Multi-Destination Communication Over Single-Hop Lightwave WDM Networks

George N. Rouskas and *Mostafa H. Ammar*
College of Computing, Georgia Institute of Technology, Atlanta, GA 30332-0280

Abstract

We address the open issue of providing efficient mechanisms for multi-destination communication over one class of lightwave WDM architectures, namely, single-hop networks. We suggest, analyze, and optimize several alternative approaches for broadcast/multicast. One of our major contributions is the development of a suite of adaptive multicast protocols which have very good performance, are very simple to implement, and are insensitive to propagation delays.

1 Introduction

With the advent of telecommunication services and computer applications (such as distributed data processing [1], broadcast information systems [2], and teleconferencing, among others) requiring multi-destination communication, it is now likely that a significant portion of the overall traffic in future communication environments will be of the multi-destination type. It is, therefore, important that next-generation networks employ efficient multicast mechanisms [3].

The single-hop architecture for lightwave networks [4, 5, 6, 7, 8] employs Wave Division Multiplexing (WDM) to divide the enormous information-carrying capacity of fiber into multiple concurrent channels, and may deliver an aggregate throughput in the order of Terabits per second. Single-hop networks are all-optical in nature, i.e., a packet is sent from the source to the destination without passing through intermediate stations. 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, an extension of TDMA over a multichannel environment.

For *single-destination* traffic, we have developed a general framework for analyzing and optimizing the throughput performance for any transceiver tunability characteristics, and general (potentially non-uniform) traffic patterns [5]. The issue of *multi-destination* traffic has only been addressed in the context of single-hop networks in [9], where, under simplifying assumptions, upper bounds on the system performance are derived. In this paper we suggest, analyze, and optimize several alternative approaches to performing efficient broadcast/multicast over single-hop lightwave networks.

Section 2 describes the network model, and Section 3 analyzes and optimizes the performance of schedules suitable for mixed (single- and multi-destination) traffic. In Section 4 we develop a suite of adaptive multicast protocols. Section 5 presents some numerical results, and Section 6 concludes the paper.

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, $\lambda_1, \ldots, \lambda_C$. In general, $C \leq N$. The properties of the network depend on whether the receivers only, or the transmitters only are tunable; we refer to these systems as FT-TR and TT-FR, respectively. In the former case, wavelength $\lambda(i) \in \{\lambda_1, \ldots, \lambda_C\}$ is assigned to the fixed transmitter of station i, and the receivers are tunable over all wavelengths $\lambda_c, c = 1, \ldots, C$. Similarly for TT-FR systems. Due to space constraints, only tunable-receiver networks will be considered here; a discussion on TT-FR networks can be found in [10].

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. The network operates in a slotted mode [11], with a slot time equal to the packet transmission time plus the tuning time. We define σ_i and ρ_i as the probability that a new single-destination and multi-destination packet, respectively, arrives at station i during a slot time. p_{ij} is the probability that an arriving single-destination packet is destined to station j, and $\sum_j p_{ij} = 1$. Each station has N buffers, one for storing packets destined to each of the $N-1$ possible destinations, and one for storing multi-destination packets. Each buffer can hold one packet; packets arriving to a full buffer are lost. This is an extension of the models in [5, 12].

Time slots are grouped in frames of M slots. A schedule indicates, for all i and j, which slots within a frame can be used for transmissions from i to j; it can be described by variables $\delta_{ij}^{(t)}, t = 1, \ldots, M$, called *permissions*, and defined as

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

Whenever $C < N$, a number of fixed receivers or transmitters may have to be assigned the same wavelength $\lambda_c, c = 1, \ldots, C$. We let R_c and X_c, subsets of $\{1, \ldots, N\}$, denote the set of receivers and transmitters, respectively, sharing wavelength λ_c.

We will be concerned with evaluating the performance of the various schedules in terms of throughput, defined as the expected number of packets successfully received per slot. Under certain circumstances to be discussed shortly, 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 higher than the number of channels, C, if $C < N$.

