Scheduling Combined Unicast and Multicast Traffic in WDM Networks*

Zeydy Ortiz, George N. Rouskas, Harry G. Perros
Department of Computer Science
North Carolina State University
Raleigh, NC 27695-7534, USA
{zortizl,rouskas,hp}@csc.ncsu.edu

Abstract

We study 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. 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 under very limited circumstances. On the other hand, treating unicast traffic as multicast traffic produces short schedules in most cases. Alternatively, scheduling and transmitting each traffic component separately is also a good choice.

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, otherwise a collision will occur. 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 contribute to the need for algorithms to appropriately schedule multicast transmissions.

In a previous paper [4], 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 [4] 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, since 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 the network 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 in [5, 2]. However, [5] does not take into consideration the latency associated with tuning to different channels, while in [2] the average number of channels utilized in the network is only one. On the other hand, the scheduling policies presented in this paper are based on an algorithm designed to mask the tuning latency, and they can fully utilize the resources available in the network.

In Section 2 we present the network and traffic models used in this study, and we summarize earlier results. In Section 3, we present three strategies for handling combined unicast and multicast traffic. In Section 4, we compare these three strategies through extensive numerical experiments to determine which one yields the shortest schedule, and we conclude the paper in Section 5.

2 SYSTEM MODEL

We consider an optical broadcast WDM network with a set $\mathcal{N} = \{1, \cdots, N\}$ of nodes and a set $\mathcal{C} = \{\lambda_1, \cdots, \lambda_C\}$ of wavelengths, where $C \leq N$. 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, $c = 1, \ldots, C$, 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 $g \subseteq \mathcal{N} = \{1, 2, \cdots, N\}$ represent the destination set of a multicast packet and $| g |$ denote its cardinality. Also, we let G represent the number of currently active multicast groups.

In this paper, we assume that there is a $C \times N$ unicast traffic demand matrix $\mathbf{A} = [a_{cj}]$, where a_{cj} is the total amount of unicast traffic 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 number of multicast packets 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 [7].

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 note that, at very high data rates, receiver tuning latency becomes 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 broadcast WDM environment will be inefficient.

The problem of constructing schedules for transmitting unicast traffic in this network environment has been addressed in [6] where arbitrary traffic demands and arbitrary transceiver tuning latencies were considered. The algorithms presented in [6] 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 [6]. For presentation purposes, we introduce the following operation: $\mathcal{S} \leftarrow Sched(\mathbf{A}, \Delta)$. The $Sched(\cdot)$ operation takes as arguments a unicast traffic demand matrix $\mathbf{A}$ and the transceiver tuning latency Δ, and it applies the algorithms in [6] to obtain a schedule $\mathcal{S}$ for clearing matrix $\mathbf{A}$.

The authors have previously considered the problem of scheduling multicast traffic in broadcast optical networks [4]. There, a virtual receiver $V \subseteq \mathcal{N}$ was

defined as a set of *physical* receivers that behave identically in terms of tuning. 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, $\mathcal{V}^{(k)} = \{V_1^{(k)}, V_2^{(k)}, \cdots, V_k^{(k)}\}$. 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 $g \cap V_l^{(k)} \neq \phi$ (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 members 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 [4]). For presentation purposes, we introduce another operation, $VR(\cdot)$, which takes as arguments a multicast traffic matrix $\mathbf{M}$ and the tuning latency Δ, and which applies the heuristics in [4] to construct a near-optimal virtual receiver set $\mathcal{V}^{(k^\star)}$ for $\mathbf{M}$: $\mathcal{V}^{(k^\star)} \leftarrow VR(\mathbf{M}, \Delta)$.

