ABSTRACT

ZEYDY ORTIZ-LAUREANO. Techniques to Support Multicast Traffic in Single-Hop WDM Optical Networks. (Under the direction of Professor Harry G. Perros and Professor George N. Rouskas.)

Many applications and telecommunications services in future high-speed networks will require some form of multipoint communication. 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.

We consider the problem of supporting multipoint communication at the media access control (MAC) layer of broadcast-and-select Wavelength Division Multiplexed (WDM) networks. In this environment, bandwidth consumption and channel utilization arise as two conflicting objectives in the design of scheduling algorithms for multicast traffic. We present a new technique for the transmission of multicast packets which is based on the concept of a *virtual* receiver. This is a set of physical receivers which behave identically in terms of tuning. We focus on the problem of optimally selecting the virtual receivers, and prove that it is $\mathcal{NP}$-complete. We then present four heuristics of varying degrees of complexity for obtaining virtual receivers that provide a good balance between the two conflicting objectives.

The dynamic nature of multicast traffic could affect the balance obtained with the virtual receivers when the network conditions change. We study the sensitivity to changes of the virtual receiver sets and the cost associated with handling the changes. Also, the cost of three different approaches to handling the changes is analyzed. Finally, we study the performance of various strategies for scheduling a combined load of unicast and multicast traffic in a broadcast WDM network. Three different scheduling strategies are presented, namely: separate scheduling of unicast and multicast traffic, treating multicast traffic as a number of unicast messages, and treating unicast traffic as multicasts of size one. Performance is measured in terms of schedule length which directly affects both aggregate network throughput and average packet delay.

TECHNIQUES TO SUPPORT MULTICAST TRAFFIC IN SINGLE-HOP WDM OPTICAL NETWORKS

by

Zeydy Ortiz-Laureano

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Computer Science

Raleigh

1998

APPROVED BY:

Co-Chair of Advisory Committee

Co-Chair of Advisory Committee

God blessed me a wonderful family:
my parents, Ismael and Olga,
two sisters, Olga Liz and Katherine,
a loving husband, Robinson,
and now a son, Michael Robinson.
For their love, patience, and encouragement
I affectionately dedicate this work
to my growing family.

BIOGRAPHY

Zeydy Ortiz-Laureano was born in San Juan, Puerto Rico, on September 29, 1970. She graduated with top honors from the University of Puerto Rico, Mayaguez Campus in June 1992 when she completed her Bachelor of Science in Computer Engineering. Zeydy was awarded a National Physical Science Consortium Fellowship to pursue doctoral studies in Computer Science. She received a Master's of Computer Science degree in December 1993 from Texas A&M University. Zeydy joined the Computer Science department at North Carolina State University to complete her studies. Her research interests include High-Speed Networks, Multicasting, and Distributed Systems.

ACKNOWLEDGEMENTS

My graduate educational experience has been greatly enhanced by the help and guidance of my advisors Drs. H. G. Perros and G. N. Rouskas. I appreciate their intellectual exchanges, valuable suggestions, critical reviews, and technical assistance. Their questions, issues, and experience have been most helpful in ensuring the quality of this work. My appreciation also goes to Drs. A. A. Nilsson and A. J. Rindos for their encouragement and professional advice. I want to recognize the help of Scott F. Callicutt for setting up my account in the Multimedia Lab and answering my questions whenever I had a problem.

Finally, I want to acknowledge the research support I have received from the National Physical Science Consortium. Joan Balaris and L. Nan Snow were instrumental in getting through the administrative procedures. I thank the Graduate School and the Computer Science Department at North Carolina State University for participating in the Consortium and providing tuition and fees for my graduate work. The excellent job of the Graduate Fellowship Coordinator, David Shafer, helped me focus on my studies. In addition, I thank the National Security Agency for providing the stipend support and introducing me to the area of Networks and Distributed Systems.

Contents

List of Tables vii

List of Figures viii

1 Introduction 1

2 WDM Networks and Multicasting 5
- 2.1 Properties of Optical Networks . . . 5
- 2.2 Architectures of WDM Optical Networks . . . 6
  - 2.2.1 Broadcast-and-Select and Wavelength-Routing Networks . . . 7
  - 2.2.2 Single-Hop and Multihop Networks . . . 7
  - 2.2.3 $FT^iTT^j - FR^mTR^n$ Networks . . . 8
- 2.3 Challenges in Media Access Control Protocol Design . . . 8
- 2.4 Multicasting . . . 10
- 2.5 Network Support for Multicasting . . . 11

3 System Model 13

4 Related Work 16
- 4.1 Media Access Control protocols for Unicast Traffic . . . 17
- 4.2 Media Access Control protocols for Multicast Traffic . . . 19

5 Support for Multicast Traffic 21
- 5.1 Virtual Receiver Sets . . . 22
- 5.2 Lower Bounds on the Schedule Length . . . 25
  - 5.2.1 Monotonicity Properties of the Lower Bounds . . . 27
- 5.3 The Virtual Receiver Set Problem . . . 29
  - 5.3.1 Special Cases . . . 31
- 5.4 Optimization Heuristics for *VRSP* . . . 33
  - 5.4.1 The *Greedy JOIN (G-JOIN)* Heuristic . . . 33
  - 5.4.2 The *Random JOIN (R-JOIN)* Heuristic . . . 36
  - 5.4.3 The *Greedy SPLIT (G-SPLIT)* Heuristic . . . 36
  - 5.4.4 The *Random SPLIT (R-SPLIT)* Heuristic . . . 37
- 5.5 Numerical Results . . . 38

5.6 Discussion of Results 39

6 Dynamic Multicast Traffic **46**
6.1 Sensitivity Analysis 47
6.2 Changes in Multicast Traffic 50
6.2.1 Changes in Group Composition 50
6.2.2 Bandwidth Changes 52
6.2.3 Effect of Changes on the Lower Bound 53
6.3 Handling Changes in Multicast Traffic 54
6.3.1 Cost Analysis 54
6.3.2 Approach 1: Use Initial Virtual Receiver Assignment 55
6.3.3 Approach 2: Recalculate the Virtual Receiver Assignment 55
6.3.4 Approach 3: Rearrange Virtual Receiver Sets 55
6.4 Numerical Results 56
6.4.1 Discussion of Results 58
6.4.2 Cost Comparison 63
6.5 Concluding Remarks 64

7 Scheduling Combined Unicast and Multicast Traffic **66**
7.1 Transmission Strategies for Combined Unicast and Multicast Traffic 67
7.1.1 Strategy 1: Separate Scheduling 67
7.1.2 Strategy 2: Multicast Traffic Treated as Unicast Traffic 68
7.1.3 Strategy 3: Unicast Traffic Treated as Multicast Traffic 70
7.2 Numerical Results 71
7.2.1 Detailed Comparisons 73
7.2.2 Summary 81
7.3 Concluding Remarks 83

8 Conclusions and Future Research **86**

Bibliography **89**

A A Branch-and-Bound Technique for *VRSP* **93**

B An Alternative Transmission Strategy for Combined Unicast and Multicast Traffic **96**
B.1 Strategy 4: Adding Unicast Traffic to the Virtual Receivers 96
B.2 Numerical Results 98
B.3 Concluding Remarks 100

List of Tables

6.1 Cost Comparison of the Approaches to Handling Multicast Traffic Changes 63

7.1 Behavior of strategies under varying parameters (↑: increase, ↓: decrease, —: no change) 81
7.2 Best strategy when $\bar{g}$ and s are varied 83
7.3 Best strategy when N and G are varied 84
7.4 Best strategy when N and C are varied 84

List of Figures

3.1 Example of a Single Hop Broadcast-and-Select network with $N = 5$ nodes and $C = 2$ channels 14

5.1 Example schedule for a network with $N = 5$, $C = 2$, $\Delta = 2$, and 2 virtual receivers 24
5.2 The *G-JOIN* heuristic for *VRSP* 34
5.3 Optimality property of *G-JOIN* (sample network with $N = 30, C = 15, G = 30$) 36
5.4 Heuristic comparison for $C = 3$ channels, $G = 10$ (uniform case) 39
5.5 Heuristic comparison for $C = 10$ channels, $G = 10$ (uniform case) 40
5.6 Heuristic comparison for $C = 10$ channels, $G = 20$ (uniform case) 40
5.7 Heuristic comparison for $C = 10$ channels, $G = 30$ (uniform case) 41
5.8 Heuristic comparison for $N = 50$ nodes, $G = 25$, and $N = 100$ nodes, $G = 50$ (uniform case) 41
5.9 Heuristic comparison for $C = 10$ channels, $G = 10$ (hot-spot case) 42
5.10 Heuristic comparison for $C = 10$ channels, $G = 10$ (hot-spot case, new traffic matrix) 42

6.1 Definition of $\epsilon^{+}(\cdot)$ for a network with $N = 5$, $C = 2$, $\Delta = 2$, and 2 virtual receivers 50
6.2 Decision tree to determine the type of change to perform with the values of p, q, and r 57
6.3 Comparison of Approaches when Nodes Join Multicast Groups 59
6.4 Comparison of Approaches when Nodes Leave Multicast Groups 59
6.5 Comparison of Approaches when Multicast Groups are Created 60
6.6 Comparison of Approaches when Multicast Groups are Eliminated 60
6.7 Comparison of Approaches when the Group Composition is Changed 61
6.8 Comparison of Approaches when the Bandwidth of a Channel is Changed by 50% 61
6.9 Comparison of Approaches when the Bandwidth of a Group is Changed by 50% 62
6.10 Comparison of Approaches when the Bandwidth of Channels and Groups is Changed by 50% 62

6.11 Comparison of Approaches when All Changes are Performed with Equal Probability 65

7.1 Comparison of strategies for $N = 20$, $G = 30$, $C = 10$, $\Delta = 4$, and $\bar{g} = 0.25N$ (base case) 74
7.2 Comparison of strategies for $N = 20, G = 30, C = 10, \Delta = 4, \bar{g} = 0.10N$ . . . 75
7.3 Comparison of strategies for $N = 20, G = 30, C = 10, \Delta = 4, \bar{g} = 0.50N$ . . . 75
7.4 Comparison of strategies for $N = 20, G = 30, C = 10, \Delta = 1, \bar{g} = 0.25N$ . . . 76
7.5 Comparison of strategies for $N = 20, G = 30, C = 10, \Delta = 16, \bar{g} = 0.25N$ . . 76
7.6 Comparison of strategies for $N = 20, G = 30, C = 5, \Delta = 4, \bar{g} = 0.25N$ . . . 77
7.7 Comparison of strategies for $N = 20, G = 30, C = 15, \Delta = 4, \bar{g} = 0.25N$ . . . 77
7.8 Comparison of strategies for $N = 20, G = 10, C = 10, \Delta = 4, \bar{g} = 0.25N$ . . . 78
7.9 Comparison of strategies for $N = 20, G = 20, C = 10, \Delta = 4, \bar{g} = 0.25N$ . . . 78
7.10 Comparison of strategies for $N = 30, G = 30, C = 10, \Delta = 4, \bar{g} = 0.25N$ . . . 79
7.11 Comparison of strategies for $N = 40, G = 30, C = 10, \Delta = 4, \bar{g} = 0.25N$ . . . 79
7.12 Comparison of strategies for $N = 50, G = 30, C = 10, \Delta = 4, \bar{g} = 0.25N$ . . . 82
7.13 Comparison of strategies with hotspots for unicast traffic ($N = 20, G = 30, C = 10, \Delta = 4, \bar{g} = 0.25N$) 82

A.1 Enumeration of the virtual receiver sets for $N = 4$ 95

B.1 Comparison of Strategy 4 for $N = 20, G = 30, C = 10, \Delta = 4, \bar{g} = 0.25N$ . . 98
B.2 Comparison of Strategy 4 for $N = 20, G = 30, C = 10, \Delta = 4, \bar{g} = 0.10N$ . . 99
B.3 Comparison of Strategy 4 for $N = 20, G = 30, C = 10, \Delta = 4, \bar{g} = 0.50N$ . . 99
B.4 Comparison of Strategy 4 for $N = 20, G = 10, C = 10, \Delta = 4, \bar{g} = 0.25N$ . . 101
B.5 Comparison of Strategy 4 for $N = 20, G = 20, C = 10, \Delta = 4, \bar{g} = 0.25N$ . . 101

Chapter 1

Introduction

While applications drive the development for faster and more efficient network technology, network technology on the other hand opens up the opportunity for the development of new applications. Applications that were not feasible or even imaginable a few years ago are now widely used. The most current example is the development of multimedia applications for the World Wide Web. Web browsers permit us to receive not only text-based information but also audio and video from a wide variety of sources, such as, research institutions, government, businesses, and individuals. These new applications are pushing the limits on current networks, since they require a great amount of bandwidth and have specific quality-of-service requirements.

The increasing demand makes imperative the use of some new technology that is not only capable of meeting today's demands but is also flexible to accommodate tomorrow's growth. The telecommunications industry has already looked at ways to increase their capacity by deploying huge amounts of fiber optic cables. Long-haul routes for the main long distance telephone carriers, AT&T, MCI and Sprint, use fiber optic WDM links to carry the traffic. The data communications industry has also looked for solutions in optical technology. Fiber Distributed Data Interface (FDDI) and distributed-queueing dual bus (DQDB) use fiber to operate at higher data rates than what was possible before their introduction. However, the potential of fiber optic technology has not been exploited because of the limitations imposed by the electronics. Optical networks, and in particular all-optical WDM networks, offer the possibility to tap into the huge bandwidth available in a fiber.

