Multi-Destination Communication Over Tunable-Receiver Single-Hop WDM Networks

George N. Rouskas   Mostafa H. Ammar

TR-96-12

June 18, 1996

Abstract

The importance of computer applications and telecommunication services relying on efficient multi-destination communication is growing rapidly. It is now likely that a significant portion of the overall traffic in future communication environments will be of the multi-destination type. At the same time, lightwave technology is emerging as a promising candidate for implementing the next generation of multiuser high-speed networks. In this paper we address the issue of providing efficient mechanisms for multi-destination communication over one class of lightwave WDM architectures, namely, single-hop networks with tunability provided only at the receiving side. We distinguish a number of multicast traffic types, and we suggest, analyze, and optimize alternative broadcast/multicast approaches for each type. One of our major contributions is the development of a suite of adaptive multicast protocols which are simple to implement, and have good performance under changing multicast traffic conditions.

Department of Computer Science
North Carolina State University
Raleigh, NC 27695

1 Introduction

The single-hop architecture for lightwave networks [1] employs Wave Division Multiplexing (WDM) to divide the enormous information-carrying capacity of single mode fiber into multiple concurrent channels, and may deliver an aggregate throughput that can grow with the number of wavelengths deployed, and can be in the order of Terabits per second. Single-hop networks are all-optical in nature, i.e., the entire path between the source and the destination is optical, and no conversion back and forth between electronics and photonics takes place within the network. For a successful packet transmission, one of the transmitters of the source and one of the receivers of the destination must operate on the same wavelength. Thus, tunable transceivers are required, as well as some form of coordination among stations wishing to communicate. We focus on a wavelength-time assignment of the optical bandwidth, whereby time is slotted and each station may transmit only in slots specified by a predetermined schedule. This is an extension of TDMA over a multichannel environment.

Several single-hop architectures have been proposed in the literature, and have been extensively studied for *single-destination* traffic [2, 3, 4, 5, 6]. In particular, the authors have developed a general framework for analyzing and optimizing the throughput [7] and delay [8] performance of single-hop networks using TDM schedules, for any number of wavelengths, any transceiver tunability characteristics, and general (potentially non-uniform) traffic patterns. It is, however, widely believed [9, 10] that with the advent of computer applications and telecommunication services such as distributed data processing [11], broadcast information systems [12], and teleconferencing, a significant portion of the overall traffic in future broadband networks will be of the multi-destination type. As a result, a significant amount of research has been aimed at designing efficient multicast mechanisms for next-generation communications environments [13, 14, 15, 16]. In the context of single-hop networks, the issue of *multi-destination* traffic has been addressed in [17], where a control-channel based multicast protocol is presented, and in [18], where we investigated the throughput characteristics of various multicast schemes.

In this paper we suggest, analyze, and optimize a number of alternative approaches to performing efficient broadcast/multicast over single-hop lightwave networks employing tunable receivers. The main difference between this work and the one in [18] is our use of a more realistic model to evaluate the protocols and the emphasis on the delay performance of the various approaches.

This paper is organized as follows. In Section 2 we introduce the network model, and the various schedules we consider. In Section 3 we present a classification of multi-destination traffic that takes into account the length of multicast sessions and the size of multicast groups, and argue for the need to provide a variety of multicast mechanisms addressing the requirements of each type of multi-destination traffic. Section 4 describes an optimization heuristic for constructing schedules that minimize the average packet delay, and in Section 5 a suite of adaptive multicast protocols is developed. We present numerical results in Section 6, and conclude with a summary of our work in Section 7.

2 System Model

We consider a network of N stations, each equipped with one transmitter and one receiver, interconnected through a passive broadcast optical medium that can support C wavelengths, $\lambda_1, \lambda_2, \ldots, \lambda_C$. In this work we only consider fixed-transmitter, tunable-receiver (FT-TR) systems (a study of multicasting in tunable-transmitter, fixed-receiver (TT-FR) networks can be found in [19]). In such a system, we let $\lambda(i) \in \{\lambda_1, \ldots, \lambda_C\}$ denote the channel [1] assigned to the fixed transmitter of station i. The receivers, on the other hand, are optical filters that can be tuned to, and receive from any and all wavelengths $\lambda_c, c = 1, \ldots, C$. An important parameter in such a system is the *tuning latency*, the time it takes a receiver to tune from one wavelength to another; the importance of the value of tuning latency relative to the packet transmission time will be discussed shortly.

We distinguish between single- and multi-destination packets; the latter need to be delivered to a number of stations, members of a *multicast group*. A *multicast address* is associated with each multicast group. We assume that any station, upon receiving a packet having a multicast address as a destination address, is capable of determining whether it is a member of the multicast group or not; in the latter case it discards the packet.

The network operates in a slotted mode with a slot time equal to the packet transmission time plus the maximum tuning latency; the maximum is taken over all receivers and all wavelength pairs. A *collision* occurs when two or more transmitters access the same channel during a slot. When two or more sources transmit to the same destination on different channels the result is a *destination conflict*. All packets involved in a collision or destination conflict are considered lost [2]; recovery is assumed to take place via a higher level protocol.

We define σ_i and ρ_i as the probability that a new single-destination and multi-destination packet, respectively, is generated at station i during a slot time. We let p_{ij} denote the probability that a new single-destination packet is destined to station j. Similarly, q_{ig} will denote the probability that

[1] Throughout this paper we will use the terms "wavelength" and "channel" interchangeably.

[2] In some cases, one of the packet involved in a destination conflict may be successfully received; this will happen if it is the only packet transmitted on the channel the receiver listens to in that slot.

a new multi-destination packet is addressed to multicast group $g \subseteq \{1, \ldots, N\}$; obviously then,

$$\sum_{j=1}^{N} p_{ij} = 1 \; ; \quad \sum_{g} q_{ig} = 1 \qquad i = 1, \ldots, N \tag{1}$$

If the number of wavelengths, C, is equal to the number of stations, N, the fixed transmitter at each station i is assigned a unique wavelength, $\lambda(i) \in \{\lambda_1, \ldots, \lambda_C\}$, or *home* channel. However, from a technical point of view, having N channels is difficult to achieve, as current WDM technology supports only a small number of wavelengths in a single mode fiber, making these architectures unsuitable for anything but trivial networks. Furthermore, in order to keep channel utilization at high levels, the maximum tuning latency 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 optical filters [20]; in other words, the fastest tunable receivers may only tune over a small portion of the useful optical bandwidth. As a result, even if a large number of wavelengths could be supported within a fiber, because of the need to minimize the time spent for retuning within each slot by employing *very fast* tunable optical receivers, only a subset of these channels would be usable. Therefore, in this work we consider the general case, $C \leq N$.

Whenever $C < N$, a number of transmitters have to be assigned the same wavelength $\lambda_c, c = 1, \ldots, C$. We let X_c, a subset of $\{1, \ldots, N\}$, denote the set of transmitters sharing wavelength λ_c:

$$X_c = \{i \mid \lambda(i) = \lambda_c\} \quad c = 1, \ldots, C \tag{2}$$

2.1 Transmission Schedules

The media access scheme we consider is an extension of *weighted* Time Division Multiplexing (TDM) over a multi-channel environment. In such a scheme, time slots are grouped in frames of $M \geq N$ slots. A *transmission schedule* indicates, for all i and j, which slots during a frame can be used for transmissions from i to j; it can be described by variables $\delta_{ij}^{(t)}$, called *permissions*, and defined as

$$\delta_{ij}^{(t)} = \begin{cases} 1, & \text{if station } i \text{ has permission to transmit to station } j \text{ in slot } t \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

This description implies that, if $\delta_{ij}^{(t)} = 1$ for some slot t, then receiver j will tune to channel $\lambda(i)$ in slot t. The permissions repeat over time so that $\delta_{ij}^{(t+kM)} = \delta_{ij}^{(t)}, k = 0, 1, 2, \cdots$. We also define the following two variables:

$$\zeta_i^{(t)} = \begin{cases} 1, & \text{if } \sum_{j=1}^{N} \delta_{ij}^{(t)} \geq 1 \\ 0, & \text{otherwise} \end{cases} \qquad \forall \; i, t \tag{4}$$

$$G_i^{(t)} = \begin{cases} \{j \mid \delta_{ij}^{(t)} = 1\}, & \text{if } \zeta_i^{(t)} = 1 \\ \phi, & \text{otherwise} \end{cases} \qquad \forall\ i,t \tag{5}$$

Whenever $\zeta_i^{(t)} = 1$ we say that transmitter i is *active* in slot t. In that case, $G_i^{(t)}$, the *transmitting set* of i, is the set or receivers which will be tuned to channel $\lambda(i)$ in slot t.

