Communication Networks

Maximizing Multicast Throughput in WDM Networks with Tuning Latencies Using the Virtual Receiver Concept[*]

ZEYDY ORTIZ, GEORGE N. ROUSKAS, HARRY G. PERROS
Department of Computer Science, North Carolina State University, Raleigh, NC 27695-7534, USA

Abstract. We consider the problem of supporting multipoint communication at the MAC layer of broadcast WDM networks. We first introduce the *multicast throughput* as the performance measure of interest and we show that it depends on two measures that have previously been considered in isolation, namely, the efficient use of channel bandwidth and wavelength throughput. 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 demonstrate that the number of virtual receivers naturally captures the performance of the system in terms of multicast throughput. Consequently, we focus on the problem of optimally selecting the virtual receivers to maximize multicast throughput, and we prove that it is $\mathcal{NP}$-complete. Finally, we present four heuristics of varying degree of complexity for obtaining a set of virtual receivers that provide near-optimal performance in terms of multicast throughput.

1 INTRODUCTION

Many applications and telecommunication services in future high-speed networks will require some form of multipoint communication [1]. 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, 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 multiwavelength optical 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 [9, 3] such that a *single* transmission of a multicast packet can reach all receivers in the packet's destination set simultaneously. Its efficient use of network resources makes 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. On the other hand, waiting until all receivers become available before scheduling a multicast packet results in low wavelength throughput, especially for medium to large size multicast groups. To improve the situation, it was proposed in [5] to partition a multicast group into several sub-groups, and to transmit a packet once to each sub-group. This approach leads to higher wavelength throughput despite the fact that each packet is transmitted multiple times, indicating the existence of a tradeoff between wavelength throughput and the degree of efficiency in using the bandwidth. Also, [6] presents an approach similar to the one in [5] in that a packet is also transmitted multiple times, until all members of its multicast group receive the packet. Instead of partitioning the multicast group in advance, however, each receiver follows a set of rules to listen to a packet transmission in each slot.

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

[*] This work was supported by the National Physical Science Consortium, the National Security Agency, and the NSF under grant NCR-9701113. An earlier version of this paper was presented at ACM SIGCOMM '97.

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 the scheduling algorithms in [10] that have been shown to work well under non-negligible tuning latencies. We also introduce the *multicast throughput* as a performance measure that naturally captures the tradeoff between two conflicting objectives: the efficient use of channel bandwidth and wavelength throughput. Hence, our main focus is to select a partition of physical receivers into virtual receivers so as to maximize multicast throughput.

In section 2 we introduce the concepts of multicast throughput and 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 some 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: $X_c = \{i : \lambda(i) = \lambda_c\}$. We also 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.

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 $\mid g \mid$ to denote the cardinality of group g.

Let G represent the number of currently active multicast groups, and $\mathbf{M} = [m_{ig}]$ be a $N \times G$ multicast traffic demand matrix, where m_{ig} is the number of multicast packets originating at source i and destined to multicast group g. 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 $\mathbf{A} = [a_{cg}]$, where a_{cg} is the total amount of traffic to multicast group g carried by channel λ_c: $a_{cg} = \sum_{i \in X_c} m_{ig} \ \forall \ c, g$. We also let M denote the total traffic demand over all channels and groups: $M = \sum_{i=1}^{N} \ \sum_g m_{ig} = \sum_{c=1}^{C} \ \sum_g a_{cg}$.

We define the *wavelength throughput* $S, S \leq C$ of the network as the average number of packet transmitted on the C channels per unit of time (slot). We note, however, that while high wavelength throughput is certainly desirable, this traditional definition of throughput does not accurately reflect the performance of a network with multicast traffic, as it fails to capture the *degree of efficiency* in the use of channel bandwidth. A measure of this efficiency is the average number $\bar{l}$ of times a packet is transmitted before it is received by all members of its multicast group. Thus, both S and $\bar{l}$ are important in characterizing the performance of the network. For example, a system that can achieve high wavelength throughput only by unnecessarily replicating each multicast packet (resulting in a high $\bar{l}$ value) may actually be inferior to one with a somewhat lower wavelength throughput but which is very efficient in how it transmits packets (i.e., it achieves a very low value for $\bar{l}$).