One of the areas that would benefit from the development of faster and more efficient network technologies is the area of group communication in distributed computing.

Applications that make use of group communication include software for collaborative work, distributed data processing, and wide scale information dissemination. Also, telecommunication services in future high-speed networks like video distribution and teleconferencing, would benefit from improved technologies. All of these applications will require some form of multipoint communication [2, 32]. 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 [16], these problems take on new significance, and raise a number of challenging issues that require novel solutions. In this thesis, we focus on the problem of supporting multipoint communication at the media access control (MAC) layer of broadcast-and-select wavelength division multiplexed (WDM) networks [19] when tunable receivers are available.

In multiwavelength optical broadcast-and-select networks, information transmitted on any channel is broadcast to the entire set of nodes. However, 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 [24, 5] 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 [25, 3, 6]. Although a similar algorithm has been developed for multicast traffic [5], the achievable channel utilization can be very low.

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 [25, 3, 6] 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.

Multicast traffic is dynamic in nature. A multicast group is formed, nodes join and leave the group at different times, and finally the multicast group is eliminated from the system when no longer active. Also, the amount of multicast traffic carried by a channel and destined to a multicast group changes with time. All these changes impact the balance achieved with the formation of virtual receiver sets. The network could ignore the effect of the changes by continuing to use the existing virtual receiver set. However, the length of the schedule may be longer than necessary thus increasing delay and decreasing throughput. Recalculating the virtual receiver sets will once again achieve the desired balance but the heuristics are computationally-intensive. We present an approach that reduces the cost of recalculating the virtual receivers for every change in traffic while minimizing the impact of the change on the schedule length.

With new services and uses for technology, a mixed scenario of unicast and multicast traffic is the one more likely to be encountered in practice. The overall objective is still to minimize the length of the schedule produced for the traffic since aggregate network throughput and average packet delay are directly affected by its length. The question then becomes how to treat the unicast and multicast traffic in order to produce the shortest schedule. On one hand, treating multicast traffic as unicast traffic does not take advantage of the broadcast capability of the network and wastes resources. On the other hand, the transmission of unicast traffic to a virtual receiver penalizes the nodes in the virtual receiver that are not the recipients of the traffic. Our goal then is to investigate different strategies that minimize the schedule length.

The organization of the thesis is as follows. In the next chapter, we present background information on optical networks and on multicasting. We discuss the characteristics of optical networks, their classification, and the challenges presented in providing a multiple access scheme. Also, we discuss the uses of multicasting and network support for multicasting. Chapter 3 presents the system model and the notation used in this thesis. Chapter 4 is a survey of related work in this area. The survey includes a summary of the research done in the area of scheduling unicast traffic and multicast traffic. Chapter 5 discusses the problem of supporting multicast traffic. We first discuss different approaches to the problem and introduce the concept of *virtual* receivers as an alternative solution. Lower bounds on the schedule length, and some important properties of the bounds are derived. The problem of optimally selecting a virtual receiver set is formulated and shown to be $\mathcal{NP}$-complete. Heuristics for this problem are presented along with some numerical results.

A branch-and-bound technique developed to prune the search tree used to find the optimal partition of the receivers is presented in Appendix A.

In Chapter 6 we study the effect of changes in multicast traffic on the lower bound of the schedule length. The sensitivity to changes of the virtual receiver sets is first analyzed. Then, we present an analysis of the impact of changes in group composition and changes in bandwidth on the requirements of channels and virtual receivers. Three different approaches to handling the changes are presented along with their associated cost.

Finally, we discuss the performance of various strategies for scheduling a combined load of unicast and multicast traffic in Chapter 7. These strategies are: separate scheduling of unicast and multicast traffic, treating multicast traffic as a number of unicast messages, and treating unicast as multicasts of size one. A lower bound on the schedule length for each strategy is first obtained. Subsequently, the strategies are compared against each other using extensive simulation experiments in order to establish the regions of operation, in terms of number of relevant system parameters, for which each strategy performs best. An alternative strategy is discussed and analyzed in Appendix B.

Multipoint communication support in optical networks has just recently been addressed and there are many issues that remain to be examined. We summarize our contribution to this area in Chapter 8 and identify several directions for future research.

Chapter 2

WDM Networks and Multicasting

Optical networks promise to be a good technological choice for applications that rely on multipoint communication. These applications often require large bandwidth to operate. Optical networks offer the potential of meeting the demands of these applications and, in some cases, decrease the added impact of point-to-multipoint communication. Experimental prototype demonstrations have already shown the feasibility of exploiting the capabilities of optical network technology.

In this chapter, we discuss features of optical networks that provide the means to support multipoint communication. In sections 2.1–2.3 we present an overview of WDM optical networks, the different architectures of the network, and the problems encountered in this environment for multiple access. Sections 2.4 and 2.5 discuss multicasting.

2.1 Properties of Optical Networks

Optical fiber technology offers several advantages over traditional transmission media, such as copper twisted pair and coaxial cable. First, optical fiber can offer a huge bandwidth; approximately 30 Tbps in the low-loss regions centered at 1300 nm and 1500 nm. The error bit rate in those regions is several orders of magnitude lower than that of the other media. In terms of its physical characteristics, optical fiber is smaller in size and lighter in weight. Signals being transmitted on optical fiber are not affected by external electromagnetic fields and do not cause interference when the system has been properly dimensioned. These transport properties, physical properties, and network capabilities of optical networks make them an attractive possibility to meet the demands of current and future traffic. These properties also enable designers to simplify the protocols used for these networks. For instance, given the low error rate, protocols can focus on transmitting the packets through the network and leave error checking mechanisms to higher layers.

2.2 Architectures of WDM Optical Networks

The architectures of optical networks can be classified according to the multiplexing technique used. Several of the multiplexing techniques that have been used in traditional networks have also been proposed for optical networking. These includes Time Division Multiplexing (TDM), Space Division Multiplexing (SDM), and Code Division Multiaccess (CDMA). There are several shortcomings to these techniques including dispersion, distance, and synchronization problems among others [16, 13].

Wavelength Division Multiplexing (WDM) techniques seem to be the preferred choice to fully exploit the potential of fiber optics. WDM divides the optical spectrum into channels. Each channel is characterized by the wavelength (or the frequency) on which the transmissions can be made. For clear transmission, the channels must have some minimum separation. Optical filters must be sensitive enough to be able to distinguish the transmissions in a channel.

WDM architectures for optical networks can be classified as Broadcast-and- Select networks or Wavelength-Routing networks. Alternatively, they can be classified according to the number of hops that the signal must travel in the optical domain. Additionally, the structure of the network interface unit further defines a WDM architecture. Other criteria have also been used [19, 20, 1] to classify the architecture of WDM optical networks. Some of these criteria, however, are not extensively used in the literature nor are they relevant to our thesis and, therefore, they are not included in this discussion. In the following subsections, we discuss the features of some of these WDM architectures.

Finally, we note that different types of architectures of WDM optical networks can be combined into a single network. For example, whilst broadcast-and-select WDM Networks may be suitable for local- and metropolitan-area networks (LANs and MANs), wavelength-routing WDM networks are better suited for wide-area networks (WANs) and they can both be combined in a network. For instance, the architecture proposed by the ARPA sponsored Consortium for Wideband All-Optical Networks [11] is a hierarchical architecture that includes passive broadcast LANs, passive wavelength-routed WANs and configurable wavelength-routed WANs.

2.2.1 Broadcast-and-Select and Wavelength-Routing Networks

Broadcast-and-Select and Wavelength Routing WDM Networks appear to be the most promising architectures for optical networks. If the signal is sent to all the stations in the network, the architecture is said to be broadcast-and-select. Broadcast-and-select optical networks are implemented with a passive star coupler. The star coupler combines the transmissions from every node into a single signal which is sent to all the nodes in the network. The filters at each node extract the signal from the channel to which the station is receiving transmissions. The use of the passive star allows for simplicity in delivery. Depending on the setup of the devices, a Broadcast-and-Select network could provide all-to-all connectivity. The main disadvantage of Broadcast-and-Select Networks is the power loss due to the propagation of the signal to all stations. Broadcast-and-Select WDM networks are suitable for LANs and MANs.

Wavelength Routing Networks are characterized by the use of a distinct wavelength to propagate the signal from point-to-point. They are called wavelength routing because packets may travel through intermediate nodes to reach their destinations. Intermediate nodes may need to process a packet and forward it to another wavelength or they may just act as repeaters. Wavelength-Routing WDM networks are best suited for WANs.

2.2.2 Single-Hop and Multihop Networks

Another criteria used for classification of WDM Networks is the number of hops in the optical domain that the packet traverses. In Single Hop networks, the packet remains in the optical domain from the time it enters the network until it reaches its destination. Note that it is possible that the packets go through intermediate nodes. However, these nodes will not translate the information into the electronic domain in order to forward the packet. These networks are also called All-Optical Networks.

A network in which packets may have to be converted into the electronic domain, processed and then forwarded is called a Multihop Network. Notice that there are Single-Hop and Multihop networks in the Broadcast-and-Select architecture as well as in Wavelength- Routing architecture.

2.2.3 $FT^iTT^j - FR^mTR^n$ Networks

If we wish to transmit information in different wavelengths we must have the ability to discern the signal from each of the wavelengths separately. Recent advances in technology have made it possible to produce devices capable of switching to different wavelengths for transmission or for reception. Because of the time required to tune, some devices have a wider or narrower range of operation. This range of operation will limit the number of usable wavelengths.

The structure of the network interface unit is used to classify an optical network. The notation $FT^iTT^j - FR^mTR^n$ has been adopted to mean that a node in the network has i fixed transmitters, j tunable transmitters, m fixed receiver, and n tunable receivers. If there is only one transmitter or receiver, the superscript is not used. If the node does not have any of them, the term is omitted. For instance, in LAMBDANET the nodes have one fixed transmitter and N fixed receivers and is said to be a $FT - FR^N$ optical network. RAINBOW II[10] employs one fixed transmitter and one tunable receiver per node. This system is called a $FT - TR$ system. Some other proposed systems are classified as $FTTT - FRTR$. These employ the fixed devices for transmission of control information in a specified channel and the tunable devices for data communication.

2.3 Challenges in Media Access Control Protocol Design

In a broadcast-and-select single-hop WDM network, the only way to transmit information successfully is to have both source's transmitter and destination's receiver tuned to the same channel. A very simple way of achieving this is to take the approach of Bellcore's LAMBDANET [15] where a distinct wavelength is assigned to every node in the network and each node has N receivers each, tuned to one of the N distinct wavelengths (where N is the number of nodes). There are many drawbacks to this approach. First of all, the number of wavelengths available in the low-loss region of an optical fiber is limited. Consequently, the system is not scalable. Furthermore, the system will be very expensive to built because of the many transceivers needed for each node.

Another approach that has been implemented in IBM's RAINBOW [9, 10] is to assign each node a home channel to transmit and have tunable receivers moving from channel to channel to determine if they should be listening to the transmissions in a channel. When the destination's receiver gets the connection setup request, it sends an acknowledgment and the connection is established. This approach reduces the number of transceivers per node but it may take a long time for a connection to be established.

The cost, scalability, and efficiency issues of these approaches inspired researchers to study different ways in which the physical medium can be shared efficiently. However, current optical technology imposes three limitations on the efficient use of optical networks. First, at high speed rates, the transmission time of a packet could be less than its propagation delay. This limitation hinders the use of carrier sensing in optical transmissions at high speeds. To deal with this limitation, MAC protocols were developed that provide some pretransmission coordination. By organizing the use of the channels, collisions can be prevented and the medium can be used more efficiently.

The second limitation is due to the number of wavelengths that can be used in the low-loss band. The number of wavelengths is also determined by the technology used in the tunable devices. The fidelity of the devices dictates channel separation and speed, two considerations that restrict the tuning range. The impact of this limitation has been reduced by allowing more than one node to use the same channel for transmission and/or reception. Time Division Multiplexing (TDM) is then used to allow access to the channel.

Finally, there is a high cost associated with switching to other channels. Current tunable devices have a tuning latency which is comparable to the transmission time of a packet at high speed. The effect of tuning latencies can be minimized by appropriate transmission schedules.

To overcome all of these limitations, we must develop MAC protocols that make efficient use of the media. The goal of the protocols is then to coordinate the use of the media such that the characteristics of the network are taken into consideration and problems that could result in unsuccessful transmissions are avoided. We can encounter two different problems in this environment: contention and destination conflict. These problems are discussed next.

Contention

Contention for the medium arises when two or more nodes attempt to transmit on the same wavelength simultaneously. Even when carrier sensing is used, detecting this conflict may be inefficient because the transmission time of a packet may be smaller than its propagation delay. A well designed MAC protocol will schedule transmissions such that each channel is used only by one of the nodes at any given time for transmission.

Destination Conflict

Even if more than one transmitter is prevented from using the same channel at any given time, several transmitters may attempt to transmit to the same destination using different wavelengths. This problem is known as destination conflict. Obviously, the destination node will only be able to receive at most x simultaneous transmissions if it has x receivers tuned to the appropriate channels. Destination conflicts causes the loss of packets. Some arbitration mechanism must be employed to determine who has the right to transmit to the destination at any given time.

Now, suppose that only one transmitter is given permission to transmit to a node in a given channel. If the MAC protocol does not take into consideration the effect of tuning latency, the receiver of the destination node may not be tuned to the appropriate channel in time to receive the transmission. Coordination mechanisms must take into account the tunability requirements in these cases.

