Scheduling of Multicast Traffic in Tunable-Receiver WDM Networks with Non-Negligible Tuning Latencies

Zeydy Ortiz, George N. Rouskas, Harry G. Perros
Department of Computer Science, North Carolina State University, Raleigh, NC 27695-8206
{zortizl,rouskas,hp}@csc.ncsu.edu

Abstract

We consider the problem of supporting multipoint communication at the media access control (MAC) layer of broadcast-and-select WDM networks. We first show that bandwidth consumption and channel utilization arise as two conflicting objectives in the design of scheduling algorithms for multicast traffic in this environment. We then present a new technique for the transmission of multicast packets, based on the concept of a *virtual* receiver, a set of physical receivers which behave identically in terms of tuning. We also show that the number k of virtual receivers naturally captures the tradeoff between channel utilization and bandwidth consumption. Our approach decouples the problem of determining the virtual receivers from the problem of scheduling packet transmissions, making it possible to employ existing scheduling algorithms that have been shown to successfully hide the effects of tuning latency. Consequently, we focus on the problem of optimally selecting the virtual receivers, and we prove that it is $\mathcal{NP}$-complete. Finally, we present four heuristics of varying degrees of complexity for obtaining virtual receivers that provide a good balance between the two conflicting objectives.

1 Introduction

Many applications and telecommunication services in future high-speed networks will require some form of multipoint communication [1, 12]. Examples include distributed data processing, wide scale information dissemination, video distribution, and teleconferencing. The problems associated with providing network support for multipoint communication have been widely studied within a number of different networking contexts. As current network technologies evolve to an all-optical, largely passive infrastructure [6], these problems take on new significance, and raise a number of challenging issues that require novel solutions. In this paper we consider the problem of supporting multipoint communication at the media access control (MAC) layer of broadcast-and-select wavelength division multiplexed (WDM) networks [7], when tunable receivers are available at all nodes.

In optical WDM broadcast-and-select networks, information transmitted on any channel is broadcast to the entire set of nodes, but it is only received by those with a receiver listening on that channel. This feature, coupled with tunability at the receiving end, makes it possible to design receiver tuning algorithms [3, 9] such that a *single* transmission of a multicast packet can reach all receivers in the packet's destination set simultaneously. Its minimal bandwidth requirements make this approach especially appealing for transmitting multicast traffic. However, the design of appropriate receiver tuning algorithms is complicated by the fact that (a) tunable receivers take a non-negligible amount of time to switch between channels, and (b) different multicast groups may have several receivers in common. For unicast traffic, several scheduling algorithms exist that can successfully hide the effects of relatively large (compared to the packet transmission time) values of tuning latency [2, 4, 10]. Although a similar algorithm has been developed for multicast traffic [3], the achievable channel utilization can be very low.

In this paper we present a novel solution to the problem of scheduling multicast traffic in broadcast-and-select WDM networks. Our approach is based on the concept of a *virtual* receiver, a set of physical receivers that behave identically in terms of tuning. By partitioning the set of all physical receivers into virtual receivers, we effectively transform the original network with multicast traffic, into a new network with unicast traffic. Consequently, we can take advantage of scheduling algorithms such as the ones in [2, 4, 10] that have been shown to work well under non-negligible tuning latencies. Hence, our main focus is to select a partition of physical receivers into virtual receivers so as to achieve an optimal tradeoff between two conflicting objectives: bandwidth consumption and channel utilization.

In Section 2 we present our multicast transmission model and we introduce the concept of a virtual receiver. Lower bounds on the schedule length are given in Section 3, and some important properties of the bounds are also derived. We formulate the problem of optimally selecting a virtual receiver set in Section 4, and we show that it is $\mathcal{NP}$-complete. Heuristics for this problem are developed in Section 5. We present numerical results in Section 6, and we conclude the paper in Section 7.

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 $\lambda(i) \in \mathcal{C}$. We 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 network nodes are synchronized at slot boundaries. We only consider multicast traffic in this paper, and we let $g \subseteq \mathcal{N} = \{1, 2, \cdots, N\}$ represent the destination set or multicast group of a packet. We will also use $| g |$ to denote the cardinality of group g.

Let G represent the number of currently active [1] multicast groups (that is, each of these G groups receives traffic from at least one node in the network). Under the traffic scenario we are considering, there is a $N \times G$ multicast traffic demand matrix $\mathrm{M} = [m_{ig}]$, where m_{ig} is the number of multicast packets originating at source i and destined to multicast group g. We assume that traffic matrix M is known to all nodes. Matrix M will be used in this paper as the starting point to obtain a transmission schedule that specifies when receivers tune to the various channels, and when packet transmissions should take place. Information about the traffic demands m_{ig} may be collected using a distributed reservation protocol such as HiPeR-ℓ [11].

Given the assignment of transmit wavelengths $\{X_c\}$, we construct a new $C \times G$ traffic demand matrix $\mathrm{A} = [a_{cg}]$, where a_{cg} is the total amount of traffic to multicast group g carried by channel λ_c:

$$a_{cg} = \sum_{i \in X_c} m_{ig} \quad \forall\, c, g \tag{1}$$

We also let M denote the total traffic demand over all channels and groups:

$$M = \sum_{i=1}^{N} \sum_{g} m_{ig} = \sum_{c=1}^{C} \sum_{g} a_{cg} \tag{2}$$

Finally, 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. Our work is motivated by the observation that, at very high data rates and for small packet sizes, receiver tuning latency will be significant compared to a packet transmission time. Therefore, unless techniques that can effectively overlap the tuning latency are employed, any solutions to the problem of transmitting multicast traffic in a WDM broadcast-and-select environment will be highly inefficient.

2.1 Multicast Transmission Model

Given a traffic matrix M, there are several possible approaches to delivering the multicast packets to all receivers in their corresponding multicast groups. One extreme approach would be to separately transmit a copy of a packet to each of the packet's destinations. This solution can achieve high channel utilization since a number of transmissions may take place simultaneously on different channels (using, for example, the techniques in [2, 4, 10]). Its obvious drawback is high bandwidth consumption, since all packets to a multicast group g must be transmitted exactly $| g |$ times. Another possibility would be to somehow schedule all receivers of each multicast group g such that they simultaneously tune to a channel with packets for g. This approach has minimal bandwidth requirements, since only a single copy of each packet needs to be transmitted. However, transmissions to multicast groups with at least one receiver in common cannot be scheduled simultaneously, possibly resulting in low channel utilization. An algorithm based on a similar scheduling principle was presented in [3], and it was found to utilize only one channel (out of C) on average.

As we can see, channel utilization and bandwidth consumption are two conflicting objectives arising in the design of multicast traffic scheduling techniques. The two approaches just described can be thought of as two opposite extremes, each optimizing one objective but performing poorly in terms of the other. A third possibility that might achieve a reasonable performance in terms of both objectives would be to split multicast groups with common receivers into smaller sets, and transmit packets in multiple phases. However, this approach introduces two problems: (a) how to split groups with common receivers, and (b) how to coordinate the tuning of sets of receivers among the various channels. Both problems appear to be difficult to deal with, especially in the presence of non-negligible tuning latencies and when receivers may belong to multiple multicast groups.

