carolina State University

Georgia Institute of Technology

Raleigh, NC 27695-8206

Atlanta, GA 30332-0280

## Abstract

We consider packet-switched lightwave WDM networks with stations equipped with tunable transmitters and fixed receivers. Access to each of the available channels is controlled by a *weighted* TDM scheme, whereby the channels are not necessarily shared equally among the various sources. While previous work focused on the throughput characteristics of such schemes, in this paper we study the problem of designing TDM frames to minimize the mean packet delay, as well as the mean packet loss probability given a finite buffer capacity. The corresponding optimization problems are shown to be hard allocation problems, and heuristics to obtain near-optimal solutions are developed. For (potentially non-uniform) communication patterns common to parallel and distributed computations, our approach represents a significant improvement over I-TDMA\*. In addition, the margin of improvement increases with the size of the network.

# 1 Introduction

Wave division multiplexing (WDM) is currently believed to be the most promising technology for implementing a new generation of computer communication networks that fully exploit the vast information-carrying capacity of single-mode fiber [10]. By carving the bandwidth of the optical medium into multiple concurrent channels, each on a different wavelength, WDM has the potential of delivering an aggregate throughput that can be in the order Terabits per second. At the same time, WDM has introduced a new set of media-access problems, on which a great deal of recent research has been devoted. In this paper we focus on one of the candidate network architectures, namely the single-hop systems (see [17] for an overview).

As their name implies, single-hop networks provide one hop communication between any source-destination pair, by allowing the various stations to select any of the available channels for packet transmission/reception. Access to the channels can be based on a reservation scheme that requires the use of one [11, 16, 4, 5] or more [14] separate control channels (the last reference also contains a performance comparison of some of the schemes that have appeared in the literature). Alternatively, a hybrid WDM-TDM approach may be employed, in which case the bandwidth of each channel may be preallocated to each of the sources by means of a transmission schedule that indicates the slots in which the various stations may access the available channels [6, 2].

This work explores the delay and packet loss probability behavior of transmission schedules. In particular, we are interested in developing schedules that will have good performance under the (potentially non-uniform) traffic patterns one expects to encounter in realistic parallel and distributed computing environments. Because of the complexity of the resulting optimization problems, a large part of our work is devoted to developing heuristics that (a) construct schedules to achieve near-optimal mean queue length, and (b) allocate the buffer capacity of each station to the various channels to attain an acceptable packet loss probability. Numerical results to be presented indicate that the performance gains over I-TDMA\* [2] are significant, and increase with the size of the network.

The remainder of the paper is organized as follows. In Section 2 we present the essential features of our system, and in Section 3 we show how to select the minimum frame length to ensure stability. A heuristic for the problem of minimizing the mean packet delay across the network is developed in Section 4. Section 5 addresses the problem of how to allocate

the finite buffer capacity at each station to minimize the packet loss probability. We present numerical results in Section 6, and summarize our work in Section 7.

## 2 System Model

We consider a network of  $N$  stations, each equipped with one receiver and one transmitter, interconnected through an optical broadcast medium that can support  $C$  wavelengths,  $\lambda_1, \lambda_2, \dots, \lambda_C$ . Each wavelength can be considered as a channel<sup>1</sup> operating at a data rate accessible by the electronic interfaces at each station, say a few Gigabits per second. In order to access the various channels, the stations must employ tunable transmitters and/or receivers. For simplicity, in this paper we will only consider systems with tunable transmitters and fixed receivers; all our results can be easily extended to systems with tunable receivers.

We let  $\lambda(i) \in \{\lambda_1, \dots, \lambda_C\}$ , denote the wavelength assigned to the fixed receiver of station  $i$ ; in other words,  $i$  may only receive packets transmitted on channel  $\lambda(i)$ . The tunable transmitters at each station, on the other hand, are lasers that can be tuned to, and transmit on any and all wavelengths  $\lambda_c, c = 1, \dots, C$ . An important parameter in such a system is the *tuning delay*, or the time it takes a transmitter to tune from one wavelength to another. The tuning delay may vary with the transmitters and/or the wavelength pairs.

The network operates in a slotted mode, with a slot time equal to the packet transmission time plus the maximum tuning delay; the maximum is taken over all transmitters and all wavelength pairs. All stations are synchronized to the slot boundaries [21]. A *collision* will occur if two or more transmitters access the same channel in a given slot. All packets involved in a collision are considered lost, and recovery is assumed to take place via a higher level protocol.

We define  $\sigma_i$  as the probability that a new packet arrives at station  $i$  during a slot time. We also let  $p_{ij}$  denote the probability that a packet arriving at station  $i$  has station  $j$  as its destination, and  $\sum_j p_{ij} = 1$ . Thus,  $\Sigma = [\sigma_i p_{ij}]$  is the matrix of externally offered traffic, in units of packets per slot.

---

<sup>1</sup>Throughout this paper we will use the terms “wavelength” and “channel” interchangeably.

## 2.1 Channel Sharing

If the number of wavelengths,  $C$ , is equal to the number of stations,  $N$ <sup>2</sup>, the fixed receiver at each station  $i$  is assigned a unique wavelength,  $\lambda(i) \in \{\lambda_1, \dots, \lambda_C\}$ , or *home* channel. Consequently, total *optical self-routing* [2, 7] is achieved, as a station only receives packets destined to itself. While optical self-routing is clearly a desirable feature, technical considerations make it difficult to achieve. Current WDM technology supports only a small number of wavelengths in a single mode fiber, and self-routing architectures are unsuitable for anything but trivial networks. On the other hand, in order to keep channel utilization at high levels and take full advantage of the optical bandwidth, the maximum tuning delay should only constitute a small fraction of the slot time. However, there is a tradeoff between the tuning range and tuning speed in state of the art tunable lasers and optical filters [3]; in other words, the fastest tunable transceivers may only tune over a small portion of the optical bandwidth. Even if a large number of wavelengths was available within a fiber, the need to employ *very fast* tunable optical transceivers would limit the actual number of channels that can be accessed. To allow for network scalability<sup>3</sup>, all the techniques developed in this work are applicable in the general case, i.e., for a number of wavelengths less than or equal to the number of stations.

Whenever  $C < N$ , a number of receivers have to be assigned the same wavelength, and share a single channel  $\lambda_c, c = 1, \dots, C$ . We let  $R_c$ , a subset of  $\{1, \dots, N\}$ , denote the set of receivers that share channel  $\lambda_c$ ,

$$R_c = \{j \mid \lambda(j) = \lambda_c\} \quad c = 1, \dots, C \quad (1)$$

For the moment, we will assume that  $R_c$  are given for all  $\lambda_c$ ; how these sets may be determined using some optimality criteria has been discussed in detail in [19], and is repeated, for the sake of completeness, in Appendix A.

Note that a station  $j \in R_c$  will receive all packets transmitted on channel  $\lambda_c$ , and will have to filter out those addressed to stations  $j' \in R_c, j' \neq j$ . Thus, only *partial* self-routing is achieved. On the other hand, these networks have a *limited multicast capability* [20], i.e.,

---

<sup>2</sup>With one transceiver per station, at most  $N$  channels may be accessed during any given slot, thus we do not consider the case  $C > N$ .

<sup>3</sup>The maximum number of stations in the network is limited by the power budget [12]; what is implied is that it should not be wavelength-limited

a packet transmitted on  $\lambda_c$  may reach any subset of stations in  $R_c$ . We now define

$$q_{ic} = \sigma_i \sum_{j \in R_c} p_{ij} \quad i = 1, \dots, N, \quad c = 1, \dots, C \quad (2)$$

as the probability that a packet with destination  $j \in R_c$  arrives at  $i$  within a slot. We also assume that each station has  $C$  buffers, one for storing packets that need to be transmitted on a particular channel. Having  $C$  separate queues eliminates the head of line effects of a single buffer, and helps to drastically improve the throughput and delay characteristics as demonstrated in [2] for I-TDMA\*<sup>4</sup>. We also assume that the buffer for channel  $\lambda_c$  at station  $i$  has a capacity of  $L_{ic}$  packets; packets arriving to find a full buffer are lost.