Let a *multicast completion* denote the completion of a multicast transmission of a packet to all receivers in its multicast group. We define the *multicast throughput* D of the system as the average number of multicast completions per slot. This definition of throughput is independent of how multicast is actually performed (i.e., by performing a single or multiple transmissions), and thus is applicable to any network with multicast traffic. The multicast throughput is related to wavelength throughput and the degree of efficiency through the expression: $D = S/\bar{l}$. As we can see, the multicast throughput D combines both parameters S and $\bar{l}$ in a meaningful way, and it naturally arises as the performance measure of interest in a WDM network with multicast traffic. While previous work has considered either wavelength throughput or the degree of efficiency as the objective in the design of scheduling algorithms for multicast traffic, in this paper we are interested in techniques to maximize the multicast throughput D.

2.1 THE THROUGHPUT-EFFICIENCY TRADEOFF

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 is to separately transmit a copy of a packet to each of the packet's destinations. This solution can achieve high wavelength throughput since a number of transmissions may take place simultaneously on different channels. Its main drawback is the inefficient use of channel bandwidth, since all packets to a multicast group g must be transmitted exactly $\mid g \mid$ times. Let $\bar{g}$ denote the average multicast group size. Then, even if the wavelength throughput S is equal to the number of channels C (the maximum possible), the multicast throughput is only $C/\bar{g}$. Thus, this approach is inefficient, even for relatively small multicast group sizes.

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 makes very efficient use of channel bandwidth, since only a single copy of each packet needs to be transmitted, therefore multicast throughput is equal to the achievable wavelength throughput. However, transmissions to multicast groups with at least one receiver in common cannot be scheduled simultaneously. 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. Again, therefore, multicast throughput can be very low.

As we can see, wavelength throughput and efficient use of bandwidth 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 sub-groups, and to transmit packets in multiple phases [5]. 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 wavelength throughput and efficient use of bandwidth, leading to high multicast throughput. Our work differs from previous research in two novel ways. First, instead of attempting to partition the multicast group of each packet into sub-groups (a shortsighted approach, since it considers each packet independently of others), our objective is to partition the receiver set $\mathcal{N}$ by taking into account the overall traffic offered to the network. Since our solution is obtained by considering the total traffic demands, it is expected to achieve better performance than if each packet were considered in isolation. Second, our approach 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, we can take advantage of the scheduling algorithms in [10] that have provingly optimal properties and have been shown to successfully hide the effects of tuning latency. Since we do not require the development of similar algorithms for scheduling multicast traffic (a rather difficult task [5]), we can therefore concentrate on the important problem of tradeoff selection between the two conflicting objectives, as described in the next subsection.

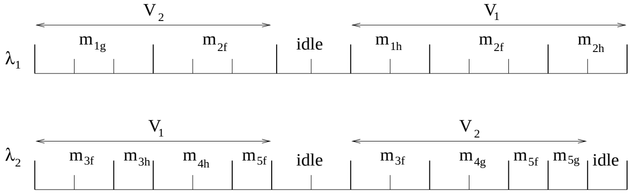

Figure 1: *Example schedule for a network with $N = 5$, $C = 2$, $\Delta = 2$, and 2 virtual receivers.*

2.2 The virtual receiver model

We define a *virtual* receiver $V \subseteq \mathcal{N}$ 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. 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. While 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) transmits 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 $V_l^{(k)}$ 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 a schedule for $N = 5$ nodes, $C = 2$ channels, three multicast groups f, g, and h, $\Delta = 2$, and the following parameters.

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

In this case the wavelength throughput is $S = 25/15 = 1.67$ since 25 packets are transmitted over 15 slots, but the multicast throughput is $D = 19/15 = 1.27$, since exactly 19 multicast transmissions, corresponding to the 19 packets of matrix **M**, are completed within 15 slots.

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 $b_{il} = \sum_{g: g \cap V_l^{(k)} \neq \phi} m_{ig}$. 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 and wavelength throughput is very low. 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, resulting in higher wavelength throughput. 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, resulting in less efficient use of bandwidth. 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 wavelength throughput and efficient use of bandwidth.

As we can see, multicast throughput is maximized only when the virtual receiver sets are selected so as to minimize the length of time to clear the traffic demands represented by traffic matrix **M**. Let's say that such a virtual receiver set and the corresponding scheduled has been obtained. This schedule achieves the multicast packet completions denoted by matrix **M** in the shortest time possible, therefore, by definition, multicast throughput is maximized. Furthermore, a short schedule length implies low delay for delivering the multicast packets. Thus, the objective of our work is to select k and the virtual receivers so as to minimize the schedule length for a given matrix **M**.