In this paper we introduce a new technique for the transmission of multicast packets that achieves a good balance between channel utilization and bandwidth consumption. Our approach differs from previous solutions to the problem of scheduling multicast traffic in that it decouples the problem of determining how many times each packet should be transmitted, from the problem of scheduling the actual packet transmissions. As a result, it can take advantage of the scheduling algorithms in [10] that have been shown to successfully hide the effects of tuning latency, allowing us to concentrate on the important problem of tradeoff selection between the two conflicting goals.

We define a *virtual* receiver $V \subseteq \mathcal{N}$ 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 listen 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.

We also define a *k-virtual receiver set* $\mathcal{V}^{(k)}$, $1 \leq k \leq N$, 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 traffic matrix M, transmission of multicast packets proceeds as follows. A schedule (how it is constructed will be discussed shortly) specifies when the virtual receivers should tune to each channel. When a virtual receiver $V_l^{(k)}$ is on channel λ_c, each node in X_c (i.e., each node with λ_c as its transmit wavelength) 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 its 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.

Figure 1 shows an example schedule for $N = 5$ nodes, $C = 2$ channels, three different multicast groups f, g, and

[1] Typically, the number G of active groups is significantly smaller than the total number 2^N of possible groups.

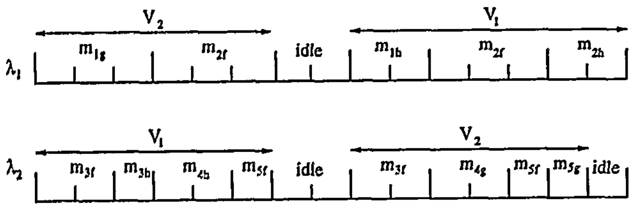

Figure 1: Example schedule for a network with $N = 5$, $C = 2$, $\Delta = 2$, and 2 virtual receivers

h, $\Delta = 2$, and the following parameters.

$$M = \begin{bmatrix} 0 & 3 & 2 \\ 3 & 0 & 2 \\ 2 & 0 & 1 \\ 0 & 2 & 2 \\ 1 & 1 & 0 \end{bmatrix} \quad \begin{array}{ll} & X_1 = \{1,2\} \\ f=\{2,3,4\} & X_2 = \{3,4,5\} \\ g=\{1,2\} & \\ h=\{4,5\} & V_1 = \{4,5\} \\ & V_2 = \{1,2,3\} \end{array} \tag{3}$$

We now observe that, given the k-virtual receiver set $\mathcal{V}^{(k)}$, a node $i \in X_c$, $c = 1, \cdots, C$, must transmit a number of packets to virtual receiver $V_l^{(k)}$,$l = 1, \cdots, k$, equal to the sum of its packets for any multicast group g with members whose receivers are in $V_l^{(k)}$. Let B= $[b_{il}]$ be the $N \times k$ matrix with

$$b_{il} = \sum_{g: g \cap V_l^{(k)} \neq \phi} m_{ig} \tag{4}$$

Quantity b_{il} represents the amount of traffic originating at source i and destined to virtual receiver $V_l^{(k)}$. By specifying the k-virtual receiver set we have effectively transformed our original network with multicast traffic matrix M, to an equivalent network with *unicast* traffic matrix B. This new network has the same number of transmitters and channels, and the same tuning latency as the original one, but only k receivers, corresponding to the k virtual receivers in $\mathcal{V}^{(k)}$. We can then immediately employ the algorithms in [10] (which were developed for unicast traffic) to construct schedules for clearing matrix B in the new network. The reader is referred to [10] for details on the optimality properties of these scheduling algorithms. For the rest of this paper we concentrate on the problem of selecting the virtual receiver set $\mathcal{V}^{(k)}$ to use.

When $k = 1$, each multicast packet is transmitted only once, but there is no transmission concurrency; only one channel is utilized at a time. For larger values of k, each of the k virtual receivers can be independently tuned to the various channels, and a higher degree of transmission concurrency can be achieved. On the short side, multicast packets may have to be transmitted multiple times when $k > 1$, since members of a multicast group g may belong to different virtual receivers. When $k = N$, each virtual receiver consists of exactly one physical receiver, and each multicast packet to group g has to be transmitted exactly $\mid g \mid$ times. Hence, the number k of virtual receivers naturally captures the tradeoff between channel utilization and bandwidth consumption. The objective of our work is to select k and the virtual receivers in a way that strikes a balance between the two conflicting goals.

Next, we obtain lower bounds on the length of schedules for clearing matrix M, and we present some properties that quantify the effect of the number k of virtual receivers on these bounds.

3 Lower Bounds on the Schedule Length

Let $\mathcal{V}^{(k)} = \{V_1^{(k)}, \cdots, V_k^{(k)}\}$ be a k-virtual receiver set. We observe that the length of any schedule cannot be smaller than the number of slots required to carry all traffic from the transmitters of any given channel to virtual receivers, yielding the *channel* bound:

$$\hat{F}_{ch}(\mathcal{V}^{(k)}) = \max_{c=1,\cdots,C} \left\{ \sum_{l=1}^{k} \sum_{g: g \cap V_l^{(k)} \neq \phi} a_{cg} \right\} \tag{5}$$

We can obtain a different lower bound by adopting a virtual receiver's point of view. Let T_l, $1 \le T_l \le C$, represent 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 $V_l^{(k)}$). Each virtual receiver $V_l^{(k)}$ needs 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_l wavelengths. We call this the *receiver* bound; it can be expressed as:

$$\hat{F}_r(\mathcal{V}^{(k)}) = \max_{l=1,\cdots,k} \left\{ \left[\sum_{c=1}^{C} \sum_{g: g \cap V_l^{(k)} \neq \phi} a_{cg} \right] + T_l \Delta \right\} \tag{6}$$

We have written the channel and receiver bounds as functions of the virtual receiver set to emphasize the fact that their values depend on the actual receivers comprising each virtual receiver, not just on the number k of virtual receivers. We now obtain the overall lower bound as:

$$\hat{F}(\mathcal{V}^{(k)}) = \max \{ \hat{F}_{ch}(\mathcal{V}^{(k)}), \hat{F}_r(\mathcal{V}^{(k)}) \} \tag{7}$$

To gain some insight into how the number k of virtual receivers may affect the relative values of the two bounds in (5) and (6), let us consider the two extreme scenarios, $k = 1$ and $k = N$. For $k = 1$, there is only one virtual receiver, $\mathcal{V}^{(1)} = \mathcal{N}$, which includes all physical receivers, and we can rewrite (5) and (6) as follows:

$$\frac{M}{C} \le \hat{F}_{ch}(\mathcal{V}^{(1)}) = \max_{c=1,\cdots,C} \left\{ \sum_{g} a_{cg} \right\} < M \tag{8}$$

$$\hat{F}_r(\mathcal{V}^{(1)}) = \left\{ \sum_{c=1}^{C} \sum_{g} a_{cg} \right\} + C\Delta = M + C\Delta \tag{9}$$