We are particularly interested in schedules which guarantee that no collisions or destination conflicts will ever occur; we thus have the following definition.

Definition 1 *A schedule provides full connectivity in the strong sense iff it satisfies the following three conditions:*

$$\forall\ i,j: \quad \{\sigma_i\ p_{ij} > 0 \ \vee\ [(\exists\ g : \rho_i q_{ig} > 0 \ \wedge\ j \in g)]\} \ \Rightarrow\ \exists\ t:\ \delta_{ij}^{(t)} = 1 \tag{6}$$

$$\sum_{i \in X_c} \zeta_i^{(t)} \leq 1 \qquad \forall\ c,t \tag{7}$$

$$\sum_{i=1}^{N} \delta_{ij}^{(t)} \leq 1 \qquad \forall\ j,t \tag{8}$$

Condition (6) specifies that, if there is single- or multi-destination traffic flowing from i to j, then there is at least one slot per frame in which i may transmit to j. This guarantees full connectivity among the network stations. Constraint (7) requires that, in a given slot t, at most one of the fixed transmitters in X_c be active; this constraint eliminates collisions. Finally, constraint (8) implies that at most one source may transmit to a given destination within a slot t, and thus no destination conflicts may occur. Unless otherwise specified, in the following the term "schedule" will be used to refer to a schedule satisfying (6), (7), and (8).

Based on the number of permissions given to the various sources, a slot t can be classified as a:

1. **Unicast slot:** There are exactly C active transmitters in slot t, and the transmitting set of each active transmitter has cardinality one. Note that, because of (7), there will be exactly one active transmitter per channel.

$$\sum_{i=1}^{N} \zeta_i^{(t)} = C \ ; \ \mid G_i^{(t)} \mid \ \leq 1 \quad i = 1, \ldots, N \tag{9}$$

2. **Broadcast slot:** There is exactly one active transmitter, k, in slot t, and its transmitting set includes all possible destinations. We will call k the *owner* of slot t; a packet transmitted by the owner of the slot will reach all stations in the network, thus the name "broadcast slot". No other station may be active in slot t as that could result in a destination conflict (see (8)).

$$\zeta_i^{(t)} = \begin{cases} 1, & \text{if } i = k \\ 0, & \text{otherwise} \end{cases} \ ; \quad G_i^{(t)} = \begin{cases} \{1, 2, \ldots, N\}, & \text{if } i = k \\ \phi, & \text{otherwise} \end{cases} \tag{10}$$

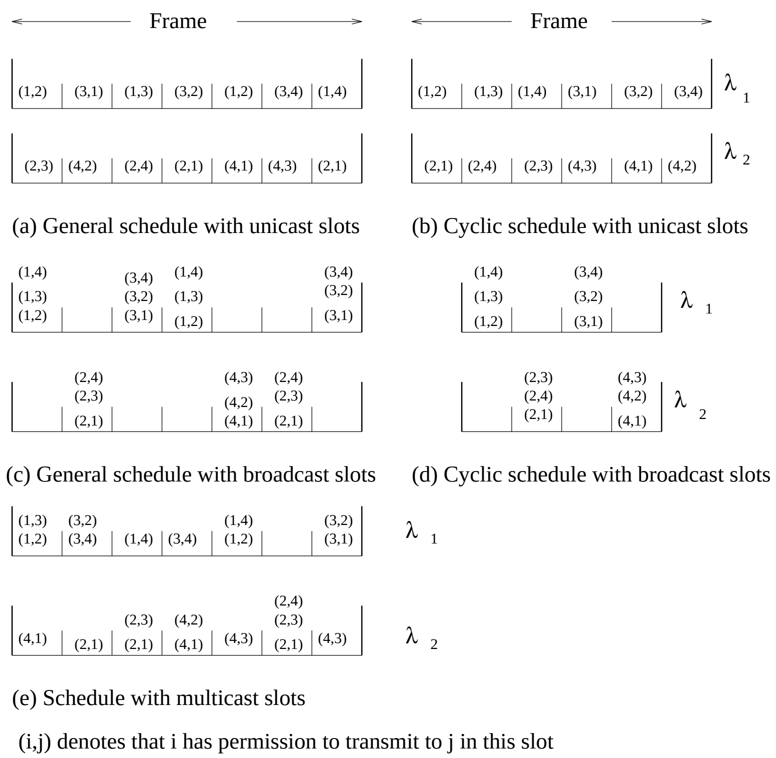

Figure 1: Schedules with: (a)-(b) unicast, (c)-(d) broadcast, and (e) multicast slots for a network with $N = 4$ stations, $C = 2$ wavelengths, $X_1 = \{1, 3\}, X_2 = \{2, 4\}$

3. **Multicast slot:** There are $m, 1 \leq m \leq C$ active transmitters in slot t. One of the active transmitters, say k, has a transmitting set with cardinality greater than one, while all others have transmitting sets with cardinality one. We say that k is the owner of the slot, and its transmission will reach all stations in $G_k^{(t)}$.

$$\sum_{i=1}^{N} \zeta_i^{(t)} = m, \ 1 \leq m \leq C; \ G_i^{(t)} = \begin{cases} g \subset \{1, 2, \ldots, N\}, \mid g \mid \geq 2, & \text{if } i = k \\ j \in \{1, 2, \ldots, N\} - g, & \text{if } i \neq k \text{ and } \zeta_i^{(t)} = 1 \\ \phi, & \text{otherwise} \end{cases} \tag{11}$$

Figure 1 shows schedules with unicast, broadcast, and multicast slots for a fixed-transmitter, tunable-receiver network with $N = 4$ stations and $C = 2$ wavelengths. Channel λ_1 is shared by the fixed transmitters of stations 1 and 3 ($X_1 = \{1, 3\}$), while channel λ_2 is shared by the transmitters of stations 2 and 4 ($X_2 = \{2, 4\}$). The cyclic schedules also shown are special cases, whereby each station may transmit to each possible destination exactly once per frame (for unicast slots), or is the owner of exactly one broadcast slot per frame.

2.2 Media Access Control Layer

The buffer capacity at source $i, i = 1, \ldots, N$, is organized into N logical queues [3]. One queue is associated with each of the possible $N - 1$ destinations; a single-destination packet addressed to j joins the queue for this station. Having separate queues eliminates the head of line effects of a single buffer, and helps to drastically improve the delay and throughput characteristics as demonstrated in [4, 8]. The N-th queue (to be called the *multicast queue* from now on) is used for storing arriving multi-destination packets regardless of their multicast group [4]. This is a model of the Media Access Control (MAC) layer, and is an extension of the models in [7, 8, 21].

2.3 Performance Parameters

We will be concerned with evaluating the performance of the various schedules in terms of average packet delay and aggregate throughput. Packet delay is defined as the number of slots elapsed between the generation of a packet at its source and the slot in which it is transmitted. (This definition ignores propagation delay; however, the latter is independent of the particular schedule used, and ignoring it will not affect our conclusions regarding the relative performance of the various schedules.) Throughput, on the other hand, is defined as the expected number of packets successfully *received* per slot. Note that a single transmission of a multi-destination packet on a given channel may be received by several receivers listening on that channel. Thus, the total throughput may be greater than the number of channels, C, though it certainly cannot be greater than the number of potential receivers, N.

As a final observation, our definition of throughput assumes that the tuning latency (which is included as part of every slot) is small compared to the packet transmission time. This assumption is reasonable for a system with a small number of channels C and for Local and Metropolitan Area Networks with relatively large packet sizes (a few thousand bytes), as the tuning latency would only take up a small fraction of the slot time. On the other hand, including the tuning latency in each and every slot of the frame would be highly inefficient for environments, such as fast packet switches, characterized by very small packet sizes (e.g., ATM cells). In these situations, the packet transmission time may be only a fraction of the tuning latency of even the fastest currently available tunable optical filters, and techniques to construct schedules to *hide* or *overlap* the tuning latency

[3] These are *logical* rather than physical queues, and may be implemented in shared memory for efficiency.

[4] Due to the large number of potential multicast groups, it is not practical to dedicate one queue for each such group. If a certain station, as is typical of a real system, maintains only a small number of active multicast sessions, say m, we may use m queues for multi-destination packets, one for each multicast session, as will be discussed later.

(see [22, 23, 24, 25]) would be more appropriate.