2.1 Transmission Modes

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 a multicast communication. One way to guarantee that a packet will be received by all current members of its multicast group would be to have the source broadcast the packet to the entire network. The receivers would then use the multicast address to filter out any packets they do not need. If, however, 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 the packet's multicast group. We now define the following transmission modes to exploit the inherent multicast capability of FT-TR networks (formal definitions in terms of $\delta_{ij}^{(t)}$ can be found in [10]):

Broadcast mode. One station, i, the *owner* of slot t, is given permission to transmit to all stations (which have to tune their receivers to $\lambda(i)$, the transmit wavelength of i, in that slot). No other station may transmit in slot t.

Multicast mode. One station, i, the owner of slot t, may transmit to a multicast group g. Other stations may transmit to destinations not in the multicast group, in one-to-one mode (to be described shortly), and in channels other than $\lambda(i)$.

In a multicast slot only current members of a multicast group, g, tune their receivers to the source's transmit wavelength; other receivers may tune to other wavelengths. 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 and, consequently, higher throughput. However, 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 such groups explodes with the number of stations. In Section 4 we present several adaptive protocols to dynamically update the permissions in each multicast slot according to the current multicast groups. Since the protocols are based on the assumption that a source will have $L > 1$ packets to transmit to the same multicast group, broadcast slots may still need to be used in situations where this assumption does not hold (for instance, when multicast is used for response collection [13]).

Finally, the one-to-one mode [5] will be used to transmit single-destination packets in FT-TR networks.

One-to-One mode. The one-to-one mode is such that within a slot t (a) exactly C permissions are given to different source-destination pairs, one per channel, (b) no transmitter is given more than one permission, and (c) no two stations may transmit a packet to the same destination.

A schedule will be called "one-to-one", "broadcast", or "multicast" if it consists entirely of one-to-one, broadcast, or multicast slots, respectively. Figure 1 demonstrates the various transmission modes for a FT-TR network with $N = 4$ stations, $C = 2$ wavelengths, $X_1 = \{1, 3\}$ and $X_2 = \{2, 4\}$. The cyclic schedules also shown are special cases, whereby each station may transmit to each possible destination exactly once per frame (for the one-to-one mode), or has exactly one broadcast slot per frame (for the broadcast mode).

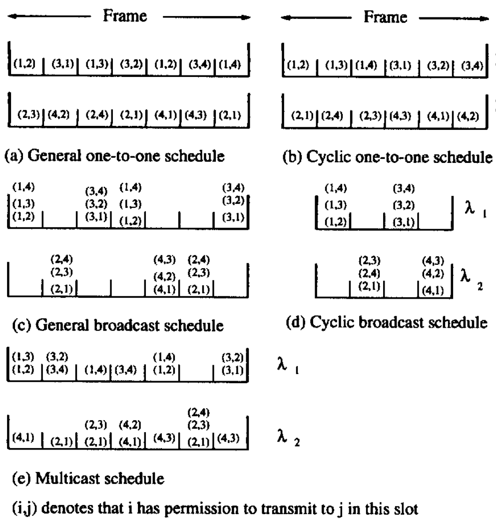


Figure 1: Transmission Modes

3 Analysis and Optimization

In any realistic environment, the traffic pattern will typically consist of a mix of point-to-point and multicast traffic. As a result, for the case of FT-TR networks that we are considering, we need schedules with both one-to-one and broadcast slots. One-to-one slots will be used exclusively for single-destination packets, while broadcast slots will be used solely for broadcasting multi-destination packets to the network. Note that, for a given schedule, the throughput of single-destination traffic will be independent of the amount of multi-destination traffic, and vice versa.