In (8) we have assumed that no single channel will carry all traffic, and thus the channel bound will be strictly less than M, while in (9) we have assumed that at least one transmitter at each channel will have traffic for at least one multicast group, and thus $T_l = C$. Obviously, the receiver bound dominates in this case, even if $\Delta = 0$ or $T_l < C$. On the other hand, for $k = N$, the virtual receiver set is $\{\{1\}, \cdots, \{N\}\}$, and (5) and (6) become:

$$\hat{F}_{ch}(\mathcal{V}^{(N)}) = \max_{c=1,\cdots,C} \left\{ \sum_{g} \mid g \mid a_{cg} \right\} \tag{10}$$

$$\hat{F}_r(\mathcal{V}^{(N)}) = \max_{l=1,\cdots,N} \left\{ \left[\sum_{c=1}^{C} \sum_{g: l \in g} a_{cg} \right] + T_l \Delta \right\} \tag{11}$$

It is not clear from (10) and (11) which bound dominates in this case. The channel bound in (10) depends on the number of receivers in each multicast group g, since packets to g must be individually transmitted to each member of the group. On the other hand, the receiver bound depends on (a) the value of the tuning latency Δ, and (b) the amount of traffic destined to each receiver. In general, we expect the channel bound (10) to be the dominant one when $k = N$, *unless $\Delta \gg 1$ and/or there is a hot-spot receiver*, i.e., one that is a member of a large number of multicast groups.

The following lemma establishes a lower bound on the length of any schedule for matrix M. We note, however, that this absolute lower bound is not necessarily achievable.

Lemma 3.1 *Regardless of the method used to transmit multicast packets, a lower bound on the length of any schedule to clear matrix* M, *is given by:*

$$\hat{F} = \max\left\{\hat{F}_r(\mathcal{V}^{(N)}), \hat{F}_{ch}(\mathcal{V}^{(1)})\right\} \quad (12)$$

Proof. The length of any schedule for M cannot be smaller than the number of multicast packets to be transmitted on any channel, which is given by $\hat{F}_{ch}(\mathcal{V}^{(1)})$ in (8). 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 $\hat{F}_r(\mathcal{V}^{(N)})$ in (11). □

3.1 Monotonicity Properties of the Lower Bounds

Let us now study the behavior of the receiver and channel bounds as a function of the number k of virtual receivers. Intuitively, the smaller (larger) the number of virtual receivers, the larger (smaller) the number of physical receivers within each virtual receiver, and the larger (smaller) the number of multicast groups with members within each virtual receiver. Consequently, we expect the receiver bound to increase as the number of virtual receivers decreases, and vice versa. By applying a similar argument we expect that the channel bound move in the opposite direction, that is, it should decrease as the number of virtual receivers decreases, and vice versa. Returning to expressions (8) - (11), we note that the two special cases $k = 1$ and $k = N$ appear to confirm our intuition, since we immediately obtain that

$$\hat{F}_{ch}(\mathcal{V}^{(1)}) \leq \hat{F}_{ch}(\mathcal{V}^{(N)}) \text{ and } \hat{F}_r(\mathcal{V}^{(N)}) \leq \hat{F}_r(\mathcal{V}^{(1)}) \quad (13)$$

In the general case, however, the lower bounds in (5) and (6) are strongly dependent on the actual virtual receiver set $\mathcal{V}^{(k)}$. As a result, the qualitative arguments we presented above cannot be used to draw similar conclusions for virtual receiver sets with an arbitrary number k, $1 < k < N$, of virtual receivers. In fact, it is possible that there exist two virtual receiver sets, one with k and one with $k' > k$ virtual receivers, such that the receiver bound of the k-virtual receiver set is smaller than the receiver bound of the k'-virtual receiver set; similarly for the channel bound.

Although given two arbitrary virtual receiver sets there is no way to reach *a priori* any conclusions regarding the relative ordering of their channel and receiver bounds, the two bounds do exhibit behavior that is in agreement with the intuitive arguments discussed above when two special operations are applied to virtual receiver sets. The two operations are:

- *JOIN*$(\mathcal{V}^{(k)}, n)$, $1 \leq n < k \leq N$. *JOIN* creates a $(k-n)$-virtual receiver set by replacing any $n+1$ of the virtual receivers in $\mathcal{V}^{(k)}$ with their union, and keeping the other $k - n - 1$ virtual receivers the same.
- *SPLIT*$(\mathcal{V}^{(k)}, n)$, $1 \leq k < k + n \leq N$. *SPLIT* creates a $(k+n)$-virtual receiver set by arbitrarily splitting any virtual receiver in $\mathcal{V}^{(k)}$ with at least $n+1$ physical receivers into $n+1$ virtual receivers, and keeping the other $k-1$ virtual receivers the same.

The following lemma states the monotonic behavior of the channel and receiver bounds when the *JOIN* operation is applied.

Lemma 3.2 (Monotonicity Property of *JOIN*) *Let $\mathcal{V}^{(k)}$ be a k-virtual receiver set, and let $\mathcal{V}^{(k-n)}, 1 \leq n < k$, be the $(k-n)$-virtual receiver set obtained by applying the* JOIN$(\mathcal{V}^{(k)}, n)$, $1 \leq n < k \leq N$, *operation. Then, we have that*

$$\hat{F}_{ch}(\mathcal{V}^{(k-n)}) \leq \hat{F}_{ch}(\mathcal{V}^{(k)}); \hat{F}_r(\mathcal{V}^{(k-n)}) \geq \hat{F}_r(\mathcal{V}^{(k)}) \quad (14)$$

Proof. Let $\mathcal{V}^{(k)} = \{V_1^{(k)}, \cdots, V_k^{(k)}\}$ be the initial k-virtual receiver set. Without loss of generality, we assume that the $(k-n)$-virtual receiver set is formed by taking the union of the last $n+1$ virtual receivers of $\mathcal{V}^{(k)}$ (if that is not the case, we can always relabel the virtual receivers). Hence, we have that

$$V_1^{(k-n)} = V_1^{(k)}, \ \cdots, \ V_{k-n-1}^{(k-n)} = V_{k-n-1}^{(k)},$$
$$V_{k-n}^{(k-n)} = V_{k-n}^{(k)} \cup \cdots \cup V_k^{(k)} \quad (15)$$

Then, the relative values of the channel and receiver bounds for the k- and $(k-n)$-virtual receiver sets depend only on the contributions of virtual receivers $V_{k-n}^{(k)}, \cdots, V_k^{(k)}$ and $V_{k-n}^{(k-n)}$, respectively, to these bounds.

Let us first consider the receiver bound in (6). By construction, the value of the term within the brackets in (6) for $V_{k-n}^{(k-n)}$ is at least equal to the value of the same term for any of $V_{k-n}^{(k)}, \cdots, V_k^{(k)}$. Also, the number of channels to which virtual receiver $V_{k-n}^{(k-n)}$ has to tune is at least equal to the maximum number of channels to which any of the virtual receivers $V_{k-n}^{(k)}, \cdots, V_k^{(k)}$ have to tune. Therefore, the receiver bound for $\mathcal{V}^{(k-n)}$ cannot be smaller than that for $\mathcal{V}^{(k)}$. Thus, the second inequality in (14) holds. For the first inequality in (14), note that the nodes in X_c, $c = 1, \cdots, C$, will transmit a number of packets to virtual receiver $V_{k-n}^{(k-n)}$ which is at most equal to the sum of the packets they would transmit to virtual receivers $V_{k-n}^{(k)}, \cdots, V_k^{(k)}$ (refer to (5)). Therefore, the first inequality in (14) also holds true. □

As a consequence of the monotonicity property of *JOIN*, if we start with the N-virtual receiver set $\mathcal{V}^{(N)}$ and apply an arbitrary sequence of *JOIN* operations, we will obtain a sequence of virtual receiver sets, each with a smaller number of virtual receivers, such that the channel (receiver) bound of any virtual receiver set in the sequence is no greater (smaller) than the channel (receiver) bound of the previous set in the sequence. A similar monotonicity property holds for the *SPLIT* operation and is stated in the following lemma. Lemma 3.3 is in a sense the inverse of Lemma 3.2. Its proof is omitted since it is very similar to that of Lemma 3.2.