3 Schedules for Mixed Single/Multi-Destination Traffic

When only single-destination traffic is offered to the network, schedules with unicast slots only are sufficient: a source i will transmit a packet for j in a slot t such that $\delta_{ij}^{(t)} = 1$. These schedules may also be used for multi-destination traffic. In this case, a source would have to transmit a multi-destination packet multiple times, once to each member in the packet's multicast group. Obviously, this approach does not take advantage of the inherent multicast capabilities of tunable-receiver single-hop networks. We now describe how we may use broadcast or multicast slots to carry multi-destination packets.

Let g be the multicast group of a multi-destination packet originating at station i. Typically, the members of g are not known in advance; also, group membership may change during the life of the multicast communication. One way to guarantee that a packet will be received by all current members of its multicast group would be to have station i broadcast the packet to the entire network. The receivers would then use the multicast address to filter out any packets they do not need. This can be achieved by setting aside some of the slots of the schedule as broadcast slots with i as their owner; source i would then use its broadcast slots to transmit its multi-destination packets. However, if the average size of a multicast group is small compared to the number of stations in the network (a situation that often arises in distributed computing systems), an approach that attempts to deliver all multicast packets to all possible destinations would be extremely wasteful in terms of bandwidth. Ideally, we would like to have schedules that allow a source to deliver a packet only to the current members of a packet's multicast group.

To this end, we may allocate a number of slots in the schedule as multicast slots with i as their owner; the transmitting set of i in such a slot t would be $G_i^{(t)} = g$. As a result, only current members of multicast group g would tune their receivers to i's transmit wavelength in these slots. Other stations may tune their receivers to one of the remaining wavelengths (see (11)) allowing sources other than i to transmit single-destination packets. Thus, unlike broadcast slots in which all receivers are tuned to a certain source's wavelength and no other communication may take place, multicast slots provide for transmission concurrency.

Allocating at least one slot per frame for transmissions to each possible multicast group would be impractical even for networks of moderate size, as the number of possible multicast groups increases very rapidly with the number of stations. Typically, though, at any given time, a source participates in a small number of multicast sessions. But as multicast groups of existing sessions change over time, or as active sessions terminate and new ones start, the schedule has to be modified to reflect these changes. Thus, unless a mechanism for dynamically updating the permissions in the multicast slots of the frame is available, the only practical approaches to multicasting would be to use either unicast or broadcast slots.

As we can see, the nature of multi-destination traffic is of crucial importance in designing the schedules; we may classify this traffic along two dimensions:

- **Length of multicast session:** In a typical distributed computing environment (such as when multicast is used for "response collection" applications [26]) the multicast session is short, involving the exchange of a few multi-destination packets. In other situations, however, the multicast session may be quite long, requiring the transmission of a large number of packets to the same multicast group; examples include video and audio conferencing applications.
- **Size of multicast group:** In distributed systems, the average size of the multicast group is often small compared to the number of stations in the network. On the other hand, applications that require the sending of messages to all or a large number of the stations in the network do exist (for instance, the update of routing information).

We now distinguish three types of multi-destination traffic:

- **Type 1.** Multicast sessions are relatively short, and multicast groups are likely to include a large number of stations. In this case, schedules with both unicast (for single-destination packets) and broadcast (for multi-destination packets) slots are most appropriate.
- **Type 2.** Multicast sessions are relatively short, while the average multicast group contains a few stations. Since broadcast slots might lead to substantial underutilization of bandwidth, we believe that schedules with unicast slots only should be used and that multi-destination packets should be transmitted to each of their destinations individually (this is not expected to have a severe effect on performance due to the small number of stations per group).
- **Type 3.** Multicast sessions are relatively long. Schedules with both unicast and multicast slots may be used, and we have developed a suite of protocols to dynamically update the permissions in each multicast slot according to the current multicast group.

Our conclusions are summarized in Table 1. The next section describes a way to construct schedules for Type 1 traffic so that the average packet delay is minimized, while Section 5 presents a number of adaptive protocols for Type 3 traffic. How to obtain schedules with unicast slots only, that minimize the average packet delay for Type 2 traffic is discussed in Appendix A.1.

		Average Duration of Multicast Session	
		Short	Long
Average Size of Multicast Group	Small	Unicast Slots	Unicast and Multicast Slots
	Large	Unicast and Broadcast Slots	

Table 1: Schedules appropriate for the various types of multi-destination traffic

4 Delay Minimization for Type 1 Traffic

In this section we restrict our attention to schedules with unicast and broadcast slots only. Our objective is to determine 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, σ_i, ρ_i, p_{ij}, and $q_{ig}, \forall\ i, j, g$, and assuming that buffers of infinite capacity are available at each station, find a schedule with unicast and broadcast slots such that the network-wide average packet delay is minimized.*

From previous experience with a similar problem (namely, the problem of constructing optimal schedules with unicast slots only for tunable-transmitter, fixed-receiver systems [8]) we expect Problem 1 to be a very hard allocation problem. We now develop an optimization heuristic to obtain schedules that not only perform well, but also guarantee a designer-specified level of performance for each type of traffic (single- or multi-destination). The heuristic is based on a decomposition of the problem into two manageable subproblems, namely, the problems of finding optimal schedules assuming each type of traffic is offered to the network in isolation. Optimization of unicast slot schedules for single-destination traffic only ($\rho_i = 0\ \forall\ i$) is presented in Appendix A.1, while optimized broadcast slot schedules for multi-destination traffic only ($\sigma_i = 0\ \forall\ i$) are derived in Appendix A.2. Then, the two schedules are appropriately *merged* into a final schedule for the mixed traffic at hand.

Let S_1 and S_2 be two schedules of frame lengths M_1 and M_2, respectively. Without loss of generality, assume that $M_1 \geq M_2$ and $M_1 = mM_2$. If m is an integer, merging of S_1 and S_2 is performed by inserting one slot of S_2 after every m slots of S_1, resulting in a new schedule S, of frame length $M = M_1 + M_2$. Schedule merging can be easily generalized to situations where M_1 is

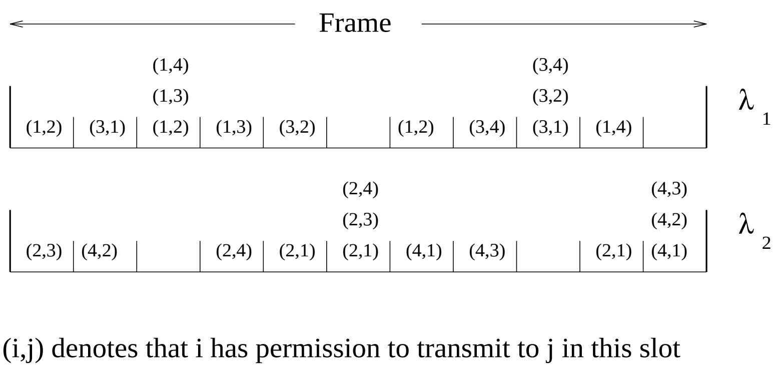

Figure 2: Schedule merging

not an integer multiple of M_2. In Figure 2 we show the result of merging the unicast slot schedule of Figure 1(a) ($M_1 = 7$), with the cyclic broadcast slot schedule of Figure 1(d) ($M_2 = 4$).

Consider a unicast slot schedule, S, optimized for single-destination traffic only. Since there are no broadcast slots, multi-destination traffic experiences infinite delay. If we merge S with $l, l \geq 1$, frames of a broadcast slot schedule, S', optimized for multi-destination traffic, we effectively provide slots in which multi-destination packets may be transmitted, thus improving their delay performance. As l increases, the merged schedule will tend to favor multi-destination packets (note that as $l \rightarrow \infty$, the resulting schedule will be indistinguishable from an S' schedule, in which case single-destination traffic will suffer). Therefore, we must choose an l such that both the overall delay, $D_{overall}$, is low and the delay of single-destination traffic, D_{single}, and the delay of multi-destination traffic, D_{multi} is acceptable by higher layer applications. Our approach is outlined in the following Schedule-Merging Heuristic (SMH). Note that the stopping rule in Step 4 guarantees that the final schedule will contain broadcast slots.

Schedule-Merging Heuristic (SMH)

1. Given single-destination traffic parameters σ_i and p_{ij}, and the number of wavelengths, C, obtain an optimized unicast slot schedule, S_0, of frame length M, as in Appendix A.1.

2. Given ρ_i and C obtain a broadcast slot schedule, S'_0, of frame length M', optimized for multi-destination traffic only (as in Appendix A.2). Set $l \leftarrow 1$.