## 2.2 Transmission Schedules

Time Division Multiplexing (TDM) is a well-known media access scheme. The Interleaved TDMA (I-TDMA\*) protocol [2] is an extension of TDM over a multi-channel environment. In I-TDMA\*, time slots are grouped into frames of  $N$  or  $N - 1$  slots (depending on whether  $C < N$  or  $C = N$ , respectively), and each station has a chance to transmit on each channel exactly once per frame. Intuitively, I-TDMA\* will have very good performance under uniform traffic (i.e., when  $\sigma_i = \sigma_k, p_{ij} = p_{kl} \forall i, j, k, l$ ). It is expected, however, and will be demonstrated later, that its performance will degrade under non-uniform traffic loads one expects to encounter in realistic distributed and parallel computing environments [8]. Here, we are concerned with *weighted* TDM schemes, a generalization of I-TDMA\*, whereby stations do not share the channels equally.

In a weighted TDM scheme each frame consists of  $M \geq N$  slots. Within each frame, source  $i$  is allowed to transmit on channel  $\lambda_c$  in exactly  $a_{ic}(M)$  slots,  $1 \leq a_{ic}(M) \leq M$ . In these slots,  $i$  may transmit a packet to any station<sup>5</sup>,  $j \in R_c$ , with a receiver listening on wavelength  $\lambda_c$ . A *transmission schedule* indicates, for all  $i$  and  $c$ , which slots within a frame can be used for transmissions from  $i$  on wavelength  $\lambda_c$ , and can be described by the variables

---

<sup>4</sup>Note that a station does not need to support  $C$  distinct buffers; one physical buffer is sufficient. Packets to be transmitted on the same wavelength may be appropriately linked to give the appearance of  $C$  buffers.

<sup>5</sup>Or subset of stations, for a multicast packet.

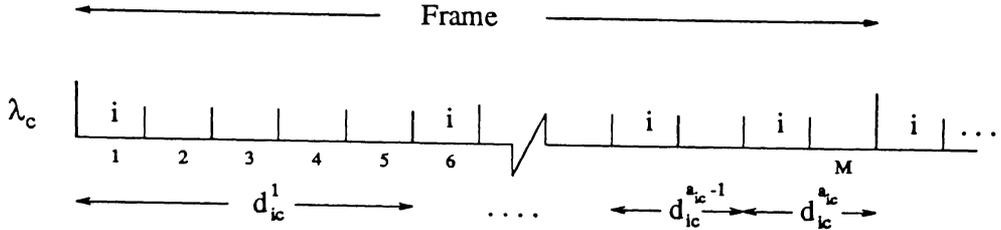


Figure 1: Definition of  $d_{ic}^{(k)}$  for  $k = 1, \dots, a_{ic}(M)$

$\delta_{ic}^{(t)}$ ,  $t = 1, 2, \dots, M$ , called *permissions*, and defined as

$$\delta_{ic}^{(t)} = \begin{cases} 1, & \text{if station } i \text{ has permission to transmit on channel } \lambda_c \text{ in slot } t \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

Obviously,  $a_{ic}(M) = \sum_{t=1}^M \delta_{ic}^{(t)}$ . For  $k = 1, 2, \dots, a_{ic}(M)$ , we let  $d_{ic}^{(k)}$  denote the distance, in slots, between the beginning of the  $k$ -th slot that  $i$  has permission to transmit on  $\lambda_c$ , and the beginning of the next such slot, in the same or the next frame (see Figure 1).

**Definition 1** *A schedule of frame length  $M$  provides full connectivity in the strong sense iff it satisfies the following three conditions:*

$$q_{ic} > 0 \Rightarrow a_{ic}(M) \geq 1 \quad \forall i, c \quad (4)$$

$$\sum_{c=1}^C \delta_{ic}^{(t)} \leq 1 \quad \forall i, t \quad (5)$$

$$\sum_{i=1}^N \delta_{ic}^{(t)} = 1 \quad \forall c, t \quad (6)$$

Condition (4) specifies that, if the traffic originating at station  $i$  and terminating at stations listening on wavelength  $\lambda_c$  is nonzero, then there is at least one slot per frame in which  $i$  may transmit on wavelength  $\lambda_c$ . This guarantees full connectivity among the network stations. Constraint (5) requires that each station be given permission to transmit on at most one channel within a slot  $t$ . Finally, constraint (6) implies that exactly one source may transmit on a given channel within a slot  $t$ . The last two constraints guarantee a collision-free operation.

**Definition 2** *A schedule of frame length  $M$  provides full connectivity in the weak sense iff it satisfies (4).*

Schedules providing full connectivity in the weak sense do not guarantee collision-free operation, but may be defined for very short frame lengths. As a result, it has been shown [18, 19] that these schedules may significantly improve the overall network performance in terms of both throughput and response collection cost [1]. Techniques to construct such schedules were developed in [18, 19]. Since these techniques may directly be applied to this work, in this paper we will focus on schedules which provide full connectivity in the strong sense. By summing over all  $t = 1, \dots, M$ , constraints (5) and (6) imply that the following two conditions hold:

$$\sum_{c=1}^C a_{ic}(M) \leq M \quad \forall i \quad (7)$$

$$\sum_{i=1}^N a_{ic}(M) = M \quad \forall c \quad (8)$$

The first condition specifies that a source may not be given permission to transmit in more than  $M$  slots within a frame, while the second requires that exactly  $M$  slots contain permissions for transmission on each channel. In [18, Appendix B] it has been shown that (7) and (8) are also sufficient for constructing a schedule that satisfies (5) and (6), leading to the following corollary:

**Corollary 1** *A schedule of frame length  $M$  providing full connectivity in the strong sense exists iff (4), (7), and (8) are satisfied.*

I-TDMA\* is a special case of such a schedule with  $M = N$  (for  $C < N$ ) or  $M = N - 1$  (for  $C = N$ ), and  $a_{ic}(M) = 1 \quad \forall i, c$ .

Figure 2 illustrates the schedules providing full connectivity in the strong sense for a network with  $N = 4$  stations and  $C = 2$  wavelengths. Channel  $\lambda_1$  is shared by the receivers of stations 1 and 3 ( $R_1 = \{1, 3\}$ ), while channel  $\lambda_2$  is shared by the receivers of stations 2 and 4 ( $R_2 = \{2, 4\}$ ). Figure 2(a) shows the I-TDMA\* schedule, whereby each station is given exactly one permission within a frame to transmit on each channel. Figure 2(b) shows one possible weighted TDM schedule which gives a different number of permissions per frame to each source-channel pair. Also note that no collisions are possible under either schedule.

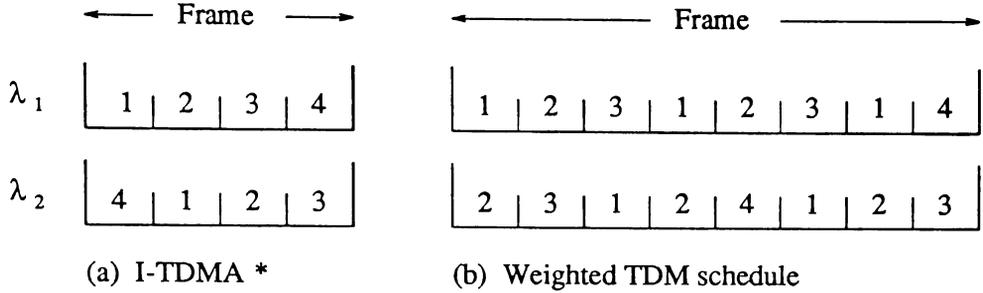


Figure 2: (a) I-TDMA\*, and (b) weighted schedule providing full connectivity in the strong sense, for a network with  $N = 4, C = 2, R_1 = \{1, 3\}, R_2 = \{2, 4\}$

### 3 Selecting the Frame Length to Insure Stability

Let us now suppose that the  $C$  buffers at each station have infinite capacity ( $L_{ic} = \infty \forall i, c$ ), and that the sets,  $R_c$ , of stations sharing channel  $\lambda_c$  have been decided upon. Observe that the buffers for distinct source-destination pairs do not interact, and thus are independent. Consequently, if the frame length is  $M$ , the following condition is necessary and sufficient for stability <sup>6</sup> [15]:

$$Mq_{ic} < a_{ic}(M) \quad \forall i, c \quad (9)$$

The above requires that the number of slots within a frame in which station  $i$  is permitted to transmit on channel  $\lambda_c$  be greater than the number of packets destined to a receiver listening on  $\lambda_c$  that are expected to arrive at station  $i$  during a number of slots equal to the frame length. As we shall see, the poor performance of I-TDMA\* under non-uniform traffic loads arises from its failure to satisfy the above condition, even in situations when the load offered to each channel is less than 1.