We also note that, traffic demands may vary over time, resulting in a different matrix **M**, for which the current receiver set is suboptimal. However, in this paper we assume that matrix **M** represents the long-term traffic in the network, and that changes in this traffic take place over longer time scales. Although we expect that the instantaneous traffic offered to the network will vary, we assume that the network will operate with the same virtual receiver sets until the long-term behavior of the traffic changes. At that time, a new set of virtual receivers may be computed (using the techniques developed here) and communicated to the nodes (using the techniques described in [2]), effectively reconfiguring the network to adapt to the new traffic pattern. Therefore, this work is only concerned with the problem of obtaining the virtual receiver sets for a given matrix **M**.

3 LOWER BOUNDS ON 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:

$$\hat{F}_{ch}(\mathcal{V}^{(k)}) \quad = \quad \max_{c=1,\cdots,C} \left\{ \sum_{l=1}^{k} \sum_{g:g\cap V_l^{(k)} \neq \phi} a_{cg} \right\} \qquad (2)$$

We can obtain a different lower bound by adopting a virtual receiver's point of view. Let $T_l,\ 1 \leq T_l \leq C$, 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:

$$\hat{F}_r(\mathcal{V}^{(k)}) = \max_{l=1,\cdots,k} \left\{ \left[\sum_{c=1}^{C} \sum_{g:g\cap V_l^{(k)} \neq \phi} a_{cg} \right] + T_l \Delta \right\} \quad (3)$$

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:

$$\hat{F}(\mathcal{V}^{(k)}) \quad = \quad \max\left\{ \hat{F}_{ch}(\mathcal{V}^{(k)}), \hat{F}_r(\mathcal{V}^{(k)}) \right\} \qquad (4)$$

To gain some insight into how the number k of virtual receivers may affect the relative values of the two bounds in (2) and (3), let us consider the two extreme scenarios, $k = 1$ and $k = N$. For $k = 1$, there is only one virtual receiver, $\mathcal{N}$, which includes all physical receivers, and we can rewrite (2) and (3) as follows:

$$\frac{M}{C} \leq \hat{F}_{ch}(\mathcal{V}^{(1)}) = \max_{c=1,\cdots,C} \left\{ \sum_{g} a_{cg} \right\} < M \qquad (5)$$

$$\hat{F}_r(\mathcal{V}^{(1)}) = \left\{ \sum_{c=1}^{C} \sum_{g} a_{cg} \right\} + C\Delta = M + C\Delta \qquad (6)$$

In (5) we have assumed that no single channel will carry all traffic, and thus the channel bound will be strictly less than M, while in (6) 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 (2) and (3) become:

$$\hat{F}_{ch}(\mathcal{V}^{(N)}) = \max_{c=1,\cdots,C}\left\{\sum_{g} |\ g\ |\ \ a_{cg}\right\} \tag{7}$$

$$\hat{F}_{r}(\mathcal{V}^{(N)}) = \max_{l=1,\cdots,N}\left\{\left[\sum_{c=1}^{C}\ \sum_{g:l\in g} a_{cg}\right]\ +\ T_l\Delta\right\} \tag{8}$$

It is not clear from (7) and (8) which bound dominates in this case. The channel bound in (7) 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 (7) 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:*

$$\hat{F} \quad = \quad \max\left\{\hat{F}_r(\mathcal{V}^{(N)}), \hat{F}_{ch}(\mathcal{V}^{(1)})\right\} \tag{9}$$

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 (5). 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 (8). **QED.**

3.1 MONOTONICITY PROPERTIES

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, while 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.

