



Scheduling Combined Unicast and Multicast Traffic in Broadcast WDM Networks

Zeydy Ortiz†

IBM, RTP, NC 27709, USA

E-mail: zortiz@us.ibm.com

George N. Rouskas, Harry G. Perros

North Carolina State University, Raleigh, NC 27695-7534, USA

E-mail: {rouskas,hp}@csc.ncsu.edu

Received July 21, 1999; Revised December 13, 1999

Abstract. This paper studies the performance of various strategies for scheduling a combined load of unicast and multicast traffic in a broadcast WDM network. The performance measure of interest is schedule length, which directly affects both aggregate network throughput and average packet delay. Three different scheduling strategies are presented, namely: separate scheduling of unicast and multicast traffic, treating multicast traffic as a number of unicast messages, and treating unicast traffic as multicasts of size one. A lower bound on the schedule length for each strategy is first obtained. Subsequently, the strategies are compared against each other using extensive simulation experiments in order to establish the regions of operation, in terms of a number of relevant system parameters, for which each strategy performs best. Our main conclusions are as follows. Multicast traffic can be treated as unicast traffic, by replicating all multicast packets, under very limited circumstances. On the other hand, treating unicast traffic as a special case of multicast traffic with a group of size 1, produces short schedules in most cases. Alternatively, scheduling and transmitting each traffic component separately is also a good choice.

Keywords: broadcast optical networks, wavelength division multiplexing (WDM), scheduling, multicast

1 Introduction

The ability to efficiently transmit a message addressed to multiple destinations has become increasingly important with the emergence of telecommunication services and computer applications requiring support for multipoint communication [1]. These applications include teleconferencing, distributed data processing, and video distribution. Traditionally, without network support for multicasting, a multi-destination message is replicated and transmitted individually to all its recipients. This method, however, consumes more bandwidth than necessary. Bandwidth consumption constitutes a problem since most of the applications requiring multipoint communication support typically consume a large amount of bandwidth. An alternative solution is to broadcast a multi-destination message to all nodes in the network. The problem is that nodes not addressed in the message will have to dedicate

resources to receive and process the message. In a multi-channel environment we could arrange for all nodes addressed in a multi-destination message to receive such communication over a previously determined channel. The coordination must be carefully made such that the use of the channels in the system is maximized.

In an optical broadcast network using wavelength division multiplexing (WDM) the available bandwidth is divided into channels. In order to communicate in this multi-channel environment, a transmitter and a receiver of the interested parties must be tuned to a common channel. Also, while the transmission is taking place, no other transmission may be made in that channel to avoid collisions. With current technology, we must take into consideration the time required for a transceiver to tune to a different channel since this time may be comparable to a packet's transmission time. These three factors

†This work was performed while the author was with the Department of Computer Science, North Carolina State University.

contribute to the need for algorithms to appropriately schedule multicast transmissions.

In a previous paper [5], we studied the problem of scheduling multicast traffic in broadcast-and-select networks employing WDM. We found that in this environment we must balance two conflicting objectives: low bandwidth consumption and high channel utilization. Bandwidth consumption can be high if a multi-destination message is always replicated and transmitted separately to each recipient. On the other hand, attempts to coordinate the addressed nodes so that a single transmission of a multicast packet be sufficient can lead to low channel utilization; in other words, it is possible that only a small number of channels carry transmissions at any given time, defeating the original purpose of a multi-channel environment. In [5] we introduced and studied the concept of a virtual receiver which can be used to provide a good balance between the two objectives.

In this paper, we focus on the problem of scheduling both unicast and multicast traffic. With new services and uses for technology, a mixed traffic scenario is the one more likely to be encountered in practice. Thus, the issue at hand is how to schedule traffic in order to efficiently utilize resources. In our case, efficiency is measured in terms of the length of the schedule produced: the shorter the schedule length, the higher the overall network throughput and the lower the average delay experienced by a message. The problem of scheduling unicast and multicast traffic has been studied by Rouskas and Ammar in [7] and Borella and Mukherjee in [3]. However, the work in [7] does not take into consideration the latency associated with tuning to different channels, while the scheduling policies presented in this paper are based on an algorithm designed to mask the tuning latency. In [3] the average number of channels utilized in the network is only one; our approach, on the other hand, can fully utilize the resources available in the network.

The paper is organized as follows. In the next section, we present the network and the traffic model used in the study. Also, we summarize some important earlier results that will be used in this paper. In Section 3, we present three different strategies for handling combined unicast and multicast traffic. The lower bound on the schedule length for each of the strategies is derived as well. In Section 4, we compare these three strategies through extensive numerical experiments in order to determine which one yields the shortest schedule. Our conclusions are discussed in Section 5.

2 System Model and Review of Previous Relevant Results

We consider an optical broadcast WDM network with a set $\mathcal{N} = \{1, \dots, N\}$ of end nodes and a set $\mathcal{C} = \{\lambda_1, \dots, \lambda_C\}$ of wavelengths, where $C \leq N$, as shown in Fig. 1. Each node is equipped with one fixed transmitter and one tunable receiver. The tunable receivers can tune to, and listen on any of the C wavelengths. The fixed transmitter at station i is assigned a home channel $\lambda(i) \in \mathcal{C}$. Let X_c , $c = 1, \dots, C$, denote the set of nodes with λ_c as their home channel: $X_c = \{i : \lambda(i) = \lambda_c\}$. The network is packet-switched, with fixed-size packets. Time is slotted, with a slot time equal to the packet transmission time, and all the nodes are synchronized at slot boundaries. We assume that the traffic offered to the network is of two types: unicast and multicast. We let $g \subseteq \mathcal{N} = \{1, 2, \dots, N\}$ represent the destination set of a multicast packet and $|g|$ denote its cardinality. Also, we let G represent the number of currently active¹ multicast groups.

We let integer $\Delta \geq 1$ represent the normalized tuning latency, expressed in units of packet transmission time. Parameter Δ is the number of slots a tunable receiver takes to tune from one wavelength to another. We observe that at very high data rates (i.e., 1 Gb/s and beyond), receiver tuning latency is significant when compared to packet transmission time. Therefore, unless techniques that can effectively overlap the tuning latency are employed, any solution to the problem of transmitting traffic in a WDM broadcast-and-select environment will be highly inefficient.

In this paper, we assume that there is a $C \times N$ unicast traffic demand matrix $\mathbf{A} = [a_{cj}]$, where a_{cj} is a measure of the *long-term* amount of unicast traffic

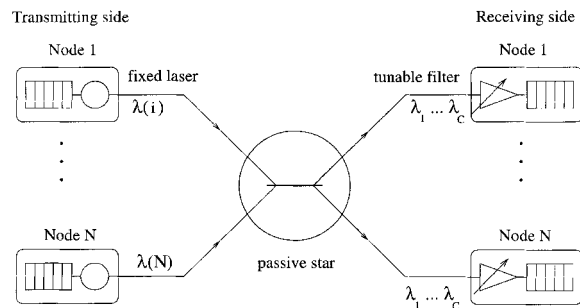


Fig. 1. A broadcast optical network with N nodes and C channels.

destined to receiver j and carried by channel λ_c . There is also a $C \times G$ multicast traffic demand matrix $\mathbf{M} = [m_{cg}]$, with m_{cg} representing the long-term amount of multicast traffic originating at nodes whose home channel is λ_c and destined to multicast group g . We assume that traffic matrices \mathbf{M} and \mathbf{A} are known to all nodes. Information about the traffic demands $\{a_{cj}\}$ and $\{m_{cg}\}$ may be collected using a distributed reservation protocol such as HiPeR- ℓ [9]. Due to time-varying conditions in the network, these traffic matrices will change over time. However, these changes will take place in time scales much larger than the time needed to transmit a single schedule of packets, as defined in the next section. Thus, despite small short-term fluctuations, the traffic carried by the network will be characterized for a sufficiently large amount of time by these traffic matrices, which will be assumed to be fixed during this time. Once the changes in the network traffic patterns become significant, these changes may be communicated to all nodes through a reservation protocol. Then, the algorithms described in the next section (to obtain a new virtual receiver set and schedule) will be run again for the new traffic matrix. These algorithms can be run in the background (i.e., while normal data transmission takes place) so that they will not affect the operation of the network.

2.1 Transmission Schedules

The problem of constructing schedules for transmitting unicast traffic in this network environment has been addressed by Azizoglu, Barry, and Mokhtar [2], Borella and Mukherjee [4], Rouskas and Sivaraman [8], and Pieris and Sasaki [6]. In the paper by Rouskas and Sivaraman [8], the authors address a fairly general version of the problem, as they consider arbitrary traffic demands and arbitrary transceiver tuning latencies. The algorithms presented in [8] yield optimal schedules when the traffic demands satisfy certain optimality conditions. A number of heuristics were also presented for the general case, and they were shown to produce schedules of length very close to (and in many cases equal to) the lower bound. In this paper, we will make extensive use of the algorithms in [8]. For presentation purposes, we introduce the following operation:

$$\mathcal{C} \leftarrow \text{Sched}(\mathbf{A}, \Delta). \quad (1)$$

The $\text{Sched}(\cdot)$ operation takes as arguments a unicast traffic demand matrix \mathbf{A} and the transceiver tuning

latency Δ , and it applies the Bandwidth Limited Scheduling Heuristic (*BLSH*) algorithm (for a bandwidth-limited network) or the Tuning Limited Scheduling Heuristic (*TLSH*) algorithm (for a tuning-limited network) presented in [8] to obtain a schedule \mathcal{C} for clearing matrix \mathbf{A} . The number of nodes and channels of the network are implicitly defined in the dimensions of matrix \mathbf{A} .

In this paper, we will also make use of the results presented by Ortiz, Rouskas and Perros [5], where the problem of scheduling multicast traffic in broadcast optical networks was considered. Specifically, a multicast schedule can be obtained by first partitioning the receivers into a set of *virtual* receivers and then using the scheduling algorithms in [8] which were developed for unicast traffic. A virtual receiver $V \subseteq \mathcal{N}$ is defined as a set of *physical* receivers that behave identically in terms of tuning. Specifically, if virtual receiver V must tune, say, from channel λ_c to channel $\lambda_{c'}$ starting at time t , then all physical receivers in V are taken off-line for tuning to $\lambda_{c'}$ between t and $t + \Delta$. Similarly, if virtual receiver V must remain tuned to channel $\lambda_{c'}$ for a certain number of slots (packet transmissions), then all physical receivers in V remain tuned to $\lambda_{c'}$ during those slots. Thus, from the point of view of coordinating the tuning of receivers to the various channels, all physical receivers in V can be logically thought of as a single receiver.