3. Merge one frame of S_0 with l frames of S'_0 to produce a new schedule, S_l, of frame length $M + lM'$.

4. If $l = 1$ or ($D_{overall}(S_l) < D_{overall}(S_{l-1})$ *and* $D_{single}(S_l)$) and $D_{multi}(S_l)$ acceptable, set $l \leftarrow l + 1$ and repeat from Step 3. Otherwise, stop; the best schedule to use is S_{l-1}.

5 Adaptive Multicast Protocols for Type 3 Traffic

We now present a suite of adaptive multicast protocols for FT-TR networks which assume that each station, i, is the owner of b_i multicast slots per frame. The protocols are adaptive in the sense that the transmissions allowed in these multicast slots are not specified in advance; instead, they are dynamically updated to reflect the current members of multicast groups. A station, i, may transmit to any multicast group in its b_i multicast slots $t_1, \ldots, t_{b_i}$, and as the multicast group changes, the permissions in each of these slots (see (11)) also change so that the overall throughput is maximized.

5.1 The Basic Idea

The operation of the protocols is based on the assumption that a source, i, will typically transmit P consecutive packets, $P > 1$, to the same multicast group, g. This is true, for example, in the case of bulk arrivals, i.e., when a long message has to be fragmented into a number of fixed-size packets or in the case where multicast is being used to provide a conferencing service. P need not be constant; for example, P could be an integer in the range $[P_{min}, P_{max}]$, where P_{min} and P_{max} may correspond to the minimum and maximum message size, respectively. We will now describe the basic idea behind the operation of the protocols by considering the transmissions in source station i's multicast slots; similar observations can be made for other stations' multicast slots. We will also assume that i transmits to only one multicast group at a time. The protocols, however, can be easily extended to handle a source that maintains a small number, m, of multicast sessions simultaneously; in these situations, it would be beneficial to the source to have m rather than 1 multicast queues, one for the packets belonging to each session (see also the discussion in Section 2.2).

In the first multicast slot with i as its owner all stations tune their receivers to $\lambda(i)$, the transmit wavelength of i. Let g be the multicast group to which the packet transmitted by i in that slot is addressed, and let P be the total number of packets i will transmit to the same group; $g = \phi$ if no packet is transmitted by i in that slot. Suppose that $\mid g \mid < N - 1$, and consider a station $j \neq i$. If $j \in g$ then j will continue listening to $\lambda(i)$ in subsequent multicast slots of i. However, if $j \notin g$, j is free to tune its receiver to the transmit wavelength of another station, k, in subsequent multicast slots of i. If k has a single-destination packet for j, and provided that $\lambda(k) \neq \lambda(i)$, it can transmit it in i's multicast slots, thus increasing channel utilization.

After i transmits all P packets to the same multicast group g, it will not be able to transmit to a group $g' \neq g$, unless all stations not in g are somehow notified. We therefore require that all stations tune their receivers to $\lambda(i)$ in specified multicast slots of i, called *synchronization* slots [5] (as explained, the first multicast slot is a synchronization slot). The F multicast slots of i between synchronization slots are called *free* as receivers not in g are free to tune to any wavelength other than $\lambda(i)$. F is a network-wide constant and thus, all stations can synchronize by tuning to $\lambda(i)$ in synchronization slots.

Note that i may start transmitting packets to a new multicast group only in a synchronization slot. If F is large relative to P, the number of consecutive packets to the same multicast group, i will, on average, have to wait for a considerable number of slots to start transmitting to a new group. On the other hand, if F is very small relative to P there will be unnecessarily many synchronization slots in which no transmissions by stations other than i are allowed. F will, in general, be a function of P, as well as of the propagation delay (more on this later), and must be carefully selected in order to maximize the overall throughput.

We have not yet discussed how a receiver $j \notin g$ selects a transmitter $k \neq i$ to tune to in i's free multicast slots. There are two issues that need to be considered. First, $\lambda(k)$ must be different than $\lambda(i)$ to prevent packet loss due to collisions in free multicast slots. Second, k must also be informed of j's decision. Real-time negotiation between j and other stations to determine k is impractical because of the propagation delays involved.

To solve the first problem we start with a unicast slot schedule, S, of frame length M, which satisfies (6), (7) and (8); let a_i be the number of slots per frame in which i may transmit under S, $a_i = \sum_{t=1}^{M} \sum_{j=1}^{N} \delta_{ij}^{(t)}$. We then specify $b_i, b_i < a_i$, of these slots as multicast slots with i as their owner. Let t be one of these b_i slots and consider a station $j \neq i$ which, according to schedule S, has to tune its receiver to station k in slot t. If t is a synchronization slot, or if t is a free slot but $j \in g$, j will ignore the permissions specified by S and, in slot t, it will tune to $\lambda(i)$ instead. However, if t is a free multicast slot and $j \notin g$, j will tune to $\lambda(k)$ as S specifies. Note that, since S satisfies (7) and both i and k are given permission to transmit in the same slot t, we have $\lambda(i) \neq \lambda(k)$.

5.2 Determining Group Membership

Since all stations execute the same protocol, the problem of informing k about j's decision is now partially solved: k knows that j will tune to $\lambda(k)$ in slot t if (a) t is a free slot, and (b) $j \notin g$. Deciding about (a) is done by k as part of the protocol for tuning its own receiver. Thus the problem reduces to how k may determine whether j is in the multicast group g or not. We now describe three protocols which differ in their assumptions about k's knowledge regarding membership in the

[5] Synchronization slots are actually broadcast slots.

multicast groups of packets originating at station $i \neq k$.

Global-knowledge Multicast Protocol (GMP). Station k maintains tables to map a multicast address in a packet originating at i into the station-members of the multicast group. By listening to a synchronization slot of i it can tell whether j is in the multicast group or not. Since k must have similar tables for all i, this protocol may be very expensive in terms of memory requirements, as well as in terms of the communication cost for building and maintaining the tables.

Control-packet Multicast Protocol (CMP). Station k has no knowledge about the members of multicast groups of packets originating at i [6]. However, before transmitting a packet to a new multicast group, g, i will first transmit, in a synchronization slot, a control packet with information about the members of g. Following the control packet transmission, i will transmit the P packets to g as discussed above. Station k uses the control packet to associate g with the group members. This protocol incurs the overhead of one extra packet, but this is not expected to be a problem, especially if $P \gg 1$. In addition, this protocol does not require building and maintaining potentially large global tables at each station.

Probabilistic Multicast Protocol (PMP). Station k has no way of finding out whether j belongs to g or not. It will transmit a packet to j in a free multicast slot of i, if it has one, with probability r [7]. No overhead in terms of memory or control packets is incurred, but the selection of an appropriate value for r is crucial in order to minimize packet loss due to destination conflicts (if $j \in g$, j will tune to $\lambda(i)$ in free multicast slots of i and k's transmissions in these slots will be wasted). In general, r should represent the probability that j does *not* belong to g. If $\bar{\eta}$ is the average number of stations in a multicast group, we set $r = 1 - \frac{\bar{\eta}}{N-1}$.

5.3 Effect of Propagation Delay

Under either GMP or CMP a transmitter $k \neq i$ must receive the packet transmitted by i in a synchronization slot before it can determine whether the stations to which it is scheduled to transmit in the next free multicast slots of i belong to g or not. Figure 3 illustrates how propagation delay may become a problem. In this figure we show a synchronization slot of i followed by F free slots and another synchronization slot; the horizontal axis represents time increasing from left to right. The transmitters of both i and k are synchronized at the beginning of each slot. But a packet transmitted by i will not be heard by the receiver of k until τ_{ik} slots later, where τ_{ik} is the propagation delay from i to k in slots. In the scenario of Figure 3, by the time k receives the packet

[6] Except, of course, for deciding whether k itself is in the multicast group or not.

[7] A schedule with PMP may not satisfy constraint (8).

transmitted by i in the first synchronization slot, free slot t_1 has already passed by its transmitter. Since at the beginning of t_1, k does not know whether $j \in g$ it may not transmit a packet to it.

As a result of the propagation delays some of the free multicast slots may not be used for single-destination transmissions; the longer the propagation delays the less free slots that may be utilized. In the extreme case when all F free slots are within a propagation delay, neither GMP nor CMP will be able to capitalize on the availability of free slots to improve the throughput. Thus, F is indeed a function of the propagation delay as mentioned earlier. Observe, though, that the propagation delay will have a negative effect only if it increases beyond the number of slots between *consecutive multicast* slots with the same owner. Going back to Figure 3, if τ_{ik}, in slots, is less than the distance between the first synchronization slot and t_1, k will be able to transmit in t_1, as well as in all other free slots (if $j \notin g$). Otherwise, k will still be able to use slots $t_2, \ldots, t_F$ as long as τ_{ik} is further increased by less than the distance between t_1 and t_2, and so on. By assigning multicast slots to i so that they are spaced out in the frame we can make the distance between two consecutive multicast slots much larger than one slot. We then expect GMP and CMP to be largely insensitive to propagation delays.