Let M denote the number of slots per frame, including one-to-one and broadcast slots, and a_{ij} be the number of one-to-one slots within a frame in which i may transmit to j. Let $d_{ij}^{(k)}, k = 1, \ldots, a_{ij}$, denote the distance, in slots, between the beginning of the $(k-1)$-th such slot (or the a_{ij}-th slot of the previous frame, if $k = 1$) and the beginning of the k-th slot. i will have a packet to transmit to j in the k-th slot in a frame, if at least one packet arrived in the previous $d_{ij}^{(k)}$ slots. The throughput of single-destination traffic originating at i and destined to j would then be

$$T_{ij} = \frac{1}{M} \sum_{k=1}^{a_{ij}} 1 - (1 - \sigma_i p_{ij})^{d_{ij}^{(k)}} \qquad (2)$$

Let b_i be the number of broadcast slots within a frame of which i is the owner (i.e., the only station which can transmit in these slots), and define $d_{i,b}^{(m)}, m = 1, \ldots, b_i$, in a way similar to $d_{ij}^{(k)}$. The average number of multi-destination packets *transmitted* by i per slot is

$$T_{i,b} = \frac{1}{M} \sum_{m=1}^{b_i} 1 - (1 - \rho_i)^{d_{i,b}^{(m)}} \qquad (3)$$

A multi-destination packet will be received by all stations in its multicast group; in terms of throughput, then, only the size of a group is important. If $\bar{\eta}$ is the average size of a multicast group, the aggregate network throughput can be computed by (2), (3), and

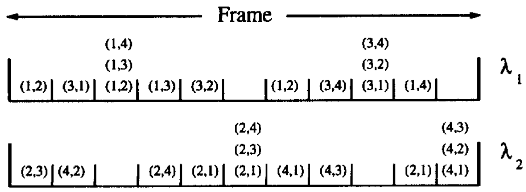
$$T_{total} = T_{multi} + T_{single} = \sum_{i=1}^{N} \bar{\eta} T_{i,b} + \sum_{i=1}^{N} \sum_{j=1}^{N} T_{ij} \quad (4)$$

3.1 Schedule Optimization

From previous experience [5] we expect the problem of obtaining schedules that optimize the throughput of mixed traffic to be a hard allocation problem. We now develop an optimization heuristic to construct schedules that not only perform very well, but also guarantee a specified level of throughput for each class of traffic. The heuristic is based on a decomposition of the problem into two manageable subproblems, namely, the problems of finding optimal schedules assuming each class of traffic is offered to the network in isolation. Optimization of schedules for single-destination traffic only ($\rho_i = 0 \; \forall \; i$) is presented in [5], while schedules optimal for multi-destination traffic only ($\sigma_i = 0 \; \forall \; i$) are derived in Appendix A. Then, the two schedules are *merged* so that the aggregate throughput is maximized.

Let S_1 and S_2 be two schedules of frame lengths M_1 and M_2, respectively. Without loss of generality, let $M_1 \geq M_2$ 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 is not an integer. In Figure 2 we show the result of merging the schedules of Figures 1(a) and 1(d).

Consider a one-to-one schedule, S, optimized for single-destination traffic only. We may now merge S with $l, l \geq 1$, frames of a broadcast schedule, S', optimized for multi-destination traffic, effectively providing slots in which multi-destination packets may be transmitted. As l increases, the merged schedule will tend to favor multi-destination packets (note that as $l \to \infty$, the resulting schedule will be indistinguishable from an S' schedule) in which case the throughput of single-destination traffic may suffer. Therefore,



(i,j) denotes that i has permission to transmit to j in this slot

Figure 2: Schedule merging

we must choose l so that both the total throughput, T_{total}, is high and the throughput of single-destination traffic, T_{single}, is at least α percent of T_{total}, where α is a specified parameter. Our approach is outlined in the following Schedule-Merging Heuristic (SMH). Note that the stopping rule at Step 4 guarantees that the final schedule will have broadcast slots.