A k -virtual receiver set $\mathcal{V}^{(k)}$, $1 \leq k \leq N$, is defined as a partition of the set \mathcal{N} of receivers into k virtual receivers, $\mathcal{V}^{(k)} = \{V_1^{(k)}, V_2^{(k)}, \dots, V_k^{(k)}\}$. Given a k -virtual receiver set $\mathcal{V}^{(k)}$ and a multicast traffic matrix \mathbf{M} , transmission of multicast packets proceeds as follows. When a virtual receiver $V_l^{(k)}$ is on channel λ_c , each transmitter in X_c (i.e., each transmitter tuned to wavelength λ_c) will transmit all its multicast packets to groups g such that $g \cap V_l^{(k)} \neq \emptyset$ (i.e., at least one member of g has a receiver in $V_l^{(k)}$). All receivers in $V_l^{(k)}$ will have to filter out packets addressed to multicast groups of which they are not a member, but they are guaranteed to receive the packets for all groups of which they are members.

Given matrix \mathbf{M} , our previous work focused on how to select a virtual receiver set so as to achieve a good balance between two conflicting objectives: channel utilization and bandwidth consumption (for more details, see [5]). For presentation purposes, we introduce another operation, $VR(\cdot)$, which takes as arguments a multicast traffic \mathbf{M} and the tuning latency

Δ , and which applies the *Greedy-JOIN* heuristic in [5] to construct a near-optimal virtual receiver set $\mathcal{V}^{(k^*)}$ for matrix \mathbf{M} :

$$\mathcal{V}^{(k^*)} \leftarrow VR(\mathbf{M}, \Delta). \quad (2)$$

Once the k^* -virtual receiver set $\mathcal{V}^{(k^*)}$ has been determined, we construct a $C \times k^*$ matrix $\mathbf{B} = [b_{cl}]$ where

$$b_{cl} = \sum_{g: g \cap V_l^{(k^*)} \neq \emptyset} m_{cg}. \quad (3)$$

An element b_{cl} of this new matrix represents the amount of multicast traffic originating at channel λ_c and destined to virtual receiver $V_l^{(k^*)}$. Thus, by specifying the k^* -virtual receiver set $\mathcal{V}^{(k^*)}$, we have effectively transformed our original network with multicast traffic matrix \mathbf{M} , to an equivalent network with *unicast* traffic matrix \mathbf{B} . This new network has the same number of transmitters and channels and the same tuning latency as the original one. However, it only has k^* receivers, corresponding to the k^* virtual receivers in $\mathcal{V}^{(k^*)}$. We can now employ the algorithms in [8] to construct schedules for clearing matrix \mathbf{B} in this new network. In summary, the construction of a schedule for the transmission of multicast traffic matrix \mathbf{M} , involves three steps: applying the operation $VR(\mathbf{M}, \Delta)$, determining matrix \mathbf{B} from the resulting virtual receiver set $\mathcal{V}^{(k^*)}$, and finally applying the $Sched(\mathbf{B}, \Delta)$ operation. We will use $MSched(\mathbf{M}, \Delta)$ to denote this sequence of operations resulting in a schedule \mathcal{C} for \mathbf{M} . We have:

$$\mathcal{C} \leftarrow MSched(\mathbf{M}, \Delta). \quad (4)$$

We note that the *Greedy-JOIN* heuristic was one of a suite of heuristics developed in [5] to find good virtual receiver sets. The complexity of the heuristics ranged from $O(CN)$ to $O(GN^3)$ (the complexity of *Greedy-JOIN* is $O(N^3)$). Experimental results presented in [5] showed that the *Greedy-JOIN* heuristic performed best (in terms of the quality of the virtual receiver set constructed) compared to the other heuristics, over a wide range of values for the various network parameters. Since the algorithm for selecting a virtual receiver set does not need to be performed very often, and can be performed in the background, we have decided to use the *Greedy-JOIN* heuristic in this work.

2.2 Lower Bounds on the Schedule Length

We now present lower bounds on the number of slots required to clear traffic matrices \mathbf{A} and \mathbf{M} , each considered in isolation. These bounds will be used in the next section to compare various strategies for handling the combined unicast and multicast traffic. The bounds in this section have been derived in [8] and in [5]; they are included here only for the sake of completeness.

2.2.1 Unicast Traffic

Let us only consider unicast traffic. The length of any schedule for unicast traffic matrix \mathbf{A} cannot be smaller than the number of slots required to satisfy all transmissions on any given channel. In view of this, we can obtain the following *unicast channel* bound [8]:

$$\hat{H}_{ch} = \max_{c=1, \dots, C} \left\{ \sum_{j=1}^N a_{cj} \right\}. \quad (5)$$

We can obtain a different bound by adopting a receiver's point of view. Let T_j , $1 \leq T_j \leq C$, represent the number of channels to which receiver j must tune (these are the transmit channels of nodes that have packets for j , i.e., those channels λ_c such that $a_{cj} > 0$). Each receiver j needs at least a number of slots equal to the number of packets it has to receive, plus the number of slots required to tune to each of the T_j wavelengths. Therefore, we can obtain the following *unicast receiver* bound:

$$\hat{H}_r = \max_{j=1, \dots, N} \left\{ \sum_{c=1}^C a_{cj} + T_j \Delta \right\}. \quad (6)$$

We can now obtain the overall lower bound \hat{H} for clearing matrix \mathbf{A} :

$$\hat{H} = \max\{\hat{H}_{ch}, \hat{H}_r\}. \quad (7)$$

Considering only the unicast traffic, a network is called *tuning limited* when $\hat{H} = \hat{H}_r > \hat{H}_{ch}$ and *bandwidth limited* otherwise. Tuning limited networks are greatly affected by the tuning latency. The schedule length of bandwidth limited networks, however, depends only on the traffic requirements of the channels.

2.2.2 Multicast Traffic

Let us now assume that only multicast traffic is transmitted through the optical network. Let

$\mathcal{V}^{(k)} = \{V_1^{(k)}, \dots, V_k^{(k)}\}$ be a k -virtual receiver set. Since the virtual receiver concept effectively transforms the multicast traffic matrix \mathbf{M} to unicast traffic matrix \mathbf{B} , two bounds similar to the ones presented in the previous subsection can be obtained for clearing matrix \mathbf{M} . The main difference is that we must now consider virtual, rather than physical, receivers.

As before, the schedule length must be at least equal to the number of slots required to carry all traffic from the transmitters of any given channel to the virtual receivers. This yields the following *multicast channel* bound:

$$\begin{aligned} \hat{F}_{ch}(\mathcal{V}^{(k)}) &= \max_{c=1, \dots, C} \left\{ \sum_{l=1}^k b_{cl} \right\} \\ &= \max_{c=1, \dots, C} \left\{ \sum_{l=1}^k \sum_{g: g \cap V_l^{(k)} \neq \phi} m_{cg} \right\}. \end{aligned} \quad (8)$$

Let $T'_l, 1 \leq T'_l \leq C$, be the number of channels to which virtual receiver $V_l^{(k)}$ must tune (these are the transmit channels of nodes that have packets for multicast groups with at least one member in the virtual receiver $V_l^{(k)}$). As in the unicast traffic case, each virtual receiver $V_l^{(k)}$ needs a number of slots at least equal to the number of packets it has to receive, plus the number of slots required to tune to each of the T'_l wavelengths. Hence, we obtain the following *multicast receiver* bound:

$$\begin{aligned} \hat{F}_r(\mathcal{V}^{(k)}) &= \max_{l=1, \dots, k} \left\{ \sum_{c=1}^C b_{cl} + T'_l \Delta \right\} \\ &= \max_{l=1, \dots, k} \left\{ \left[\sum_{c=1}^C \sum_{g: g \cap V_l^{(k)} \neq \phi} m_{cg} \right] + T'_l \Delta \right\}. \end{aligned} \quad (9)$$

We have written the channel and receiver bounds as functions of the virtual receiver set since their values strongly depend on the actual receivers comprising each virtual receiver. The overall lower bound for multicast traffic matrix \mathbf{M} for the virtual receiver set is $\mathcal{V}^{(k)}$, is given by

$$\hat{F}(\mathcal{V}^{(k)}) = \max \left\{ \hat{F}_{ch}(\mathcal{V}^{(k)}), \hat{F}_r(\mathcal{V}^{(k)}) \right\}. \quad (10)$$

3 Transmission Strategies for Combined Unicast and Multicast Traffic

In this section we present three different strategies for scheduling and transmitting an offered load of combined unicast and multicast traffic. These are: separate scheduling, treating multicast as unicast traffic, and treating unicast as multicast traffic. These strategies were selected because they provide an intuitive solution to handling unicast and multicast traffic. We assume that the unicast and multicast traffic demands are given by matrices \mathbf{A} and \mathbf{M} respectively. Based on the results of the previous section, we derive lower bounds on the schedule length for each strategy. All three strategies use the algorithms in [8] to schedule packet transmissions. Since the lower bound accurately characterizes the scheduling efficiency of the algorithms in [8], the lower bounds will provide insight into the relative merits of each strategy. In the following, we will use $L_{ch}^{(i)}, L_r^{(i)}$, and $L^{(i)}$ to denote the channel, receiver, and overall lower bound, respectively, of strategy $i, i = 1, 2, 3$.

3.1 Strategy 1: Separate Scheduling

Our first strategy for transmitting the combined traffic offered to the network is to separately schedule the unicast and multicast matrices. That is, each traffic matrix is considered in isolation, and the appropriate scheduling techniques from [8,5] are applied to each traffic component. The two schedules are then used in sequence. This is a straightforward approach and involves the following operations: $Sched(\mathbf{A}, \Delta)$ and $MSched(\mathbf{M}, \Delta)$. Since at the end of the first schedule (say, the one for unicast traffic) the receivers may not be tuned to the channels required to start the next schedule (say, the one for multicast traffic), a sufficient number of slots for receiver retuning must be added between the two schedules.² Thus, we get a lower bound on the length of time it takes to clear matrices \mathbf{A} and \mathbf{M} under this approach as ($\mathcal{V}^{(k^*)}$ is the near-optimal virtual receiver set for matrix \mathbf{M}):

$$L^{(1)} = \hat{H} + \hat{F}(\mathcal{V}^{(k^*)}) + \Delta. \quad (11)$$

We note that the separate scheduling strategy achieves a lower bound which is equal to the sum of the best lower bounds for each traffic component in isolation (plus Δ slots to account for the retuning between the schedules). However, it could be possible

to obtain a schedule of smaller length by mixing together both traffic types. First, using a single schedule would eliminate the need for the Δ slots between the two schedules. We also note that a schedule may include some empty slots or gaps. These empty slots could be used to carry traffic of the other type, thus reducing the overall schedule length. However, the new schedule must still ensure that there are no channel or destination conflicts. The next two strategies combine both traffic types to produce a single schedule.