Lemma 3.3 (Monotonicity Property of *SPLIT*) *Let $\mathcal{V}^{(k)}$ be a k-virtual receiver set, and let $\mathcal{V}^{(k+n)}, 1 \leq n < k$,* be the $(k+n)$-virtual receiver set obtained by applying the SPLIT$(\mathcal{V}^{(k)}, n)$, $1 \leq k < k+n \leq N$, operation. Then, we have that

$$\hat{F}_{ch}(\mathcal{V}^{(k+n)}) \geq \hat{F}_{ch}(\mathcal{V}^{(k)}); \hat{F}_r(\mathcal{V}^{(k+n)}) \leq \hat{F}_r(\mathcal{V}^{(k)}) \qquad (16)$$

4 The Virtual Receiver Set Problem

Our objective is to determine a virtual receiver set such that the length of the schedule to transmit the multicast demand matrix M is minimum over all virtual receiver sets. Unfortunately, given a virtual receiver set, the length of the corresponding schedule is not known until after we run the algorithms in [10]. However, we have found that the lower bound accurately characterizes the scheduling efficiency of our algorithms, since, on the average, the schedules produced by the algorithms in [10] are very close to (and in many cases equal to) the lower bound. Based on this observation, we will instead seek a virtual receiver set that minimizes the lower bound in (7), a known quantity, rather than the actual schedule length. This problem, which we will call the *Virtual Receiver Set Problem (VRSP)* arises naturally as a decision problem, and can be formally expressed as follows.

Problem 4.1 (VRSP) *Given N nodes, C channels, transmitter sets X_c, tuning latency Δ, G multicast groups, a multicast traffic demand matrix M, and a real number F, does there exist a k-virtual receiver set $\mathcal{V}^{(k)}$, $1 \leq k \leq N$, such that the lower bound in (7) $\hat{F}(\mathcal{V}^{(k)}) \leq F$?*

We proceed to show that the following simpler version of *VRSP*, whereby the value of k is fixed to 2, is $\mathcal{NP}$-complete.

Problem 4.2 (2-VRSP) *Given N nodes, C channels, transmitter sets X_c, tuning latency Δ, G multicast groups, a multicast traffic demand matrix M, and a real number F, does there exist a 2-virtual receiver set $\mathcal{V}^{(2)}$ such that the lower bound in (7) $\hat{F}(\mathcal{V}^{(2)}) \leq F$?*

Theorem 4.1 2-VRSP *is $\mathcal{NP}$-complete.*

Proof. It is easy to see that *2-VRSP* is in the class $\mathcal{NP}$, since a nondeterministic algorithm need only guess a 2-virtual receiver set and verify in polynomial time that the lower bound in (7) is at most F.

We now transform the *PARTITION* problem [5] to *2-VRSP*. Let $S = \{1, 2, \cdots, k\}, k \geq 3$, be the set of elements of weights w_n, $n = 1, \cdots, k$, making up an arbitrary instance of *PARTITION*, and let $W = \sum_{n=1}^{k} w_n$. We construct an instance of *2-VRSP* as follows. The network has $N = k$ nodes, $C = 3$ channels, $G = k$ groups $g_n = \{n\}$, $n = 1 \cdots, k$, and the tuning latency $\Delta = 0$. We let the transmitter sets be $X_1 = \{1, \cdots, \lceil \frac{k}{3} \rceil\}$, $X_2 = \{\lceil \frac{k}{3} \rceil + 1, \cdots, \lceil \frac{2k}{3} \rceil\}$, and $X_3 = \{\lceil \frac{2k}{3} \rceil + 1, \cdots, k\}$. The multicast demand matrix M$= [m_{i,g_n}]$ is such that

$$m_{i,g_n} = \frac{w_n}{k}, \quad i, n = 1, \cdots, k \qquad (17)$$

Finally, we let $F = \frac{W}{2}$.

It is obvious that this transformation can be performed in polynomial time. We also note that, for $k \geq 3$, the channel bound in (5) becomes

$$\hat{F}_{ch}(\mathcal{V}^{(2)}) = \left\lceil \frac{k}{3} \right\rceil \sum_{n=1}^{k} \frac{w_n}{k} = \left\lceil \frac{k}{3} \right\rceil \frac{W}{k} \leq \frac{W}{2} \qquad (18)$$

independently of the actual virtual receiver sets. As a result, a feasible 2-virtual receiver set will exist for this instance of *2-VRSP* as long as the receiver bound in (6) satisfies $\hat{F}_r(\mathcal{V}^{(2)}) < \frac{W}{2}$. Hence, we only consider the receiver bound in the following discussion.

We now show that a feasible 2-virtual receiver set exists for the above instance of *2-VRSP* if and only if set S has a partition. If S has a partition S_1, S_2, then $\sum_{n \in S_1} w_n = \sum_{n \in S_2} w_n = \frac{W}{2}$. Then, the 2-virtual receiver set with $V_1^{(2)} = S_1$ and $V_2^{(2)} = S_2$, is a feasible receiver set for *2-VRSP* since the receiver bound in (6) evaluates to:

$$\begin{aligned} \hat{F}_r(\mathcal{V}^{(2)}) &= \max \left\{ \sum_{n \in S_1} \sum_{i=1}^{k} \frac{w_n}{k}, \sum_{n \in S_2} \sum_{i=1}^{k} \frac{w_n}{k} \right\} \\ &= \max \left\{ \sum_{n \in S_1} w_n, \sum_{l \in S_2} w_l \right\} = \frac{W}{2} \qquad (19) \end{aligned}$$

Conversely, let $\mathcal{V}^{(2)} = \{V_1^{(2)}, V_2^{(2)}\}$ be a feasible virtual receiver set for *2-VRSP*. We have that

$$\hat{F}_r(\mathcal{V}^{(2)}) = \max \left\{ \sum_{n \in V_1^{(2)}} w_n, \sum_{n \in V_2^{(2)}} w_n \right\} \leq \frac{W}{2} \qquad (20)$$

Since the sum of all w_n is equal to W, (20) can only hold with equality in which case we have that $\sum_{n \in V_1^{(2)}} w_n = \sum_{l \in V_2^{(2)}} w_n = \frac{W}{2}$. Hence, $V_1^{(2)}$, $V_2^{(2)}$ is a partition of S. □

4.1 Special Cases

Although *VRSP* is $\mathcal{NP}$-complete in the general case, there do exist two interesting special cases for which the optimal solution can be obtained in polynomial time. These are discussed in the following two subsections.