Once the $k^\star$-virtual receiver set $\mathcal{V}^{(k^\star)}$ has been determined, we construct a $C \times k^\star$ matrix $\mathbf{B} = [b_{cl}]$ where $b_{cl} = \sum_{g: g \cap V_l^{(k^\star)} \neq \phi} m_{cg}$. An element b_{cl} of this new matrix represents the amount of multicast traffic originating at channel λ_c and destined to virtual receiver $V_l^{(k^\star)}$. Thus, by specifying the $k^\star$-virtual receiver set $\mathcal{V}^{(k^\star)}$, 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 $k^\star$ receivers, corresponding to the $k^\star$ virtual receivers in $\mathcal{V}^{(k^\star)}$. We can now employ the algorithms in [6] 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 $\mathcal{V}^{(k^\star)}$, 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 $\mathcal{S}$ for $\mathbf{M}$: $\mathcal{S} \leftarrow MSched(\mathbf{M}, \Delta)$.

3   TRANSMISSION STRATEGIES

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 **A** and **M** respectively. Lower bounds on the schedule length for each strategy have been obtained (see [3]) but are omitted due to space limitations.

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 [6, 4] 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. 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).

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: $a_{cj}^{(2)} = a_{cj} + \sum_{g:j \in g} m_{cg}$. Given $\mathbf{A}^{(2)}$, we construct a transmission schedule by applying the operator for unicast traffic, $Sched(\mathbf{A}^{(2)}, \Delta)$.

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 **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:

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

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 $k^\star$-virtual receiver set obtained from matrix $\mathbf{M}$.

4   NUMERICAL RESULTS

In this section we investigate 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:

$$p_{cg} \quad = \quad \begin{cases} \frac{C + c - \lfloor \frac{g}{\lceil G/C \rceil} \rfloor + 1}{C}, & c < \lfloor \frac{g}{\lceil G/C \rceil} \rfloor \\ \frac{c - \lfloor \frac{g}{\lceil G/C \rceil} \rfloor + 1}{C}, & \text{otherwise} \end{cases} \tag{2}$$

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:

$$s \quad = \quad \frac{C \bar{g} G \bar{m}}{C \bar{g} G \bar{m} + C N \bar{a}} \; 100\% \tag{3}$$

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:

$$q_{cj} = \begin{cases} \frac{C + c - \lfloor \frac{j}{\lfloor N/C \rfloor} \rfloor + 1}{C}, & c < \lfloor \frac{j}{\lfloor N/C \rfloor} \rfloor \\ \frac{c - \lfloor \frac{j}{\lfloor N/C \rfloor} \rfloor + 1}{C}, & \text{otherwise} \end{cases} \tag{4}$$

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 (4), the entry $a_{cj}, j = 1, \cdots, 5$, 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 $a_{cj}, j = 6, \cdots, N$, 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 $N, G, C, \Delta, \bar{g}$, 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 $S^\star$ be the schedule length of the strategy with the lower schedule length, i.e., $S^\star = \min\{S^{(1)}, S^{(2)}, S^{(3)}\}$. Then, for each strategy i, we compute the quantity $D^{(i)} = \frac{S^{(i)} - S^\star}{S^\star} 100\%$, which indicates how far is the schedule length of the ith 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 $N, G, C, \Delta, \bar{g}$, 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.

4.1   Detailed Comparisons

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

ures present the performance of the various strategies relative to each other. 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 is referred to as "Unicast" and Strategy 3 is referred to as "Multicast".

Figure 1 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. Figures 2 to 12 give results in which only one of the parameters has been changed while the remaining parameters are the same as those in Figure 1. Specifically, Figures 2 and 3 show the cases in which we vary $\bar{g}$. In Figures 4 and 5 we varied Δ. The number of channels is varied in Figures 6 and 7, while the number of multicast groups is changed in Figures 8 and 9. The next three figures, namely 10, 11, and 12, show results when the number of nodes is increased. Below, we discuss the results presented in Figures 1–12 for each strategy separately. More detailed explanations that take into account the various lower bounds can be found in [3].

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. Recall that Δ slots are added to the optimal bounds for unicast and multicast traffic, while the lower bounds for the other two strategies do 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 or $\bar{g}$ increases, the difference from the best strategy, $\bar{D}^{(2)}$, increases (and in some cases it increases dramatically). This behavior can be explained by noting that in this strategy, all multicast packets are 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 is more multicast traffic or more recipients per packet. Similar observations apply when N is increased.