3.2 Strategy 2: Multicast Traffic Treated as Unicast Traffic

Our second approach is to treat multicast traffic as unicast traffic by replicating a packet for a multicast group g to all the members of g . In essence, using this strategy we create a new $C \times N$ unicast matrix $\mathbf{A}^{(2)} = [a_{cj}^{(2)}]$ where each element $a_{cj}^{(2)}$ represents the number of packets originating at channel λ_c and destined to physical receiver j :

$$a_{cj}^{(2)} = a_{cj} + \sum_{g:j \in g} m_{cg}. \quad (12)$$

Given matrix $\mathbf{A}^{(2)}$, we construct a transmission schedule by applying the operator for unicast traffic $Sched(\mathbf{A}^{(2)}, \Delta)$.

By considering the amount of traffic carried by each channel, we can obtain the channel bound for this strategy:

$$\begin{aligned} L_{ch}^{(2)} &= \max_{c=1, \dots, C} \left\{ \sum_{j=1}^N a_{cj}^{(2)} \right\} \\ &= \max_{c=1, \dots, C} \left\{ \sum_{j=1}^N a_{cj} + \sum_{j=1}^N \sum_{g:j \in g} m_{cg} \right\} \leq \max_{c=1, \dots, C} \left\{ \sum_{j=1}^N a_{cj} \right\} \\ &\quad + \max_{c=1, \dots, C} \left\{ \sum_{j=1}^N \sum_{g:j \in g} m_{cg} \right\} \\ &= \hat{H}_{ch} + \hat{F}_{ch}(\mathcal{V}^{(N)}). \end{aligned} \quad (13)$$

Similarly, we can obtain the receiver bound by accounting for the traffic plus tuning requirements of each (physical) receiver. Let us define $T_j^{(2)}$ as the number of channels to which (physical) receiver j must tune during the schedule according to the new unicast matrix $\mathbf{A}^{(2)}$. Recall that T_j (respectively T_j') is the number of channels to which receiver j must tune based on the requirements of traffic \mathbf{A} (respectively

\mathbf{M}). Obviously, we have that $T_j^{(2)} = T_j + T_j' - x_j$, where x_j is the number of channels in common between the tuning requirements of j for \mathbf{A} and \mathbf{M} . We have:

$$\begin{aligned} L_r^{(2)} &= \max_{j=1, \dots, N} \left\{ \sum_{c=1}^C a_{cj}^{(2)} + T_j^{(2)} \Delta \right\} \\ &= \max_{j=1, \dots, N} \left\{ \sum_{c=1}^C a_{cj} + \sum_{c=1}^C \sum_{g:j \in g} m_{cg} \right. \\ &\quad \left. + (T_j + T_j' - x_j) \Delta \right\} \\ &= \max_{j=1, \dots, N} \left\{ \left[\sum_{c=1}^C a_{cj} + T_j \Delta \right] \right. \\ &\quad \left. + \left[\sum_{c=1}^C \sum_{g:j \in g} m_{cg} + T_j' \Delta \right] - x_j \Delta \right\} \\ &\leq \hat{H}_r + \hat{F}_r(\mathcal{V}^{(N)}) - \min_{j=1, \dots, N} \{x_j \Delta\} \\ &\leq \hat{H}_r + \hat{F}_r(\mathcal{V}^{(N)}). \end{aligned} \quad (14)$$

In (13) and (14) above, $\hat{F}_{ch}(\mathcal{V}^{(N)})$ and $\hat{F}_r(\mathcal{V}^{(N)})$ are the channel and receiver bounds, respectively, on clearing matrix \mathbf{M} when the virtual receiver set is $\mathcal{V}^{(N)} = \{\{1\}, \dots, \{N\}\}$, i.e., when there are N virtual receivers, each consisting of exactly one physical receiver. These bounds can be obtained from (8) and (9), respectively, by letting $\mathcal{V}^{(k)} = \mathcal{V}^{(N)}$.

From (13) and (14) we may obtain a lower bound for Strategy 2:

$$\begin{aligned} L^{(2)} &\leq \max \left\{ \hat{H}_{ch} + \hat{F}_{ch}(\mathcal{V}^{(N)}), \hat{H}_r + \hat{F}_r(\mathcal{V}^{(N)}) \right\} \\ &\leq \hat{H} + \max \left\{ \hat{F}_{ch}(\mathcal{V}^{(N)}), \hat{F}_r(\mathcal{V}^{(N)}) \right\}. \end{aligned} \quad (15)$$

This strategy may result in a lower bound that is lower than $L^{(1)}$ when the unicast traffic is tuning limited (i.e., $\hat{H} = \hat{H}_r > \hat{H}_{ch}$). In this case, $L^{(1)} > L_r^{(2)} > L_{ch}^{(2)}$ must hold. We know that $L^{(1)} > L_r^{(2)}$ for tuning limited networks because $\hat{F}(\mathcal{V}^{(k^*)}) + \Delta > \hat{F}_r(\mathcal{V}^{(N)})$ is true for any $\Delta > 0$ (see [5]). To satisfy the condition that $L_r^{(2)} > L_{ch}^{(2)}$, we must have a network where $k = N$ or where $\hat{F}_{ch}(\mathcal{V}^{(N)}) - \hat{F}_r(\mathcal{V}^{(N)}) < \hat{H}_r - \hat{H}_{ch}$. In the latter case, the difference between the unicast bounds must

compensate for the difference between the multicast bounds to obtain a better lower bound in $L^{(2)}$. We may also obtain a better lower bound in a bandwidth limited network whenever $\hat{F}(\mathcal{V}^{(k^*)}) = \hat{F}(\mathcal{V}^{(N)})$. The Δ additional slots required for the first strategy will make $L^{(1)} > L^{(2)}$.

3.3 Strategy 3: Unicast Traffic Treated as Multicast Traffic

This strategy, in a sense, is the dual of the previous one. The unicast traffic is treated as multicast traffic by considering each individual destination node as a multicast group of size one. Given that initially there are G multicast groups (i.e., matrix \mathbf{M} has dimensions $C \times G$), this approach transforms the original network into a new network with multicast traffic only and with $G + N$ multicast groups (the groups of the original network plus N new groups $\{j\}$, one for each destination node j). The multicast traffic demands of the new network are given by a new $C \times (G + N)$ matrix $\mathbf{M}^{(3)} = [m_{cg}^{(3)}]$ whose elements are defined as follows:

$$m_{cg}^{(3)} = \begin{cases} m_{cg}, & g = 1, \dots, G \\ a_{cj}, & g = G + j, j = 1, \dots, N. \end{cases} \quad (16)$$

We can then use the new matrix $\mathbf{M}^{(3)}$ to obtain a schedule for the combined unicast and multicast traffic: $MSched(\mathbf{M}^{(3)}, \Delta)$. The near-optimal $k^{(3)}$ -virtual receiver set obtained from matrix $\mathbf{M}^{(3)}$, however, will in general be quite different from the k^* -virtual receiver set obtained from matrix \mathbf{M} . Consequently, we cannot express the channel and receiver bounds for this strategy as a function of the channel and receiver bounds for matrix \mathbf{M} as we did with Strategy 2 in (13) and (14).

We could still express the lower bound for this strategy in terms of $k^{(3)}$ using equations (8), (9), and (10). These equations, however, will not allow us to compare the lower bound for the different strategies. Therefore, we now obtain a lower bound for Strategy 3 as (see [5 Lemma 3.1]):

$$L^{(3)} = \max \left\{ \hat{F}_r^{(3)}(\mathcal{V}^{(N)}), \hat{F}_{ch}^{(3)}(\mathcal{V}^{(1)}) \right\}, \quad (17)$$

where $\hat{F}_r^{(3)}$ and $\hat{F}_{ch}^{(3)}$ represent the corresponding bounds on matrix $\mathbf{M}^{(3)}$. Expression (17) can be explained by noting that the length of any schedule cannot be smaller than the number of multicast

packets to be transmitted on any channel, which is given by $\hat{F}_{ch}^{(3)}(\mathcal{V}^{(1)})$. Similarly, the length of any schedule cannot be smaller than the sum of the number of packets destined to a particular receiver plus the receiver's tuning requirements, as expressed by $\hat{F}_r^{(3)}(\mathcal{V}^{(N)})$.

Expanding expression (17), we obtain:

$$\begin{aligned} \hat{F}_r^{(3)}(\mathcal{V}^{(N)}) &= \max_{j=1, \dots, N} \left\{ \left[\sum_{c=1}^C \sum_{g:j \in \mathcal{G}_g} m_{cg}^{(3)} \right] + T_j^{(3)} \Delta \right\} \\ &= \max_{j=1, \dots, N} \left\{ \left[\sum_{c=1}^C \sum_{g:j \in \mathcal{G}_g} m_{cg} \right] \right. \\ &\quad \left. + \left[\sum_{c=1}^C a_{cj} \right] + T_j^{(3)} \Delta \right\} \\ &\leq \hat{F}_r(\mathcal{V}^{(N)}) + \hat{H}_r. \end{aligned} \quad (18)$$

$$\begin{aligned} \hat{F}_{ch}^{(3)}(\mathcal{V}^{(1)}) &= \max_{c=1, \dots, C} \left\{ \sum_g m_{cg}^{(3)} \right\} \\ &= \max_{c=1, \dots, C} \left\{ \left[\sum_g m_{cg} \right] + \left[\sum_{j=1}^N a_{cj} \right] \right\} \\ &\leq \hat{F}_{ch}(\mathcal{V}^{(1)}) + \hat{H}_{ch}. \end{aligned} \quad (19)$$

We thus have:

$$L^{(3)} \leq \hat{H} + \max \left\{ \hat{F}_r(\mathcal{V}^{(N)}), \hat{F}_{ch}(\mathcal{V}^{(1)}) \right\}. \quad (20)$$

We note, however, that, unlike the other bounds presented in this section, the bound in (20) is not tight and may not be achievable.