4.1.1 All-to-All Broadcast

The first special case we study is the all-to-all broadcast problem, whereby there is a single multicast group $g = \mathcal{N}$ encompassing all nodes in the network. We let m_i denote the number of broadcast packets originating at node i, and $M = \sum_{i=1}^{N} m_i$. For a k-virtual receiver set, the two bounds (5) and (6) can be rewritten as

$$\hat{F}_{ch}(\mathcal{V}^{(k)}) = k \max_{c=1,\cdots,C} \left\{ \sum_{i \in X_c} m_i \right\} \qquad (21)$$

$$\hat{F}_r(\mathcal{V}^{(k)}) = \sum_{i=1}^{N} m_i + C\Delta = M + C\Delta \qquad (22)$$

We note that the bounds are independent of the actual virtual receiver sets, and that only the channel bound depends on the number k of virtual receivers. Therefore, for the all-to-all broadcast case, *VRSP* reduces to obtaining the number k of virtual receivers that minimizes the overall lower bound.

To obtain the optimal value for k, we observe that the channel bound depends on the assignment of transmit wavelengths $\{X_c\}$, but that it cannot be less than $k\frac{M}{C}$. Let ϵ

be a real number such that the channel bound in (21) is equal to $k\frac{M}{C}+\epsilon$. Since the receiver bound is independent of k, the overall lower bound is minimized when $\hat{F}_{ch}(\mathcal{V}^{(k)}) \le \hat{F}_r(\mathcal{V}^{(k)})$, or equivalently, if

$$k\frac{M}{C}+\epsilon \le M + C\Delta \;\Leftrightarrow\; k \le C + \frac{C^2\Delta - C\epsilon}{M} \qquad (23)$$

Let us now further suppose that $\epsilon = 0$, that is, the broadcast traffic is completely balanced across the C channels. Then, after some algebraic manipulation of the first expression in (23) (with the equality sign), we obtain:

$$\frac{kM}{C} = \frac{kC\Delta}{k-C} \qquad (24)$$

Relationship (24) is fundamental in that it represents the point at which wavelength concurrency balances the tuning latency. Indeed, if the last quantity in (23) is integer for $\epsilon = 0$, and we set k to this value, then the resulting schedule will have length equal to the lower bound, and it will be such that exactly C (respectively, $k - C$) virtual receivers are in the receiving (respectively, tuning) state within each slot. Consequently, all $kC\Delta$ tuning slots will be overlapped with the kM slots containing packet transmissions, and vice versa.

4.1.2 Disjoint Multicast Groups

Let us now consider the case when there are $G < N$ disjoint multicast groups $g_1, \cdots, g_G$. Obviously, we also have that $g_1 \cup \cdots \cup g_G \cup f = \mathcal{N}$, where f is the (possibly empty) set of nodes which are not members of any group. Let $\mathcal{V}^{(G)}$ denote the G-virtual receiver set $\{g_1, \cdots, g_G\}$. The channel bound of $\mathcal{V}^{(G)}$ is equal to the sum of the traffic demands on the dominant channel, which is a lower bound on any k virtual receiver set. Similarly, the receiver bound of $\mathcal{V}^{(G)}$ is determined by the traffic and tuning requirements of the dominant multicast group; again, the latter is a lower bound on any k-virtual receiver set. We conclude that, when the G multicast groups are disjoint, the G-virtual receiver set where each virtual receiver corresponds to a different multicast group, is an optimal solution to *VRSP*.

5 Optimization Heuristics for *VRSP*

In this section we develop four heuristics for the optimization problem corresponding to *VRSP*. Specifically, our objective is to obtain a k-virtual receiver set, $1 \le k \le N$, such that the overall bound in (7) for the given instance of *VRSP* is minimized. Our heuristic approaches exploit the monotonicity properties stated in Lemmas 3.2 and 3.3. Although it is *not guaranteed that the heuristics will find the virtual receiver set with the minimum bound, we will prove that they do converge to a local minimum.*

5.1 The *Greedy JOIN (G-JOIN)* Heuristic

Our first approach is to start with the N-virtual receiver set $\{\{1\}, \cdots, \{N\}\}$ for which we expect the channel bound in (10) to be greater than the receiver bound in (11). We then repeatedly apply the *JOIN*$(\mathcal{V}^{(k)}, 1)$ operation to obtain a sequence of virtual receiver sets, each with one fewer virtual receiver. Because of the monotonicity property (14) of the *JOIN* operation, we expect the channel (receiver) bound to decrease (increase) after each *JOIN*, yielding a virtual receiver set with a lower overall bound. When the virtual receiver set is $\mathcal{V}^{(k)}$, we select the two (out of k) virtual receivers to join into a single virtual receiver V by employing the following greedy rule:

> Select the pair of virtual receivers such that the quantity corresponding to V's term in the receiver bound (6) for $\mathcal{V}^{(k-1)}$ is minimum over all pairs of virtual receivers in $\mathcal{V}^{(k)}$. If there are more than one pairs that achieve the minimum, select the pair that minimizes the channel bound (5) for $\mathcal{V}^{(k-1)}$. If again there is a tie, then break it arbitrarily.

In essence, the heuristic linearly searches over all possible values of k, starting with $k = N$, in an attempt to find a virtual receiver set with a low overall bound. Because of the greedy rule it uses when applying the *JOIN* operation, we call it the *Greedy JOIN (G-JOIN)* heuristic. A detailed description of the *G-JOIN* heuristic is provided in Figure 2. Regarding its complexity, we note that, for a k-virtual receiver set, Step 8 of the heuristic takes $\mathcal{O}(k^2)$ time, since one of a possible $\frac{k(k-1)}{2}$ pairs of virtual receivers must be selected. Since the while loop will be executed at most N times, the overall complexity is $\mathcal{O}(N^3)$.

We now state and prove the optimality property of the *G-JOIN* heuristic.

The Greedy JOIN (G-JOIN) **Heuristic**
Input: N, C, X_c, $c = 1, \cdots, C$, G multicast groups, and multicast traffic matrix $\mathbf{M}$
Output: A virtual receiver set

1. begin
2. Set $k = N$
3. Set $\mathcal{V}^{(k)} = \{\{1\}, \cdots, \{N\}\}$
4. Set $\hat{F}_{ch} = \hat{F}_{ch}(\mathcal{V}^{(k)})$
5. Set $\hat{F}_r = \hat{F}_r(\mathcal{V}^{(k)})$ // Because of (10) and (11), // we expect that $\hat{F}_{ch} \ge \hat{F}_r$ at this step
6. while $\hat{F}_{ch} > \hat{F}_r$ do
7. Set $k = k - 1$
8. Select two virtual receivers in $\mathcal{V}^{(k+1)}$ using the greedy rule described in Section 5.1
9. Set $\mathcal{V}^{(k)}$ to the set resulting from $\mathcal{V}^{(k+1)}$ by joining the two virtual receivers in Step 8
10. Set $\hat{F}_{ch} = \hat{F}_{ch}(\mathcal{V}^{(k)})$
11. Set $\hat{F}_r = \hat{F}_r(\mathcal{V}^{(k)})$
12. end while
13. Return the virtual receiver set with the smallest overall bound among $\mathcal{V}^{(k)}$ and $\mathcal{V}^{(k+1)}$
14. end of algorithm

Figure 2: The *G-JOIN* heuristic for *VRSP*

Lemma 5.1 *The* G-JOIN *heuristic in Figure 2 returns a virtual receiver set that achieves a local minimum with respect to the lower bound in (7).*

Proof. We first observe that, because of (8) and (9), if the value of k in the *G-JOIN* heuristic becomes 1, then the receiver bound will be greater than the channel bound, the condition of the while loop in Figure 2 will become false, and the algorithm will terminate. Therefore, the heuristic will always return a valid virtual receiver set.