2.4 Multicasting

In our daily life we find the need to communicate with more than one person at the same time. Group communication could be achieved if we have point-to-multipoint connections available. Application developers and researchers saw the need to provide tools for group communication. Operating systems like Amoeba [28], and protocols like ISIS [4], Horus [33], and others [8, 12, 23] already provide mechanisms that facilitate the propagation of information from one sender to several destinations. These tools could be used in applications like air traffic control, industrial automation, transaction processing, stock market trading, intelligent highways, medical monitoring, and replicated database systems. Moreover, they are used for providing high service availability, reliability, distributing trust, and for the emerging generation of intelligent network and collaborative work applications.

Initially, most of the support provided was at the application layer. The tools provide primitives that give the user the illusion of simultaneous communication with several other users or applications. Some of the inefficiency of the applications were attributed to the number of messages the underlying communication protocols had to produce in order to support multicasting. Typically, a multicast message was replicated for each of the members of the multicast group. In a broadcast environment, that approach meant that a single message was received by a node in the network many times but it was only accepted when it was directly addressed to the node. Multicast support allowed the transmission of one message to all members. Eventually, the protocols were integrated more with the computer technology used and they were made more efficient. Thus, the load to the network and the performance of the applications improved.

Broadcast-and-Select Single-Hop WDM Networks provide the potential for supporting multicast transmissions efficiently. With the careful coordination of transmitters and receivers in the network, multicast transmissions can be successfully achieved. The next section briefly introduces the various approaches used to support multicasting and how broadcast-and-select optical networks can provide the support.

2.5 Network Support for Multicasting

The problems associated with providing network support to multipoint communications have been widely studied within a number of different networking contexts [32, 34, 2]. In broadcast networks, multipoint communication is supported by sending the message to all destinations and by providing a mechanism to select only those multicast packets of interest to the node. Special addresses are used to identify and select the packets. Switched or routed networks will waste resources if the packets are sent to all destinations. In these environments, multicast packets are replicated along the route to each of the nodes in a multicast group. Multicast trees are built to deliver these packets.

Even though broadcast-and-select single-hop networks provide the capability of broadcasting a packet to all nodes with a single transmission, broadcasting comes with two disadvantages for the nodes not interested in the multicast message. First, a node will spend time tuning to a wavelength which may not carry packets addressed to the node. Second, the time spent receiving the multicast packet could have been better utilized receiving relevant packets.

From the point of view of coordination, broadcasting requires that all nodes are tuned to the same wavelength before transmission can begin. Therefore, broadcasting a multicast message in these environment is not the most efficient way to support multicasting. Message routing and/or building multicast trees cannot be applied to these networks because they provide all-to-all connectivity. Consequently, new techniques and mechanisms must be developed for broadcast-and-select single-hop networks.

Chapter 3

System Model

In this chapter we present the network model and the notation that is used in this thesis. 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: $X_c = \{i : \lambda(i) = \lambda_c\}$. Figure 3.1 shows a Single-Hop Broadcast-and-Select Network where $X_1 = \{1, 2\}$ and $X_2 = \{3, 4, 5\}$.

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 assume that the traffic offered to the network is of two types: unicast and multicast. For multicast traffic 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 [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 $\mathbf{A} = [a_{ig}]$, where a_{ig} is the number of multicast packets originating at source i and destined to multicast group g. Also, there is a $N \times N$ unicast traffic demand matrix $\mathbf{Z} = [z_{ij}]$, where z_{ij} is the number of unicast packets destined to receiver j from source node i. We assume that traffic

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

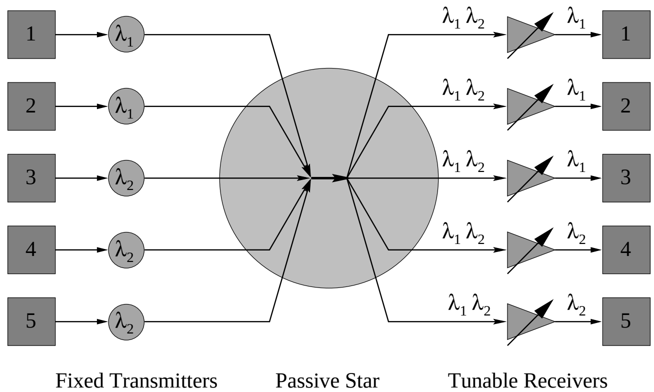

Figure 3.1: Example of a Single Hop Broadcast-and-Select network with $N = 5$ nodes and $C = 2$ channels

matrices $\mathbf{A}$ and $\mathbf{Z}$ are known to all nodes. Information about the traffic demands a_{ig} and z_{ij} may be collected using a distributed reservation protocol such as HiPeR-ℓ [27].

Given the assignment of transmit wavelengths $\{X_c\}$, we construct the $C \times G$ *collapsed* multicast traffic demand matrix $\mathbf{M} = [m_{cg}]$, where m_{cg} is the total amount of traffic to multicast group g carried by channel λ_c:

$$m_{cg} = \sum_{i \in X_c} a_{ig} \quad \forall\ c, g \tag{3.1}$$

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

$$M = \sum_{i=1}^{N} \sum_{g} a_{ig} = \sum_{c=1}^{C} \sum_{g} m_{cg} \tag{3.2}$$

Similarly, we construct the $C \times N$ *collapsed* unicast traffic demand matrix $\mathbf{U} = [u_{cj}]$, where u_{cj} is the total amount of unicast traffic carried by channel λ_c for receiver j:

$$u_{cj} = \sum_{i \in X_c} z_{ij} \quad \forall\ c, j \tag{3.3}$$

Matrices $\mathbf{M}$ and $\mathbf{U}$ will be used in this thesis as starting points to obtain a transmission schedule that specifies when receivers tune to the various channels, and when packet transmissions should take place.

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.

Chapter 4

Related Work

The main focus of this thesis is on multipoint communication in Broadcast-and-Select Single-Hop WDM networks. However, in the next section we present a summary of the key findings on scheduling unicast traffic in optical networks for the following two reasons. First, the problems associated with scheduling are common to both unicast and multicast traffic. Even though the coordination required for multidestination communication is more complex, we still have to prevent collisions and avoid conflicts for successful transmission of packets. Also, our solution to scheduling multicast traffic relies on the solution for scheduling unicast traffic. The technique we present in this thesis transforms multicast traffic into unicast traffic for scheduling purposes. Section 4.2 presents the results of previous research on scheduling multicast traffic.

Earlier proposals for Media Access Control (MAC) protocols for optical networks were based on the assumption of all-to-all communications with uniform traffic and negligible tuning latency. Furthermore, some of the earlier proposals assumed that only the next packet to be transmitted should be scheduled. In the literature review given in the next two sections we only consider system models which are similar to ours. The traffic pattern is non-uniform and the tuning delay is non-negligible (except when noted). Transmissions are scheduled in blocks so that the head-of-line problem is avoided and tuning latencies are better masked. Also, 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. The related work reported on unicast transmissions are based on TT-FR systems but can be easily adapted to FT-TR systems. For multicast transmissions, the focus is on FT-TR systems only.

4.1 Media Access Control protocols for Unicast Traffic

Coordination of transmitters and receivers in all-optical networks is crucial for the successful transmission of packets in the network. As explained in Section 2.3, collisions and destination conflicts arise when access has not been coordinated. Also, the non-negligible tuning latency expected should be taken into consideration. The problem of constructing schedules for transmitting unicast traffic in this network environment has been addressed by Azizoglu, Barry, and Mokhtar [3], Borella and Mukherjee [6], Rouskas and Sivaraman [25, 26], and Pieris and Sasaki [22].

Rouskas and Sivaraman [25] define tight lower bounds for schedule length and present a heuristic that construct near-optimal schedules. The lower bound on the schedule length for unicast traffic is determined by the number of slots required by a node to receive all packets destined for it (including slots for tuning to the different channels) and the number of slots required for a channel to transmit all packets. If a channel requires $z > y$ slots, some transmitter in the channel will not be able to transmit all its packets. Similarly, if the frame has only y slots but a node needs $x > y$ slots, the node will not be able to receive some packets. The bounds are defined using the unicast traffic matrix $\mathbf{U}$ as a starting point.

Since the length of any schedule cannot be smaller than the number of slots required to satisfy all transmissions on any given channel, the *unicast channel* bound is defined as [26]:

$$\hat{H}_{ch} = \max_{c=1,\cdots,C} \left\{ \sum_{j=1}^{N} u_{cj} \right\} \tag{4.1}$$

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

$$\hat{H}_{r} = \max_{j=1,\cdots,N} \left\{ \sum_{c=1}^{C} u_{cj} + T_j \Delta \right\} \tag{4.2}$$

The overall lower bound $\hat{H}$ for clearing matrix $\mathbf{U}$ is:

$$\hat{H} = \max \left\{ \hat{H}_{ch}, \hat{H}_{r} \right\} \tag{4.3}$$

These bounds are used in Chapter 7 to analyze strategies for scheduling unicast and multicast traffic in the network.

The algorithm developed by Rouskas and Sivaraman constructs schedules for unicast traffic of length equal to the lower bound when certain optimality conditions are satisfied. Under different conditions, they developed heuristics that produce schedules of length very close to the lower bound. The heuristics consists of determining the best ordering of the nodes to receive transmissions that will produce a minimum lower bound. Instead of evaluating all possible $N!$ combinations, the heuristics construct the schedules by incorporating the information of the transmissions of one node at a time. The ordering produced is the same used in all channels. This ordering facilitates the coordination of tuning to other channels. In this thesis, we will make extensive use of the algorithms in [26]. For presentation purposes, we introduce the following operation:

$$\mathcal{S} \quad \leftarrow \quad Sched(\mathbf{U}, \Delta) \tag{4.4}$$

The $Sched(\cdot)$ operation takes as arguments a unicast traffic demand matrix $\mathbf{U}$ and the transceiver tuning latency Δ, and it applies the Bandwidth Limited Scheduling Heuristic (*BLSH*) algorithm (for a bandwidth-limited network where $\hat{H} = \hat{H}_{ch} > \hat{H}_r$) or the Tuning Limited Scheduling Heuristic (*TLSH*) algorithm (for a tuning-limited network where $\hat{H} = \hat{H}_r > \hat{H}_{ch}$) presented in [26] to obtain a schedule $\mathcal{S}$ for clearing matrix $\mathbf{U}$. The number of nodes and channels of the network are implicitly defined in the dimensions of matrix $\mathbf{U}$.

Borella and Mukherjee [6] designed a heuristic based on the MULTIFIT algorithm [7] to produce the schedule. Transmissions for the node with greater requirement are scheduled first in each of the channels. The transmissions for the others are then accommodated as possible. The single reservation scheduling proposed in [3] uses a similar idea. The transmitter with the highest number of packets to transmit in λ_1 will do so first. Of the remaining transmitters, the one with the highest number of packets for destinations in λ_2 will start, and so on. The transmitters with the lowest requirements will remain idle until transmissions in a channel are about to finish (Δ slots before finishing). The idle transmitter with the highest requirement will then tune to λ_i, where λ_i is the transmit wavelength of the channel, just in time for its transmissions.

4.2 Media Access Control protocols for Multicast Traffic

Multipoint communication have been studied in different network environments. Recently, it has been considered within the optical networks environment. Several techniques have been proposed in the literature for scheduling unicast traffic. However, only three different techniques have been reported in the literature for scheduling multicast traffic in broadcast-and-select single-hop networks.

In particular, Rouskas and Ammar [24] proposed adaptive multicast protocols for FT-TR networks. The protocol produces a schedule in which each node is given permission to transmit multicast traffic in several slots per frame. The idea is that in the first slot for multicast traffic of node i, all receivers must tune to the transmit wavelength of i, $\lambda(i)$. This first slot is called the synchronization slot and is used, among other things, to identify the multicast group to which the packets are addressed. Nodes belonging to the multicast group will remain tuned to that wavelength while all others can receive unicast messages in other wavelengths. This protocol assumes that the source of multicast traffic does not have knowledge of the membership of the multicast group to which it wishes to communicate. The advantage of this "blind" approach is that it is transparent to membership changes. One of the problems with this approach is the wasted time switching wavelengths ($\Delta+1+\Delta$ slots for nodes that do not belong to the multicast group). In their work, tuning latency is considered to be negligible ($\Delta = 0$). So, the wasted time in this case corresponds to the time spent in the synchronization slot by a node that was not in the multicast group. Also, each station can transmit to one multicast group per frame unless it has non-contiguous blocks for multicast traffic in the frame. If only multicast traffic was offered, this approach does not provide any transmission concurrency.

Borella and Mukherjee [5] proposed a protocol that uses a control-channel to schedule multicast packets. Nodes request transmission time by sending a control packet indicating the destination of the packet at the head of their queues. Upon receipt of this information all nodes run a deterministic distributed algorithm to schedule the transmission of the multicast packet. With the control information, nodes determine the earliest time at which the packet can be received by all the members of the multicast group and the earliest time at which it can be transmitted. The main problem with this approach is that it reserves resources when they are not in use. If a node belongs to the multicast group addressed in the control packet, its receiver must become idle until all nodes in the group have tuned to the appropriate wavelength to receive the packet. This problem leads to poor transmission concurrency and, consequently, low channel utilization.

Note that the strategy in these two proposals is to have all the members of a multicast group tune to the same channel to receive the messages for the group in one transmission. Even though this strategy uses the minimum bandwidth required for transmission, the advantage of having multiple channels in a WDM network is not exploited. We will show later that we can exploit the transmission concurrency provided by the channels and, at the same time, limit the number of transmissions of a message with the technique presented in this thesis.