We also note that changes in parameter Δ do not significantly affect $\bar{D}^{(2)}$. Finally, as C increases, $\bar{D}^{(2)}$ tends to decrease. For the traffic matrices considered here, the network is in the bandwidth-limited region [6] when this strategy is used. Therefore, an increase in the number of available channels results in a shorter schedule length.

Unicast Traffic Treated as Multicast Traffic. This strategy is not the best choice when we have a large amount of unicast traffic (compared to
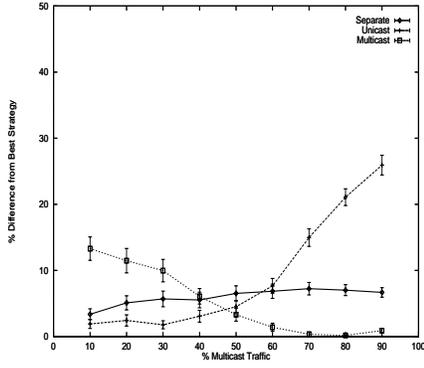
Figure 1 Comparison for $N = 20$, $G = 30$, $C = 10$, $\Delta = 4$, $\bar{g} = 0.25N$
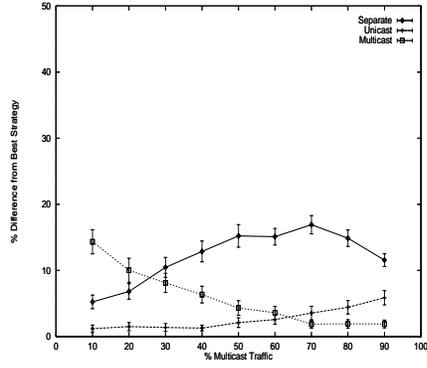
Figure 2 Comparison for $N = 20$, $G = 30$, $C = 10$, $\Delta = 4$, $\bar{g} = 0.10N$
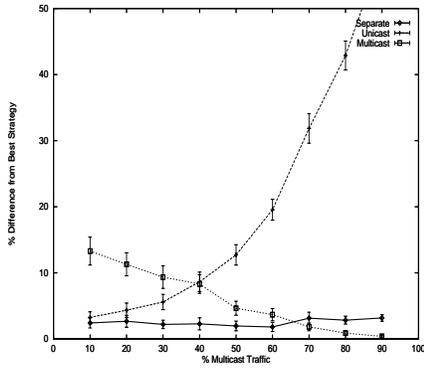
Figure 3 Comparison for $N = 20$, $G = 30$, $C = 10$, $\Delta = 4$, $\bar{g} = 0.50N$
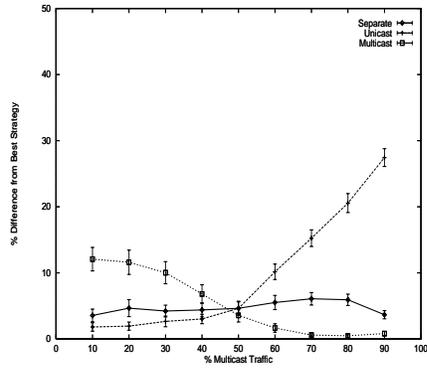
Figure 4 Comparison for $N = 20$, $G = 30$, $C = 10$, $\Delta = 1$, $\bar{g} = 0.25N$
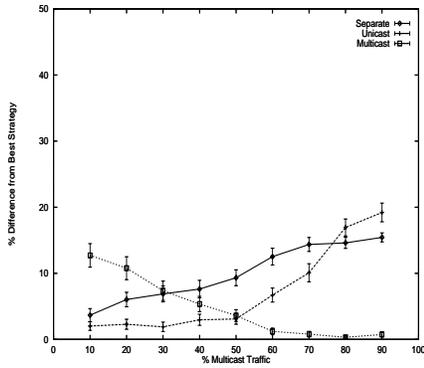