We now define  $b_{ic}(M)$  and  $f_{ic}(M)$  such that:

$$Mq_{ic} = b_{ic}(M) + f_{ic}(M), \quad b_{ic}(M) \text{ integer, and } 0 \leq f_{ij}(M) < 1 \quad (10)$$

Therefore,  $b_{ic}(M) + 1$  is a lower bound on  $a_{ic}(M)$  if the stability condition (9) is to be satisfied:

$$b_{ic}(M) + 1 \leq a_{ic}(M) \quad \forall i, c \quad (11)$$

---

<sup>6</sup>Note that, because of (8), (9) implies that  $\sum_{i=1}^N q_{ic} < 1 \forall c$ , i.e., that the load offered to each channel is less than 1. The latter, in turn, implies that  $\sum_{i=1}^N \sigma_i < C$ , or that the total offered load does not exceed the network capacity.

Because of (7) and (8) the following two conditions must hold:

$$\sum_{c=1}^C (b_{ic}(M) + 1) \leq M \quad \forall i \quad (12)$$

$$\sum_{i=1}^N (b_{ic}(M) + 1) \leq M \quad \forall c \quad (13)$$

However, since  $f_{ic}(M) < 1$ , it is easy to see that, unless  $M$  is sufficiently large, (12) and/or (13) may be violated, making it impossible to have  $ba_{ic}(M) \geq b_{ic}(M) + 1$  as required by 9 and 10. We now show how to select  $M$  so that (12) and (13) are satisfied.

**Transmissions To A Channel.** Consider channel  $\lambda_c$ , and select a frame length,  $M'_c$ , such that:

$$M'_c \sum_{i=1}^N q_{ic} \leq M'_c - N \Leftrightarrow M'_c \geq \frac{N}{1 - \sum_{i=1}^N q_{ic}} \quad (14)$$

For this frame length,  $M'_c$ , we get using (10):

$$\sum_{i=1}^N b_{ic}(M'_c) \leq M'_c \sum_{i=1}^N q_{ic} \leq M'_c - N \Leftrightarrow \sum_{i=1}^N (b_{ic}(M'_c) + 1) \leq M'_c \quad (15)$$

Thus, by selecting  $M' \geq \max_c \{M'_c\}$  we ensure that (13) is satisfied.

**Transmissions From A Station.** By proceeding as above, we consider source  $i$  and select a frame length  $M''_i$  such that

$$M''_i \sum_{c=1}^C q_{ic} \leq M''_i - C \Leftrightarrow M''_i \geq \frac{C}{1 - \sum_{c=1}^C q_{ic}} = \frac{C}{1 - \sigma_i} \quad (16)$$

We now have for source  $i$ :

$$\sum_{c=1}^C b_{ic}(M''_i) \leq M''_i \sum_{c=1}^C q_{ic} \leq M''_i - C \Leftrightarrow \sum_{c=1}^C (b_{ic}(M''_i) + 1) \leq M''_i \quad (17)$$

Thus, frame length  $M'' \geq \max_i \{M''_i\}$  ensures that (12) is satisfied. Obviously, in order to satisfy both (12) and (13),  $M$  has to be selected as

$$M \geq \max\{M', M''\} \quad (18)$$

## 4 Minimization of the Average Packet Delay

We now turn our attention to the issue of constructing schedules such that the average packet delay over all source-channel pairs is minimized. Thus, we are seeking a solution to the following optimization problem.

**Problem 1** *Given the number of stations,  $N$ , the number of available wavelengths,  $C$ , and the traffic parameters,  $\sigma_i p_{ij}$ ,  $i, j = 1, \dots, N$ , find a schedule such that the network-wide average packet delay is minimized, assuming that buffers of infinite capacity are available at each station.*

There are three dimensions to this problem <sup>7</sup>:

- the sets of receivers,  $R_c$ , sharing wavelength  $\lambda_c$ ,  $c = 1, \dots, C$ , must be constructed,
- the number of slots per frame,  $a_{ic}(M)$ , allocated to each source-channel pair  $(i, \lambda_c)$  must be obtained, and
- a way of placing the  $a_{ic}(M)$  slots within the frame, for all  $i, \lambda_c$ , must be determined.

A similar problem was addressed in [13] for a single-channel network, and it was shown that the optimization yields a very hard allocation problem. The corresponding multi-channel optimization problem is expected to be even harder as the minimization is over all possible partitions of the set of receivers,  $\{1, 2, \dots, N\}$ , into sets  $R_c$ ,  $c = 1, \dots, C$ . Our approach, then, is to first present a heuristic to obtain near-optimal schedules assuming that sets  $R_c$  are known. The issue of constructing these sets to achieve load balancing is addressed in Appendix A.

Recall that the buffers for each source-channel pair are independent; therefore, if we consider each channel in isolation, all the results obtained in [13] will be applicable. We now review these results, which provide a lower bound on the multi-channel problem, *given a partition of  $\{1, \dots, N\}$  into sets  $R_c$ .*

Consider channel  $\lambda_c$ ; it has been shown that the average packet delay is minimized when:

---

<sup>7</sup>This is just a logical decomposition of the optimization problem. The order in which the three subproblems are presented is irrelevant as the subproblems are interdependent, and an exact solution method would simultaneously resolve all of them.

- The percentage of time station  $i$  is permitted to transmit on channel  $\lambda_c$  is [13, Eq. (3.3)]

$$x_{ic} = q_{ic} + \left(1 - \sum_{k=1}^N q_{kc}\right) \frac{\sqrt{1 - q_{ic}}}{\sum_{k=1}^N \sqrt{1 - q_{kc}}} \quad \forall i \quad (19)$$

- For each source,  $i$ , the  $a_{ic}(M)$  permissions for  $i$  to transmit on channel  $\lambda_c$  are equally spaced within the frame, i.e.,

$$\forall i : d_{ic}^{(k)} = d_{ic} = \frac{1}{x_{ic}}, \quad k = 1, \dots, a_{ic}(M) \quad (20)$$

Note that  $x_{ic}$  and  $d_{ic}$  are independent of  $M$ . Given a frame length,  $M$ , satisfying (18), we assign a number of slots to the source-channel pair  $(i, \lambda_c)$  such that <sup>8</sup>:

$$b_{ic}(M) + 1 \leq a_{ic} \leq \lceil M x_{ic} \rceil \quad (21)$$

and constraints (7) and (8) hold.

## 4.1 Slot Allocation

Once  $a_{ic}(M)$  have been determined for all  $i$  and  $\lambda_c$ , we need to construct the schedule so that the permissions assigned to each source-channel pair are placed within the frame according to (20). However, this is not feasible in general, even in the single-channel case, as  $d_{ic}$  may not be integers. Even if they are, scheduling the transmissions between all sources and channels in equally spaced slots may violate constraints (5) and (6). To overcome this problem, a golden-ratio policy was developed in [13], which requires that the frame length be a Fibonacci number. It was also shown that this policy places the various permissions within the frame in intervals close to the ones dictated by (20), and, as a result, it achieves an average packet delay very close to the lower bound.

Our approach is to use the golden ratio policy to place the permissions within each channel independently of the others. This, however, may result in allocations that violate (5). In other words, considering channels in isolation may cause a source to be assigned to transmit on two or more channels in the same slot. If this occurs, we must rearrange the schedule to remove these violations (recall that, from Corollary 1, this is always possible, since  $a_{ic}(M)$  satisfy both (7) and (8), and thus, a schedule providing full connectivity in the strong sense

---

<sup>8</sup>Because of (11), we may not use  $\lfloor M x_{ij} \rfloor$  as a lower bound for  $a_{ic}(M)$ , as that may be equal to  $b_{ic}(M)$ .

always exists). To this end, we use algorithm REARRANGE, described in [19], with a worst case complexity of  $O(N^2M^2)$ .

We now propose the following Slot Allocation Heuristic.