4 Numerical Results

In this section we investigate numerically the behavior of the three strategies for a wide range of traffic loads and network parameters. Our objective is to determine which strategy produces the shortest schedule. Results are obtained by varying the following parameters: the number of nodes N in the optical network, the number of channels C , the tuning latency Δ , the number of different multicast groups G , the average number of nodes \bar{g} per multicast group, and the amount of multicast traffic as a percentage of the total traffic, s .

Specifically, in our experiments the parameters were varied as follows: $N = 20, 30, 40, 50$, network nodes, $G = 10, 20, 30$, multicast groups, $C = 5, 10, 15$, channels, and $\Delta = 1, 4, 16$, slots. The average group size \bar{g} was varied so that it accounted for 10%, 25% and 50% of the total number of network nodes N . For each multicast group, the number of members x in the group was selected randomly from the uniform distribution $[1, 2\bar{g} - 1]$. Some network nodes may not belong to any of the multicast groups.

The multicast traffic matrix was constructed as follows. Let p_{cg} be the probability that channel λ_c will have traffic for multicast group g . Then, with probability p_{cg} , m_{cg} was set equal to a randomly selected value from the uniform distribution $[1, 20]$, and with probability $1 - p_{cg}$ it was set equal to zero. The probability p_{cg} was calculated as follows (in our experiments we always assume that $G \geq C$):

$$p_{cg} = \begin{cases} \frac{C + c - \lfloor \frac{g}{G/C} \rfloor + 1}{C}, & c < \lfloor \frac{g}{G/C} \rfloor \\ \frac{c - \lfloor \frac{g}{G/C} \rfloor + 1}{C}, & \text{otherwise.} \end{cases} \quad (21)$$

Parameter s represents the percentage of total traffic due to multicast. It can be obtained as the ratio of the total multicast traffic (as seen by the receivers) to the total traffic in the network:

$$s = \frac{C\bar{g}G\bar{m}}{C\bar{g}G\bar{m} + CN\bar{a}} 100\%, \quad (22)$$

where \bar{m} and \bar{a} denote the average of the entries in the multicast and the unicast matrices, respectively. The percentage s of multicast traffic was varied from 10% to 90%. From the value assigned to $N, C, G, \bar{m}, \bar{g}$, and s , we can use the above equation to calculate \bar{a} . Let q_{cj} be the probability that channel λ_c has traffic for receiver j . The probability q_{cj} was calculated as follows:

$$q_{cj} = \begin{cases} \frac{C + c - \lfloor \frac{j}{N/C} \rfloor + 1}{C}, & c < \lfloor \frac{j}{N/C} \rfloor \\ \frac{c - \lfloor \frac{j}{N/C} \rfloor + 1}{C}, & \text{otherwise.} \end{cases} \quad (23)$$

Then, with probability q_{cj} the corresponding entry of the unicast traffic matrix a_{cj} was set to a randomly selected number from the uniform distribution $[1, 2\bar{a} - 1]$, and with probability $1 - q_{cj}$ it was set equal to zero.

We also investigated the effects of hot-spots by introducing hot nodes which receive a larger amount of traffic compared to non-hot nodes. Specifically, we let the first five nodes of the network be the hot nodes.

The average number of unicast packets received by these nodes was set to $1.5\bar{a}$. Therefore, with probability q_{cj} , given by (23), the entry $a_{cj}, j = 1, \dots, 5$ was set to a randomly selected number from the uniform distribution $[1, 2(1.5\bar{a}) - 1]$, and with probability $1 - q_{cj}$ it was set to zero. The remaining $N - 5$ nodes receive an average number of unicast packets equal to $(\frac{N-7.5}{N-5})\bar{a}$. For these nodes with probability q_{cj} , the entry $a_{cj}, j = 6, \dots, N$ was set to a randomly selected value from the uniform distribution $[1, 2(\frac{N-7.5}{N-5})\bar{a} - 1]$, and with probability $1 - q_{cj}$ it was set equal to zero. Note that the overall average number of unicast packets remains equal to \bar{a} , as in the non-hot-spot case.

For each combination of values for the input parameters N, G, C, Δ, \bar{g} and s , we construct the individual multicast groups, the multicast traffic matrix, \mathbf{M} , and the unicast matrix, \mathbf{A} , using random numbers as described above. When constructing a case, we require that all nodes receive transmissions (unicast and/or multicast packets) and that all channels have packets to transmit. Based on all these values, we then obtain $S^{(i)}$, the schedule length of the i -th strategy, $i = 1, 2, 3$. Let S^* be the schedule length of the strategy with the lower schedule length, i.e., $S^* = \min\{S^{(1)}, S^{(2)}, S^{(3)}\}$. Then, for each strategy i , we compute the quantity $D^{(i)} = \frac{S^{(i)} - S^*}{S^*} 100\%$, which indicates how far is the schedule length of the i -th strategy from the best one. Due to the randomness in the construction of the multicast groups and of matrices \mathbf{M} and \mathbf{A} , each experiment associated with a specific set of values for N, G, C, Δ, \bar{g} , and s is replicated 100 times. For each strategy i , we finally compute $\bar{D}^{(i)} = \frac{1}{100} \sum_{j=1}^{100} D_j^{(i)}$, where $D_j^{(i)}$ is obtained from the j -th replication. All figures in this section plot $\bar{D}^{(i)}, i = 1, 2, 3$, against the percentage s of multicast traffic offered to the network.

The results presented below are organized as follows. In Section 4.1 we give some representative detailed comparisons of the three strategies obtained by varying only one of the parameters s, \bar{g}, Δ, C, G and N at a time. In Section 4.2 we summarize our findings, and we discuss under which conditions each strategy gives the shortest schedule.

4.1 Detailed Comparisons

The results are presented in Figs. 2–12. In each figure, we plot $D^{(i)}, i = 1, 2, 3$, against s indicated as ‘‘% Multicast Traffic’’. In other words, the figures present

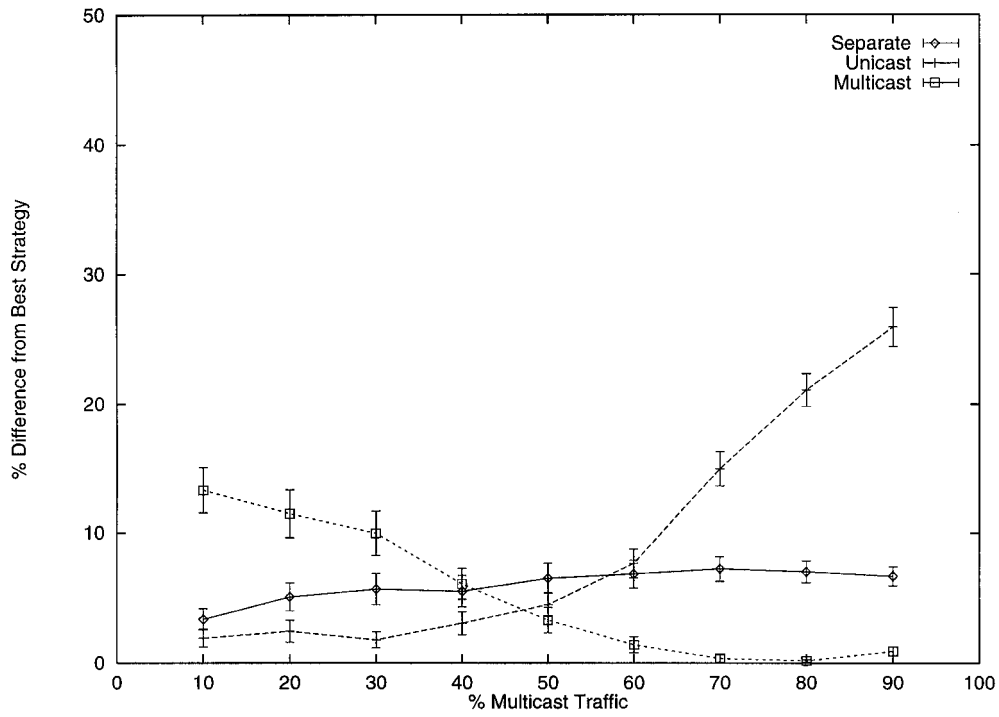


Fig. 2. Comparison of strategies for $N = 20, G = 30, C = 10, \Delta = 4$, and $\bar{g} = 0.25N$.

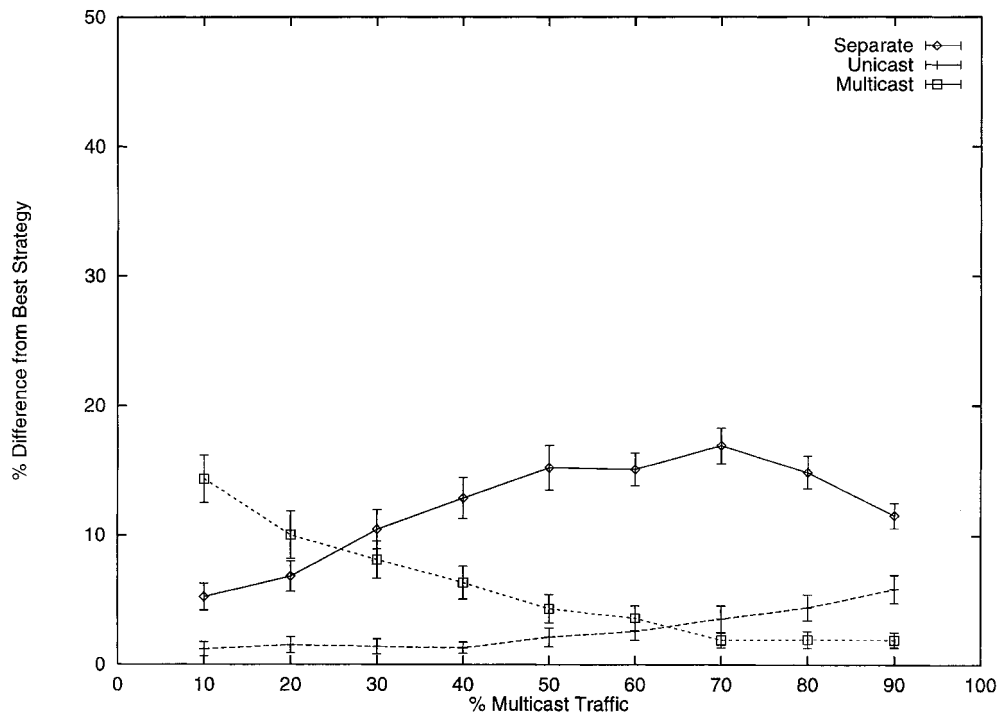


Fig. 3. Comparison of strategies for $N = 20, G = 30, C = 10, \Delta = 4, \bar{g} = 0.10N$.