Figure 5 Comparison for $N = 20$, $G = 30$, $C = 10$, $\Delta = 16$, $\bar{g} = 0.25N$
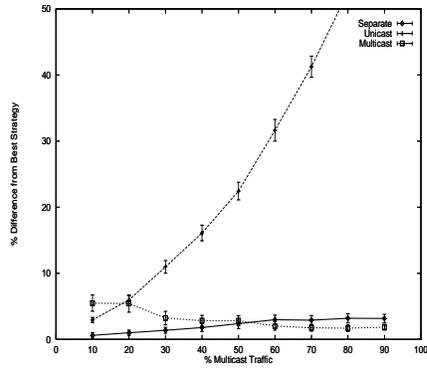
Figure 6 Comparison for $N = 20$, $G = 30$, $C = 5$, $\Delta = 4$, $\bar{g} = 0.25N$
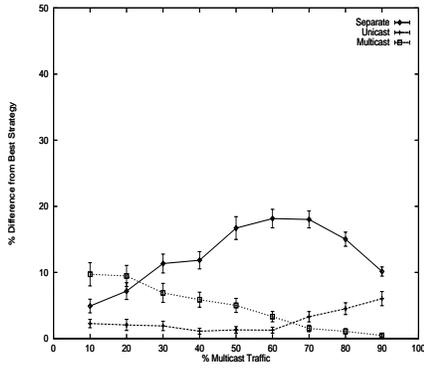
Figure 7 Comparison for $N = 20$, $G = 30$, $C = 15$, $\Delta = 4$, $\bar{g} = 0.25N$
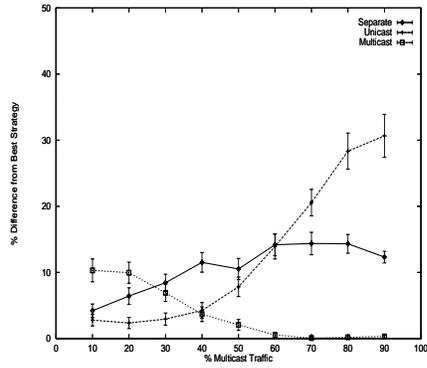
Figure 8 Comparison for $N = 20$, $G = 10$, $C = 10$, $\Delta = 4$, $\bar{g} = 0.25N$
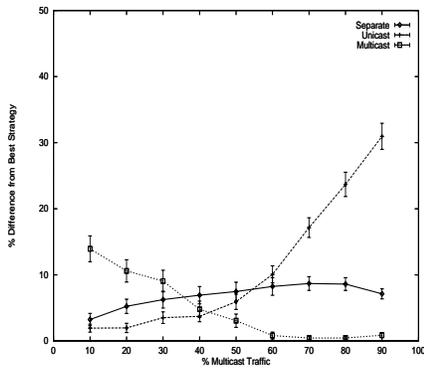
Figure 9 Comparison for $N = 20$, $G = 20$, $C = 10$, $\Delta = 4$, $\bar{g} = 0.25N$
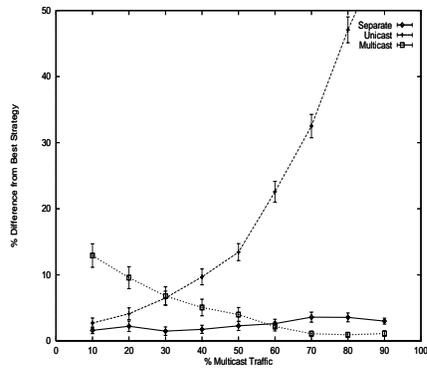
Figure 10 Comparison for $N = 30$, $G = 30$, $C = 10$, $\Delta = 4$, $\bar{g} = 0.25N$
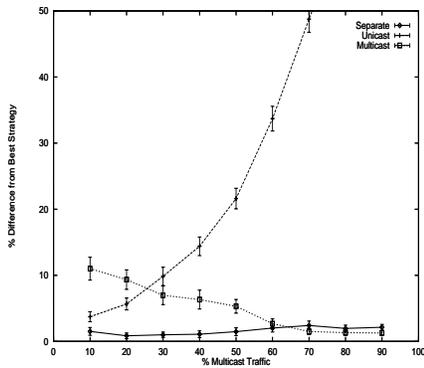
Figure 11 Comparison for $N = 40$, $G = 30$, $C = 10$, $\Delta = 4$, $\bar{g} = 0.25N$
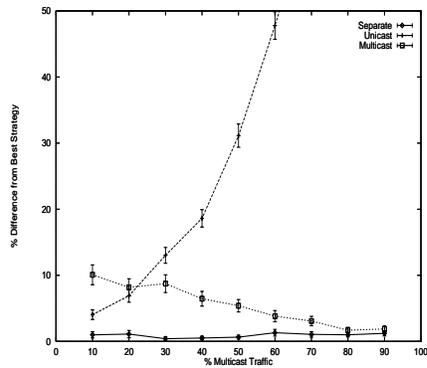
Figure 12 Comparison for $N = 50$, $G = 30$, $C = 10$, $\Delta = 4$, $\bar{g} = 0.25N$
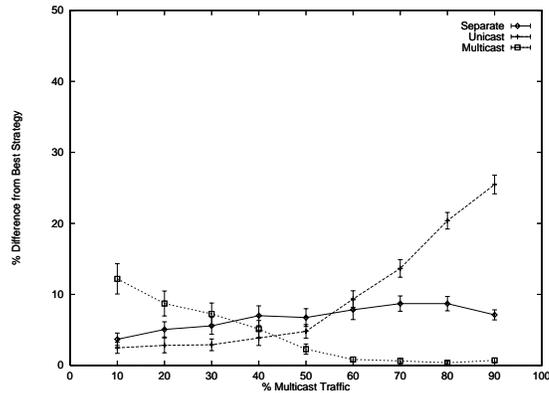
Figure 13 Comparison with hotspots for unicast traffic ($N = 20, G = 30, C = 10, \Delta = 4, \bar{g} = 0.25N$)