SMH

1. Given σ_i, p_{ij}, and C, obtain an optimized one-to-one schedule, S_0, of frame length M [5].
2. Given ρ_i and C obtain a broadcast schedule, S'_0, of frame length M', optimized for multi-destination traffic only (Appendix A). Set $l \leftarrow 1$.
3. Merge one frame of S_0 with l frames of S'_0 to produce a schedule, S_l, of frame length $M + lM'$.
4. If $l = 1$ or $(T_{total}(S_l) > T_{total}(S_{l-1})$ *and* $T_{single}(S_l) \geq \alpha T_{total}(S_l))$ set $l \leftarrow l+1$ and repeat from Step 3. Otherwise, stop; the best schedule is S_{l-1}.

4 Adaptive Multicast Protocols

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, and as the multicast group changes, the permissions in each of these slots also change so that the overall throughput is maximized.

4.1 The Basic Idea

The operation of the protocols is based on the assumption that a source, i, will typically transmit L consecutive packets, $L > 1$, 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 in a number of fixed-size packets. L need not be constant; we assume that L is equally likely to be any integer in the range $L_{min} \leq L \leq L_{max}$. L_{min} and L_{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 i's multicast slots; similar observations can be made for other stations' multicast slots.

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 L 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 $| g | < N-1$, 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 L 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 (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. F will, in general, be a function of L (i.e., L_{min}, L_{max}), 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 one-to-one schedule, S, of frame length M, and 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 the one-to-one 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 is a one-to-one schedule and both i and k are given permission to transmit in the same slot t, we have $\lambda(i) \neq \lambda(k)$.

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

Global-knowledge Multicast Protocol (GMP). k maintains tables to map a multicast address in a packet originating at i into the stations-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). k has no knowledge about the members of multicast groups of packets originating at i [1]. 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 L packets to g as discussed above. 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 $L_{min} \gg 1$. In addition, this protocol does not require building and maintaining potentially large global tables at each station.

Probabilistic Multicast Protocol (PMP). k has no way to find 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 q. No overhead in terms of memory or control packets is incurred, but the selection of an appropriate value for q 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, q should represent the probability that j will *not* belong to g. If $\bar{\eta}$ is the average number of stations in a multicast group, we set $q = 1 - \frac{\bar{\eta}}{N-1}$.

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

4.3 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 transmitters of both i and k are synchronized at the beginning of each slot. But a packet transmitted by i will not be received 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 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 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, GMP will not 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. 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, and make GMP and CMP largely insensitive to propagation delays. Under PMP, on the other hand, a station does not need to wait until it receives the packet transmitted in the synchronization slots; thus, the throughput of PMP is not affected by propagation delays at all.

The algorithms used by the various transmitters and receivers for tuning 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.

The throughput performance of adaptive protocols was evaluated through simulation (with confidence of 99% in less than 1% variation from the mean).

5 Numerical Results

Schedules Optimized for Mixed Traffic. We have used SMH to obtain optimal schedules for networks with various single-destination traffic parameters [10, 5] (not shown here due to space constraints). For the multi-destination traffic we have assumed, without loss of generality, that $\rho_i = \rho \; \forall \; i$. Since we are interested in the performance of the schedules as the amount of multi-destination traffic increases, our conclusions are valid for the general case, i.e., different ρ_i. In all our experiments we have found that schedules produced with SMH achieve a very high total throughput, and, by adjusting the value of α, they overcome the fairness problems associated with schedules optimized for only one class of traffic [10].