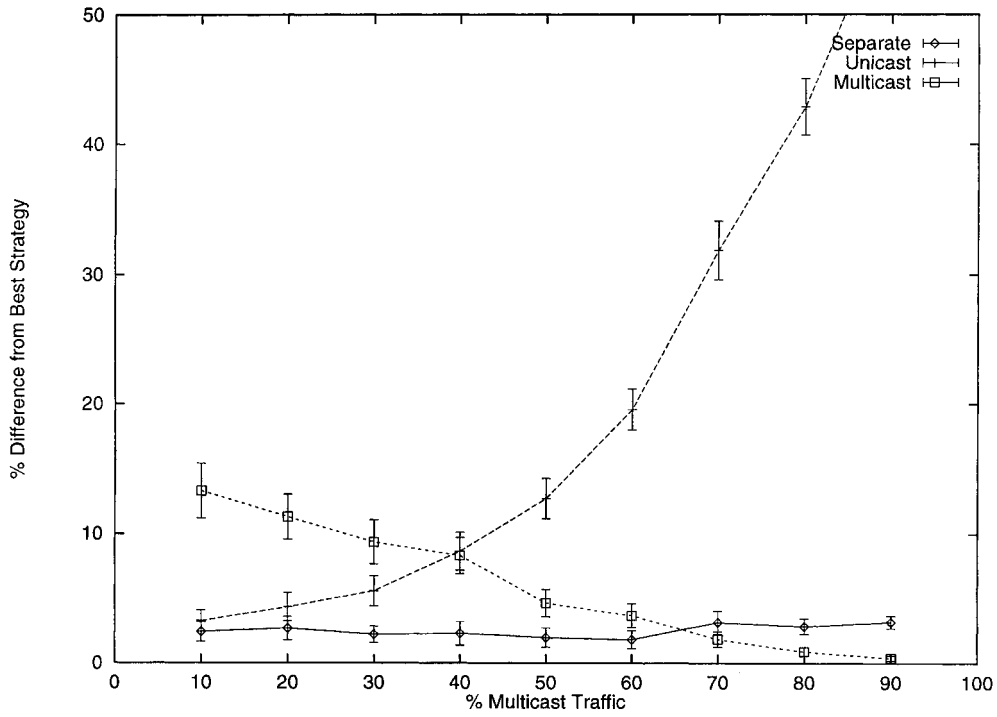


Fig. 4. Comparison of strategies for $N = 20, G = 30, C = 10, \Delta = 4, \bar{g} = 0.50N$.

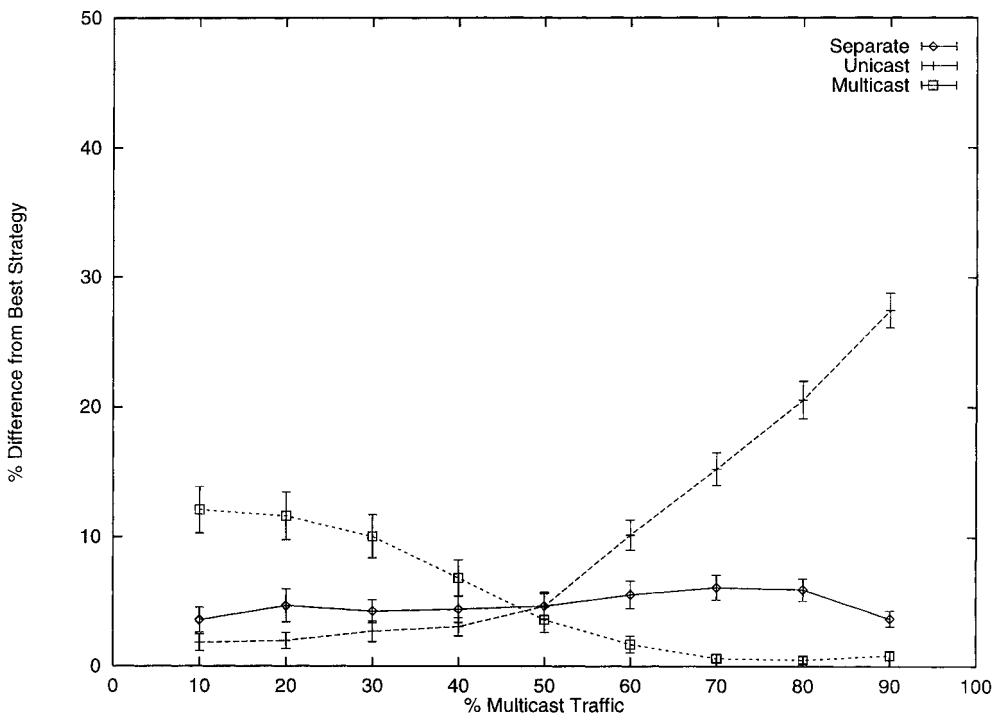


Fig. 5. Comparison of strategies for $N = 20, G = 30, C = 10, \Delta = 1, \bar{g} = 0.25N$.

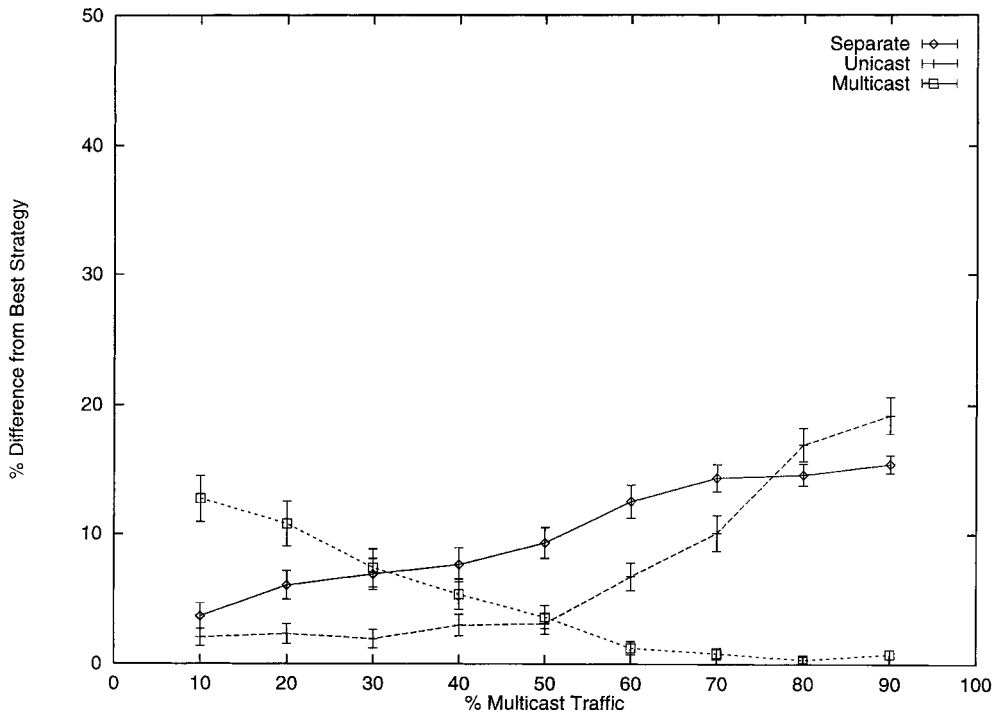


Fig. 6. Comparison of strategies for $N = 20, G = 30, C = 10, \Delta = 16, \bar{g} = 0.25N$.

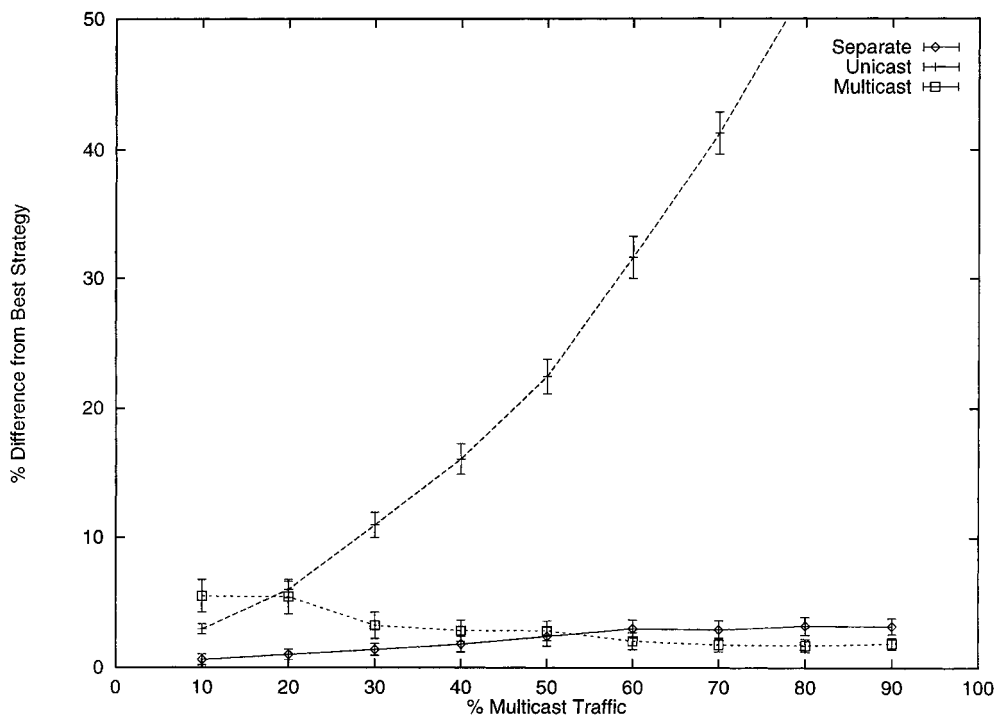


Fig. 7. Comparison of strategies for $N = 20, G = 30, C = 5, \Delta = 4, \bar{g} = 0.25N$.