Jue and Mukherjee [17] recognized the shortcomings of the protocol in [5] and they improved on it by partitioning multicast transmissions. A multicast packet is transmitted according to different scheduling policies: at the time when the first receiver is available, at the time when the last receiver is available or by combining the policies to obtain a better partition. Modiano [18] also incorporated the concept of partitioning transmissions in unscheduled multicasts. A multicast packet is retransmitted multiple times until all the nodes in the multicast group have received the transmission. Their work assume negligible tuning latency and that the number of available wavelengths is equal to the number of nodes in the network. Additionally, the heuristics presented are designed for on-line scheduling. Consequently, none of these schemes can optimize the partition of receivers because the multicast transmissions are scheduled with incomplete information. However, the most significant difference between these proposals and our work is that the partition is done for every new packet. We partition the nodes in the network taking in consideration all the multicast traffic. The overhead of computing the partition is incurred only once in a frame.

Techniques for multihop networks have also been reported in [30, 31] but they are not included here because they are outside of the scope of this thesis.

Chapter 5

Support for Multicast Traffic

A fixed transmitter-tunable receiver (FT-TR) optical network provides the flexibility of allowing multiple destination communications. A group of nodes that need to receive a message from a single source can just tune to the wavelength assigned to the source and receive the message simultaneously. However, with current technology, tuning to a wavelength is costly in terms of time spent to tune. Therefore, it is our goal to produce schedules for FT-TR networks of minimum length that mask the tuning latencies. In one hand it is desirable that the source transmit the message once to all destinations. In the other hand, to mask tuning latencies, it is optimal to have each destination tune to a channel once in a frame.

Two different approaches can be derived from the objectives of the problem: single transmission and multiple transmission. The single transmission approach requires that all destinations tune to the source's wavelength simultaneously to have the multicast message transmitted once. The multiple transmission approach consists of treating multicast messages as unicast messages and replicating the message to all destinations. In this case it is possible to create a schedule in which destinations tune to a wavelength once. The problem with these approaches, however, is that they do not utilize the resources efficiently. The single transmission approach does not allow transmission concurrency. Only one channel is used for transmission at a time and nodes that do not belong to the multicast group addressed are idle. The multiple transmission approach, in the other hand, uses a lot of bandwidth for a message. The focus of this chapter is the introduction of the concept of a *k-virtual receiver set* to provide support for multicasting such that a balance of the two conflicting objectives is achieved.

5.1 Virtual Receiver Sets

In this section we discuss alternatives to delivering multicast traffic in the network. The alternatives attempt to make a balance between channel utilization and bandwidth consumption. We introduce the concept of virtual receivers and k-virtual receiver sets to transform our problem of scheduling multicast traffic to one of finding the best way to partition the nodes to strike the balance.

Recall that there is a $N \times G$ multicast traffic demand matrix $\mathbf{A} = [a_{ig}]$, where a_{ig} is the number of multicast packets originating at source i and destined to multicast group g. We assume that traffic matrix $\mathbf{A}$ is known to all nodes. Given the traffic matrix $\mathbf{A}$, 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 [25, 6, 3]). Its obvious drawback is high bandwidth consumption, since all packets to a multicast group g must be transmitted exactly $\mid g \mid$ 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 [5], and it was found to utilize only one (out of C) channels 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 thesis 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 [25] 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 $\mathbf{M}$, 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 members of which they are not a member, but they are guaranteed to receive the packets for all groups of which they are members.

Figure 5.1 shows an example schedule for the network in Figure 3.1 with $N = 5$ nodes, $C = 2$ channels, three different multicast groups f, g, and h, $\Delta = 2$, and the following parameters.

$$\mathbf{A} = \begin{bmatrix} 0 & 3 & 2 \\ 3 & 0 & 2 \\ 2 & 0 & 1 \\ 0 & 2 & 2 \\ 1 & 1 & 0 \end{bmatrix} \quad ; \quad \begin{array}{l} \text{f=\{2,3,4\}} \\ \text{g=\{1,2\}} \\ \text{h=\{4,5\}} \end{array} \quad ; \quad \begin{array}{l} X_1 = \{1,2\} \\ X_2 = \{3,4,5\} \end{array} \quad ; \quad \begin{array}{l} V_1 = \{4,5\} \\ V_2 = \{1,2,3\} \end{array} \qquad (5.1)$$

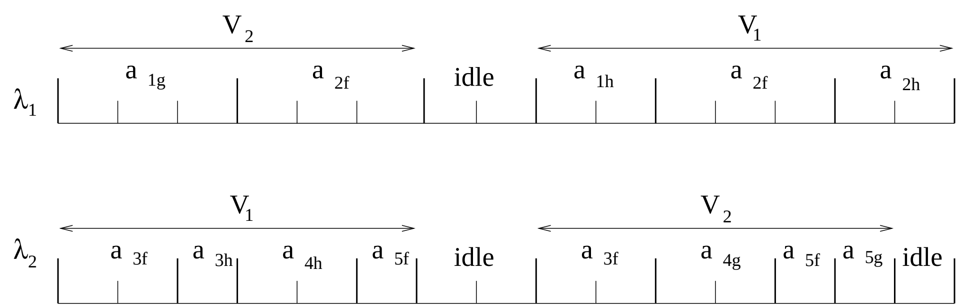

Figure 5.1: Example schedule for a network with $N = 5$, $C = 2$, $\Delta = 2$, and 2 virtual receivers

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 $V_l^{(k)}$. Channel λ_c will then carry the multicast traffic of all nodes in X_c. Let $\mathbf{B}$= $[b_{cl}]$ be the $C \times k$ matrix with

$$b_{cl} = \sum_{g: g \cap V_l^{(k)} \neq \phi} m_{cg} \tag{5.2}$$

Quantity b_{cl} represents the amount of traffic carried by channel λ_c 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 $\mathbf{A}$, to an equivalent network with *unicast* traffic matrix $\mathbf{B}$. This new network has the same number of transmitters and channels, and the same tuning latency as the original one, but only k receivers, corresponding to the k virtual receivers in $\mathcal{V}^{(k)}$. We can then immediately employ the algorithms in [25] (which were developed for unicast traffic) to construct schedules for clearing matrix $\mathbf{B}$ in the new network. Section 4.1 discusses the techniques used for scheduling unicast traffic proposed in previous research. The reader is referred to [25] for details on the optimality properties of these scheduling algorithms. For the rest of this thesis we concentrate on the problem of selecting the virtual receiver set $\mathcal{V}^{(k)}$ to use.

For presentation purposes, we introduce the operation, $VR(\cdot)$, which takes as arguments a multicast traffic $\mathbf{M}$ and the tuning latency Δ, and which applies the *Greedy-JOIN* heuristic that will be discussed in Section 5.4.1 to construct a near-optimal virtual receiver set $\mathcal{V}^{(k^\star)}$ for matrix $\mathbf{M}$:

$$\mathcal{V}^{(k^\star)} \leftarrow VR(\mathbf{M}, \Delta) \tag{5.3}$$

Constructing a schedule for the transmission of multicast traffic matrix $\mathbf{M}$, involves three steps: applying the operation $VR(\mathbf{M}, \Delta)$, determining matrix $\mathbf{B}$ from the resulting virtual receiver set $\mathcal{V}^{(k^\star)}$, and finally applying the $Sched(\mathbf{B}, \Delta)$ operation. We will use $MSched(\mathbf{M}, \Delta)$ to denote this sequence of operations resulting in a schedule $\mathcal{S}$ for $\mathbf{M}$. We have:

$$\mathcal{S} \quad \leftarrow \quad MSched(\mathbf{M}, \Delta) \tag{5.4}$$

When $k = 1$, each multicast packet is transmitted only once, but there is no transmission concurrency; only one channel is utilized at a time. This situation corresponds to the single transmission approach discussed earlier. 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 the 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. In this case, the multiple transmission approach is used. 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.

In the next section we obtain lower bounds on the length of schedules for clearing the multicast matrix $\mathbf{M}$, and we present some properties that quantify the effect of the number k of virtual receivers on these bounds.

5.2 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:

$$\hat{F}_{ch}(\mathcal{V}^{(k)}) \quad = \quad \max_{c=1,\cdots,C} \left\{ \sum_{l=1}^{k} \sum_{g: g \cap V_l^{(k)} \neq \phi} m_{cg} \right\} \tag{5.5}$$

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:

$$\hat{F}_r(\mathcal{V}^{(k)}) \;=\; \max_{l=1,\cdots,k} \left\{ \left[\sum_{c=1}^{C} \sum_{g: g\cap V_l^{(k)} \neq \phi} m_{cg} \right] \;+\; T_l \Delta \right\} \tag{5.6}$$

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:

$$\hat{F}(\mathcal{V}^{(k)}) \;=\; \max \left\{ \hat{F}_{ch}(\mathcal{V}^{(k)}), \hat{F}_r(\mathcal{V}^{(k)}) \right\} \tag{5.7}$$

To gain some insight into how the number k of virtual receivers may affect the relative values of the two bounds in (5.5) and (5.6), 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 (5.5) and (5.6) as follows:

$$\frac{M}{C} \;\leq\; \hat{F}_{ch}(\mathcal{V}^{(1)}) \;=\; \hat{F}_{ch}(\mathcal{N}) \;=\; \max_{c=1,\cdots,C} \left\{ \sum_{g=1}^{G} m_{cg} \right\} \;<\; M \tag{5.8}$$

$$\hat{F}_r(\mathcal{V}^{(1)}) \;=\; \hat{F}_r(\mathcal{N}) \;=\; \left\{ \sum_{c=1}^{C} \sum_{g=1}^{G} m_{cg} \right\} \;+\; C\Delta \;=\; M \;+\; C\Delta \tag{5.9}$$

In (5.8) we have assumed that no single channel will carry all traffic, and thus the channel bound will be strictly less than M, while in (5.9) 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 (5.5) and (5.6) become:

$$\hat{F}_{ch}(\mathcal{V}^{(N)}) \;=\; \hat{F}_{ch}(\{1\}, \cdots, \{N\}) \;=\; \max_{c=1,\cdots,C} \left\{ \sum_{g=1}^{G} \mid g \mid \; m_{cg} \right\} \tag{5.10}$$

$$\hat{F}_r(\mathcal{V}^{(N)}) \;=\; \hat{F}_r(\{1\}, \cdots, \{N\}) \;=\; \max_{l=1,\cdots,N} \left\{ \left[\sum_{c=1}^{C} \sum_{g: l\in g} m_{cg} \right] \;+\; T_l \Delta \right\} \tag{5.11}$$

It is not clear from (5.10) and (5.11) which bound dominates in this case. The channel bound in (5.10) 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 (5.10) 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 $\mathbf{M}$. We note, however, that this absolute lower bound is not necessarily achievable.

Lemma 5.2.1 *Regardless of the method used to transmit multicast packets, a lower bound on the length of any schedule to clear matrix* $\mathbf{M}$*, is given by:*

$$\hat{F} \quad = \quad \max\left\{\hat{F}_r(\mathcal{V}^{(N)}), \hat{F}_{ch}(\mathcal{V}^{(1)})\right\} \tag{5.12}$$

Proof. The length of any schedule for $\mathbf{M}$ 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 (5.8). 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 (5.11).

5.2.1 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 (5.8) – (5.11), we note that the two special cases $k = 1$ and $k = N$ appear to confirm our intuition, since we immediately obtain that

$$\hat{F}_{ch}(\mathcal{V}^{(1)}) \quad \leq \quad \hat{F}_{ch}(\mathcal{V}^{(N)}) \qquad \text{and} \qquad \hat{F}_r(\mathcal{V}^{(N)}) \quad \leq \quad \hat{F}_r(\mathcal{V}^{(1)}) \tag{5.13}$$

In the general case, however, the lower bounds in (5.5) and (5.6) 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 5.2.2 (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*

$$\hat{F}_{ch}(\mathcal{V}^{(k-n)}) \quad \leq \quad \hat{F}_{ch}(\mathcal{V}^{(k)}) \quad \text{and} \quad \hat{F}_{r}(\mathcal{V}^{(k-n)}) \quad \geq \quad \hat{F}_{r}(\mathcal{V}^{(k)}) \tag{5.14}$$

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

$$V_1^{(k-n)} \;=\; V_1^{(k)}, \quad \cdots, \quad V_{k-n-1}^{(k-n)} \;=\; V_{k-n-1}^{(k)}, \quad V_{k-n}^{(k-n)} \;=\; V_{k-n}^{(k)} \cup \cdots \cup V_k^{(k)} \tag{5.15}$$

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 (5.6). By construction, the value of the term within the brackets in (5.6) 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 (5.14) holds. For the first inequality in (5.14), 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 (5.5)). Therefore, the first inequality in (5.14) also holds true.

As a consequence of the monotonicity property of *JOIN*, if we start with the N-virtual receiver set $\mathcal{N}$ 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 5.2.3 is in a sense the inverse of Lemma 5.2.2. Its proof is omitted since it is very similar to that of Lemma 5.2.2.