PMP was devised to overcome this problem. Under this protocol, k does not need to wait until it receives the packet transmitted in the first synchronization slot. Regardless of whether $j \in g$ or not, k will transmit in slot t_1 with probability r, provided that it has a packet for j. Thus, PMP is not affected by propagation delays at all.

The algorithms used by the various transmitters and receivers for transmissions in i's multicast slots are shown in Figures 4, 5, and 6. The algorithms are very simple to implement, and thus suitable for the high-speed environment we are considering.

6 Numerical Results

We consider the 8-station ring-type, two-community-type, and two-server-type single-destination traffic matrices with probabilities p_{ij} as shown in Figures 7, 8, and 9, respectively. We let $\sigma_i = \sigma \; \forall \; i$, and $\rho_i = \rho \; \forall \; i$; this does not compromise the generality of our results as (a) the single-destination traffic characteristics are determined by p_{ij}, and (b) we are interested in the behavior of our schedules as the relative amount of single- and multi-destination traffic varies, and this is captured by the relative values of σ and ρ, and the average multicast group size, $\bar{\eta}$. In the simulations, the number of consecutive multicast packets is uniformly distributed in the range $[P_{min}, P_{max}]$. Multicast groups and group sizes are chosen at random while maintaining an average group size of $\bar{\eta}$. All simulation results presented in this section were obtained with a confidence of 95% in less

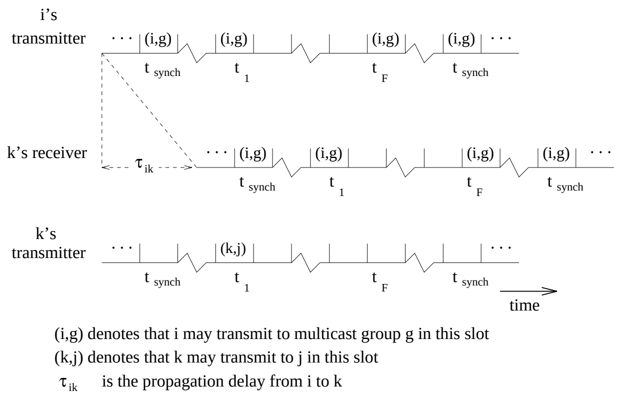

Figure 3: Effect of the propagation delay (not in scale)

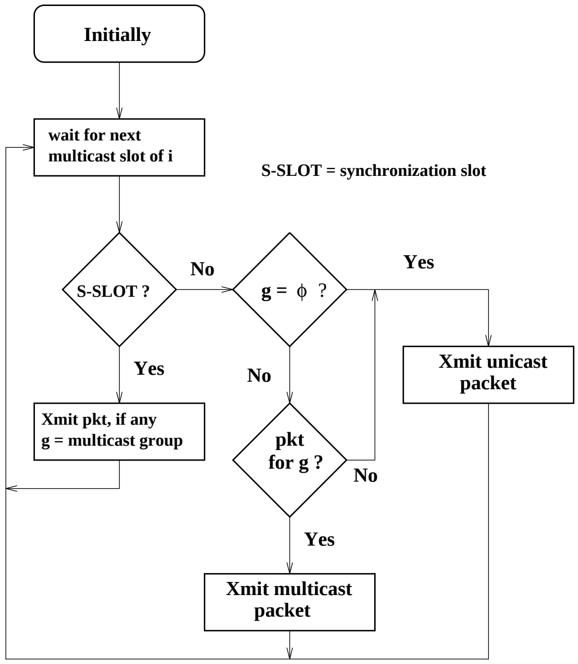

Figure 4: Algorithm executed by i's transmitter for transmission in i's multicast slots.

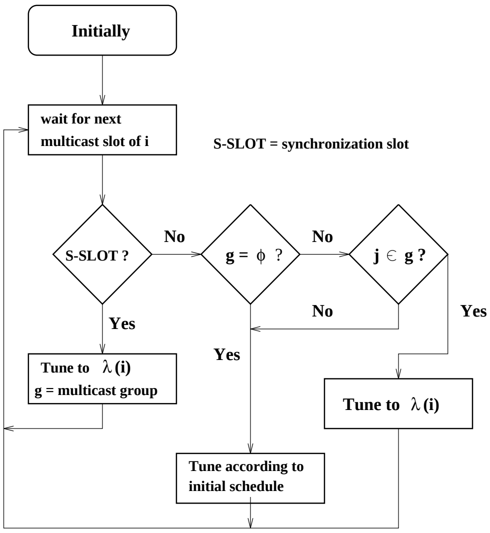

Figure 5: Algorithm executed at j's receiver for tuning in i's multicast slots.

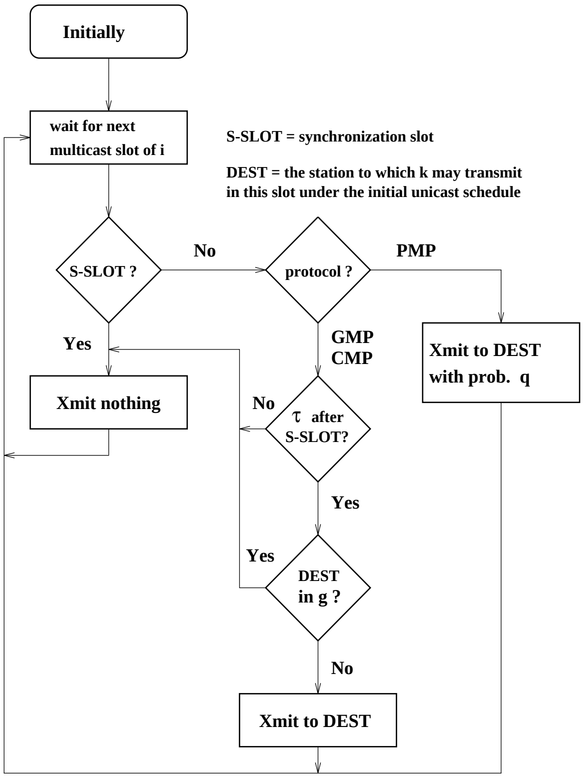

Figure 6: Algorithm executed by k's transmitter for transmission in i's multicast slots ($k \neq i$).

than 1% variation from the mean.

Figures 10, 12, and 14 plot the delay, and Figures 11, 13, and 15 plot the throughput against the average multicast group size for the stated traffic matrix and values for parameters C, σ, and ρ. Each figure shows three delay or throughput curves, each curve corresponding to one of the multicast approaches presented earlier, namely, schedules with unicast slots only, schedules with unicast and broadcast slots, and schedules with unicast and multicast slots with the GMP adaptive multicast protocol. Similar results have been obtained for various other traffic matrices, and a wide range of values for parameters C, σ and ρ.

From Figures 11, 13, and 15 we see that throughput increases with average multicast group size, and that the three approaches have almost identical performance in terms of throughput (note that when the average size of the multicast groups is 7, in effect we have a broadcast traffic scenario). However, the three approaches differ drastically in terms of their delay behavior, as evident in Figures 10, 12, and 14; this behavior can be explained by considering the way multicast packets are handled, and is the subject of the following discussion.

First observe that when schedules with unicast and broadcast slots are used, the delay is independent of the multicast group size. Recall that these schedules are constructed using the Schedule-Merging Heuristic (SMH) of Section 4, and that SMH inserts a number of broadcast slots according only to parameters σ and ρ (corresponding to the offered unicast and multicast traffic load, respectively). These broadcast slots are used for sending multicast packets, thus the delay of each packet is independent of the number of recipients. On the other hand, when schedules with unicast slots only are used, a multicast packet has to be transmitted multiple times, once to each of the members in its multicast group. Hence, as the size of the multicast groups increases, delay also increases, resulting in the curves shown in Figures 10, 12, and 14. As we can see, for small multicast group sizes, there is a penalty for using broadcast slots (all receivers tune to a certain wavelength, despite the fact that the multicast packet is addressed to a fraction of all possible destinations) making these schedules inferior to schedules with unicast slots only. However, with large multicast groups, transmitting a packet multiple times results in a waste of network resources; consequently, there is a point at which the two curves intersect, and beyond which it is preferable to use schedules with both unicast and broadcast slots. These results confirm our intuition regarding the factors affecting the design of schedules for a particular system (see Section 3 and Table 1). From the figures, it is also clear that the point at which one approach outperforms the other is not fixed, but depends on the various system and traffic parameters.