As an example, Figure 7 plots the throughput of schedules with one-to-one and broadcast slots optimized with SMH ($\alpha = 45\%$, $N = 8, C = 4$, ring-type single-destination traffic matrix). Note that, when $\rho > 0$, packets transmitted within a broadcast slot are received by multiple stations, and the total throughput is higher than the number of channels, 4. The dip in throughput at $\rho = 0.15$ can be explained by considering the operation of SMH, which guarantees that $T_{single}(S_l) \geq \alpha T_{total}(S_l)$. Let $l^{(1)}, l^{(2)}$ be the values of l selected by SMH for $\rho^{(1)}$ and $\rho^{(2)}$, respectively, with $\rho^{(1)} < \rho^{(2)}$. In general, $l^{(1)} \leq l^{(2)}$, unless the above condition is violated for $\rho = \rho^{(2)}, l = l^{(1)}$, in which case SMH is forced to select $l^{(2)} < l^{(1)}$, effectively decreasing (increasing) the throughput of multi-destination (single-destination) traffic to satisfy the condition.

Adaptive Protocols. We consider a 20-station network with ring-type single-destination traffic matrix [10], and maximum size of a multicast group equal to 5. We set $L_{min} = 30$, $L_{max} = 50$, $\rho = 0.4$. We also assume that the propagation delay from i's transmitter to j's receiver, $\tau_{ij} = \tau \; \forall \; i, j$. Figure 8 plots the throughput of PMP as F, the number of free slots between two synchronization slots increases from 5 to 100 (recall that PMP is totally insensitive to propagation delays). From the figure it is clear that the value of F affects both the single- and multi-destination traffic throughput, as discussed in Section 4.1. The highest overall throughput is obtained for $F = 50$.

The throughput of GMP ($F = 50$) as a function of the

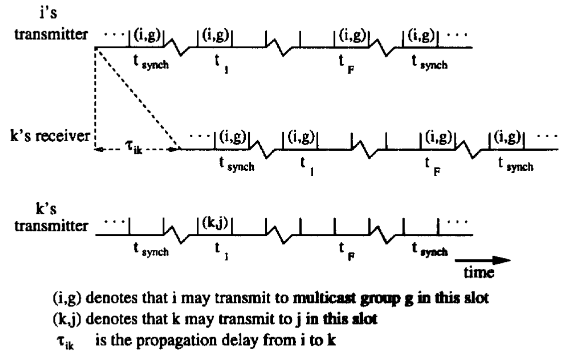


Figure 3: Effect of propagation delay (not in scale)

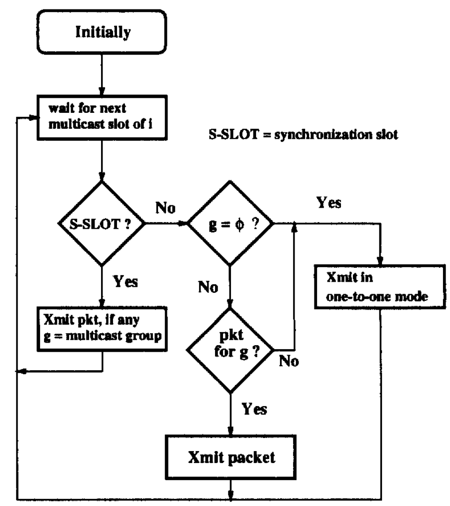


Figure 4: Algorithm executed by i's transmitter for transmission in i's multicast slots.

propagation delay, τ, is plotted in Figure 9, where τ is given in slots. For 424-bit packets (the size of an ATM cell), 1Gbps data rates, and speed of light in the fiber $v = 2 \times 10^8$ m/s, the range of propagation delays plotted represent delays over distances up to 17Km, reasonable for LANs/MANs. Figure 9 confirms our assertion that GMP is, to a large degree, insensitive to propagation delays; single-destination throughput only slightly decreases as propagation delays increase. The throughput of GMP is also higher, across the whole range of propagation delays, than the throughput of schedules with one-to-one and broadcast slots optimized with SMH (not shown). GMP increases the throughput by permitting single-destination packet transmissions in multicast slots. The real advantage of the adaptive protocols, however, lies on their adapt-