Lemma 5.2.3 (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*

$$\hat{F}_{ch}(\mathcal{V}^{(k+n)}) \geq \hat{F}_{ch}(\mathcal{V}^{(k)}) \quad \text{and} \quad \hat{F}_r(\mathcal{V}^{(k+n)}) \leq \hat{F}_r(\mathcal{V}^{(k)}) \tag{5.16}$$

5.3 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 $\mathbf{A}$ 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 [25]. 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 [25] 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 (5.7), 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 5.3.1 (VRSP) *Given N nodes, C channels, transmitter sets X_c, tuning latency Δ, G multicast groups, a multicast traffic demand matrix A, 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 (5.7) $\hat{F}(\mathcal{V}^{(k)}) \leq F$?*

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 5.3.2 (2-VRSP) *Given N nodes, C channels, transmitter sets X_c, tuning latency Δ, G multicast groups, a multicast traffic demand matrix A, and a real number F, does there exist a 2-virtual receiver set $\mathcal{V}^{(2)}$ such that the lower bound in (5.7) $\hat{F}(\mathcal{V}^{(2)}) \leq F$?*

Theorem 5.3.1 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 (5.7) is at most F.

We now transform the *PARTITION* problem [14] 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 $\mathbf{A} = [a_{i,g_n}]$ is such that

$$a_{i,g_n} = \frac{w_n}{k}, \quad i, n = 1, \cdots, k \tag{5.17}$$

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 (5.5) becomes

$$\hat{F}_{ch}(\mathcal{V}^{(2)}) = \left\lceil \frac{k}{3} \right\rceil \sum_{n=1}^{k} \frac{w_n}{k} = \left\lceil \frac{k}{3} \right\rceil \frac{W}{k} \leq \frac{W}{2} \tag{5.18}$$

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 (5.6) 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 (5.6) evaluates to:

$$\hat{F}_r(\mathcal{V}^{(2)}) = \max \left\{ \sum_{n \in S_1} \sum_{i=1}^{k} \frac{w_n}{k}, \sum_{n \in S_2} \sum_{i=1}^{k} \frac{w_n}{k} \right\} = \max \left\{ \sum_{n \in S_1} w_n, \sum_{l \in S_2} w_l \right\} = \frac{W}{2} \tag{5.19}$$

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

$$\hat{F}_r(\mathcal{V}^{(2)}) = \max \left\{ \sum_{n \in V_1^{(2)}} w_n, \sum_{n \in V_2^{(2)}} w_n \right\} \leq \frac{W}{2} \tag{5.20}$$

Since the sum of all w_n is equal to W, (5.20) 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. □

5.3.1 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.

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 a_i denote the number of broadcast packets originating at node i, and $A = \sum_{i=1}^{N} a_i$. For a k-virtual receiver set, the two bounds (5.5) and (5.6) can be rewritten as

$$\hat{F}_{ch}(\mathcal{V}^{(k)}) = k \max_{c=1,\cdots,C} \left\{ \sum_{i \in X_c} a_i \right\} \tag{5.21}$$

$$\hat{F}_r(\mathcal{V}^{(k)}) \quad = \quad \sum_{i=1}^{N} a_i \; + \; C\Delta \quad = \quad M \; + \; C\Delta \tag{5.22}$$

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 such that the channel bound in (5.21) is equal to $k\frac{M}{C} + \epsilon$. Since the receiver bound is independent of k, the overall lower bound is minimized when $\hat{F}_{ch}(\mathcal{V}^{(k)}) \leq \hat{F}_r(\mathcal{V}^{(k)})$, or equivalently, if

$$k \, \frac{M}{C} \; + \; \epsilon \quad \leq \quad M \; + \; C\Delta \quad \Leftrightarrow \quad k \quad \leq \quad C \; + \; \frac{C^2\Delta - C\epsilon}{M} \tag{5.23}$$

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 (5.23) (with the equality sign), we obtain:

$$\frac{kM}{C} \quad = \quad \frac{kC\Delta}{k - C} \tag{5.24}$$

Relationship (5.24) is fundamental in that it represents the point at which wavelength concurrency balances the tuning latency. Indeed, if the last quantity in (5.23) 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. The result derived for this case can be used when information about membership in multicast groups is not available. All multicast packets are broadcasted to all nodes but we still take advantage of virtual receivers to maximize resource utilization.

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*.

5.4 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, $1 \leq k \leq N$, such that the overall bound in (5.7) for the given instance of *VRSP* is minimized. Our heuristic approaches exploit the monotonicity properties stated in Lemmas 5.2.2 and 5.2.3. 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.

5.4.1 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 (5.10) to be greater than the receiver bound in (5.11). 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 (5.14) 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 (5.6) 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 (5.5) 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

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 (5.10) and (5.11), we expect that $\hat{F}_{ch} \geq \hat{F}_r$ at this step
6. while $\hat{F}_{ch} > \hat{F}_r$ do
7. Set $k = k - 1$
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 5.2: The *G-JOIN* heuristic for *VRSP*

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 5.2. 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 5.4.1 *The* G-JOIN *heuristic in Figure 5.2 returns a virtual receiver set that achieves a local minimum with respect to the lower bound in (5.7).*

Proof. We first observe that, because of (5.8) and (5.9), 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 5.2 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:

$$\hat{F}_{ch}(\mathcal{V}^{(N)}) \;\geq\; \cdots \;\geq\; \hat{F}_{ch}(\mathcal{V}^{(k^\star+1)}) \;\geq\; \hat{F}_{ch}(\mathcal{V}^{(k^\star)}) \tag{5.25}$$

and

$$\hat{F}_{r}(\mathcal{V}^{(N)}) \;\leq\; \cdots \;\leq\; \hat{F}_{r}(\mathcal{V}^{(k^\star+1)}) \;\leq\; \hat{F}_{r}(\mathcal{V}^{(k^\star)}) \tag{5.26}$$

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

$$\hat{F}_{ch}(\mathcal{V}^{(k^\star+1)}) \;>\; \hat{F}_{r}(\mathcal{V}^{(k^\star+1)}) \quad \text{and} \quad \hat{F}_{ch}(\mathcal{V}^{(k^\star)}) \;\leq\; \hat{F}_{r}(\mathcal{V}^{(k^\star)}) \tag{5.27}$$

From (5.25) – (5.27) 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 $\mathcal{V}^{(k^\star)}$. □

Figure 5.3 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 5.2, and we plotted the receiver and channel bounds of the resulting k-virtual receiver sets, $k = N, \cdots, 1$, in Figure 5.3. The monotonic behavior of the two bounds, derived in Section 5.2.1, 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

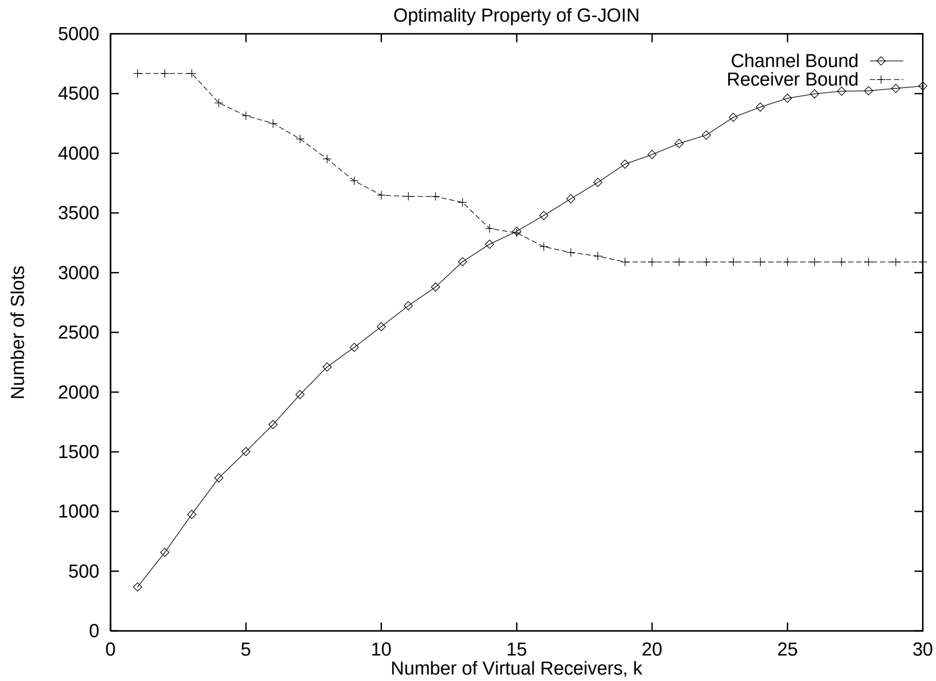

Figure 5.3: Optimality property of *G-JOIN* (sample network with $N = 30, C = 15, G = 30$)

improved any further; in Figure 5.3 we show the virtual receiver sets for all possible values of k to illustrate the monotonicity properties of the two bounds.)

5.4.2 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 5.2) 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 5.4.1 also holds for *R-JOIN*.

5.4.3 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 (5.9) is greater than the channel bound (5.8) 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 (5.6) 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 r that has the most multicast groups in common with i or j. If r has more multicast groups in common with i (respectively, j), put it in a virtual receiver set with i (j). If r has the same number of groups in common with i and j (or it has nothing in common) then compute the receiver bound (5.6) for the virtual receiver set of i and j if r was added to the set, and add r 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 5.4.2 *The* G-SPLIT *heuristic returns a virtual receiver set that achieves a local minimum with respect to the lower bound in (5.7).*

5.4.4 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 (5.6) 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 5.4.2 also holds for *R-SPLIT*.

5.5 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 (5.12) be the lower bound on an instance of *VRSP*, and let $\hat{F}(\mathcal{V}^{(k)})$ be the lower bound in (5.7) 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** [1] and random multicast groups, as follows. The elements of each matrix **M** 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 5.4 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 (obtained through the branch-and-bound technique described in Appendix A) is from the lower bound in (5.12).

Figures 5.5 - 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).

Figure 5.8 shows the behavior of the heuristics for $N = 50, G = 25$, $N = 100, G =$

[1]Recall that if the multicast traffic matrix **A** and the sets X_c are given, matrix **M** is completely specified. We have decided to generate matrix **M** directly to reduce the number of parameters to be controlled.

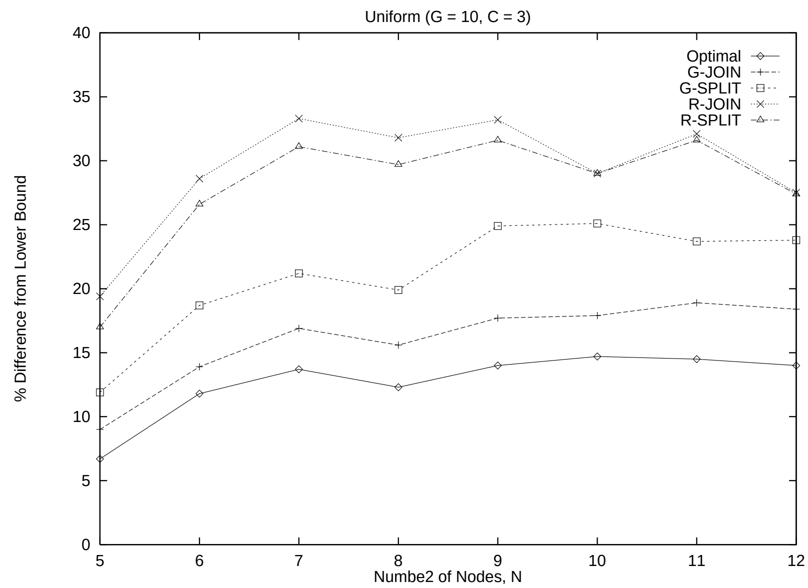

Figure 5.4: Heuristic comparison for $C = 3$ channels, $G = 10$ (uniform case)

50 and several number of channels, $C = 5, 10, 15, 20$. The results shown in this graph will let us study the impact of the number of channels on the lower bound.

To study how the existence of hot spot receivers affects our heuristics, in Figure 5.9 we plot the difference from the lower bound against N for $C = 10, G = 10$. Finally, in Figure 5.10 we again plot a hot spot case with $C = 10, G = 10$, but this time the traffic matrix $\mathbf{M}$ 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.

From these figures we can make several observations regarding the behavior of the heuristics and the effect of varying the parameters. These observations are discussed in detail in the next section.