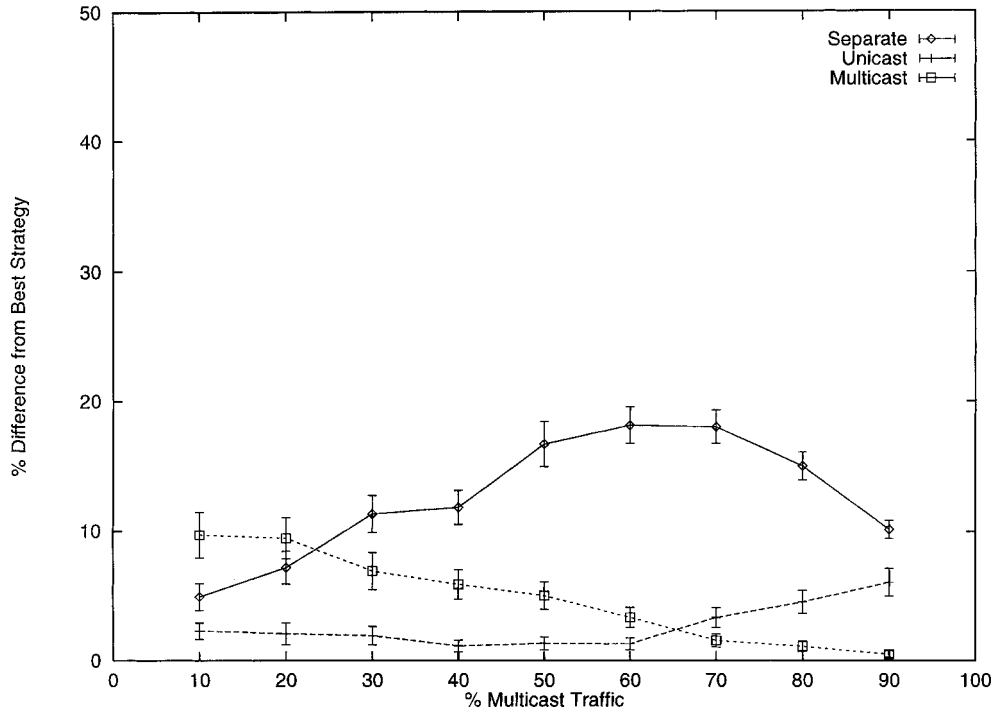


Fig. 8. Comparison of strategies for $N = 20, G = 30, C = 15, \Delta = 4, \bar{g} = 0.25N$.

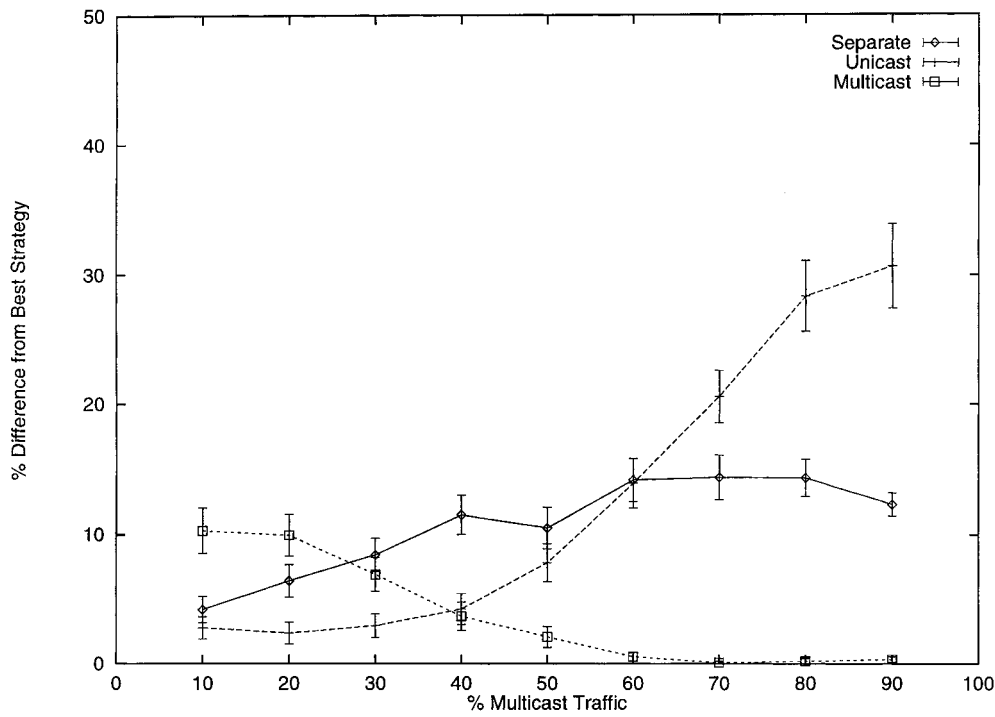


Fig. 9. Comparison of strategies for $N = 20, G = 10, C = 10, \Delta = 4, \bar{g} = 0.25N$.

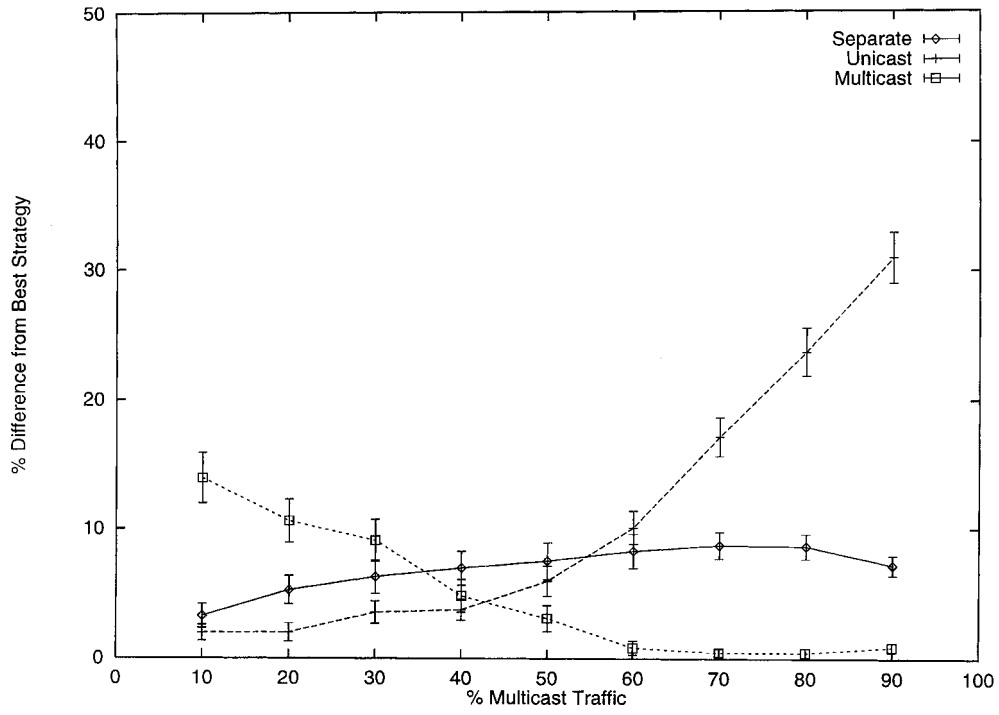


Fig. 10. Comparison of strategies for $N = 20, G = 20, C = 10, \Delta = 4, \bar{g} = 0.25N$.

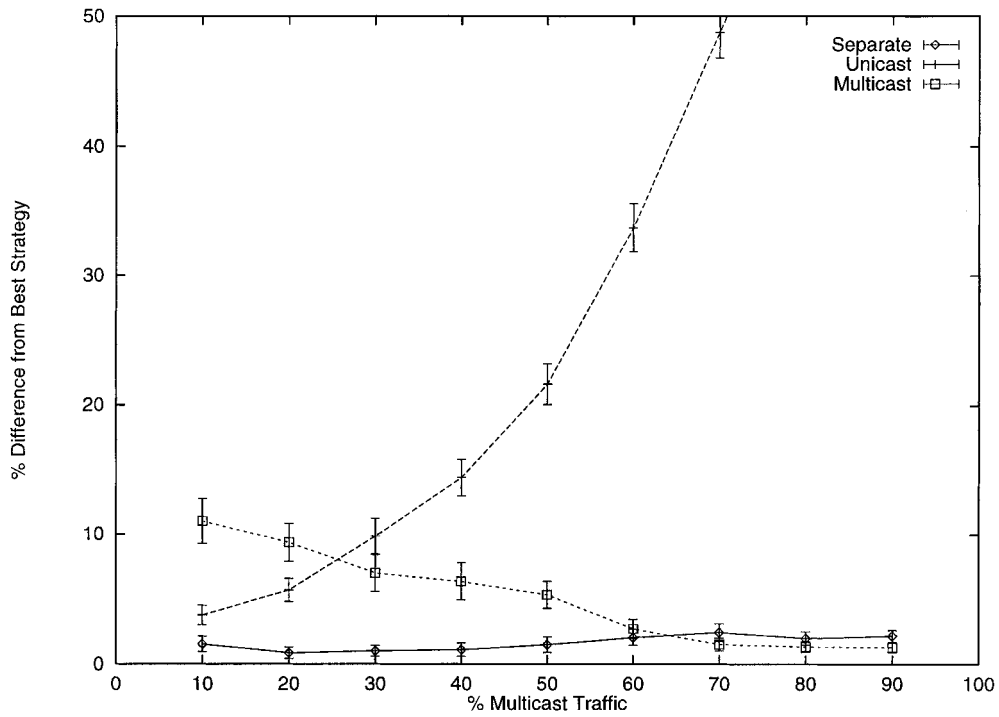


Fig. 11. Comparison of strategies for $N = 40, G = 30, C = 10, \Delta = 4, \bar{g} = 0.25N$.