### Slot Allocation Heuristic (SAH)

1. If  $C < N$ , use the Weight Balancing Heuristic (WBH, see Appendix A) to determine the set of receivers,  $R_c, c = 1, \dots, C$ , that share each channel.
2. Select the smallest Fibonacci number,  $M$ , that satisfies (18), and obtain  $a_{ic}(M)$  from (21) so that (7) and (8) hold.
3. Let  $c = 1$ , and use the golden ratio policy [13] to place the  $a_{ic}(M), i = 1, \dots, N$ , slots in a frame for transmissions on channel  $\lambda_c$ . Repeat for  $c = 2, \dots, C$  to obtain an initial schedule,  $S_0(M)$ .
4. Run algorithm REARRANGE, described in [19], to perturb  $S_0$ , producing a schedule,  $S(M)$ , satisfying constraints (5) and (6).
5. Repeat Steps 2 through 4 for the next Fibonacci number, up to an upper limit,  $M_{max}$ . Select the frame length,  $M$ , and schedule,  $S(M)$ , that yields the lowest average delay.

## 5 Minimization of the Packet Loss Probability

In the previous section we presented a heuristic to minimize the average packet delay, assuming that the each station has  $C$  infinite-size queues. Typically, however, the total number of buffers available at each station, which we will denote by  $L_{i,max}$ , is finite. The problem that arises then, and which we will address in this section, is now described.

**Problem 2** *Given the number of stations,  $N$ , the number of available wavelengths,  $C$ , the traffic parameters,  $\sigma_i p_{ij}, i, j = 1, \dots, N$ , and the maximum number of buffers at each station,  $L_{i,max}, i = 1, \dots, N$ , (a) find a schedule, and (b) for all stations  $i$ , determine the buffer size,  $L_{ic}$ , for packets waiting for transmission on channel  $\lambda_c$ , so that  $\sum_{c=1}^C L_{ic} = L_{i,max}$ , and the network-wide probability of packet loss is minimized.*

Thus, in addition to the three subproblems that comprise Problem 1, the  $L_{i,max}$  buffers at each station have to be optimally partitioned into  $C$  queues. Recall, however, that a solution to Problem 1 minimizes the average packet delay, or, equivalently, *the expected queue size* of the  $CN$  buffers. We conjecture that, *regardless of the buffer sizes*,  $L_{ic} \forall i, c$ , the packet loss probability is minimized when the conditions specified by (19) and (20) are satisfied. Our approach, then, WILL be to assume that the schedule for Problem 2 will be constructed as discussed in the previous section, and will then consider how to partition the buffers at each station so as to minimize the packet loss probability.

## 5.1 Analysis

We now derive a lower bound for the packet loss probability. This lower bound is based on the observation that the mean queue length for the source-channel pair  $(i, \lambda_c)$  is minimized when  $d_{ic}$  in (20) is integer, and  $i$  is assigned to transmit on  $\lambda_c$  in slots which are exactly  $d_{ic}$  slots apart. Since the buffers for each source-channel pair are independent, we may consider pair  $(i, \lambda_c)$  in isolation.

We observe the system at the instants just before the beginning of slots in which  $i$  may transmit on  $\lambda_c$ . Consider the  $l$ -th such slot. We define  $r_{ic}^{(l)}(n, L_{ic})$  as the probability that  $i$  has  $n$  packets in its buffer (of size  $L_{ic}$ ) for  $\lambda_c$  at the beginning of the  $l$ -th slot,  $0 \leq n \leq L_{ic}$ . We also define  $P_{ic}(v)$  as the probability that  $v$  packets for  $\lambda_c$  arrive at  $i$  in the  $d_{ic}$  slots between the beginning of the  $l$ -th slot and the beginning of the  $(l+1)$ -th slot:

$$P_{ic}(v) = \begin{cases} \binom{d_{ic}}{v} q_{ic}^v (1 - q_{ic})^{d_{ic}-v}, & 0 \leq v \leq d_{ic} \\ 0, & \text{otherwise} \end{cases} \quad (22)$$

$P_{ic}(v > d_{ic})$ , the probability that more than  $v$  packets arrive at  $i$  in the  $d_{ic}$  slots can be similarly defined.

$i$  will have  $n, n = 1, \dots, L_{ic} - 1$ , packets for  $\lambda_c$  at the beginning of the  $l$ -th slot if (a)  $i$  had  $n + 1$  packets at the beginning of slot  $l - 1$ , transmitted one on  $\lambda_c$  in the  $(l - 1)$ -th slot, and no packets arrived since, and (b)  $i$  had  $n - v$  packets, transmitted one, and  $v + 1$  packets arrived<sup>9</sup>. Similar observations can be made for  $r_{ic}^{(l)}(L_{ic}, L_{ic})$ . We can then write the

---

<sup>9</sup>Except when  $v = n$ , in which case we require that  $n$  packets arrive. In (23) this case is covered by using  $\min\{n, v + 1\}$ .

following set of recursive equations for  $l = 2, 3, \dots$

$$\begin{aligned} r_{ic}^{(l)}(n, L_{ic}) &= r_{ic}^{(l-1)}(n+1, L_{ic})P_{ic}(0) \\ &+ \sum_{v=0}^n r_{ic}^{(l-1)}(n-v, L_{ic})P_{ic}(\min(n, v+1)) \quad n = 1, \dots, L_{ic} - 1 \end{aligned} \quad (23)$$

$$r_{ic}^{(l)}(L_{ic}, L_{ic}) = \sum_{v=0}^{L_{ic}} r_{ic}^{(l-1)}(L_{ic}-v, L_{ic})P_{ic}(\geq \min(L_{ic}, v+1)) \quad (24)$$

$$r_{ic}^{(l)}(0, L_{ic}) = 1 - \sum_{n=1}^{L_{ic}} r_{ic}^{(l)}(n, L_{ic}) \quad (25)$$

$$r_{ic}^{(1)}(n, L_{ic}) = 0 \quad 1 \leq n \leq L_{ic}, \quad r_{ic}^{(1)}(0, L_{ic}) = 1 \quad (\text{Initial Conditions}) \quad (26)$$

The initial conditions (26) are obtained by assuming that the frame starts at a slot in which  $i$  may transmit on  $\lambda_c$ .

If the stability condition (9) is satisfied, the system will eventually reach a steady state such that:

$$r_{ic}(n, L_{ic}) = \lim_{l \rightarrow \infty} r_{ic}^{(l)}(n, L_{ic}), \quad n = 1, \dots, L_{ic} \quad (27)$$

As an example, when  $L_{ic} = 1$  we have that

$$r_{ic}(1, 1) = 1 - r_{ic}(0, 1) = 1 - (1 - q_{ic})^{d_{ic}} \quad (28)$$

while for  $L_{ic} = 2$  we get

$$r_{ic}(2, 2) = r_{ic}(2, 2)P_{ic}(1) + P_{ic}(\geq 2) \quad (29)$$

$$r_{ic}(1, 2) = r_{ic}(2, 2)\{P_{ic}(0) - P_{ic}(1)\} + P_{ic}(1) \quad (30)$$

In general,  $r_{ic}(n, L_{ic})$  may be determined by either solving the set of linear equations that result from (23), (24), and (25), when we replace  $r_{ic}^{(l)}(n, L_{ic})$  with  $r_{ic}(n, L_{ic})$ , or by iteratively solving (23), (24), and (25) for  $r_{ic}^{(l)}(n, L_{ic}), l = 2, 3, \dots$ , until they converge to  $r_{ic}(n, L_{ic})$ . Then the probability of a packet arriving at station  $i$  been lost, *given that the packet is destined to a receiver listening on  $\lambda_c$  and the buffer for that channel has a capacity of  $L_{ic}$  packets* is

$$Q_{ic}(L_{ic}) = \sum_{n=0}^{L_{ic}} r_{ic}(n, L_{ic})P_{ic}(> L_{ic} - n) \quad (31)$$

The probability that a packet arriving at station  $i$  has to be transmitted on channel  $\lambda_c$  is just  $q_{ic}/\sigma_i$ . Therefore, the probability of packet loss for packets arriving at station  $i$  given a partition of the  $L_{i,max}$  buffers into  $C$  queues of sizes  $L_{i1}, \dots, L_{iC}$  is

$$Q_i(L_{i1}, \dots, L_{iC}) = \frac{1}{\sigma_i} \sum_{c=1}^C q_{ic} Q_{ic}(L_{ic}) \quad L_{i1} + \dots + L_{iC} = L_{i,max} \quad (32)$$

## 5.2 Buffer Partitioning

We now concentrate on the problem of partitioning the buffers available at the network stations so that the packet loss probability is minimized. As before, due to buffer independence, we may consider each source  $i$  separately. Our problem then reduces to obtaining, for all  $i$ , queue sizes  $L_{i1}, \dots, L_{iC}$ , such that  $Q_i$  as given in (32) is minimized. One straightforward method is to enumerate all potential partitions,  $L_{i1}, \dots, L_{iC}$ , of the  $L_{i,max}$  buffers into  $C$  queues and evaluate the packet loss probability of each using (32). The best partition is the one that yields the minimum such value.