5.6 Discussion of Results

Examining Figure 5.4, we observe the behavior of the heuristics and the optimal solution with respect to the lower bound. Our first observation is that the lower bound does

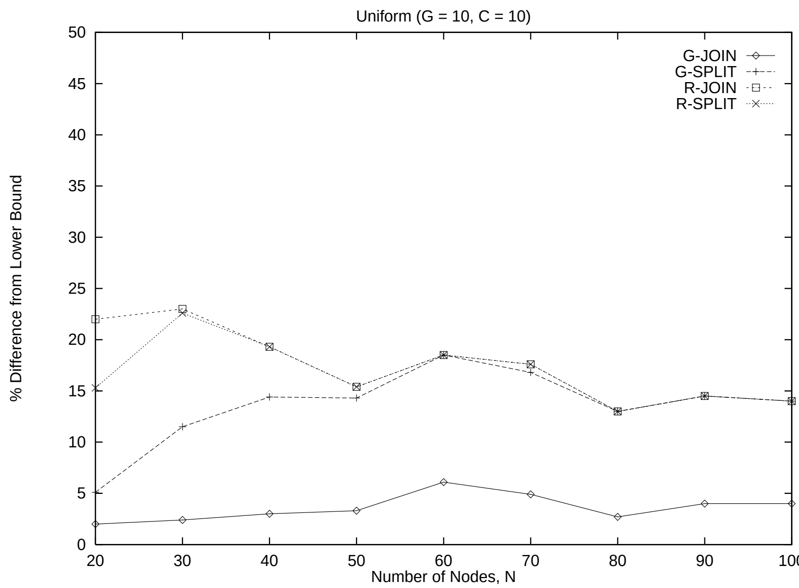

Figure 5.5: Heuristic comparison for $C = 10$ channels, $G = 10$ (uniform case)

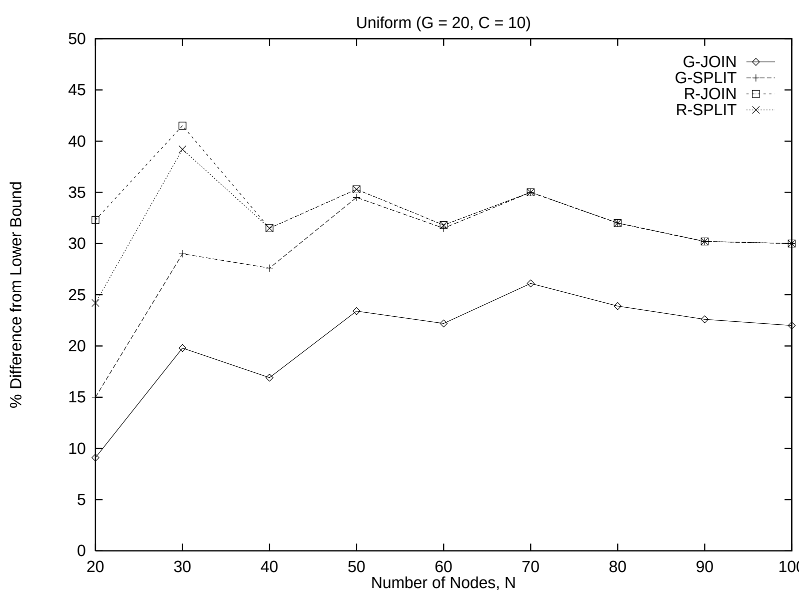

Figure 5.6: Heuristic comparison for $C = 10$ channels, $G = 20$ (uniform case)

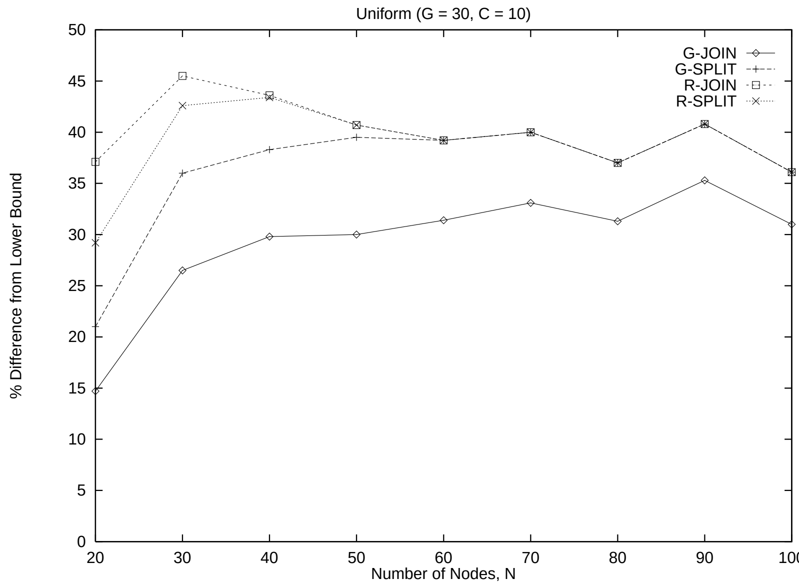

Figure 5.7: Heuristic comparison for $C = 10$ channels, $G = 30$ (uniform case)

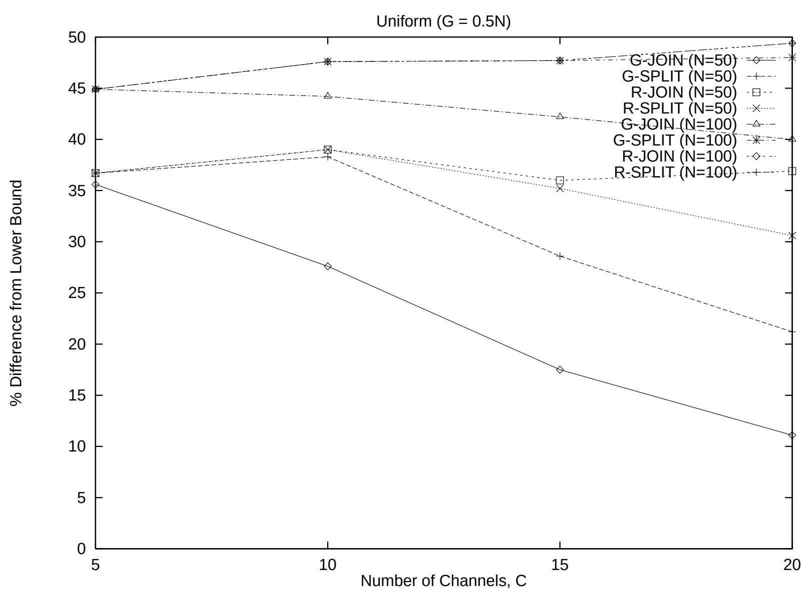

Figure 5.8: Heuristic comparison for $N = 50$ nodes, $G = 25$, and $N = 100$ nodes, $G = 50$ (uniform case)

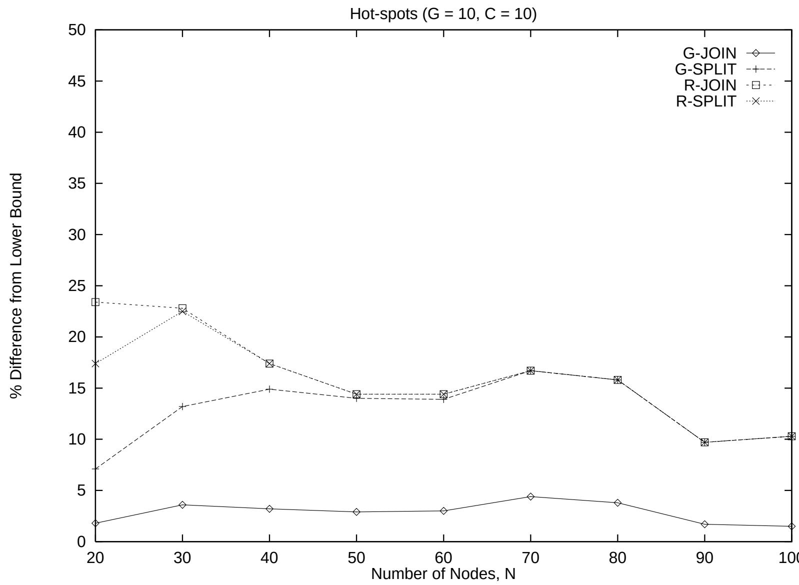

Figure 5.9: Heuristic comparison for $C = 10$ channels, $G = 10$ (hot-spot case)

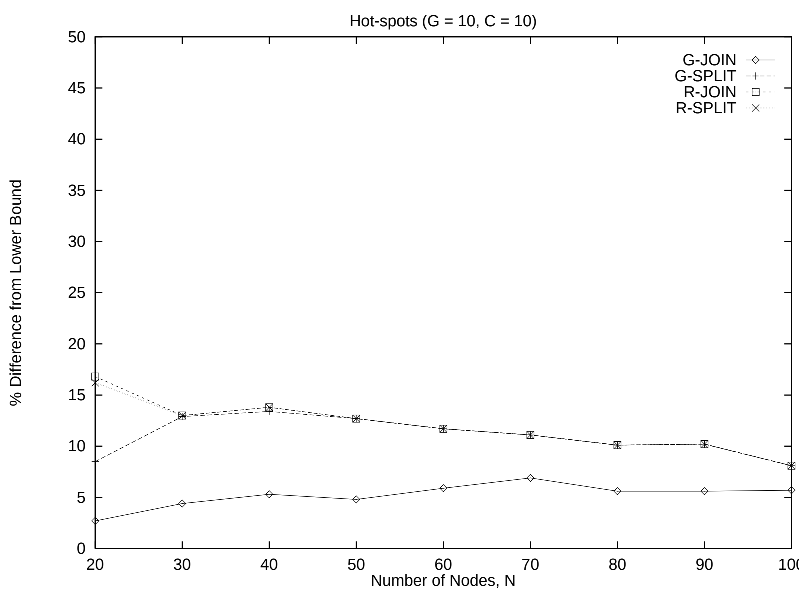

Figure 5.10: Heuristic comparison for $C = 10$ channels, $G = 10$ (hot-spot case, new traffic matrix)

not accurately characterize the optimal solution to *VRSP*, since the value at the optimal can be up to 15% higher than the lower bound. The key to interpret the results discussed next is to understand why the optimal solution, and consequently any solution, differs from the lower bound. The lower bound in (5.12) is only telling us the slots required by the dominant channel [2] or the dominant receiver [3] for its transmissions. However, in order to have the dominant channel transmit multicast traffic only once, all the members of a multicast group must be tuned to the channel simultaneously. Keeping together the members of a group decreases the likelihood of having simultaneous transmissions when multicast groups are not disjoint. Therefore, the lower bound on the schedule length in this case is determined by $\hat{F}_r(\mathcal{V}^{(1)})$.

Similarly, to have the dominant receiver get only packets addressed to itself, the dominant receiver must not get messages for multicast groups to which it does not belong. Placing the dominant receiver with another physical receiver that is in another multicast group will increase the receiver bound of the set beyond the lower bound. When nodes belong to many groups, the way to achieve this minimum is to leave the dominant receiver in a set by itself. However, that implies that every channel must transmit multicast packets for the dominant receiver at least twice. Hence, the channel bound for every channel increases. Every time a multicast group have members in different virtual receiver sets, the channel bound increases. The optimal solution may be suboptimal from the point of view of the dominant channel or receiver but the solution is optimal overall. The lower bound can only be achieved if we could make changes such that even when one of the bounds increases, the bound that dominates in (5.7) remains unchanged.

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 other figures) suggests. This assumption is further supported by the fact that the behavior of the optimal solution in Figure 5.4 appears to be similar to that of the four heuristics.

Regarding the relative performance of the heuristics, we 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

[2] The dominant channel is the channel that must transmit the maximum number of multicast messages.

[3] The dominant receiver is the receiver that needs the most slots for receiving messages from all its multicast groups.

R-JOIN, 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). From the behavior of the random heuristics we can conclude that partitioning the nodes arbitrarily will not produce the desired balance. Even a little guidance on how to partition could help in obtaining a better solution.

Now, notice that with the greedy heuristics the relative performance of the strategies is reversed; *G-JOIN* produces bounds closer to the lower bound (and the best behavior among all four heuristics) compared to the bounds produced by *G-SPLIT*. The higher complexity of *G-SPLIT* does not pay off in terms of performance. The greedy rule used in *G-SPLIT* is based on membership information rather than in the impact of the split in the lower bound. That impact is only used to break ties. This behavior illustrates once more that we need to make a careful selection of the virtual receiver sets to achieve a good balance of our objectives. Any partition will not be satisfactory.

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 in these cases.

Figures 5.5 - 5.7 allow us to observe the behavior when the number of multicast groups is increased. In all cases the heuristic behaved similarly. However, we note that when G is increased, the solutions are farther from the lower bound. The number of multicast groups to which a node belongs increases with G. Consider any two nodes i and j. Let $S_i = \{g : i \in g\}$ and $S_j = \{g : j \in g\}$ be the set of multicast groups to which i and j belong respectively. Since these nodes belong to $\frac{G}{2}$ multicast groups in average, we can find two nodes such that $S_i \neq S_j$ and $S_i \cap S_j \neq \phi$. Consequently, $S_i - S_j \neq \phi$. By increasing G we also increase the number of groups in the difference. Having more groups in the difference implies that placing these nodes in the same virtual receiver set will increase the receiver bound in the set and eventually, the overall lower bound.

The effect of increasing the number of channels in the lower bound can be seen in Figure 5.8. In this figure the lower bound decreases as C increases. From the output of the heuristics we noticed that the number, k, of virtual receiver sets produced was centered around C. This result confirms our intuition that the best balance is achieved when most of the channels are utilized. So, in average, there are N/C physical receivers in a virtual receiver set. When C increases, the number of physical receivers in a virtual receiver set decreases. As discussed earlier, if the dominant receiver is placed with other nodes that increase the receiver bound of the set, the overall lower bound increases. However, with less physical receivers per virtual receiver we have the chance to place the dominant receiver with others that belong only to the groups it belongs keeping the overall lower bound closer to the lower bound of the problem. We don't notice much difference for the cases where $N = 100$ because $N >> C$.

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 5.9 we plot the difference from the lower bound against N for $C = 10, G = 10$. Comparing the results to Figure 5.5 we see that the behavior is similar. Finally, in Figure 5.10 we again plot a hot spot case with $C = 10, G = 10$, but this time the traffic matrix $\mathbf{M}$ is generated differently. In this case, 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 5.10 to Figures 5.5 and 5.9, 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.

Chapter 6

Dynamic Multicast Traffic

In Chapter 5 we introduced the concept of virtual receivers in order to support multicast traffic in optical networks. The receivers in the network are partitioned into sets that behave identically in terms of tuning. To form the virtual receiver sets, we use information about the current network conditions, namely, the multicast traffic matrix, **M**, and the multicast group membership of the nodes.