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.

Hotspots. Finally, in Figure 13 we show the behavior of the three strategies for the hotspot pattern described earlier. Except for the unicast traffic matrix **A**, the remaining parameters are the same as those in Figure 1. We note that the results obtained in Figure 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.

In Table 1, 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. Tables 2 and 3 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

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

Table 1 Best strategy when $\bar{g}$ and s are varied

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

Table 2 Best strategy when N and G are varied

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

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

Table 3 Best strategy when N and C are varied

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 1). 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 2). In this case, the network operates in the tuning limited region [6].

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 Figures 1–13). 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., $s \geq 40\%$.

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, treat-

ing 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 [6] must be run twice, once for unicast traffic and once for multicast traffic.

REFERENCES

[1] M. Ammar, G. Polyzos, and S. Tripathi (Eds.). Special issue on network support for multipoint communication. *IEEE Journal Selected Areas in Communications*, 15(3), April 1997.

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

[3] Z. Ortiz, G. N. Rouskas, and H. G. Perros. Scheduling of combined unicast and multicast traffic in broadcast WDM networks. Technical Report TR-97-09, North Carolina State University, Raleigh, NC, 1997.

[4] Z. Ortiz, G. N. Rouskas, and H. G. Perros. Scheduling of multicast traffic in tunable-receiver WDM networks with non-negligible tuning latencies. In *Proceedings of SIGCOMM '97*, pages 301–310. ACM, September 1997.

[5] G. N. Rouskas and M. H. Ammar. Multi-destination communication over tunable-receiver single-hop WDM networks. *IEEE Journal on Selected Areas in Communications*, 15(3):501–511, April 1997.

[6] G. N. Rouskas and V. Sivaraman. Packet scheduling in broadcast WDM networks with arbitrary transceiver tuning latencies. *IEEE/ACM Transactions on Networking*, 5(3):359–370, June 1997.

[7] V. Sivaraman and G. N. Rouskas. HiPeR-ℓ: A High Performance Reservation protocol with ℓook-ahead for broadcast WDM networks. In *Proceedings of INFOCOM '97*, pages 1272–1279. IEEE, April 1997.