Let $k^* \geq 1$ be the value of k upon termination of the *G-JOIN* heuristic. Because of the monotonicity property of the *JOIN* operation, the sequence of virtual receiver sets constructed by *G-JOIN* are such that:

$$\hat{F}_{ch}(\mathcal{V}^{(N)}) \geq \cdots \geq \hat{F}_{ch}(\mathcal{V}^{(k^*+1)}) \geq \hat{F}_{ch}(\mathcal{V}^{(k^*)}) \qquad (25)$$

and

$$\hat{F}_r(\mathcal{V}^{(N)}) \leq \cdots \leq \hat{F}_r(\mathcal{V}^{(k^*+1)}) \leq \hat{F}_r(\mathcal{V}^{(k^*)}) \qquad (26)$$

Since the heuristic terminates when the condition of the while loop becomes false, we also have that

$$\hat{F}_{ch}(\mathcal{V}^{(k^*+1)}) > \hat{F}_r(\mathcal{V}^{(k^*+1)}); \hat{F}_{ch}(\mathcal{V}^{(k^*)}) \leq \hat{F}_r(\mathcal{V}^{(k^*)}) \qquad (27)$$

From (25) – (27) it immediately follows that (a) the overall lower bound of $\mathcal{V}^{(k^*+1)}$ is minimum among virtual receiver sets $\mathcal{V}^{(N)}, \cdots, \mathcal{V}^{(k^*+1)}$, since the channel bound decreases from $\hat{F}_{ch}(\mathcal{V}^{(N)})$ to $\hat{F}_{ch}(\mathcal{V}^{(k^*+1)})$ and the receiver bound increases from $\hat{F}_r(\mathcal{V}^{(N)})$ to $\hat{F}_r(\mathcal{V}^{(k^*+1)})$, but the latter is not greater than $\hat{F}_{ch}(\mathcal{V}^{(k^*+1)})$, and (b) any virtual receiver set obtained from $\mathcal{V}^{(k^*)}$ will not have a smaller overall lower bound since $\hat{F}_r(\mathcal{V}^{(k^*)}) \geq \hat{F}_{ch}(\mathcal{V}^{(k^*)})$ and the monotonicity property of *JOIN* guarantees that the receiver bound of any subsequent virtual receiver set may not decrease. Therefore, we cannot do any better by using *JOIN* operations, and the heuristic terminates by returning the virtual receiver set with the smallest lower bound among $\mathcal{V}^{(k^*+1)}$ and $\mathcal{V}^{(k^*)}$. □

5.2 The *Random JOIN (R-JOIN)* Heuristic

This heuristic is very similar to *G-JOIN*. The main difference is that, when the virtual receiver set is $\mathcal{V}^{(k)}$, we randomly select two of the k virtual receivers to join into a single virtual receiver. As a result, the complexity is $\mathcal{O}(CN)$, since Step 8 of the *R-JOIN* heuristic (compare to Figure 2) takes constant time, and the execution time of the while loop is dominated by the computation of the new channel bound at Step 10. Because of its low running time requirement, this heuristic allows us to study the tradeoff between speed of execution and the quality of the final solution. An optimality property similar to the one in Lemma 5.1 also holds for *R-JOIN*.

5.3 The *Greedy SPLIT (G-SPLIT)* Heuristic

The *Greedy SPLIT (G-SPLIT)* heuristic is similar to *G-JOIN*, but it works in the opposite direction, searching from smaller to larger values of k. Specifically, it starts with the 1-virtual receiver set $\mathcal{N} = \{1, 2, \cdots, N\}$, and repeatedly applies the $SPLIT(\mathcal{V}^{(k)}, 1)$ operation to obtain a sequence of virtual receiver sets, each with one more virtual receiver. Recall that the receiver bound (9) is greater than the channel bound (8) when $k = 1$. The heuristic continues until (a) a virtual receiver set is found such that its channel bound is greater than or equal to its receiver bound, or (b) $k = N$, whichever occurs first. When the virtual receiver set is $\mathcal{V}^{(k)}$, we apply the following greedy rule for splitting one of its virtual receivers into two sets.

> Let $V_l^{(k)}$ be a virtual receiver with cardinality $n > 1$ such that the quantity corresponding to $V_l^{(k)}$'s term in the receiver bound (6) is maximum over all virtual receivers in $\mathcal{V}^{(k)}$ with cardinality greater than one. Select two receivers in $V_l^{(k)}$ that have the least number of multicast groups in common, say, i and j. Repeat the following for all other receivers in $V_l^{(k)}$. Find a receiver m that has the most multicast groups in common with i or j. If m has more multicast groups in common with i (respectively, j), put it in a virtual receiver set with i (j). If m has the same number of groups in common with i and j (or it has nothing in common) then compute the receiver bound (6) for the virtual receiver set of i and j if m was added to the set, and add m to the set that has the smaller bound.

Selecting and splitting one of the virtual receivers of a k-virtual receiver set takes time $\mathcal{O}(GN^2)$, and thus, the overall complexity of this heuristic is $\mathcal{O}(GN^3)$.

Because of the monotonicity property of *SPLIT*, the *G-SPLIT* heuristic has the following optimality property, stated without proof.

Lemma 5.2 *The* G-SPLIT *heuristic returns a virtual receiver set that achieves a local minimum with respect to the lower bound in (7).*

5.4 The *Random SPLIT (R-SPLIT)* Heuristic

The *Random SPLIT (R-SPLIT)* heuristic operates exactly like *G-SPLIT*, but uses a different rule for splitting a virtual receiver when the virtual receiver set is $\mathcal{V}^{(k)}$, $k < N$. Let $V_l^{(k)}$ be a virtual receiver with cardinality $n > 1$ such that the quantity corresponding to $V_l^{(k)}$'s term in the receiver bound (6) is maximum over all virtual receivers in $\mathcal{V}^{(k)}$ with cardinality greater than one. A random integer between 1 and $n-1$ is chosen, say, p, and then p elements of $V_l^{(k)}$ are randomly selected to form a new virtual receiver. Since, in the worst case, the value of p will be one for all k, and the heuristic may not terminate until $k = N$, its complexity is $\mathcal{O}(N^2)$. An optimality property similar to the one in Lemma 5.2 also holds for *R-JOIN*.