The approach just described is obviously not feasible as it is prohibitively time consuming even for moderate values of  $C$  and  $L_{i,max}$ . We thus adopt a heuristic approach aimed at obtaining a near-optimal partition. Our approach is based on the observation that  $Q_{ic}(L') \leq Q_{ic}(L)$  for  $L' > L$ , and that  $L_{ic} \geq 1 \forall c$ . The following heuristic starts with queue sizes  $L_{ic} = 1 \forall c$  and, assuming that there are more buffers to be allocated, iteratively increments the queue size for the channel  $\lambda_c$  that results in the highest decrease for the packet loss probability as given in (32).

### Buffer Partitioning Heuristic (BPH)

1. Initialize  $L_{i1} \leftarrow \dots \leftarrow L_{iC} \leftarrow 1$  and  $L_{i,max} \leftarrow L_{i,max} - C$ . Repeat Step 2 while  $L_{i,max} \geq 1$ .
2. Find the channel  $\lambda_c$  for which  $|q_{ic}(Q_{ic}(L_{ic} + 1) - Q_{ic}(L_{ic}))|$  has the greatest value. Set  $L_{ic} \leftarrow L_{ic} + 1$  and  $L_{i,max} \leftarrow L_{i,max} - 1$ .

The complexity of the heuristic in terms of how many times (31) needs to be computed can be determined as follows. At the initialization step,  $Q_{ic}(1) \forall c$  have to be evaluated. Assuming that  $L_{i,max} > C$ , the first time Step 2 will be executed, the  $C$  values  $Q_{ic}(2) \forall c$  will also be computed. From then on, each time Step 2 is executed only the value of  $Q_{ic}(l)$

for the channel  $\lambda_c$  of which the queue was last incremented needs to be computed. Step 2 will be repeated  $L_{i,max} - C - 1$  times, which is the number of times (31) will be used with  $L_{ic} > 2$ . Observe also that it is possible to trade the quality of the heuristic for speed by allocating buffers to the various queues in chunks of size greater than one in Step 2.

## 6 Numerical Results

### 6.1 Average Packet Delay

We consider the 8-station mesh type, disconnected type, ring type and two-server traffic matrices shown in Figures 3, 4, 5, and 6, respectively; the numbers in the matrices represent  $p_{ij}$ , the probability that a packet arriving at station  $i$  is addressed to station  $j$ . We let  $\sigma_i = \sigma \forall i$ . This does not compromise the generality of our results, as the traffic characteristics are determined by  $p_{ij}$ . Figure 4 also shows the weighted TDM schedule of frame length  $M = 21$  produced by the Slot Allocation Heuristic (SAH) for the disconnected type traffic matrix,  $C = 8$  available wavelengths, and  $\sigma = 0.70$ . We can see that, overall, SAH places the slots assigned to each source-destination pair so that their distances are very close to the ones dictated by 20. Similar observations can be made about the schedules for other traffic matrices and different number of wavelengths not shown here.

For these four 8-node matrices we used SAH to produce optimized schedules for different numbers of wavelengths,  $C = 2, 4, 8$ , and various values of  $\sigma$ . We then obtained through simulation the delay and throughput curves of these schedules as  $\sigma$  increases from 0.01 to 0.99, shown in Figures 3 through 6; the delay is given in slots, and the throughput in packets successfully received per slot. A value of 100 in the delay plots denotes an infinite value for the average packet delay experienced under the given schedule and the given offered load,  $\sigma$ . The value of  $M_{max}$  in SAH was set to 2,584. In the figures we also plot the delay and throughput curves of the I-TDMA\* schedule for the corresponding traffic matrices; for a fair comparison, whenever  $C < N$ , we used the same sets of receivers sharing each channel for I-TDMA\* as for the optimized schedules (i.e., those produced by WBH in Appendix A).

From the figures it is immediately evident that the schedules constructed by SAH outperform I-TDMA\* by a wide margin, in terms of both delay and throughput. In particular, there are situations, as in Figure 5, when an optimized schedule with  $C = 2$  channels achieves

better performance than the I-TDMA\* schedule with the maximum number of channels,  $C = 8$ . The poor performance of I-TDMA\* even at very low offered loads are due to the fact that it assigns exactly one slot per frame to each source-channel pair. As a result, the queue for source-channel pair  $(i, \lambda_c)$  will build up whenever  $Mq_{ic} > 1$ , i.e., when the average number of packets for  $\lambda_c$  arriving at source  $i$  within a frame is greater than 1. In Figure 5, and for the I-TDMA\* schedule of frame length  $M = 7$  ( $C = 8$  channels), this condition is satisfied when  $\sigma > .204$  for the source-channel pairs for which  $q_{ic} = p_{ic} = 0.7$ . The optimized schedules, on the other hand, assign a larger number of slots to source-channel pairs with high values of  $q_{ij}$ , and thus, are able to operate at significantly higher offered loads before their delay behavior is affected. In addition, the slots assigned to a given source-channel pair are placed in almost equal distances within the frame, which also guarantees a near-optimal performance in terms of delay (see (20)).

The limitations of I-TDMA\*, and the potential for improvement by using the optimized schedules are more pronounced when one considers larger size networks. In Figure 7 we plot the delay and throughput curves for a 20-station ring-type traffic matrix (not shown here, but similar to the one in Figure 5). As we can see, the delay under an I-TDMA\* schedule with the maximum number of channels ( $C = 20$ ) grows without bound even for offered loads  $\sigma < 0.1$ . In contrast, an optimized schedule with as few as  $C = 3$  channels experiences finite delays for  $\sigma = 0.1$ , while one with  $C = 20$  channels may operate under loads as high as  $\sigma = 0.9$ .

## 6.2 Packet Loss

We again consider the 8- and 20-station traffic matrices studied in the previous section, but we now assume that each station employs a finite number of buffers,  $L_{i,max}$ . Without loss of generality, we let  $L_{i,max} = L_{max} \forall i$ . Our objective is to compare the packet loss probability under two scenarios: (a) when the  $L_{i,max}$  buffers available at each station are allocated according to the Buffer Partitioning Heuristic (BPH), and (b) when the  $L_{i,max}$  buffers are equally partitioned among the various channels. Only optimized schedules are considered in this section.

Figures 8 to 10 plot the packet loss probability curves against the total number of buffers at each station,  $L_{max}$ , for various traffic matrices and various system parameters ( $C = 8, \sigma = 0.7$

0	0.33	0.33	0	0.34	0	0	0
0.33	0	0	0.33	0	0.34	0	0
0.33	0	0	0.33	0	0	0.34	0
0	0.33	0.33	0	0	0	0	0.34
0.34	0	0	0	0	0.33	0.33	0
0	0.34	0	0	0.33	0	0	0.33
0	0	0.34	0	0.33	0	0	0.33
0	0	0	0.34	0	0.33	0.33	0

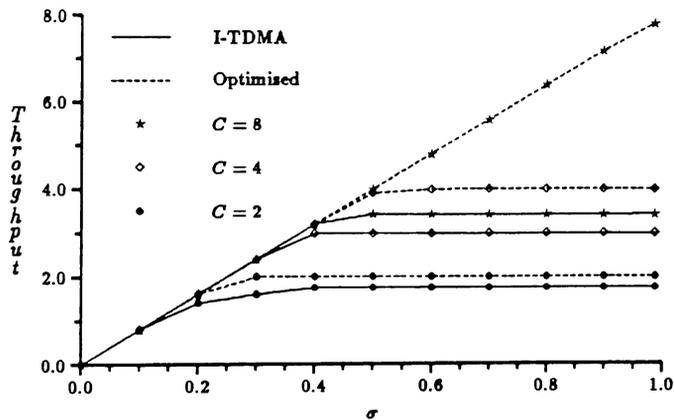
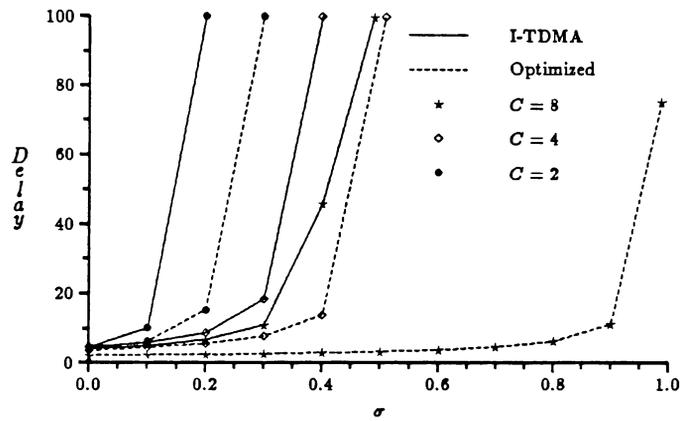