In a connection-oriented environment, the schedule produced using the current traffic matrix could be used repeatedly for a relatively long time, i.e. until a connection terminates or a new one arrives. The information about the number of packets to transmit is based on the equivalent bandwidth negotiated at connection setup. However, even in this environment, the multicast traffic offered to the network may change due to dynamic membership changes in the system. Nodes entering or leaving a multicast group may affect the amount of traffic a virtual receiver must receive. Additionally, the formation of new multicast groups or the elimination of groups affects the amount of traffic as well.

A channel in the network may have to change its requirements because a node has been added to that channel or maybe a failure occurred in one of the nodes transmitting in that channel. Also, a multicast group may need more or less bandwidth in response to events at the application layer. Naturally, when the required bandwidth of channels or multicast groups changes the offered traffic to the network is affected.

The dynamic nature of multicast traffic motivates the analysis of the sensitivity of the virtual receiver sets to changes in the traffic offered to the network. In the multi-destination environment, the traffic offered to the network depends on both the amount of traffic to a multicast group and the number of nodes in the multicast group. Ideally, we will know beforehand the network conditions in order to allocate the necessary resources for the transmission. In our case, knowing this information enables us to optimize the formation of virtual receiver sets. However, in practice, the network conditions change dynamically. Since the assignment of nodes to virtual receivers is based on those conditions, using a static assignment may not well work under varying conditions.

In this chapter we will discuss the effect of changes in the multicast traffic offered to the network in the lower bound of the schedule length. In the next section we analyze the sensitivity to changes of the virtual receiver sets. We then describe and discuss how changes in multicast traffic affect the the requirements of channels and virtual receivers and their effect on the lower bound. Section 6.3 presents alternatives to dealing with those changes and the cost associated with them. Finally, we present numerical results to evaluate the cost performance of the different alternatives.

6.1 Sensitivity Analysis

When transmitting multicast traffic in a broadcast-and-select optical network, we propose the use of virtual receivers to minimize the impact of multiple transmissions and maximize the use of resources in the system. We construct the k-virtual receivers set, $\mathcal{V}^{(k)}$, based on the $C \times G$ collapsed multicast traffic matrix $\mathbf{M} = [m_{cg}]$ defined in Chapter 3. Using the sets we then construct an equivalent $C \times k$ unicast traffic matrix $\mathbf{B} = [b_{cl}]$. Algorithms for producing schedules for unicast traffic are then used.

Changes in multicast traffic may affect the lower bound on the schedule length, $\hat{F}(\mathcal{V}^{(k)})$, when certain conditions are met. Recall, from Equation (5.7), that the lower bound is the maximum of the channel and the receiver bound. The channel bound is determined by the number of slots required by the dominant channel to carry all its traffic and, similarly, the receiver bound is the corresponding number of slots for the dominant receiver. Therefore, if changes in multicast traffic affect the dominant channel (dominant receiver) in a bandwidth-limited network (tuning-limited network), the lower bound is affected and the schedule length changes. Changes in multicast traffic that do not affect the dominant channel or the dominant receiver also have the potential of changing the lower bound. We discuss in this section the conditions that affect the lower bound.

Suppose that a change in the multicast traffic offered to the network results in a change in the lower bound of δ slots. The new lower bound is: $\hat{F}'(\mathcal{V}^{(k)}) = \hat{F}(\mathcal{V}^{(k)}) + \delta$. How many slots can we add to or eliminate from a channel or virtual receiver in the system to cause $\hat{F}(\mathcal{V}^{(k)})$ become $\hat{F}'(\mathcal{V}^{(k)})$? Since every change does not affect the lower bound, we may add $\epsilon^{+}(\cdot)$ slots from the requirements of a channel or a virtual receiver before there is any noticeable change. Also, after $\epsilon^{-}(\cdot)$ slots have been eliminated from the requirements of a channel or virtual receiver, eliminating more slots will not affect the lower bound. Each channel and virtual receiver in the network will have an associated $\epsilon^{+}(\cdot)$ and $\epsilon^{-}(\cdot)$. We define and discuss these terms below.

We denote the number of slots that can be added to channel λ_c before affecting the lower bound as $\epsilon^{+}(\lambda_c)$; similarly, $\epsilon^{+}(V_l)$ denotes the corresponding number of slots that virtual receiver V_l can add. These quantities are computed as follows:

$$\epsilon^{+}(\lambda_c) = \hat{F}(\mathcal{V}^{(k)}) - \sum_{l=1}^{k} b_{cl} \tag{6.1}$$

$$\epsilon^{+}(V_l) = \hat{F}(\mathcal{V}^{(k)}) - \left[\sum_{c=1}^{C} b_{cl} + T_l \Delta\right] \tag{6.2}$$

They are simply the difference between the lower bound of the schedule length for the equivalent unicast traffic matrix $\mathbf{B}$ and the traffic requirement of that channel or virtual receiver. The term T_l refers to the number of channels in which virtual receiver V_l receives packets ($b_{cl} > 0$). For a channel, the number of slots to add translate in an additional $\epsilon^{+}(\lambda_c)$ packets since a slot time in our model is equal to the packet transmission time. In the case of the virtual receivers, some of the additional $\epsilon^{+}(V_l)$ slots may be needed to tune to a channel λ_c in which V_l was not receiving transmissions previously (i. e., such that $b_{cl} = 0$).

In a bandwidth-limited network, $\epsilon^{+}(\lambda_c) = 0$ for the dominant channel because its traffic requirement is $\hat{F}_{ch}(\mathcal{V}^{(k)}) = \hat{F}(\mathcal{V}^{(k)})$. Therefore, any increase in the traffic carried by channel λ_c will affect the lower bound by $\delta = \gamma(\lambda_c)$ slots where $\gamma(\lambda_c)$ is the number of slots added or eliminated to channel λ_c due to the change in multicast traffic. However, if we increase the traffic of channel $\lambda_d \neq \lambda_c$ by $\gamma(\lambda_d)$, $\hat{F}(\mathcal{V}^{(k)})$ will only increase if we add enough traffic to that channel to require more slots than the dominant channel (i. e., $\gamma(\lambda_d) > \epsilon^{+}(\lambda_d)$.) Similarly, a virtual receiver V_l that will require an additional $\gamma(V_l)$ slots will only affect the lower bound if $\gamma(V_l) > \epsilon^{+}(V_l)$. Note that in this kind of network, $\epsilon^{+}(V_l) \geq \hat{F}_{ch}(\mathcal{V}^{(k)}) - \hat{F}_r(\mathcal{V}^{(k)})$ since the amount of traffic received by any virtual receiver is at most $\hat{F}_r(\mathcal{V}^{(k)})$.

When there is a decrease in traffic, the lower bound will decrease only if the dominant channel is affected and no other channel in the network will be carrying as much traffic as the dominant channel. We define $\epsilon^-(\lambda_c)$ as the maximum number of slots that, when eliminated from a dominant channel λ_c, decreases the lower bound and is computed as follows:

$$\epsilon^-(\lambda_c) = -min\left\{\min_{d:d\neq c} \epsilon^+(\lambda_d), \min_{l=1,\cdots,k} \epsilon^+(V_l)\right\} \tag{6.3}$$

We defined in Equations (6.1) and (6.2) the difference between the requirements of each channel and virtual receiver in the system from the requirement of the dominant channel as $\epsilon^+(\cdot)$. The channel or virtual receiver with the minimum $\epsilon^+(\cdot)$ (excluding the dominant channel λ_c), would become dominant if we eliminated $\epsilon^+(\cdot)$ slots or more from channel λ_c. Thus, $-\epsilon^+(\cdot)$ defines how much we could affect the lower bound by decreasing the number of slots required by channel λ_c. If $\epsilon^-(\lambda_c) = 0$, there are several dominant channels in the system or $\hat{F}_{ch}(\mathcal{V}^{(k)}) = \hat{F}_r(\mathcal{V}^{(k)})$. In this case, all of the channels and/or virtual receivers that define the lower bound will have to decrease their traffic to cause a decrease in the lower bound. Traffic eliminated in the other channels and virtual receivers will not change the lower bound so we can eliminate all their traffic and still require at least $\hat{F}(\mathcal{V}^{(k)})$ slots. For these other channels and receivers, the corresponding $\epsilon^-(\cdot)$ is 0 since the minimum of the $\epsilon^+(\cdot)$ is the one for channel λ_c.

A similar definition of terms can be applied to a tuning-limited network. Figure 6.1 illustrates the definition of $\epsilon^+(\cdot)$ for each of the channels and virtual receivers of the tuning-limited network in Figure 3.1 and defined in the example in Section 5.1 (see Equation (5.1) for the definition of $\mathbf{A}$, the multicast group membership, X_c, and the virtual receiver sets). For this example, the equivalent unicast traffic matrix $\mathbf{B}$ becomes:

$$\mathbf{B} = \begin{bmatrix} 7 & 6 \\ 6 & 6 \end{bmatrix} \tag{6.4}$$

Virtual receiver V_1 is the dominant receiver and requires 17 slots (13 slots to receive multicast packets and 4 to tune to the different channels). We could add 1 slot to virtual receiver V_2 and 4 or 5 slots to channels λ_1 and λ_2 respectively before the lower bound increases. Eliminating one or more slots from V_1 decreases $\hat{F}(\mathcal{V}^{(k)})$ since $\epsilon^-(V_1) = -\epsilon^+(V_2)$.

In the next section we describe the different events that could cause a change in the multicast traffic offered to the network and define how those events affect the traffic.

Figure 6.1: Definition of $\epsilon^{+}(\cdot)$ for a network with $N = 5$, $C = 2$, $\Delta = 2$, and 2 virtual receivers

6.2 Changes in Multicast Traffic

When dealing with multicast traffic we must keep in mind that a change in the network conditions may affect the traffic offered to the network even when the actual number of packets to be transmitted remain unchanged. Of course, changing the number of packets to transmit will change the traffic offered to the network. In this section we analyze the effect of changing the group composition and the bandwidth requirements in the lower bound of the schedule length. We define the number $\gamma(\cdot)$ of slots that an event adds or eliminates from the traffic requirements of channels and virtual receivers and how it is related to the change in the lower bound.

6.2.1 Changes in Group Composition

In distributed applications, the membership of a multicast group varies frequently. Participants are able to join and/or leave multicast groups at different times throughout the duration of the multicast transmission. Also, the creation of a multicast group has to occur at some point in time before transmissions begin. Setting up a new group in the network involves allocating resources to the group and, in our case, including its transmissions in the schedules. At the end of the transmission, a multicast group may be eliminated and the resources reserved for the group relinquished.

All these changes in the composition of multicast groups may cause a change in the traffic offered to the network. Adding a new member to a group increases the number of recipients of a packet. Accordingly, deleting a member, decreases the number of recipients. The change in number of recipients affects the number of times a packet must be replicated. Adding a group will definitely increase the traffic in the network while eliminating one causes a decrease. When using virtual receivers for multicast transmissions, the effect of these changes is noticed when the change affects the dominant channel or virtual receiver, or the extra traffic due to g is able to compensate for the difference in the bounds.

Membership Changes

The events of joining or leaving an existing multicast group have the potential of affecting the lower bound when we use virtual receiver sets to accommodate the multicast traffic in the network. Since nodes are partitioned into virtual receivers, membership changes will not affect the system when more than one node in a virtual receiver belong to the multicast group in question. In this case, $\gamma(\cdot) = 0$ and $\hat{F}'(\mathcal{V}^{(k)}) = \hat{F}(\mathcal{V}^{(k)})$. The effect can only be seen when a node that is joining or leaving a group is part of a virtual receiver set in which no other node belongs to the multicast group. In other words, a node i that joins (leaves) multicast group g may increase (decrease) the lower bound if i is in virtual receiver V_l and no other node in V_l belongs to g. The extra traffic due to g is:

$$|\gamma(V_l)| = \sum_{c=1}^{C} m_{cg} + x_l \Delta \tag{6.5}$$

$$|\gamma(\lambda_c)| = m_{cg} \tag{6.6}$$

The term x_l refers to the number of extra channels to which V_l must now tune because of the traffic for g (i. e., number of channels for which $b_{cl} = 0$ and $m_{cg} > 0$). When joining a group, $\gamma(\cdot) = |\gamma(\cdot)|$, but when leaving a group, $\gamma(\cdot) = -|\gamma(\cdot)|$.

Group Formation and Elimination

When groups are formed or eliminated from the system, all the virtual receivers that have nodes belonging to the multicast group are affected. In this case we will always meet the condition that no other node in the virtual receiver belonged to the multicast group before the formation of the group. Similarly, no node will still belong to g when the group is eliminated from the system. Equation (6.7) defines the traffic due to g that affects each one of the virtual receivers with a node that participates in g and Equation (6.8) defines the same quantity for all channels λ_c in which $m_{cg} \neq 0$.

$$|\gamma(V_l)| = \sum_{c=1}^{C} m_{cg} + x_l \Delta \tag{6.7}$$

$$m_{cg} \leq |\gamma(\lambda_c)| \leq |g|\, m_{cg} \tag{6.8}$$

The actual value of $|\gamma(\lambda_c)|$ depends on the number of virtual receivers affected. If all nodes in g are in the same virtual receiver, $|\gamma(\lambda_c)| = m_{cg}$; if all nodes in g are in different virtual receivers, $|\gamma(\lambda_c)| = |g|\, m_{cg}$. In general, the actual change will be somewhere in between these two extreme values. Again, when a group is formed, $\gamma(\cdot) = |\gamma(\cdot)|$ but when a group is eliminated $\gamma(\cdot) = -\,|\gamma(\cdot)|$.