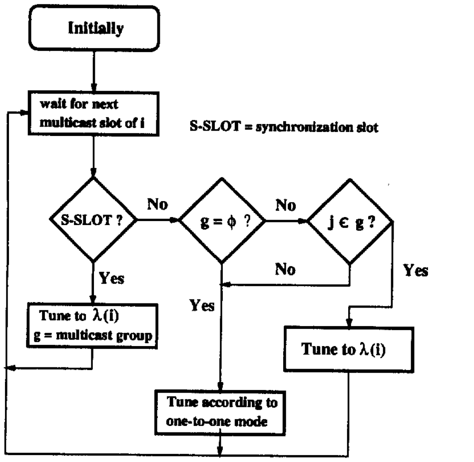


Figure 5: Algorithm executed at j's receiver for tuning in i's multicast slots.

ability. These protocols maintain very good performance even under changing multi-destination traffic characteristics (e.g., average multicast group size and ρ_i), while for schedules with one-to-one and broadcast slots it would be necessary to rerun SMH to construct a new optimized schedule.

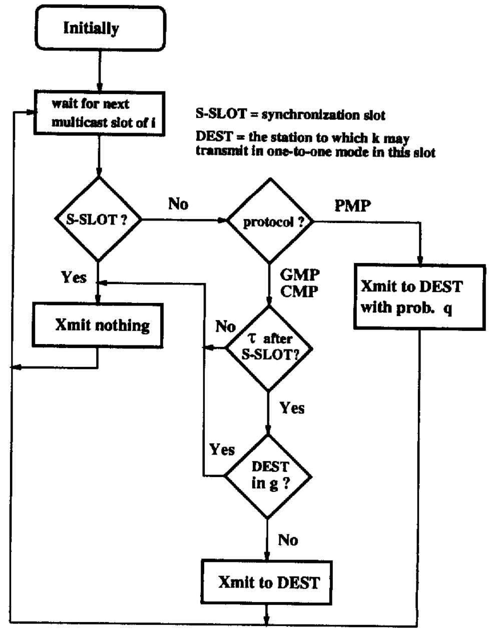


Figure 6: Algorithm executed by k's transmitter for transmission in i's multicast slots ($k \neq i$).

6 Concluding Remarks

We have addressed the problem of carrying both multi- and single-destination traffic over single-hop WDM networks. We presented an optimization heuristic, as well as adaptive multicast protocols to obtain schedules that maximize the aggregate throughput. Our results indicate that slot assignment adaptability is both desirable and feasible for multi-destination traffic.

References

[1] 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.

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

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

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

[5] G. N. Rouskas and M. H. Ammar. Analysis and optimization of transmission schedules for single-hop WDM networks. *INFOCOM '93*, pages 1342–1349. Also submitted to the IEEE/ACM Transactions on Networking.

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

[7] P. A. Humblet, R. Ramaswami, and K. N. Sivarajan. An efficient communication protocol for high-speed packet-switched multichannel networks. *IEEE JSAC*, 11(4):568–578, 1993.

[8] K. Bogineni, K. M. Sivalingam, and P. W. Dowd. Low-complexity multiple access protocols for wavelength-division multiplexed photonic networks. *IEEE JSAC*, 11(4):590–604, 1993.

[9] M. Borella and B. Mukherjee. Multicasting in a WDM local lightwave network. *ITC '93*.

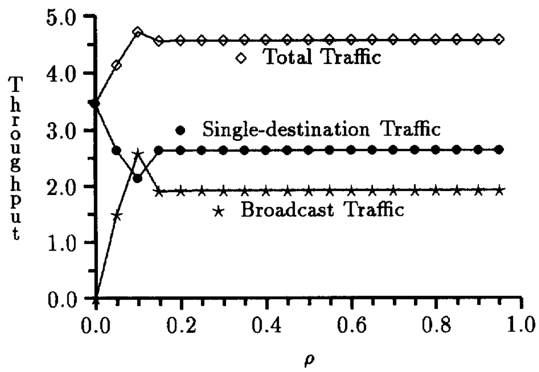


Figure 7: Throughput of schedules with one-to-one and broadcast slots optimized with SMH, $N = 8, C = 4, \alpha = 0.45$ (ring-type single-destination matrix).