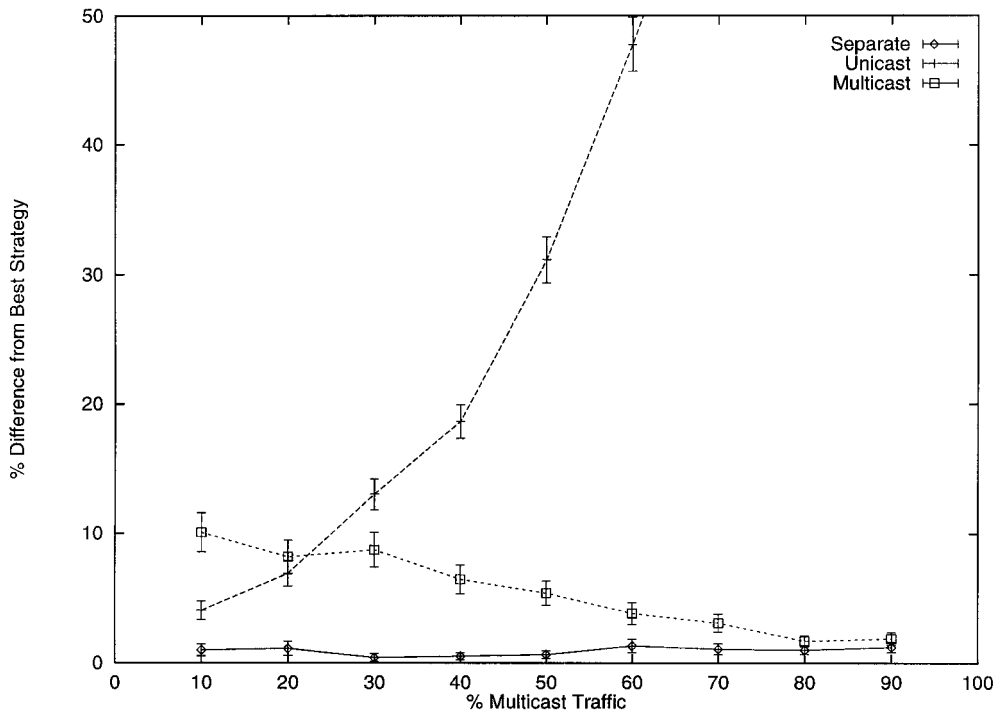


Fig. 12. Comparison of strategies for $N = 50, G = 30, C = 10, \Delta = 4, \bar{g} = 0.25N$.

the performance of the various strategies relative to each other. 95% confidence intervals are also shown in each figure. For presentation purposes, we use the following abbreviations for the names of the three strategies in the figures and tables: Strategy 1 is referred to as ‘‘Separate’’; Strategy 2, where multicast traffic is treated as unicast traffic, is referred to as ‘‘Unicast’’; and Strategy 3, where unicast traffic is treated as multicast traffic is referred to as ‘‘Multicast’’.

Fig. 2 gives the results for the case where $N = 20, G = 30, C = 10, \Delta = 4$, and $\bar{g} = 0.25N$. We note that Strategy 2 is the best strategy for $s < 50\%$, but that Strategy 3 becomes the best one for $s \geq 50\%$. This figure represents our base case. Figs. 3 to 12 give results in which only one of the parameters has been changed while the remaining parameters are the same as those in Fig. 2. Specifically, Figs. 5 and 6 show the cases in which we vary \bar{g} . In Figs. 5 and 6 we varied Δ . The number of channels is varied in Figs. 7 and 8, while the number of multicast groups is changed in Figs. 9 and 10. The next two figures, namely 11 and 12, show results when the number of nodes is increased.

Below, we discuss the results presented in Figs. 2–12 for each strategy separately.

Separate Scheduling.

Even though the behavior of Strategy 1 (relative to the others) appears to be unaffected by the different parameters, we noticed changes related to the tuning latency, as expected. When Δ was increased, $\bar{D}^{(1)}$ had a tendency to increase. From the expression (11) for $L^{(1)}$, we note that Δ slots are added to the optimal bounds for unicast and multicast traffic, while the lower bounds for the other two strategies did not have this component. It is thus expected for $\bar{D}^{(1)}$ to be sensitive to this parameter. Increasing s or C did not change the behavior of $\bar{D}^{(1)}$, except for large values of Δ ($\Delta = 16$). In these cases, the increase observed can be attributed to the large Δ .

Multicast Traffic Treated as Unicast Traffic.

For this strategy, we note that as s increases, the difference from the best strategy, $\bar{D}^{(2)}$, increases (and in some cases it increases dramatically). Changes to s only affect the value of the unicast lower bound, \hat{H} ,

because \hat{H}_r and \hat{H}_{ch} depend on \bar{a} . Increasing s causes \bar{a} to decrease and, consequently, \hat{H} decreases. However, the multicast component of the traffic is relatively larger and more important in these cases. Recall that when $k^* \neq N$, $\hat{F}_{ch}(\mathcal{V}^{(N)}) > \hat{F}_r(\mathcal{V}^{(N)})$ and $\hat{F}_{ch}(\mathcal{V}^{(N)}) > \hat{F}(\mathcal{V}^{(k^*)})$. Therefore, the lower bound for this strategy is dominated by $\hat{F}_{ch}(\mathcal{V}^{(N)})$. Compared to the lower bounds of the other strategies, $L^{(2)}$, and consequently $S^{(2)}$, has a higher value.

The increase in $\bar{D}^{(2)}$ observed when \bar{g} increases can be explained by noting that in this strategy, a single multicast packet is replicated to every member of a multicast group and transmitted independently. Therefore, it is only natural to expect that the schedule length increases when there are more recipients. The same applies when N is increased.

We note that as Δ is increased, $\bar{D}^{(2)}$ remains the same in most cases or decreases slightly. The tuning latency affects all receiver bounds. However, $S^{(2)}$ is less affected because $\hat{F}_r(\mathcal{V}^{(N)}) < \hat{F}_r(\mathcal{V}^{(k^*)})$.

When C increases, $\bar{D}^{(2)}$ decreases. Again, $\hat{F}_{ch}(\mathcal{V}^{(N)}) > \hat{F}_r(\mathcal{V}^{(N)})$ and $\hat{F}_{ch}(\mathcal{V}^{(N)}) > \hat{F}(\mathcal{V}^{(k^*)})$ when $k^* \neq N$. Both receiver bounds, $\hat{F}_r(\mathcal{V}^{(N)})$ and $\hat{F}_r(\mathcal{V}^{(k^*)})$, increase when C is increased. But since $S^{(2)}$ is determined by the channel bound, $\hat{F}_{ch}(\mathcal{V}^{(N)})$, $S^{(2)}$ remains intact while $S^{(1)}$ and $S^{(3)}$ increase. Consequently, $\bar{D}^{(2)}$ decreases.

Unicast Traffic Treated as Multicast Traffic.

This strategy is not the best choice when we have a large amount of unicast traffic (compared to multicast traffic). For small values of s , it starts as the worst strategy, but it becomes the best one for larger values of s . Changing any of the other parameters did not affect the performance of this strategy significantly. This behavior indicates that we could use this strategy in every circumstance since, even for small amounts of multicast traffic (small s), its performance is not significantly worse than that of the best strategy.

Table 1 summarizes the results presented in Figs. 2–12. The table shows the effect that increasing a

parameter has on the length of the schedule obtained from each strategy.

Hotspots.

Finally, in Fig. 13 we show the behavior of the three strategies for the hotspot pattern described earlier. Except for the unicast traffic matrix \mathbf{A} , the remaining parameters are the same as those in Fig. 2. We note that the results obtained in Fig. 13 are not different from those in previous figures where all nodes were identical (no hotspots). This result was observed for a wide range of values for the various system parameters. We conclude that, although the existence of hotspots will certainly affect the schedule length, it does not affect the relative performance of the various strategies.

4.2 Summary

In Table 2, we present the percentage of time that each strategy produced a schedule of length within 5% of the best schedule, for various values of \bar{g} and s and for all values of the other parameters N , G , C , and Δ .³ Tables 3 and 4 present similar results for different values of N , G , and N , C , respectively. The strategy that produced the shortest schedules in each case corresponds to the one with the highest percentage shown. A strategy whose schedule length was within 5% of the best schedule length was also considered to be the best strategy. The 5% margin, though somewhat arbitrary, provides us with an insight into the performance of the strategies. When deciding which strategy to implement in an actual system, we may settle for one that produces the shortest schedules under most conditions while producing schedules within 5% of the best under other conditions. Below, we discuss under what conditions each of the three strategies is best.

Separate Scheduling.

Overall, separate scheduling is effective in producing short schedules. Compared to Strategy 3, this strategy is better when there is a larger amount of unicast traffic, when there are many multicast groups (G is

Table 1. Behavior of strategies under varying parameters (\uparrow : increase, \downarrow : decrease, $-$: no change).

Strategy	$s \uparrow$	$\bar{g} \uparrow$	$\Delta \uparrow$	$C \uparrow$	$G \uparrow$	$N \uparrow$
Separate	—	—	—	—	—	—
Unicast	\uparrow	\uparrow	—	\downarrow	—	\uparrow
Multicast	—	—	—	—	—	—

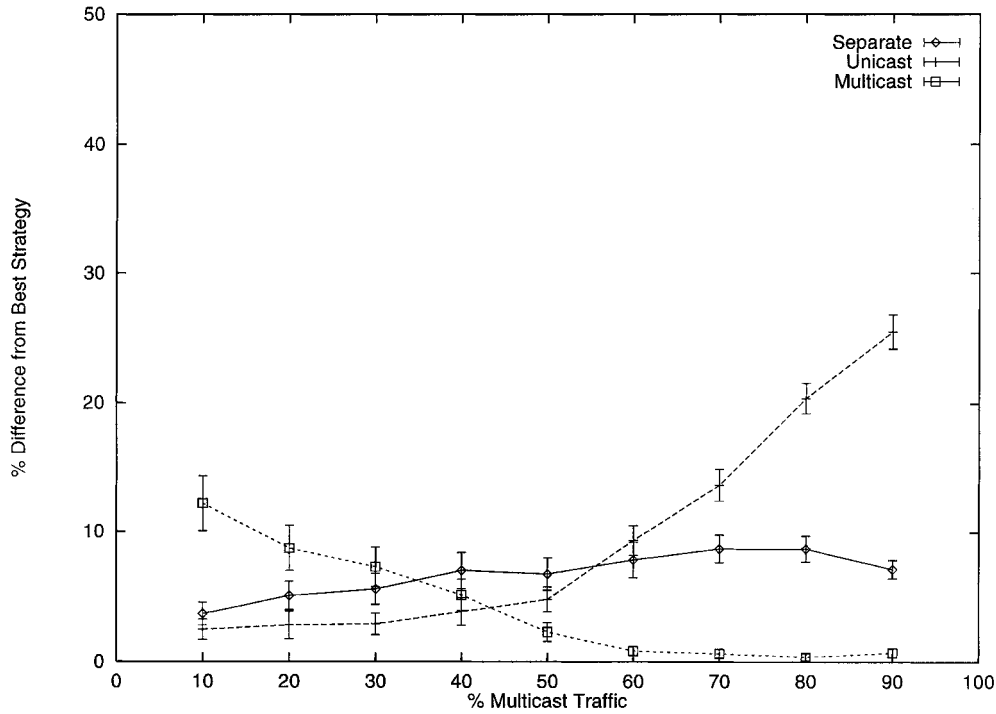


Fig. 13. Comparison of strategies with hotspots for unicast traffic ($N = 20, G = 30, C = 10, \Delta = 4, \bar{g} = 0.25N$).

large), and when the number of channels is small compared to the number of nodes in the network.