Figure 3: 8-station mesh-type traffic matrix and delay and throughput characteristics

0	0.30	0.30	0.30	0.025	0.025	0.025	0.025
0.30	0	0.30	0.30	0.025	0.025	0.025	0.025
0.30	0.30	0	0.30	0.025	0.025	0.025	0.025
0.30	0.30	0.30	0	0.025	0.025	0.025	0.025
0.025	0.025	0.025	0.025	0	0.30	0.30	0.30
0.025	0.025	0.025	0.025	0.30	0	0.30	0.30
0.025	0.025	0.025	0.025	0.30	0.30	0	0.30
0.025	0.025	0.025	0.025	0.30	0.30	0.30	0

← Frame →

2	4	3	6	4	2	5	3	8	4	2	6	3	2	4	3	7	4	2	5	3
3	6	4	1	5	3	8	4	1	6	3	1	4	3	7	4	1	5	3	1	4
4	1	5	2	8	4	1	7	2	1	4	2	8	4	1	6	2	1	4	2	7
5	2	8	3	1	7	2	1	3	2	8	3	1	6	2	1	3	2	7	3	1
6	8	7	8	2	6	6	8	4	7	1	7	2	8	6	7	8	3	1	7	6
7	5	2	5	7	1	4	2	5	8	7	8	7	1	3	8	5	7	8	8	5
8	3	1	4	6	8	3	5	6	5	5	4	6	5	8	2	6	8	5	6	8
1	7	6	7	3	5	7	6	7	3	6	5	5	7	5	5	4	6	6	4	2

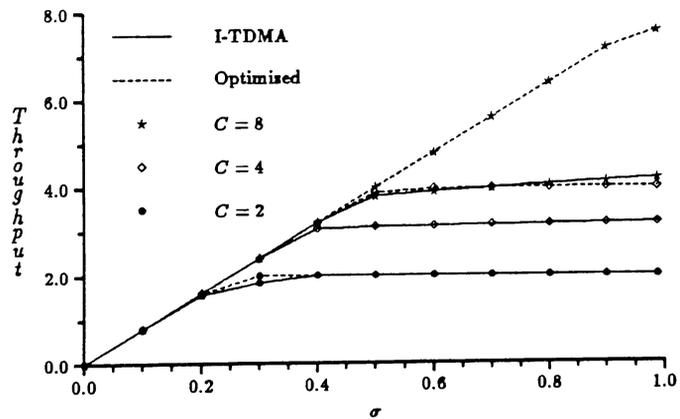
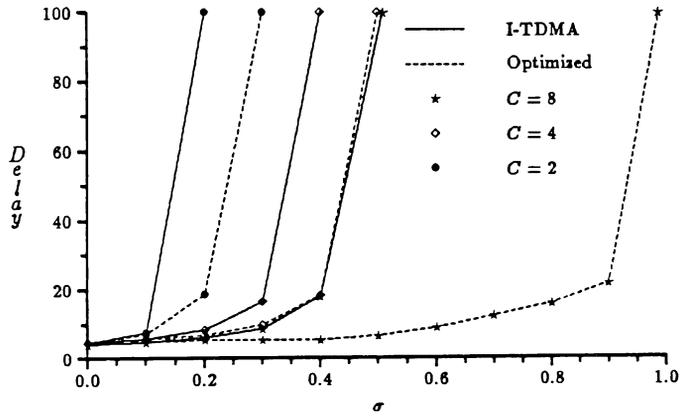


Figure 4: 8-station disconnected-type traffic matrix and delay and throughput characteristics

0	0.70	0.05	0.05	0.05	0.05	0.05	0.05
0.05	0	0.70	0.05	0.05	0.05	0.05	0.05
0.05	0.05	0	0.70	0.05	0.05	0.05	0.05
0.05	0.05	0.05	0	0.70	0.05	0.05	0.05
0.05	0.05	0.05	0.05	0	0.70	0.05	0.05
0.05	0.05	0.05	0.05	0.05	0	0.70	0.05
0.05	0.05	0.05	0.05	0.05	0.05	0	0.70
0.70	0.05	0.05	0.05	0.05	0.05	0.05	0

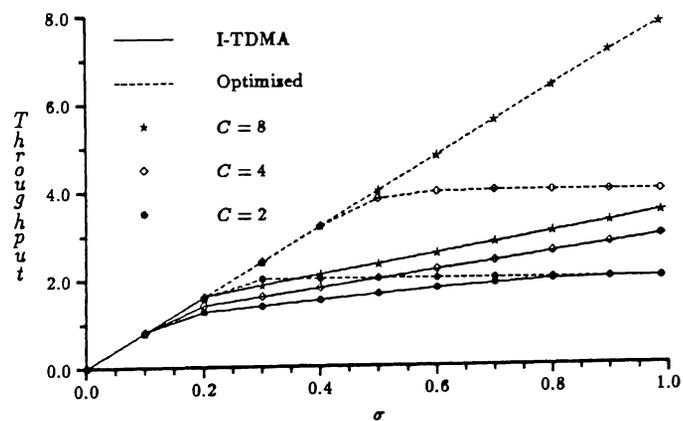
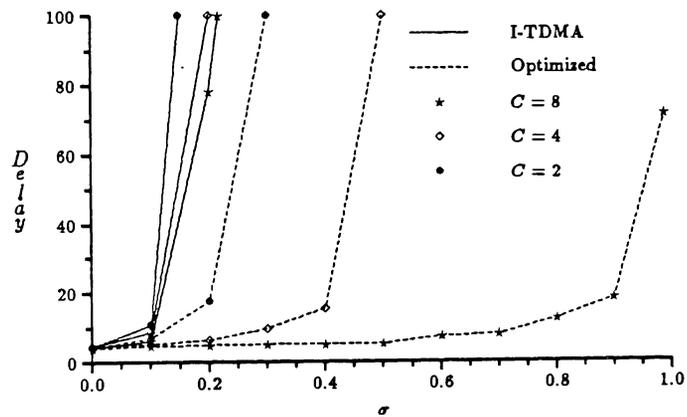


Figure 5: 8-station ring-type traffic matrix and delay and throughput characteristics

0	0.20	0.20	0.20	0.10	0.10	0.10	0.10
0.40	0	0.08	0.08	0.20	0.08	0.08	0.08
0.40	0.08	0	0.08	0.20	0.08	0.08	0.08
0.40	0.08	0.08	0	0.20	0.08	0.08	0.08
0.10	0.10	0.10	0.10	0	0.20	0.20	0.20
0.20	0.08	0.08	0.08	0.40	0	0.08	0.08
0.20	0.08	0.08	0.08	0.40	0.08	0	0.08
0.20	0.08	0.08	0.08	0.40	0.08	0.08	0