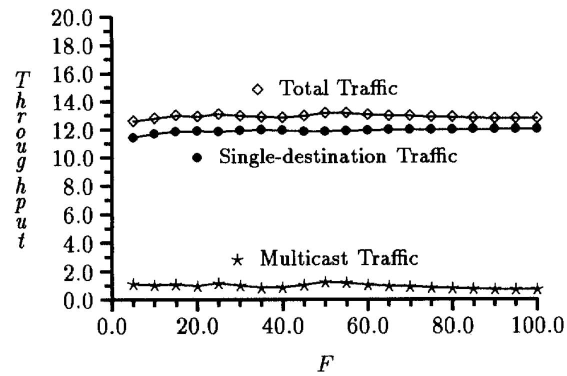


Figure 8: Throughput vs. F for PMP

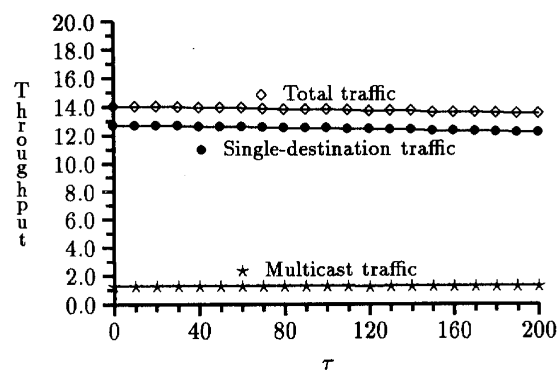


Figure 9: Throughput vs. propagation delay for GMP ($F = 50$)

[10] G. N. Rouskas and M. H. Ammar. Multi-destination communication over single-hop lightwave WDM networks. Tech. Rep. GIT-CC-93/52, Georgia Inst. Technology, Atlanta, GA, 1993.

[11] G. Semaan and P. Humblet. Timing and dispersion in WDM optical star networks. *INFOCOM '93*, pages 573–577.

[12] A. Itai and Z. Rosberg. A golden ratio control policy for a multiple-access channel. *IEEE Trans. on Autom. Control*, 29(8):712–718, Aug 1984.

[13] M. H. Ammar and G. N. Rouskas. On the performance of protocols for collecting responses over a multiple-access channel. *IEEE Trans. Commun. (to appear)*.

A Optimal Broadcast Schedules

We now address the problem of obtaining optimal broadcast schedules assuming the traffic offered to the network is of the multi-destination type only ($\sigma_i = 0 \; \forall \; i$), and is described by ρ_i.

Recall that only one station is allowed to transmit in a slot of a broadcast schedule. Given $\rho_i \; \forall i$, the problem of obtaining an optimal broadcast schedule is equivalent to the single-channel problem in [12]. There, it was shown that the percentage of time, x_i, that station i should be given permission to transmit is:

$$x_i = \frac{\ln(1-\rho_i)}{\sum_{j=1}^{N} \ln(1-\rho_j)} \tag{5}$$

x_i is independent of the frame length M. Given a Fibonacci number $M \geq N$ [12], we assign b_i broadcast slots to station i such that

$$\lfloor Mx_i \rfloor \leq b_i \leq \lceil Mx_i \rceil \; \forall \; i, \quad \text{and} \quad \sum_{i=1}^{N} b_i = M \tag{6}$$

We then use the golden-ratio policy, also developed in [12], to place the b_i slots, $i = 1, \ldots, N$, within the frame. As an example, for a network with $N = 4, C = 2$, and $\rho_1 = \rho_2 = \rho_3 = 0.19, \rho_4 = 0.1$, the optimal schedule is as in Figure 1(c). If $\rho_i = \rho \; \forall \; i$, the optimal broadcast schedule is a cyclic one, as in Figure 1(d).