Multicast Traffic Treated as Unicast Traffic.

Strategy 2 is best when there is a small amount of multicast traffic in the network and the size of the multicast groups is small (see Table 2). This result is not surprising since replicating a multicast packet increases the requirements in the network and it can only be used efficiently in very limited situations. Also, this strategy is useful when the ratio of nodes to channels is small, i.e., N/C is close to 1 (see Table 3).

In this case, the network operates in the tuning limited region.

Unicast Traffic Treated as Multicast.

Strategy 3 produces schedules of short length in most situations. Even when the strategy does not produce the best schedule, the resulting schedule has a length no more than 20% larger than that of the best schedule (see Figs. 2–13). Strategy 3 gives good results when G is small, i.e., $G \leq N/2$, when C is large, i.e., $C \geq N/2$, and when the amount of unicast traffic is small, i.e., $s \geq 40\%$.

Table 2. Best strategy when \bar{g} and s are varied.

	$s = 10, 20, 30\%$			$s = 40, 50, 60\%$			$s = 70, 80, 90\%$		
	Strategy	%		Strategy	%		Strategy	%	
$\bar{g} = 10\%N$	Separate	64%		Separate	31%		Separate	23%	
	Unicast	82%		Unicast	36%		Unicast	22%	
	Multicast	54%		Multicast	97%		Multicast	100%	
$\bar{g} = 25\%N$	Separate	90%		Separate	76%		Separate	59%	
	Unicast	57%		Unicast	20%		Unicast	4%	
	Multicast	41%		Multicast	93%		Multicast	98%	
$\bar{g} = 50\%N$	Separate	98%		Separate	93%		Separate	78%	
	Unicast	35%		Unicast	6%		Unicast	0%	
	Multicast	31%		Multicast	61%		Multicast	83%	

Table 3. Best strategy when N and G are varied.

	$G = 10$		$G = 20$		$G = 30$	
$N = 20$	Separate	33%	Separate	49%	Separate	46%
	Unicast	51%	Unicast	54%	Unicast	66%
	Multicast	81%	Multicast	75%	Multicast	75%
$N = 30$	Separate	53%	Separate	72%	Separate	74%
	Unicast	29%	Unicast	32%	Unicast	33%
	Multicast	79%	Multicast	71%	Multicast	68%
$N = 40$	Separate	61%	Separate	80%	Separate	87%
	Unicast	20%	Unicast	18%	Unicast	21%
	Multicast	82%	Multicast	71%	Multicast	69%
$N = 50$	Separate	68%	Separate	87%	Separate	90%
	Unicast	21%	Unicast	14%	Unicast	15%
	Multicast	78%	Multicast	63%	Multicast	75%

5 Concluding Remarks

We studied the problem of scheduling unicast and multicast traffic for transmission in a broadcast-and-select WDM network. Our goal was to create schedules that balance bandwidth consumption and channel utilization in order to efficiently use the system resources.

We presented three different strategies for scheduling a combined load of unicast and multicast traffic. These strategies are: separate scheduling, treating multicast traffic as unicast traffic, and treating unicast traffic as multicast traffic. As expected, multicast traffic should be treated as unicast traffic under very limited circumstances. More specifically, this strategy is useful only when there is a small amount of multicast traffic in the network and/or the multicast groups are small. On the other hand, if we treat unicast

traffic as multicast traffic with a multicast group of size 1, the resulting schedule has a shorter length (when compared with the schedules produced by the other strategies). This is the case especially when we have a large number of channels in the system, i.e., $C \geq N/2$ or when the number of multicast groups is small ($G \leq N/2$). Scheduling and transmitting each traffic separately also produces schedules of short length. Finally, one must also take into account memory and processing time limitations when considering which of the best two strategies to use. In particular, Strategy 3 requires storage for the $C \times (G + N)$ multicast traffic matrix when forming the virtual receiver sets, while for Strategy 1 the scheduling algorithms in [8] must be run twice, once for unicast traffic and once for multicast traffic. Since running this algorithm twice may be expensive, we believe that, for immediate deployment, Strategy 3 would be the best choice.

Table 4. Best strategy when N and G are varied.

	$C = 5$		$C = 10$		$C = 15$	
$N = 20$	Separate	73%	Separate	40%	Separate	12%
	Unicast	22%	Unicast	64%	Unicast	84%
	Multicast	88%	Multicast	69%	Multicast	74%
$N = 30$	Separate	86%	Separate	67%	Separate	47%
	Unicast	8%	Unicast	30%	Unicast	57%
	Multicast	86%	Multicast	69%	Multicast	63%
$N = 40$	Separate	90%	Separate	76%	Separate	61%
	Unicast	4%	Unicast	20%	Unicast	35%
	Multicast	90%	Multicast	65%	Multicast	67%
$N = 50$	Separate	91%	Separate	81%	Separate	69%
	Unicast	3%	Unicast	16%	Unicast	25%
	Multicast	86%	Multicast	63%	Multicast	64%

Acknowledgments

This work was supported by the National Physical Science Consortium, the National Security Agency, and the NSF under grant NCR-9701113.

Notes

1. Typically, the number G of active groups is significantly smaller than the total number 2^N of possible groups.
2. A number Δ of slots is also required at the very beginning of transmission to ensure that receivers are tuned to the channels as required by the first schedule. But these Δ initial slots are needed for all four strategies and do not affect their relative performance. Hence, we will ignore these Δ initial slots in the expressions for the various bounds presented here.
3. Even though the relative amount of multicast traffic in the network, s , is influenced by the size of the multicast groups, \bar{g} , we separate these two quantities to show that they affect the results independently.

References

- [1] M. Ammar, G. Polyzos, S. Tripathi (eds.), Special issue on network support for multipoint communication, *IEEE Journal Selected Areas in Communications*, vol. 15, no. 3, (April 1997).
- [2] M. Azizoglu, R. A. Barry, A. Mokhtar, Impact of Tuning Delay on the Performance of Bandwidth-limited Optical Broadcast Networks with Uniform Traffic, *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 5, (June 1996), pp. 935–944.
- [3] M. Borella, B. Mukherjee, A Reservation-based Multicasting Protocol for WDM Local Lightwave Networks, *Proc. of ICC '95* (Seattle, Washington, June 1995), pp. 1277–1281.
- [4] M. S. Borella, B. Mukherjee, Efficient Scheduling of Nonuniform Packet Traffic in a WDM/TDM Local Lightwave Network with Arbitrary Transceiver Tuning Latencies, *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 5 (June 1996), pp. 923–934.
- [5] Z. Ortiz, G. N. Rouskas, H. G. Perros, Scheduling of multicast traffic in tunable-receiver {WDM} networks with non-negligible tuning latencies, *Proc. of SIGCOMM '97* (Cannes, France, Sept. 1997) pp. 301–310.
- [6] G. R. Pieris, G. H. Sasaki, Scheduling Transmissions in WDM Broadcast-and-Select Networks, *IEEE/ACM Transactions on Networking*, vol. 2, no. 2 (April 1994), pp. 105–110.
- [7] G. N. Rouskas, M. H. Ammar, Multi-destination Communication Over Tunable-receiver Single-hop WDM Networks, *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 3 (April 1997), pp. 501–511.
- [8] G. N. Rouskas, V. Sivaraman, Packet Scheduling in Broadcast {WDM} Networks with Arbitrary Transceiver Tuning Latencies, *IEEE/ACM Transactions on Networking*, vol. 5, no. 3 (June 1997), pp. 359–370.
- [9] V. Sivaraman, G. N. Rouskas, HiPeR- ℓ : A High Performance Reservation Protocol with ℓ ook-ahead for broadcast WDM networks, *Proc. of INFOCOM '97* (Kobe, Japan, March 1997), pp. 1272–1279.

Zeydy Ortiz received her doctoral degree in Computer Science in May 1998 from North Carolina State University. She worked with Drs. Harry Perros and George Rouskas on multipoint communication support in optical networks. She received a Master's of Computer Science degree in December of 1993 from Texas A&M University and a Bachelor of Science in Computer Engineering from the University of Puerto Rico—Myaguez Campus in June 1992. She currently works at IBM Corp. as a Network Performance Engineer.

George N. Rouskas received the Diploma in Electrical Engineering from the National Technical University of Athens (NTUA), Athens, Greece, in 1989, and the M.S. and Ph.D. degrees in Computer Science from the College of Computing, Georgia Institute of Technology, Atlanta, GA, in 1991 and 1994, respectively. He joined the Department of Computer Science, North Carolina State University in August 1994, and he has been an Associate Professor since July 1999. His research interests include high-speed and lightwave network architectures, multipoint-to-multipoint communication, and performance evaluation.



He is a recipient of a 1997 NSF Faculty Early Career Development (CAREER) Award, and a co-author of a paper that received the *Best Paper* Award at the 1998 SPIE conference on All-Optical Networking. He also received the 1995 *Outstanding New Teacher* Award from the Department of Computer Science, North Carolina State University, and the 1994 *Graduate Research Assistant* Award from the College of Computing, Georgia Tech. He is a co-guest editor for the *IEEE Journal on Selected Areas in Communications*, Special Issue on Protocols and Architectures for Next Generation Optical WDM Networks, and is on the editorial board of the *Optical Networks Magazine*. He is a member of the IEEE, the ACM and of the Technical Chamber of Greece.

Harry G. Perros received the B.Sc. degree in Mathematics in 1970 from Athens University, Greece, the M.Sc. degree in Operational Research with Computing from Leeds University, UK, in 1971, and the Ph.D. degree in Operations Research from Trinity College, Dublin, Ireland, in 1975. From 1976 to 1982 he was an Assistant Professor in the Department of Quantitative Methods, University of Illinois at Chicago. In 1979 he spent a sabbatical term at INRIA, Rocquencourt, France. In 1982 he joined the Department of Computer Science, North Carolina State University, as an Associate Professor, and since 1988 he is a Professor. During the academic year 1988–89 he was on a sabbatical



leave of absence first at BNR, Research Triangle Park, North Carolina, and subsequently at the University of Paris 6, France. Also, during the academic year 1995–96 he was on a sabbatical leave of absence at Nortel, Research Triangle Park, North Carolina.

He has published extensively in the area of performance modeling of computer and communication systems, and has

organized several national and international conferences. He also published a monograph entitled “Queueing networks with blocking: exact and approximate solutions”, Oxford Press. He is the chairman of the IFIP W.G. 6.3 on the Performance of Communication Systems. His current research interests are in the areas of optical networks and their performance, and software performance evaluation.