6 Numerical Results

We now study the relative performance of the four heuristics for *VRSP* presented in the previous section, namely, *G-JOIN*, *R-JOIN*, *G-SPLIT*, and *R-SPLIT*. Let $\hat{F}$ in (12) be the lower bound on an instance of *VRSP*, and let $\hat{F}(\mathcal{V}^{(k)})$ be the lower bound in (7) corresponding to the k-virtual receiver set $\mathcal{V}^{(k)}$ returned by one of the heuristics. Quantity $\frac{\hat{F}(\mathcal{V}^{(k)}) - \hat{F}}{\hat{F}} 100\%$ represents how far the k virtual receiver set $\mathcal{V}^{(k)}$ is from the lower bound. We are interested in the average performance of the four heuristics, therefore, in this section we plot the above quantity (averaged over twenty random instances of *VRSP*) for various values of the number N of nodes, the number C of channels, and the number G of multicast groups.

We have generated random instances of *VRSP*, i.e., random matrices A [2] and random multicast groups, as follows.

[2] Recall that if the multicast traffic matrix M and the sets X_c are

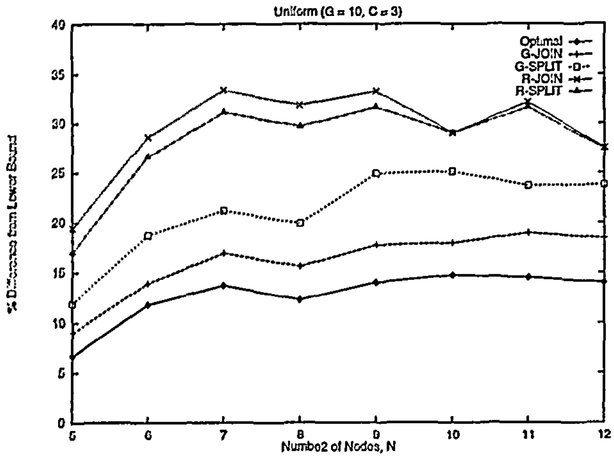

Figure 3: Heuristic comparison for $C = 3$ channels, $G = 10$ (uniform case)

The elements of each matrix A were selected as integers uniformly distributed in the range [0,20]. To construct the G multicast groups, we assigned a probability p_j to receiver j, representing the probability that the receiver would belong to a particular group. Each multicast group was determined by drawing N random numbers q_j uniformly distributed in (0,1), and including all receivers for which $q_j < p_j$ in the group. We have used two sets of values for p_j. In the *uniform* case, we let $p_j = 0.5$ for all j, that is, each receiver is equally likely to belong to a multicast group. To study how the existence of *hot spots* affects the behavior of the heuristics, we have also used $p_j = 0.6, j = 1, \cdots, 5$, and $p_j = \frac{0.5N-3}{N-5}, j = 6, \cdots, N$. In other words, the first five receivers were more likely to belong to a multicast group than the other $N - 5$ receivers; however, the average size of a multicast group was $\frac{N}{2}$, the same as for the uniform case. Finally, we have let the tuning latency $\Delta = 2$ in all test cases.

In Figure 3 we plot the performance of the four heuristics for a small number of nodes $N \leq 12$ and for $C = 3, G = 10$. The figure also shows how far the *optimal* solution is from the lower bound in (12); the optimal was obtained through a branch-and-bound technique we developed for *VRSP* [8]. Our first observation is that the lower bound does *not* accurately characterize the optimal solution to *VRSP*, since the value at the optimal can be up to 15% higher than the lower bound. Although we could not obtain the optimal solution for larger values of N, it seems reasonable to assume that the performance of our heuristics relative to the optimal solution is significantly better than what the comparison against the lower bound (in the following figures) suggests. This assumption is further supported by the fact that the behavior of the optimal solution in Figure 3 appears to be similar to that of the four heuristics.

Regarding the relative performance of the heuristics, the behavior emerging in Figure 3 is typical of the results that follow. We first note that the greedy heuristics perform better than the random ones; this is simply a reflection of the level of sophistication of the two types of heuristics. The *R-SPLIT* heuristic has a slight edge over *R-JOIN*, probably because in *R-JOIN* the two virtual receivers to be joined are chosen completely at random, while in *R-SPLIT* the virtual receiver to be split is *not* chosen randomly (although it is split randomly). On the other hand, the higher complexity of *G-SPLIT* does not pay off in terms of performance compared to *G-JOIN*, which shows the best behavior among all four heuristics.

The three Figures 4 - 6 plot the behavior of the heuristics against the number of nodes N for three values of the number of multicast groups, $G = 10, 20, 30$ ($C = 10$ in all cases). We note that the behavior of our heuristics is very similar in all cases, and that the difference from the lower bound ranges from 2% to 45%. We also note that *R-JOIN*, *R-SPLIT*, and *G-SPLIT* appear to perform identically for large values of N, while *G-JOIN* emerges as the clear winner, although not by a large margin. Similar observations can be drawn from Figure 7 where we keep the number of nodes and the number of multicast groups constant ($N = 100, G = 50$) and vary the number of channels.

In all our results so far, we have considered the situation where all receivers are equally likely to belong to a multicast group. To study how the existence of hot spot receivers affects our heuristics, in Figure 8 we plot the difference from the lower bound against N for $C = 10, G = 10$. Comparing the results to Figure 4 we see that the behavior is similar. Finally, in Figure 9 we again plot a hot spot case with $C = 10, G = 10$, but this time the traffic matrix A is generated as follows. For each multicast group g, only 2 channels (chosen randomly for each group) have packets to transmit to g. The number of packets is chosen uniformly from (1,104). In other words, each multicast group only receives traffic from a small number of channels, while in previous traffic matrices each multicast group would receive traffic from almost all channels. Comparing Figure 9 to Figures 4 and 8, however, reveals no significant differences in the behavior of the four heuristics.

Overall, our results indicate that the four heuristics can obtain virtual receiver sets with values close to the lower bound for a wide range of system and traffic parameters, and receiver characteristics. In all cases, *G-JOIN* has shown the best performance among the four heuristics, although the performance of the other three heuristics is not significantly different. Therefore, for systems with a large N over C ratio, the simplest and fastest *R-JOIN* heuristic ($\mathcal{O}(CN)$ complexity) may be the one that provides the best tradeoff between speed and quality of the final solution.

7 Concluding Remarks

We considered the problem of scheduling multicast packet transmissions in a broadcast WDM network with tunability provided at the receiving end only, and with non-negligible receiver tuning latencies. We introduced the concept of a virtual receiver as a set of physical receivers that behave identically in terms of tuning. The traffic demands between any source-virtual receiver pair is equal to the sum of the traffic originating at the source and destined to any of the multicast groups with members in the virtual receiver. A partition of the set of physical receivers into virtual receivers transforms our original network into one with the same number of transmitters but a smaller number of receivers, and unicast traffic only. Any of a number of existing algorithms can then be employed to schedule the packets transmissions in a way that hides the effects of tuning latency. We then studied the problem of optimally partitioning the set of physical receivers into virtual receivers. We proved that this problem is $\mathcal{NP}$-complete, and we showed that channel uti-

given, matrix A is completely specified. We have decided to generate matrix A directly to reduce the number of parameters to be controlled.