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 one discussed above when two special operations are applied to virtual receiver sets. The two operations are:

- *JOIN($\mathcal{V}^{(k)}, n$),* $1 \le n < k \le N$. *JOIN* creates a $(k \Leftrightarrow n)$-virtual receiver set by replacing any $n+1$ of the virtual receivers in $\mathcal{V}^{(k)}$ with their union, and keeping the other $k \Leftrightarrow n \Leftrightarrow 1$ virtual receivers the same.
- *SPLIT($\mathcal{V}^{(k)}, n$),* $1 \le k < k+n \le N$. *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 $k \Leftrightarrow 1$ 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 $\mathcal{V}^{(k-n)}, 1 \le n < k$, be the $(k \Leftrightarrow n)$-virtual receiver set obtained by applying the* JOIN($\mathcal{V}^{(k)}, n$)*, $1 \le n < k \le N$, operation. Then,*

$$\hat{F}_{ch}(\mathcal{V}^{(k-n)}) \le \hat{F}_{ch}(\mathcal{V}^{(k)})\ ;\ \hat{F}_{r}(\mathcal{V}^{(k-n)}) \ge \hat{F}_{r}(\mathcal{V}^{(k)}) \tag{10}$$

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 $(k \Leftrightarrow n)$-virtual receiver set is formed by taking the union of the last $n+1$ virtual receivers of $\mathcal{V}^{(k)}$. Hence, we have that $V_1^{(k-n)} = V_1^{(k)},\cdots,V_{k-n-1}^{(k-n)} = V_{k-n-1}^{(k)}, V_{k-n}^{(k-n)} = V_{k-n}^{(k)} \cup \cdots \cup V_k^{(k)}$. Then, the relative values of the channel and receiver bounds for the k- and $(k \Leftrightarrow n)$-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 (3). By construction, the value of the term within the brackets in (3) 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 (10) holds. For the first inequality in (10), 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 (2)). Therefore, the first inequality in (10) also holds true. **QED.**

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. This behavior is illustrated in figure 3. 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,*

$$\hat{F}_{ch}(\mathcal{V}^{(k+n)}) \geq \hat{F}_{ch}(\mathcal{V}^{(k)}); \hat{F}_r(\mathcal{V}^{(k+n)}) \leq \hat{F}_r(\mathcal{V}^{(k)}) \quad (11)$$

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, since such a schedule would maximize multicast throughput. Unfortunately, given a virtual receiver set, the length of the corresponding schedule is not known until after we run the algorithms in [10]. Thus, we will instead seek a virtual receiver set that minimizes the lower bound in (4), 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 (4) $\hat{F}(\mathcal{V}^{(k)}) \leq F$?*

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 (4) $\hat{F}(\mathcal{V}^{(2)}) \leq F$?*

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

Proof. By transformation from the *PARTITION* problem [4]; for details the reader is referred to [8]. **QED.**

Since *VRSP* is $\mathcal{NP}$-complete, we expect the problem of determining a schedule that minimizes the actual schedule length (rather than the lower bound) to be $\mathcal{NP}$-hard.

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. The first case 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 $M = \sum_{i=1}^{N} m_i$. Then, the two bounds (2) and (3) can be rewritten as

$$\hat{F}_{ch}(\mathcal{V}^{(k)}) \;=\; k \max_{c=1,\cdots,C} \left\{ \sum_{i \in X_c} m_i \right\} \quad (12)$$

$$\hat{F}_r(\mathcal{V}^{(k)}) \;=\; \sum_{i=1}^{N} m_i \,+\, C\Delta \;=\; M \,+\, C\Delta \quad (13)$$

The bounds are independent of the actual virtual receiver sets, and 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 (12) is equal to $k\frac{M}{C} + \epsilon$. Since the receiver bound is independent of k, the overall lower bound is minimized when $\hat{F}_{ch}(\mathcal{V}^{(k)}) \leq \hat{F}_r(\mathcal{V}^{(k)})$, or equivalently, if

$$k\frac{M}{C} + \epsilon \leq M + C\Delta \Leftrightarrow k \leq C + \frac{C^2\Delta \Leftrightarrow C\epsilon}{M} \quad (14)$$

Let us now consider the case when there are $G < N$ disjoint multicast groups $g_1, \cdots, g_G$. 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*. Our heuristics 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.

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 (7) to be greater than the receiver bound in (8). 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 (10) of the *JOIN* operation, we expect the channel (receiver) bound to decrease (increase) after each *JOIN*, yielding a virtual receiver set with a lower

Greedy JOIN (G-JOIN) **Heuristic**
Input: $N, C, X_c,\ c = 1, \cdots, C, G$ multicast groups, and multicast traffic matrix **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 (7) and (8),
        // we expect that $\hat{F}_{ch} \geq \hat{F}_r$ at this step
6.      while $\hat{F}_{ch} > \hat{F}_r$ do
7.          Set $k = k \Leftrightarrow 1$
8.          Select two virtual receivers in $\mathcal{V}^{(k+1)}$ using
            the greedy rule described in Section 5
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.

overall bound. When the virtual receiver set is $\mathcal{V}^{(k)}$, we select 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 (3) 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 (2) for $\mathcal{V}^{(k-1)}$. If again there is a tie, then break it arbitrarily.