Let us now turn our attention to the delay behavior of schedules with unicast and multicast slots running the Global-knowledge Multicast Protocol (GMP). The most important observation from Figures 10, 12, and 14, though, is that, although these schedules do not achieve lower delay compared to the ones discussed above, they *always* incur a delay close to that of the best static schedule. This behavior of GMP is better understood by examining the two extreme cases, namely, very small and very large multicast groups.

Let us assume that all multicast groups consist of one member, and consider the multicast slots of source i. In such a system, the free multicast slots of i behave almost like unicast slots, as one receiver (the one belonging to the current multicast group) will have to tune to $\lambda(i)$, while all others are free to tune to the home channels of other sources. However, GMP does incur some overhead compared to the pure unicast schedule, as all receivers must tune to $\lambda(i)$ during the synchronization slots of i. On the other hand, if the size of multicast groups tends to $N-1$ (a broadcast scenario), the behavior of the multicast protocol becomes similar to that of a static schedule with unicast and broadcast slots. The difference in this case is that, once all packets to the current multicast group have been transmitted, the source must wait for the next synchronization slot before it can transmit packets to a new multicast group. As a result, GMP incurs some overhead compared to the static schedule. The behavior of GMP between these two extremes is determined by which overhead is the dominant one. Thus, depending on the average multicast group size, the performance of GMP matches that of the schedule that incurs the lowest delay.

Although we have shown results only for GMP, the Control-packet Multicast Protocol (CMP) achieves very similar performance (recall that CMP differs from GMP only in that the source transmits, in a synchronization slot, an initial control packet containing information about the multicast group, before it transmits the actual multicast data packets). Finally, it has been shown [18] that PMP achieves a throughput which is slightly less than that of GMP or CMP. The disadvantage of PMP is the inevitable packet loss due to destination conflicts, leading to retransmissions and thus higher delays.

7 Concluding Remarks

We have addressed the problem of carrying both multi- and single-destination traffic over single-hop WDM networks with tunability provided at the receiving end only. We have presented a classification of multi-destination traffic that takes into account the size of multicast groups and the length of multicast sessions, and have designed multicast mechanisms addressing the requirements of each type of traffic. We have also developed a suite of adaptive multicast protocols which are simple to implement, and which can be useful under changing traffic conditions. Our main conclusion is that slot assignment adaptability is both desirable and feasible when multi-destination traffic is being carried in WDM networks.

Overall, our results indicate that the multicast protocols we have developed can successfully adapt to a wide range of multicast group sizes. However, this is only possible when the number of packets to be transmitted to the same multicast group is relatively large (in the figure presented, the values of P_{min} and P_{max}, the minimum and maximum number of packets to the same multicast group, was set to 30 and 50, respectively). Otherwise, an appropriately optimized (static) schedule with unicast or unicast and broadcast slots (depending on the expected group size) will achieve the best performance.

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 7: Ring-type matrix

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

Figure 8: Disconnected-type matrix

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 9: Two-server-type matrix

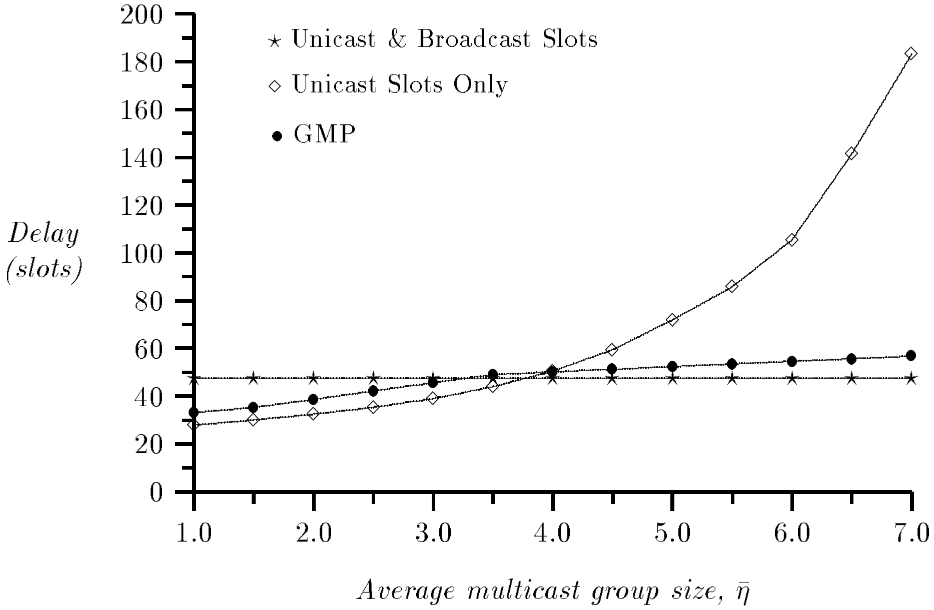

Figure 10: Delay versus average multicast group size, two-community-type matrix, $N = 8$, $C = 2$, $\sigma = 0.1$, $\rho = 0.01$ (GMP was run with: P_{min}=30, P_{max}=50, F =50)

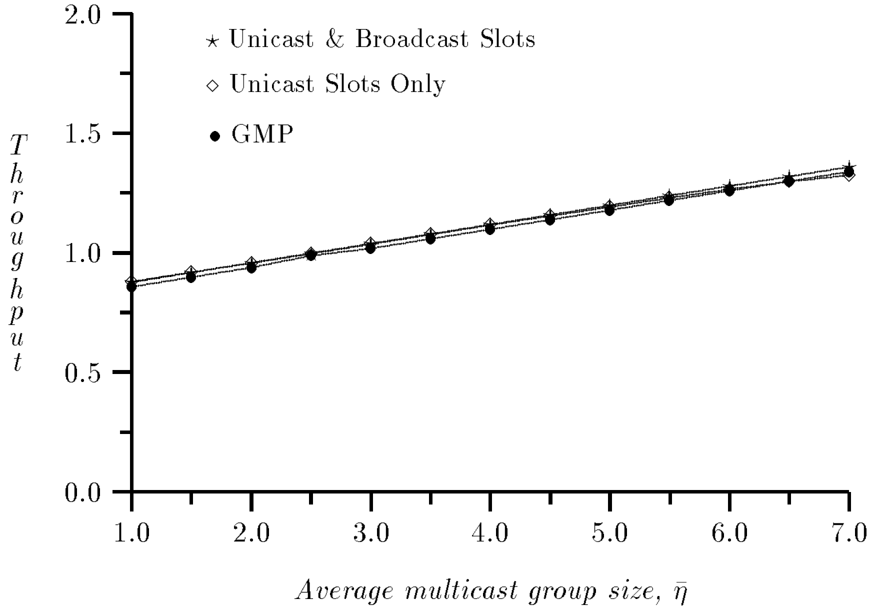

Figure 11: Throughput versus average multicast group size, two-community-type matrix, $N = 8$, $C = 2$, $\sigma = 0.1$, $\rho = 0.01$ (GMP was run with: P_{min}=30, P_{max}=50, F =50)

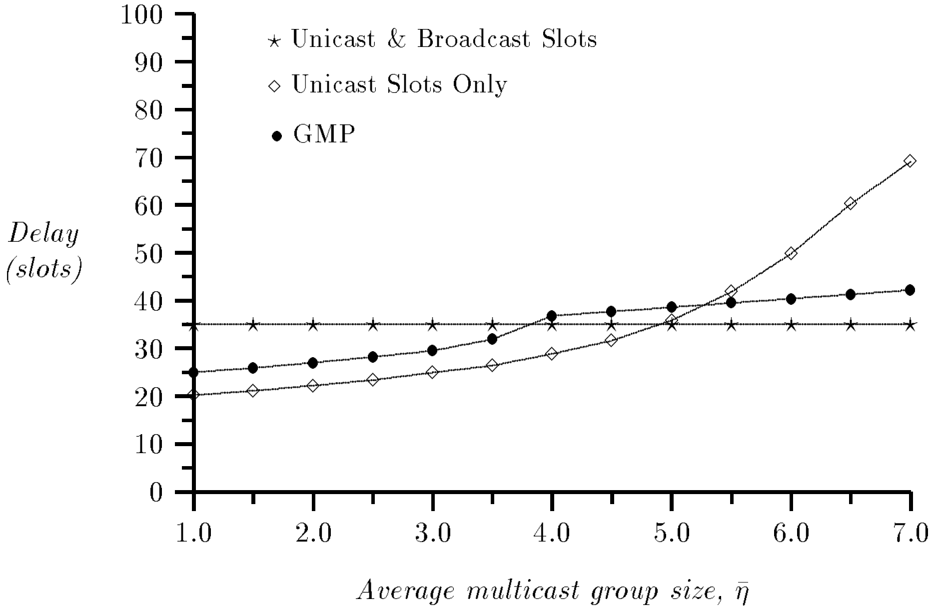

Figure 12: Delay versus average multicast group size, two-server-type matrix, $N = 8$, $C = 4$, $\sigma = 0.3$, $\rho = 0.01$ (GMP was run with: P_{min}=30, P_{max}=50, F =50)