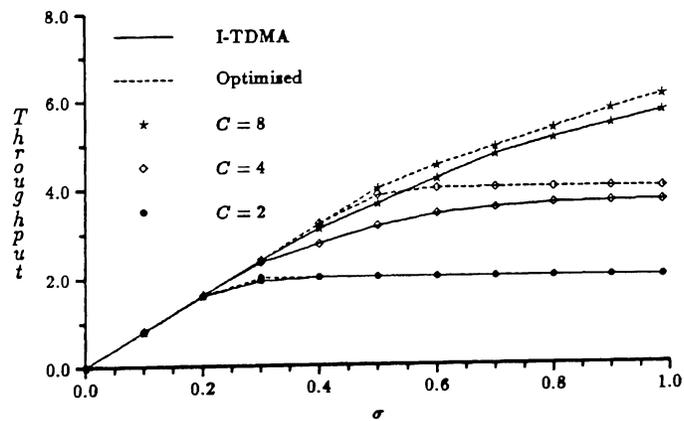
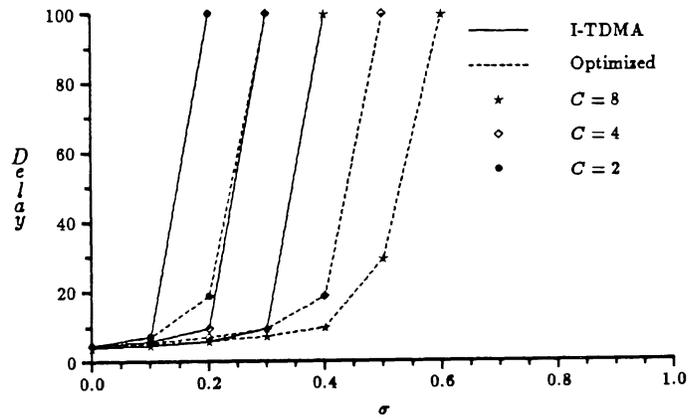


Figure 6: 8-station two-server-type traffic matrix and delay and throughput characteristics

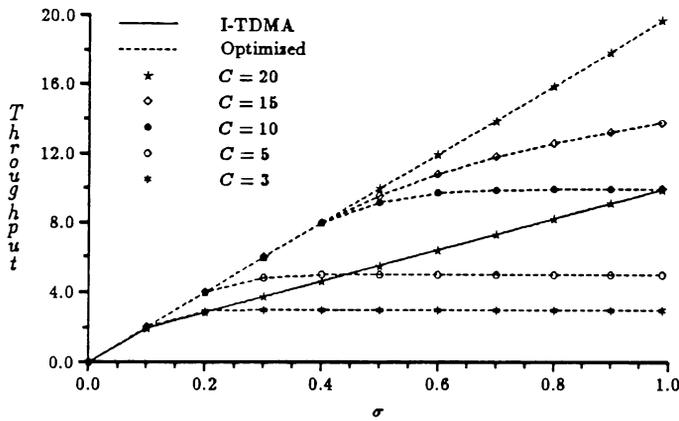
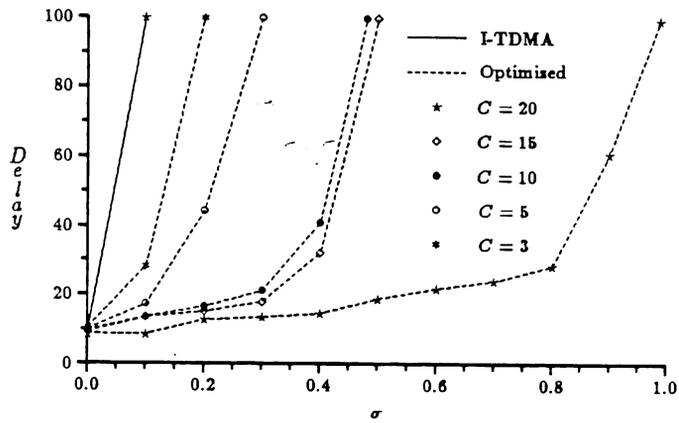


Figure 7: 20-station ring-type traffic matrix and delay and throughput characteristics

and  $C = 4, \sigma = 0.3$  for the 8-station matrices, and  $C = 10, 20, \sigma = 0.3$  for the 20-station matrix). The curves were obtained through simulation. Label “EP” is used in the figures to denote scenario (b) above, i.e., the equal sharing of buffers among the channels. The plots indicate that, as the number of buffers increases, the buffer allocation determined by BPH results in a performance improvement between one and four orders of magnitude over an equal partitioning scheme, depending on the traffic matrix and system parameters and the number of available channels.

## 7 Concluding Remarks

We have considered single-hop WDM networks in which access to the various channels is controlled by weighted transmission schedules. We have addressed the problems of minimizing the delay and packet loss probability across the network. We have developed heuristics that not only outperform previously proposed solutions, but also perform very well for communication patterns one expects to encounter in realistic environments. Techniques such as these are the first step towards lightwave WDM networks that dynamically adapt to changing traffic patterns.

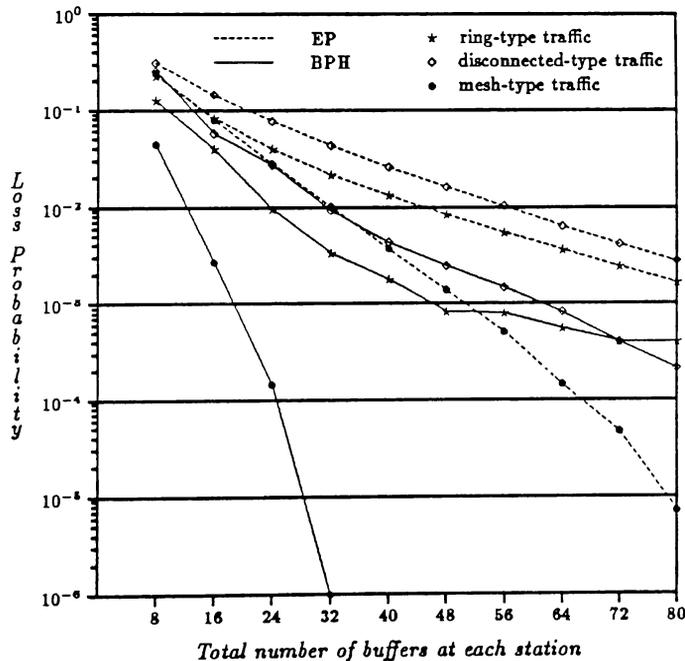


Figure 8: 8-station loss probability characteristics ( $C = 8, \sigma = 0.7$ )

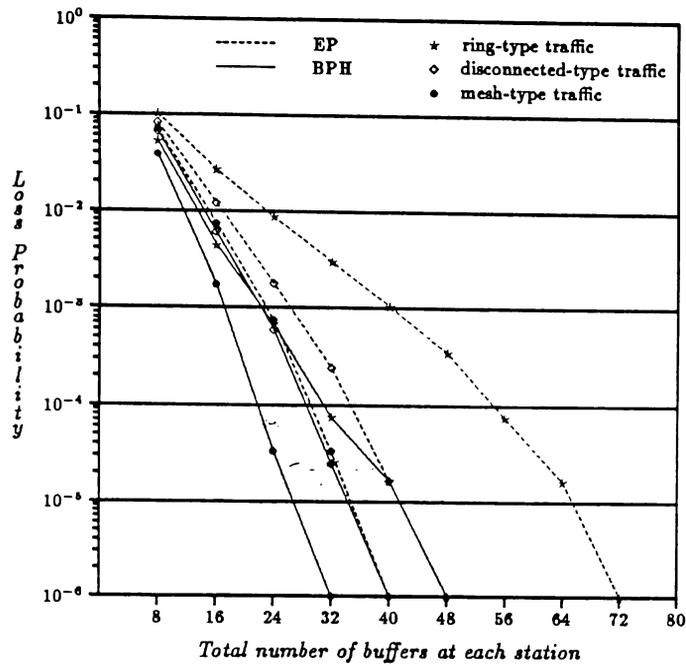


Figure 9: 8-station loss probability characteristics ( $C = 4, \sigma = 0.3$ )

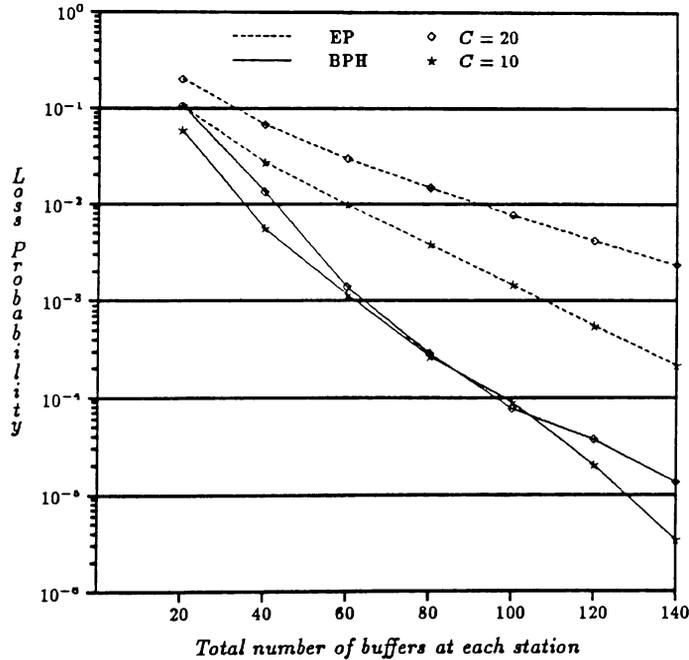


Figure 10: 20-station loss probability characteristics ( $C = 10, 20, \sigma = 0.3$ )