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.

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 (4).*

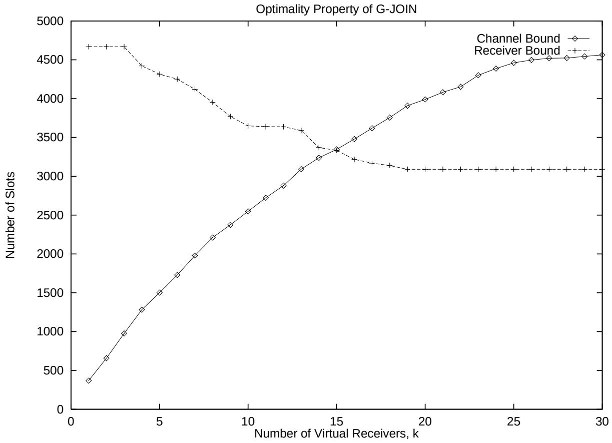

Figure 3: *Optimality property of* G-JOIN *(sample network with* $N = 30, C = 15, G = 30$*).*

Proof. We first observe that, because of (5) and (6), 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 $k^\star \geq 1$ 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:

$$\hat{F}_{ch}(\mathcal{V}^{(N)}) \geq \cdots \geq \hat{F}_{ch}(\mathcal{V}^{(k^\star+1)}) \geq \hat{F}_{ch}(\mathcal{V}^{(k^\star)}) \quad (15)$$

and

$$\hat{F}_r(\mathcal{V}^{(N)}) \leq \cdots \leq \hat{F}_r(\mathcal{V}^{(k^\star+1)}) \leq \hat{F}_r(\mathcal{V}^{(k^\star)}) \quad (16)$$

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

$$\hat{F}_{ch}(\mathcal{V}^{(k^\star+1)}) > \hat{F}_r(\mathcal{V}^{(k^\star+1)}); \hat{F}_{ch}(\mathcal{V}^{(k^\star)}) \leq \hat{F}_r(\mathcal{V}^{(k^\star)}) \quad (17)$$

From (15) – (17) it immediately follows that (a) the overall lower bound of $\mathcal{V}^{(k^\star+1)}$ is minimum among virtual receiver sets $\mathcal{V}^{(N)}, \cdots, \mathcal{V}^{(k^\star+1)}$, since the channel bound decreases from $\hat{F}_{ch}(\mathcal{V}^{(N)})$ to $\hat{F}_{ch}(\mathcal{V}^{(k^\star+1)})$ and the receiver bound increases from $\hat{F}_r(\mathcal{V}^{(N)})$ to $\hat{F}_r(\mathcal{V}^{(k^\star+1)})$, but the latter is not greater than $\hat{F}_{ch}(\mathcal{V}^{(k^\star+1)})$, and (b) any virtual receiver set obtained from $\mathcal{V}^{(k^\star)}$ will not have a smaller overall lower bound since $\hat{F}_r(\mathcal{V}^{(k^\star)}) \geq \hat{F}_{ch}(\mathcal{V}^{(k^\star)})$ 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 $\mathcal{V}^{(k^\star+1)}$ and $\mathcal{V}^{(k^\star)}$. **QED.**

Figure 3 illustrates the optimality property of *G-JOIN*, as well as the monotonicity property of the *JOIN* operation for a sample network with $N = 30$ nodes. Starting with $k = N$ virtual receivers, we applied a sequence of *JOIN* operations as dictated by the *G-JOIN* heuristic in figure 2, and we plotted the receiver and channel bounds of the resulting k-virtual receiver sets, $k = N, \cdots, 1$, in figure 3. The monotonic behavior of the two bounds, derived in section 3.1, is obvious from this figure. We also note that the overall bound decreases as k decreases from N to 15, and it increases thereafter, hence the 15-virtual receiver set is the one that achieves the minimum bound among all the virtual receiver obtained through the *JOIN* operations. (The *G-JOIN* heuristic would have stopped after constructing the virtual receiver set with $k = 14$, since the overall bound cannot be improved any further; in figure 3 we show the virtual receiver sets for all possible values of k to illustrate the monotonicity properties of the two bounds.)