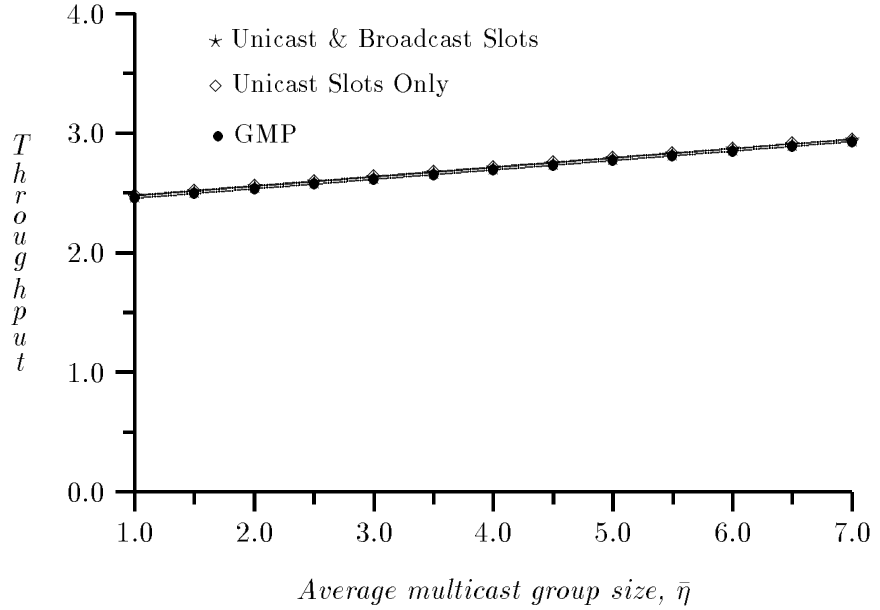

Figure 13: Throughput versus average multicast group size, two-server-type matrix, $N = 8$, $C = 4$, $\sigma = 0.3$, $\rho = 0.01$ (GMP was run with: P_{min}=30, P_{max}=50, F =50)

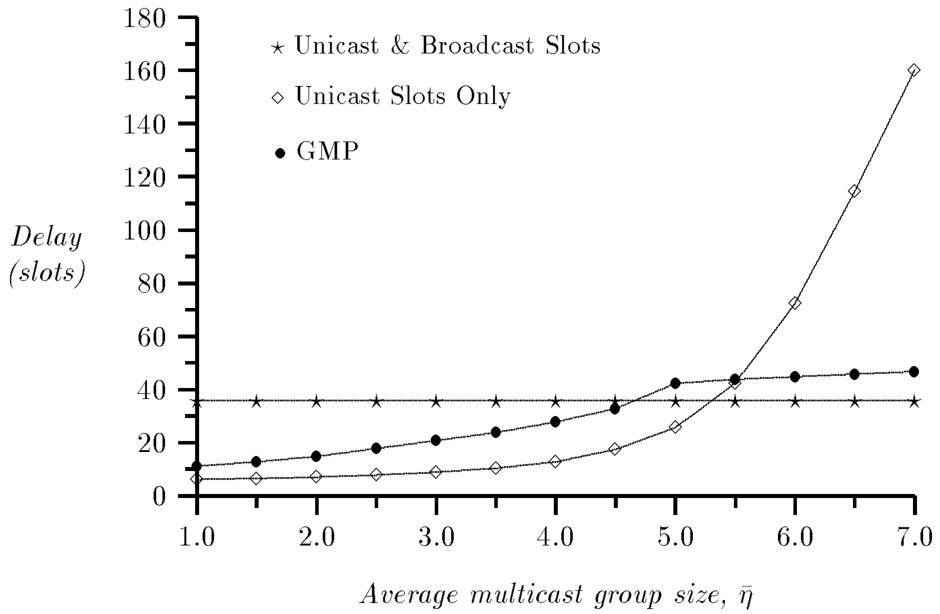

Figure 14: Delay versus average multicast group size, ring-type matrix, $N = 8$, $C = 8$, $\sigma = 0.5$, $\rho = 0.02$ (GMP was run with: P_{min}=30, P_{max}=50, F =50)

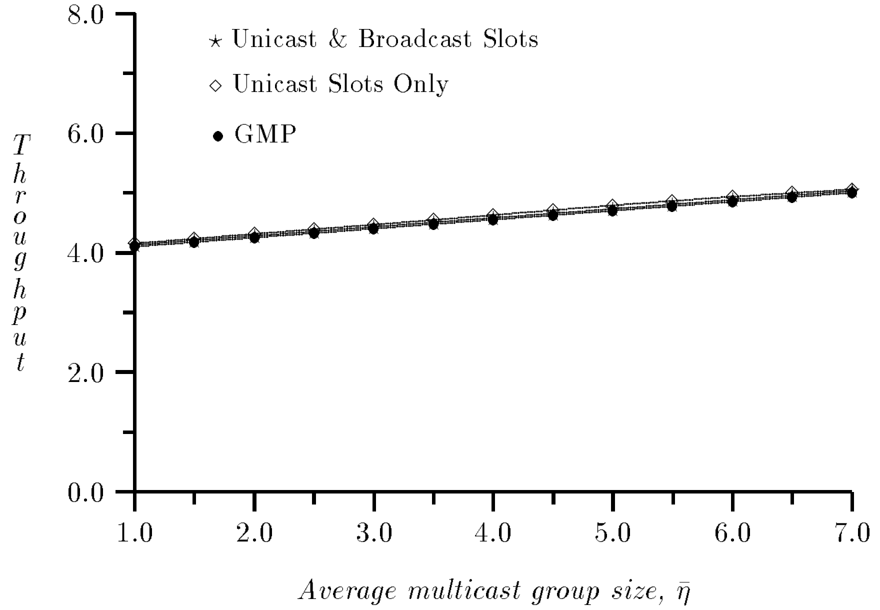

Figure 15: Throughput versus average multicast group size, ring-type matrix, $N = 8$, $C = 8$, $\sigma = 0.5$, $\rho = 0.02$ (GMP was run with: P_{min}=30, P_{max}=50, F =50)

References

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

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] J. S. Turner. New directions in communications (or which way to the information age?). *IEEE Communications Magazine*, 24(10):8–15, October 1986.

[10] J. S. Turner. The challenge of multipoint communication. In *Proceedings of 5th ITC Seminar*, pages 263–279, May 1987.

[11] M. Ahamad and M. H. Ammar. Performance characterization of quorum-consensus algorithms for replicated data. *IEEE Transactions on Software Engineering*, 15(4):492–496, April 1989.

[12] D. Gifford. Polychannel systems for mass digital communication. *Communications of the ACM*, 33:141–151, Feb 1990.

[13] C.-T. Lea. A multicast broadband packet switch. *IEEE Transactions on Communications*, pages 621–630, April 1993.

[14] B. Rajagopalan. Reliability and scaling issues in multicast communication. In *Proceedings of SIGCOMM '92*, pages 188–198. ACM, 1992.

[15] M. Decina, V. Trecordi, G. Zanolini, and D. Zucca. An approximate analysis of broadcasting in multichannel metropolitan area networks. In *Proceedings of INFOCOM '93*, pages 293–302. IEEE, 1993.

[16] S. Tridandapani and B. Mukherjee. Multicast traffic in multi-hop lightwave networks: Performance analysis and an argument for channel sharing. In *Proceedings of INFOCOM '96*. IEEE, March 1996.

[17] M. Borella and B. Mukherjee. A reservation-based multicasting protocol for WDM local lightwave networks. In *Proceedings of ICC '95*, pages 1277–1281. IEEE, 1995.