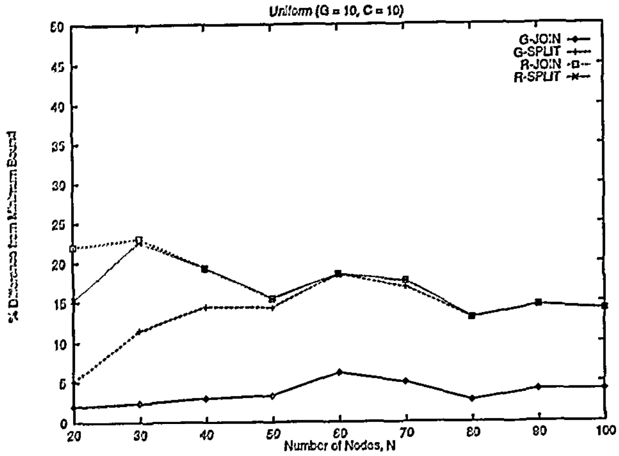

Figure 4: Heuristic comparison for $C = 10$ channels, $G = 10$ (uniform case)

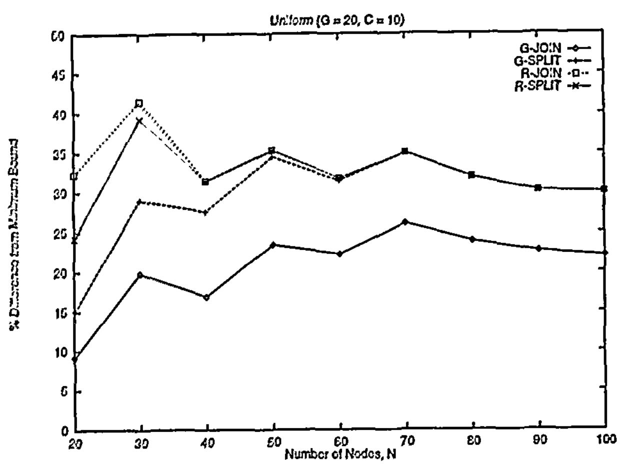

Figure 5: Heuristic comparison for $C = 10$ channels, $G = 20$ (uniform case)

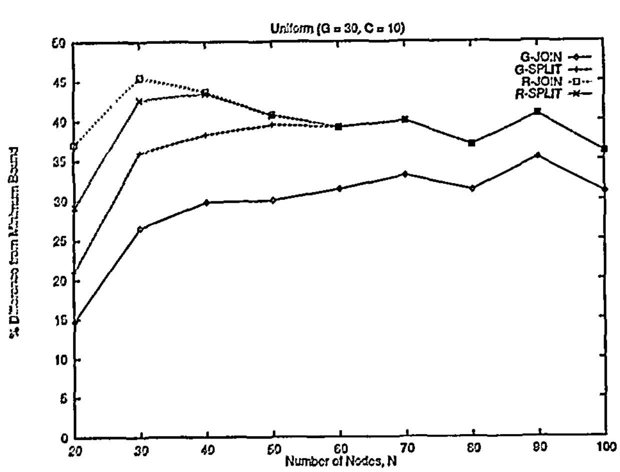

Figure 6: Heuristic comparison for $C = 10$ channels, $G = 30$ (uniform case)

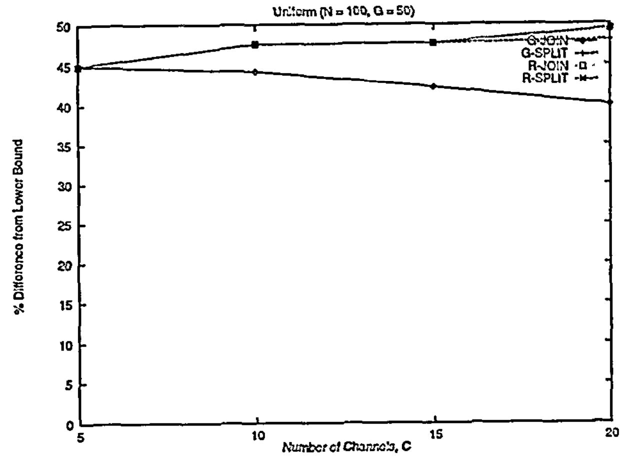

Figure 7: Heuristic comparison for $N = 100$ channels, $G = 50$ (uniform case)

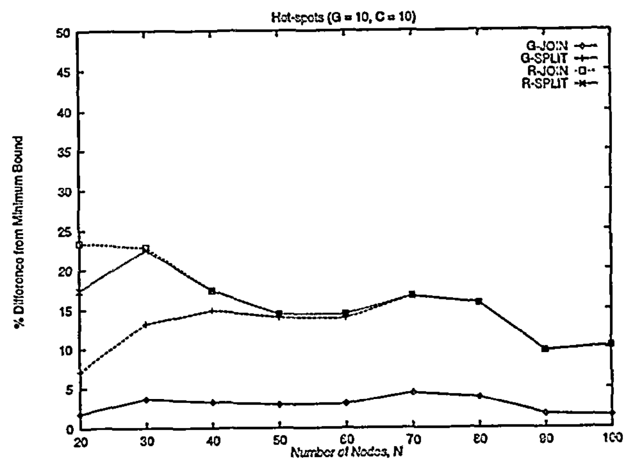

Figure 8: Heuristic comparison for $C = 10$ channels, $G = 10$ (hot-spot case)

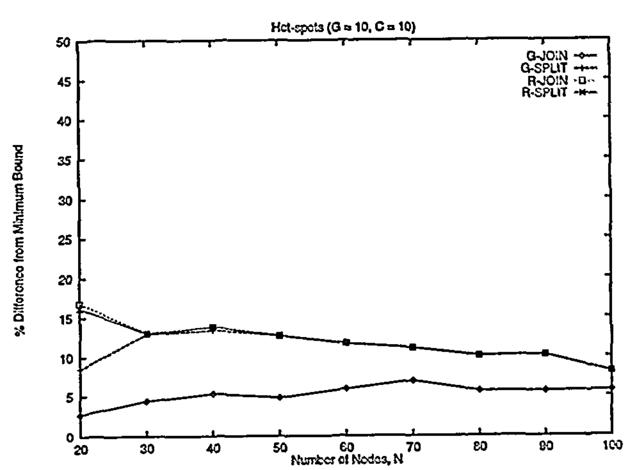

Figure 9: Heuristic comparison for $C = 10$ channels, $G = 10$ (hot-spot case, new traffic matrix)

lization and bandwidth consumption arise as two conflicting objectives in the selection of a virtual receiver set. We also developed a number of heuristics which exhibit good average performance.

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. Azizoglu, R. A. Barry, and A. Mokhtar. Impact of tuning delay on the performance of banwidth-limited optical broadcast networks with uniform traffic. *IEEE Journal on Selected Areas in Communications*, 14(5):935–944, June 1996.

[3] 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.

[4] M. S. Borella and 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*, 14(5):923–934, June 1996.

[5] M. R. Garey and D. S. Johnson. *Computers and Intractability.* W. H. Freeman and Co., New York, 1979.

[6] P. E. Green. Optical networking update. *IEEE Journal Selected Areas in Communications*, 14(5):764–779, June 1996.

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

[8] Z. Ortiz, G. N. Rouskas, and H. G. Perros. Scheduling of multicast traffic in tunable-receiver wdm networks with non-negligible tuning latencies. Technical Report TR-97-01, North Carolina State University, Raleigh, NC, 1997.

[9] 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.

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

[11] 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.

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