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. An optimality property similar to the one in Lemma 5.1 also holds for *R-JOIN*.

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 (6) is greater than the channel bound (5) 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 (3) 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 (3) for the virtual receiver set of i and j as 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 an optimality property similar to the one in Lemma 5.1.

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 (3) is maximum over all virtual receivers in $\mathcal{V}^{(k)}$ with cardinality greater than one. A random integer between 1 and $n \Leftrightarrow 1$ 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 5.1 also holds for *R-SPLIT*.

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 (9) be the lower bound on an instance of *VRSP*, and let $\hat{F}(\mathcal{V}^{(k)})$ be the lower bound in (4) 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 a large number of 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** and random multicast groups, as follows. 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 $N \Leftrightarrow 5$ 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 4 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 (9); the optimal was obtained through a branch-and-bound technique for *VRSP* (see [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 4 appears to be similar to that of the four heuristics.

Regarding the relative performance of the heuristics, the behavior emerging in figure 4 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 5 - 7 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 8 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 9 we plot the difference from the lower bound against N for $C = 10, G = 10$. Comparing the results to figure 5 we see that the behavior is similar.

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

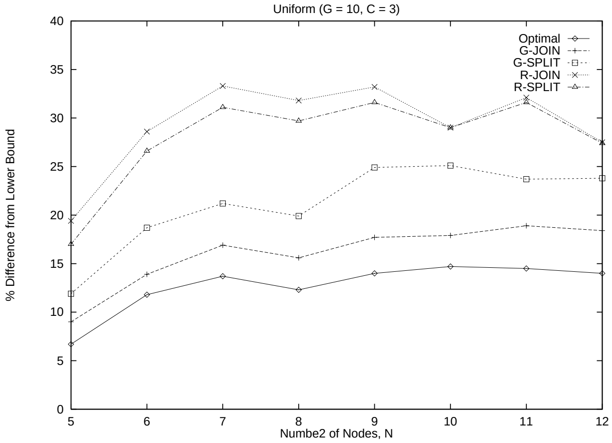

Figure 4: *Heuristic comparison for $C = 3$ channels, $G = 10$ (uniform case).*

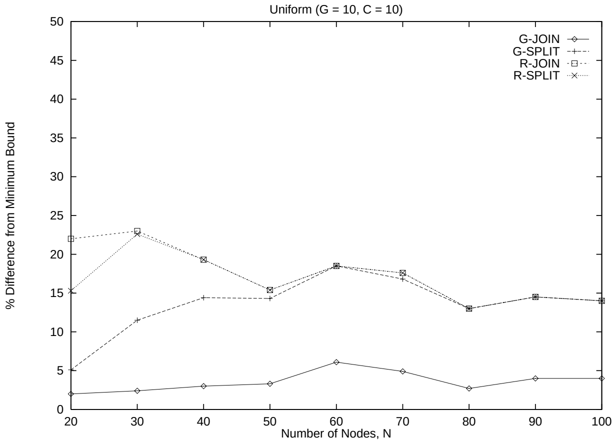

Figure 5: *Heuristic comparison for $C = 10$ channels, $G = 10$ (uniform case).*