[18] 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.

[19] G. N. Rouskas. *Single-Hop Lightwave WDM Networks and Applications to Distributed Computing*. PhD thesis, Georgia Institute of Technology, Atlanta, GA, May 1994.

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

[21] 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.

[22] G. N. Rouskas and V. Sivaraman. On the design of optimal TDM schedules for broadcast WDM networks with arbitrary transceiver tuning latencies. In *Proceedings of INFOCOM '96*, pages 1217–1224. IEEE, March 1996.

[23] 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.

[24] 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.

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

[26] M. H. Ammar and G. N. Rouskas. On the performance of protocols for collecting responses over a multiple-access channel. *IEEE Transactions on Communications*, 43(2):412–420, February 1995.

A Schedule Optimization

We now address the problem of obtaining schedules which minimize the average packet delay, assuming that each type of traffic (single- or multi-destination) is offered to the network in isolation.

A.1 Schedules With Unicast Slots Only For Single-Destination Traffic

If only single-destination traffic is offered to the network (i.e., $\rho_i = 0 \ \forall \ i$) our problem may be stated as:

Problem 2 *Given the number of stations, N, the number of available wavelengths, C, and the traffic parameters, $\sigma_i p_{ij}, i, j = 1, \ldots, N$, find a schedule that minimizes the network-wide average packet delay, assuming that buffers of infinite capacity are available at each station.*

There are three dimensions to this problem [8]:

- the sets of transmitters, X_c, sharing wavelength $\lambda_c, c = 1, \ldots, C$, must be constructed,
- the number of slots per frame, a_{ij}, allocated to each source-destination pair (i, j) must be obtained, and
- a way of placing the a_{ij} slots within the frame, for all i, j, must be determined.

This problem is somewhat different than the tunable-transmitter, fixed-receiver one addressed in [8], but is still a very hard allocation problem. Our approach, then, is to first construct sets X_c using the Weight Balancing Heuristic in [8] (which attempts to balance the traffic load across all available channels). Given these sets X_c, we now present a heuristic to obtain near-optimal schedules.

Let us consider channel λ_c in isolation. Since only stations in X_c may transmit on λ_c, and the packet arrival at station i is described by σ_i, this is exactly the single-channel problem in [21]. It was shown there that the average packet delay is minimized when the percentage of time station i is permitted to transmit on channel λ_c is [21]

$$x_i = \sigma_i + (1 - \sum_{k \in X_c} \sigma_k) \frac{\sqrt{1-\sigma_i}}{\sum_{k \in X_c} \sqrt{1-\sigma_k}} \quad \forall \ i \in X_c \tag{12}$$

[8] 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.

Note that x_i are independent of the frame length M. Once we have obtained x_i, and given M, we may obtain the number of slots per frame in which $i \in X_c$ may access channel λ_c as

$$\lfloor M\ x_i \rfloor \quad \leq \quad \alpha_i \quad \leq \lceil M\ x_i \rceil \ \ \forall\ i \in X_c\ \ ; \ \sum_{i \in X_c} \alpha_i \quad = \quad M \tag{13}$$

We then need to determine how these α_i slots should be allocated for transmissions from i to each of the potential receivers. Note that the $N-1$ queues at i do not interact with each other; thus we have again an equivalent single-channel problem, which dictates that the percentage of time station i transmits to each destination be:

$$y_{ij} \quad = \quad \sigma_i p_{ij} + (1 - \sum_{l=1}^{N} \sigma_i p_{il}) \ \frac{\sqrt{1 - \sigma_i p_{ij}}}{\sum_{l=1}^{N} \sqrt{1 - \sigma_i p_{il}}} \qquad \forall\ i \in X_c \quad j = 1, \ldots, N \tag{14}$$

Therefore, the number of slots allocated to the (i, j) pair should be

$$\lfloor \alpha_i\ y_{ij} \rfloor \quad \leq \quad a_{ij} \quad \leq \quad \lceil \alpha_i\ y_{ij} \rceil \ \ ; \ \sum_{j=1}^{N} a_{ij} \quad = \quad \alpha_i \qquad \forall\ i \in X_c \tag{15}$$

Furthermore, no receiver should be assigned to receive in more than M slots in a frame:

$$\sum_{i=1}^{N} a_{ij} \quad \leq \quad M \qquad \forall\ j \tag{16}$$

Finally, according to the results in [21], for each source-destination pair (i, j), the a_{ij} slots in which i transmits to j should be equally spaced within the frame, i.e.,

$$\forall\ i\ : \quad d_{ij}^{(k)} \quad = \quad d_{ij} \quad , \quad k = 1, \ldots, a_{ij} \tag{17}$$

Once a_{ij} have been determined for all i and j, we need to construct the schedule so that the permissions assigned to each source-destination pair be placed within the frame according to (17). However, this is not feasible in general, even in the single-channel case, as d_{ij} may not be integers. To overcome this problem, a golden-ratio policy was developed in [21], 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 (17), and, as a result, achieves an average packet delay very close to the lower bound.

Our approach is to use the golden ratio policy (which only considers frames of lengths equal to Fibonacci numbers) to place the permissions within each channel independently of the others. This, however, may result in allocations that violate (8). In other words, considering channels in isolation may cause a receiver to be assigned to tune on two or more channels in the same slot.

If this occurs, we must rearrange the schedule to remove these violations. To this end, we use algorithm REARRANGE, described in [7], 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 in [8] to determine the set of transmitters, $X_c, c = 1, \ldots, C$, that share each channel.

2. Select a Fibonacci number, M, as the frame length, and obtain $a_{ij}(M)$ from (12)-(16).

3. Let $c = 1$, and use the golden ratio policy [21] to place the $a_{ij}, i \in X_c$, slots in a frame for transmissions on channel λ_c. Repeat for $c = 2, \ldots, C$ to obtain an initial schedule, $S_0(M)$.

4. Run algorithm REARRANGE, described in [7], to perturb S_0, producing a schedule, $S(M)$, which satisfies constraints (7) and (8).

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.

In this Appendix we have assumed that $\rho_i = 0$, i.e., there is only single-destination traffic. Our approach, however, can be easily adapted to obtain schedules with unicast slots only, appropriate for networks in which Type 2 multi-destination traffic is also offered (recall that Type 2 traffic is such that the average multicast session is short, and the average multicast group size is small). In this case, in addition to σ_i and p_{ij}, we have the multi-destination traffic parameters ρ_i and q_{ig}. Since unicast slots will be used to carry multicast packets, a packet for multicast group g arriving at source i is copied into all queues j of i, such that $j \in g$ [9]. To construct a schedule for such a network we can apply the Slot Allocation Heuristic to an equivalent network with single-destination traffic only, such that its parameters, σ'_i and p'_{ij}, account for both the single- and multi-destination traffic of the original network ($\bar{\eta}$ is the average size of a multicast group):

$$\sigma'_i \quad = \quad \sigma_i \ + \ \bar{\eta}\, \rho_i \qquad \forall\ i \tag{18}$$

$$p'_{ij} \quad = \quad \frac{\sigma_i \ p_{ij} \ + \ \rho_i \ \sum_{g, j \in g} \ q_{ig}}{\sigma'_i} \qquad \forall\ i, j \tag{19}$$

[9] Therefore, there is no need for a separate multicast queue, and each source only needs $N-1$ queues.

A.2 Schedules With Broadcast Slots Only For Multi-Destination Traffic

Recall that only one station is allowed to transmit in a broadcast slot. Given $\rho_i, i = 1, \ldots, N$, the problem of obtaining an optimal broadcast schedule is then equivalent to the single-channel problem in [21]. Therefore, the percentage of time, z_i, that station i should be given permission to transmit is:

$$z_i \;=\; \rho_i + (1 - \sum_{k=1}^{N} \rho_k) \; \frac{\sqrt{1-\rho_i}}{\sum_{k=1}^{N} \sqrt{1-\rho_k}} \qquad i = 1, \ldots, N \tag{20}$$

z_i is independent of the frame length M. Given a Fibonacci number $M \geq N$ [21], we assign $b_i(M)$ broadcast slots to station i such that

$$\lfloor M z_i \rfloor \;\leq\; b_i(M) \;\leq\; \lceil M z_i \rceil \quad \forall\, i \quad \text{and} \quad \sum_{i=1}^{N} b_i(M) \;=\; M \tag{21}$$

We then use the golden-ratio policy [21] to place the b_i slots, $i = 1, \ldots, N$, within the frame.