## References

- [1] M. H. Ammar and G. N. Rouskas. On the performance of protocols for collecting responses over a multiple-access channel. In *Proceedings of INFOCOM '91*, pages 1490–1499. IEEE, April 1991. To appear in *IEEE Transactions on Communications*.
- [2] 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, 1993.
- [3] C. A. Brackett. Dense wavelength division multiplexing networks: Principles and applications. *IEEE Journal on Selected Areas in Communications*, SAC-8(6):948–964, August 1990.
- [4] 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.
- [5] 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, 1993.
- [6] I. Chlamtac and A. Ganz. Channel allocation protocols in frequency-time controlled high speed networks. *IEEE Transactions on Communications*, 36(4):430–440, April 1988.
- [7] P. W. Dowd. Random access protocols for high speed interprocessor communication based on an optical passive star topology. *Journal of Lightwave Technology*, LT-9:799–808, June 1991.
- [8] M. Eisenberg and N. Mehravari. Performance of the multichannel multihop lightwave network under nonuniform traffic. *IEEE Journal on Selected Areas in Communications*, 6(7):1063–1078, August 1988.
- [9] M. R. Garey and D. S. Johnson. *Computers and Intractability*. W. H. Freeman and Co., New York, 1979.

- [10] P. E. Green. The future of fiber-optic computer networks. *IEEE Computer*, 24(9):78–87, September 1991.
- [11] I. M. I. Habbab, M. Kavehrad, and C.-E. W. Sundberg. Protocols for very high-speed optical fiber local area networks using a passive star topology. *Journal of Lightwave Technology*, LT-5(12):1782–1793, December 1987.
- [12] P. S. Henry. High-capacity lightwave local area networks. *IEEE Communication Magazine*, pages 20–26, October 1989.
- [13] M. Hofri and Z. Rosberg. Packet delay under the golden ratio weighted TDM policy in a multiple-access channel. *IEEE Transactions on Information Theory*, IT-33(3):341–349, May 1987.
- [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, 1993.
- [15] L. Kleinrock. *Queueing Systems, Volume 1: Theory*. John Wiley & Sons, New York, 1975.
- [16] N. Mehravari. Performance and protocol improvements for very high-speed optical fiber local area networks using a passive star topology. *Journal of Lightwave Technology*, 8(4):520–530, April 1990.
- [17] B. Mukherjee. WDM-Based local lightwave networks Part I: Single-hop systems. *IEEE Network Magazine*, pages 12–27, May 1992.
- [18] G. N. Rouskas. *Single-Hop Lightwave WDM Networks and Applications to Distributed Computing*. PhD thesis, Georgia Institute of Technology, Atlanta, GA, May 1994.
- [19] G. N. Rouskas and M. H. Ammar. Analysis and optimization of transmission schedules for single-hop WDM networks. In *Proceedings of INFOCOM '93*, pages 1342–1349. IEEE, March 1993. An expanded version has been submitted to the *IEEE/ACM Transactions on Networking*.
- [20] G. N. Rouskas and M. H. Ammar. Multi-destination communication over single-hop lightwave WDM networks. In *Proceedings of INFOCOM '94*, pages 1520–1527. IEEE,

June 1994. An expanded version has been submitted to the IEEE/ACM Transactions on Networking.

- [21] G. Semaan and P. Humblet. Timing and dispersion in WDM optical star networks. In *Proceedings of INFOCOM '93*, pages 573–577. IEEE, 1993.

# A Channel Assignment

In deriving heuristics for Problems 1 and 2 we have assumed that the sets of receivers,  $R_c$ , listening on the same wavelength,  $\lambda_c, c = 1, \dots, C$ , are given. We now focus on determining how receivers are assigned to the various channels.. In [2] it is assumed that  $N = mC$ , and that  $m$  receivers share each channel. Although this is a valid assumption under the uniform traffic conditions considered there, the following two main issues need to be addressed in the general case:

- Intuitively, the externally offered traffic load must be balanced across the various channels, if the average packet delay or packet loss probability is to be minimized.
- Furthermore, whenever the total offered load is less than the network capacity (i.e.,  $\sum_{i=1}^N \sigma_i < C$ ), we would like to partition the receivers so that the load offered to each channel is less than the channel capacity <sup>10</sup> (or,  $\sum_{i=1}^N \sigma_i \sum_{j \in R_c} p_{ij} = \sum_{i=1}^N q_{ic} < 1 \forall c$ ).

The problem of determining sets  $R_c$  can be stated concisely as:

**Problem 3** *Given the number of stations,  $N$ , the number of available wavelengths,  $C$ , and the traffic parameters,  $\sigma_i p_{ij}, i, j = 1, \dots, N$ , partition the set of receivers,  $\{1, \dots, N\}$ , into  $C$  sets,  $R_1, \dots, R_C$ , with receivers in  $R_c$  sharing channel  $\lambda_c, 1 \leq c \leq C$ , so that (a) the offered load is balanced across the  $C$  channels, and (b) the stability condition  $\sum_{i=1}^N q_{ic} < 1$  holds for all channels  $\lambda_c$ .*

We will first formulate the problem in a more general context, as follows. We are given a set of  $N$  elements, and  $w_i > 0$  is the weight associated with element  $e_i$ . Our objective is to partition this set into  $C < N$  subsets  $Y_c, c = 1, \dots, C$ , such that:

$$\max_{1 \leq k, l \leq C} \left\{ \left| \sum_{e_i \in Y_k} w_i - \sum_{e_j \in Y_l} w_j \right| \right\} \text{ is minimized} \quad (33)$$

$$\forall c : \sum_{e_i \in Y_c} w_i < W, \quad W > 0 \quad (34)$$

---

<sup>10</sup>Note that this may not be feasible in general, for instance, when there is one receiver  $j$  such that  $\sum_{i=1}^N \sigma_i p_{ij} > 1$ .

Let us first consider each of these problems in isolation. For  $C = 2$ , the problem defined by (33) reduces to partitioning the set of elements into two subsets,  $Y_1$  and  $Y_2$ , such that  $|\sum_{e_i \in Y_1} w_i - \sum_{e_j \in Y_2} w_j|$  is minimized. But the problem of whether there exists a partition of the set of  $N$  elements into two subsets  $Y_1$  and  $Y_2$  such that  $\sum_{e_i \in Y_1} w_i = \sum_{e_i \in Y_2} w_i$ , is *NP*-complete [9, p. 223], and thus the minimization problem (33) for  $C = 2$  is expected to be hard. Similarly, (34) is the well-known bin packing problem, which also is *NP*-complete [9, p. 226]. Thus, we do not expect to be able to find an exact solution to Problem 3 efficiently.

We now propose the following greedy procedure to construct the  $C$  subsets. The heuristic first makes sure that each of the  $C$  elements with the largest weight is assigned to a different subset. Then, it repeats adding the element with the highest weight among the ones not yet considered to the set with the lowest total weight.

### Weight Balancing Heuristic (WBH)

1. Sort elements  $e_i$  in decreasing order of  $w_i$ . Initialize  $Y_c \leftarrow \{e_c\}, c = 1, \dots, C$ , and  $k \leftarrow C + 1$ . Note that sets  $Y_c$  are also sorted in decreasing order of  $\sum_{e_i \in Y_c} w_i$ .
2. Set  $Y_C \leftarrow Y_C \cup \{e_k\}$  and  $k \leftarrow k + 1$ . Sort  $Y_c, c = 1, \dots, C$ , in decreasing order of  $\sum_{e_i \in Y_c} w_i$ . Repeat Step 2 while  $k < N$ .

For obtaining sets  $R_c$  we used as weight of receiver  $j$  the load offered to destination  $j$ :  $w_j = \sum_{i=1}^N \sigma_i p_{ij}$ .