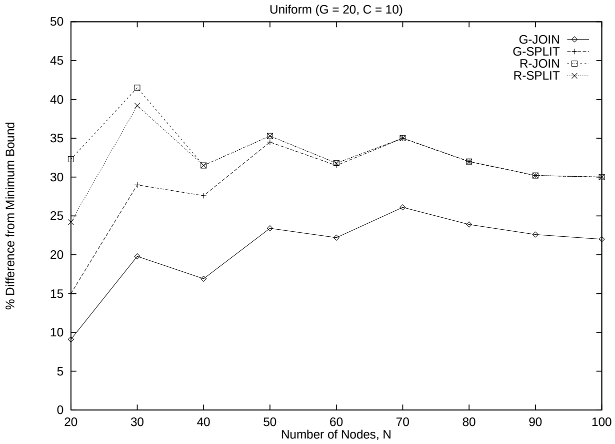

Figure 6: *Heuristic comparison for $C = 10$ channels, $G = 20$ (uniform case).*

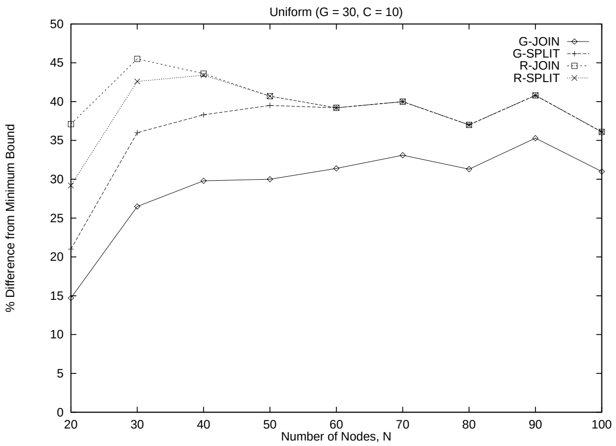

Figure 7: *Heuristic comparison for $C = 10$ channels, $G = 30$ (uniform case).*

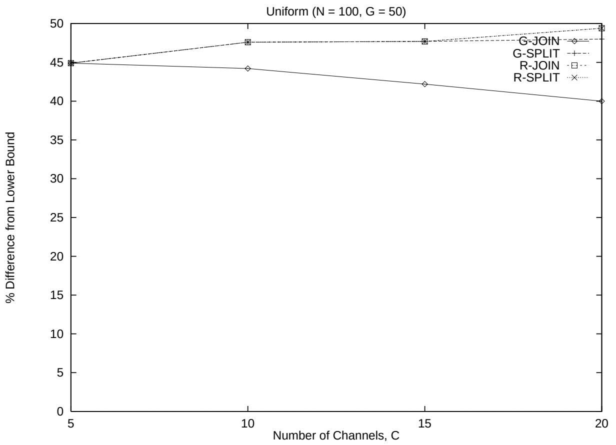

Figure 8: *Heuristic comparison for $N = 100$ nodes, $G = 50$ (uniform case).*

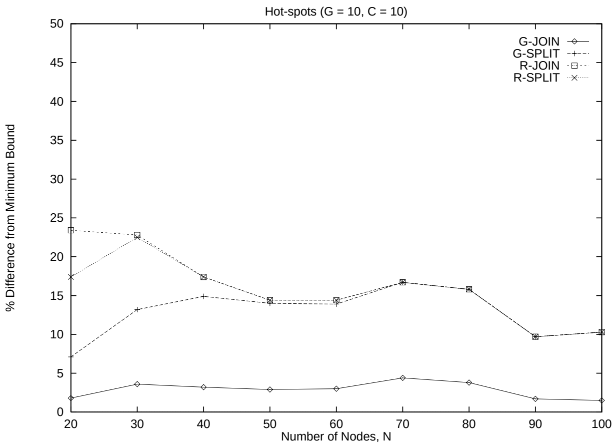

Figure 9: *Heuristic comparison for $C = 10$ channels, $G = 10$ (hot-spot case).*

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 multicast throughput as an important performance parameter in this environment, and we showed how it can capture the tradeoff between wavelength throughput and efficient use of bandwidth in a meaningful way. We also introduced the concept of a virtual receiver as a set of physical receivers that behave identically in terms of tuning. 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 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 J-SAC*, Vol. 15, No. 3, April 1997.

[2] I. Baldine and G. Rouskas. Dynamic reconfiguration policies for WDM networks. In *INFOCOM '99*, pages 313–320, 1999.

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

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

[5] J. Jue and B. Mukherjee. The advantages of partitioning multicast transmissions in a single-hop optical WDM network. In *ICC '97*, 1997.

[6] E. Modiano. Unscheduled multicasts in WDM broadcast-and-select networks. In *INFOCOM '98*, 1998.

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

[8] Z. Ortiz, G. Rouskas, and H. Perros. Scheduling of multicast traffic in tunable-receiver WDM networks with non-negligible tuning latencies. TR-97-01, NCSU, 1997.

[9] G. Rouskas and M. Ammar. Multi-destination communication over tunable-receiver single-hop WDM networks. *IEEE J-SAC*, Vol. 15, No. 3, pages 501–511, April 1997.

[10] G. Rouskas and V. Sivaraman. Packet scheduling in broadcast WDM networks with arbitrary transceiver tuning latencies. *IEEE/ACM Transactions on Networking*, Vol. 5, No. 3, pages 359–370, June 1997.

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