6.2.2 Bandwidth Changes

The traffic requirements of a channel can change with time due to several events in the network: addition of network nodes, reconfigurations, failures, etc. The traffic for a multicast group can also change due to the interaction between the participants in the group. In the following discussion, we investigate the effect of these changes on each of the channels and virtual receivers in the system. We let m'_{cg} be the new amount of traffic originating at channel λ_c and destined to group g.

Changes in Bandwidth Required by a Channel

The effect of changing the bandwidth requirements originating at channel λ_c is:

$$\gamma(\lambda_c) \quad \geq \quad \sum_{g=1}^{G} \left[m'_{cg} - m_{cg} \right] \tag{6.9}$$

$$\gamma(V_l) \quad = \quad \sum_{g:g \cap V_l \neq \phi} \left[m'_{cg} - m_{cg} \right] + x_l \Delta \tag{6.10}$$

Equality holds for $\gamma(\lambda_c)$ only when all multicast packets in channel λ_c are transmitted once. Note that changing the bandwidth of a channel may affect a large number of virtual receivers (potentially, all virtual receivers may be affected). Also, it has the potential of affecting significantly the bandwidth of the channel because of the replication that must be done.

Changes in Bandwidth Required by a Multicast Group

Changes in the bandwidth of multicast group g affects those virtual receivers with a node that belongs to g. The changes also affect the channels that have transmission for the group. In terms of the lower bound, the effect of the changes have less impact when the members of the multicast group are in a few virtual receivers (the lowest impact is when all members are in the same virtual receiver). This observation is obvious from the point of view of the requirements of the virtual receiver. For the channels, if the members of a group are spread out over different virtual receivers, each packet that is added will have to be replicated for each of the virtual receivers.

The slots added (subtracted) from the requirements of the virtual receivers and channels are:

$$\gamma(V_l) = \sum_{c=1}^{C} \left\lceil m'_{cg} - m_{cg} \right\rceil + x_l \Delta \tag{6.11}$$

$$\left\lceil m'_{cg} - m_{cg} \right\rceil \leq \gamma(\lambda_c) \leq |g| \left\lceil m'_{cg} - m_{cg} \right\rceil \tag{6.12}$$

Again, if all nodes in g are in the same virtual receiver set, $\gamma(\lambda_c) = \left\lceil m'_{cg} - m_{cg} \right\rceil$; if they are all in different sets then $\gamma(\lambda_c) = |g| \left\lceil m'_{cg} - m_{cg} \right\rceil$.

6.2.3 Effect of Changes on the Lower Bound

In section 6.1, we defined the minimum number $\epsilon^{+}(\cdot)$ of slots that need to be added to a channel or virtual receiver before $\hat{F}(\mathcal{V}^{(k)})$ increases. Also, we defined $\epsilon^{-}(\cdot)$ as the maximum number of slots that will make the lower bound decrease when eliminated from the requirements of a channel or virtual receiver. In the previous two sections we defined the number $\gamma(\cdot)$ of slots that a change adds or eliminate. To understand the relationship between these quantities, we must define how they are related to the amount of change in the lower bound, δ.

Each channel and virtual receiver will have an associated $\delta(\cdot)$ that indicates the potential effect on the lower bound. This quantity is defined as:

$$\delta(\cdot) \quad = \begin{cases} \gamma(\cdot) - \epsilon^{+}(\cdot), & \gamma(\cdot) > 0 \\ max\left\{\gamma(\cdot), \epsilon^{-}(\cdot)\right\}, & \text{otherwise} \end{cases} \tag{6.13}$$

When adding traffic ($\gamma(\cdot) > 0$), the difference between $\gamma(\cdot)$ and $\epsilon^{+}(\cdot)$ tells us how many slots are over the lower bound. When eliminating traffic, only the elimination of $\epsilon^{-}(\cdot)$ slots will make a difference. However, the change in the lower bound will be determined by the channel or receiver with the maximum change. Therefore, the actual change in the lower bound is defined as:

$$\delta = max\left\{\max_{c=1,\cdots,C}\delta(\lambda_c), \max_{l=1,\cdots,k}\delta(V_l)\right\} \tag{6.14}$$

and the new lower bound becomes $\hat{F}'(\mathcal{V}^{(k)}) = \hat{F}(\mathcal{V}^{(k)}) + \delta$.

6.3 Handling Changes in Multicast Traffic

Multicast traffic changes could be managed by using two extreme approaches: use the virtual receiver assignment already determined or recalculate the virtual receiver sets for every change. Each of these approaches offer advantages but they also have a cost associated with their implementation. In this section we discuss the components of the cost associated with approaches to handle the changes. We then discuss the advantages and disadvantages of the two extreme approaches and propose an alternative approach that attempts to minimize the disadvantages. Their cost is also examined.

6.3.1 Cost Analysis

The cost associated with the alternatives of handling changes in multicast traffic has three components: (1) processing time, (2) reconfiguration, and (3) length of the schedule. Processing time is the required time to handle each and every change. A complex heuristic may require a long processing time to produce a schedule. In the high-speed environment considered here, this time may be significant with respect to the packet transmission time in the optical medium. Reconfiguration refers to the update of information about the receivers that the nodes must maintain. In order to produce schedules and transmit the appropriate packets to a virtual receiver, the nodes need to keep information on the virtual receiver assignment of each node. When a node changes virtual receivers, packets for some of the multicast groups will no longer have to be transmitted to the virtual receiver. Finally, the length of the schedule produced is related to throughput and delay. An approach that produces a short schedule has a higher throughput and minimizes the delay. The goal of any approach to handle the changes is to minimize the three components of cost.

6.3.2 Approach 1: Use Initial Virtual Receiver Assignment

The main advantage of using the same virtual receiver assignment is that the virtual receiver algorithms need not be run, simplifying the implementation. Remember that for the transmission of multicast packets it is necessary to perform a series of steps: (1) calculate the virtual receiver sets by applying the $VR(\mathbf{M}, \Delta)$ operation, (2) transform the original multicast traffic matrix $\mathbf{M}$ into a unicast traffic matrix $\mathbf{B}$ using the resulting virtual receiver set $\mathcal{V}^{(k)}$, and (3) determine the schedule with the $Sched(\mathbf{B}, \Delta)$ operation. Since the $VR(\cdot)$ operation takes at least $\mathcal{O}(N^3)$ time [1] to execute, this approach reduces the processing time required before the transmissions. Also, this approach does not require any reconfiguration. However, after several changes, the network conditions may be significantly different from the conditions used to calculate the assignment in the first place. Therefore, the assignment is no longer optimal for the network and degradation may occur due to longer schedules.

6.3.3 Approach 2: Recalculate the Virtual Receiver Assignment

On the other hand, recalculating the virtual receiver sets every time there is a change guarantees that the set is tailored to the particular network conditions encountered at the moment. Therefore, the schedule to be produced is close to optimal. However, the algorithms must be re-run at every node and the configuration of the receivers changed.

6.3.4 Approach 3: Rearrange Virtual Receiver Sets

An alternative that attempts to strike a balance between increasing processing time and reconfiguration versus producing short schedules is to make local changes to the virtual receiver sets. The idea is to use the current assignment of virtual receiver sets to determine a new assignment that improves the lower bound without requiring $\mathcal{O}(N^3)$ steps. Applying the monotonicity property of the lower bounds discussed in section 5.2.1, we devised an algorithm that runs one iteration of the appropriate heuristic in order to decrease the impact of the changes. Since $\hat{F}_{ch}(\mathcal{V}^{(k)}) \geq \hat{F}_{ch}(\mathcal{V}(k-1))$, we use the Greedy Join heuristic (Steps 7–11 in Figure 5.2) when the lower bound is dominated by the channel bound. Similarly, when the receiver bound dominates the lower bound, we apply the Greedy

[1] Calculating the virtual receiver sets takes $\mathcal{O}(N^3)$ time when using the Greedy Join heuristic discussed in section 5.4.1

Split heuristic because $\hat{F}_r(\mathcal{V}^{(k)}) \geq \hat{F}_r(\mathcal{V}^{(k+1)})$. One iteration of the Greedy Join heuristic takes $\mathcal{O}(N^2)$ steps while one iteration of the Greedy Split heuristic takes $\mathcal{O}(GN)$ steps. In our model, the number G of active groups is smaller than the total number of possible groups and we assume that $G = cN$ for the purpose of this discussion. Therefore, the running time of this approach is also $\mathcal{O}(N^2)$.

Since we only run one iteration of the appropriate heuristic in this approach, only two virtual receivers are affected by the *JOIN*($\mathcal{V}^{(k)}, 1$) or *SPLIT*($\mathcal{V}^{(k)}, 1$) operation. The nodes from one of them will have to be reconfigured because they will change virtual receiver sets. In the best case, we will only have to reconfigure one node (because it was added to another virtual receiver or it was split from a virtual receiver). In the worst case, we will have $N - k + 1$ nodes in the virtual receiver that will need to be reconfigured when we have a k-virtual receiver set. On average, we will have about N/k nodes per virtual receiver and will only need to reconfigure that many.

6.4 Numerical Results

In this section, we compare the cost of using the three approaches in terms of the length of the schedule. The performance of the three approaches is compared with respect to the lower bound $\hat{F}$ in (5.12). Note that this lower bound does not depend on the particular virtual receiver assignment but, instead, on the requirements of channels and physical receivers in the system. We use this bound in order to fairly compare the three approaches. Recall, however, that it does not characterize the optimal schedule length for the given network conditions.

Since we have a large number of parameters to vary, we show results only for the case where $N = 20, G = 15, C = 10, \Delta = 2$ and $\bar{g} = 5$. We obtained similar results for other network configurations. We use three values, p, q, and r to determine the type of change to be performed. Figure 6.2 illustrates the decision tree used.

These values represent the probability of following a branch in the tree. With the value of p we will determine if we will change the bandwidth or change the group composition. When there is a change in bandwidth, the value of q determines if the bandwidth of a channel (a row in the traffic matrix) or the bandwidth of a group (a column in the traffic matrix) will be changed. We also use the value of r to calculate the range of the change with respect to the current traffic level. Suppose we will be changing the value of m_{cg}. We

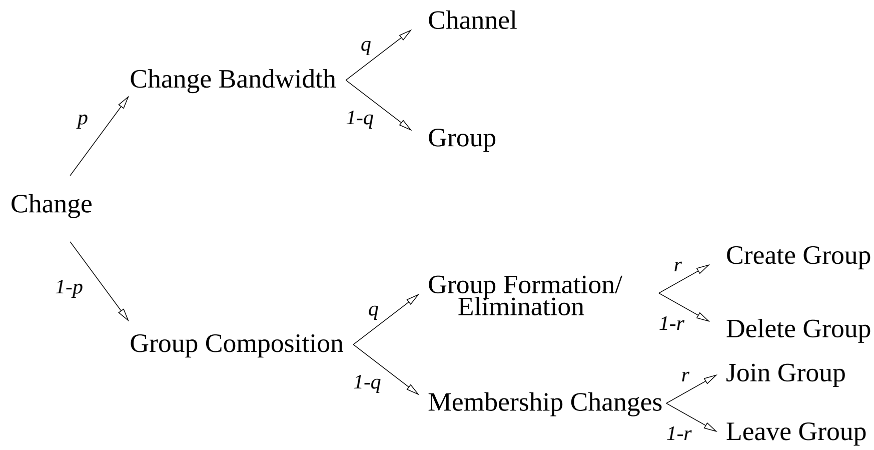

Figure 6.2: Decision tree to determine the type of change to perform with the values of p, q, and r

let $\sigma = m_{cg} * r$ and randomly select a new value for m_{cg}, m'_{cg}, between $m_{cg} - \sigma$ and $m_{cg} + \sigma$ with uniform distribution. That is, m'_{cg} = uniform$(m_{cg} - \sigma, m_{cg} + \sigma)$. Of course, if $m'_{cg} < 0$, we let $m'_{cg} = 0$.

For a change in group composition, q determines if we will create or eliminate groups or if membership within the groups will be changed. The value of r will further define the specific change to be performed. When a new group is created its traffic is defined the same way as for the original G groups (discussed shortly). The node and the group to join or leave are selected randomly with uniform distribution. However, only nodes that do not currently belong to the selected group can join the group and, similarly, only a node that belongs to the selected group can leave the group.

We created 30 cases and for each case performed 30 iterations due to the randomness of the selected change. The cumulative effect of the changes is studied by making 10 consecutive changes in traffic. For each case, we determined the initial composition of the multicast groups by selecting the number of members x in the group randomly from the uniform distribution [1, $2\bar{g}$ -1]. Then, exactly x of the N nodes are set to belong to the group. All of the nodes are required to belong to a multicast group.

The multicast traffic matrix was constructed as follows. Let t_{cg} be the probability that channel λ_c will have traffic for multicast group g. Then, with probability t_{cg}, m_{cg} was set equal to a randomly selected value from the uniform distribution [1, 20], and with probability $1 - t_{cg}$ it was set equal to zero. The probability p_{cg} was calculated as follows:

$$t_{cg} = \begin{cases} \frac{C+c-\lfloor \frac{g}{\lceil G/C \rceil} \rfloor + 1}{C}, & c < \lfloor \frac{g}{\lceil G/C \rceil} \rfloor \\ \frac{c-\lfloor \frac{g}{\lceil G/C \rceil} \rfloor + 1}{C}, & \text{otherwise} \end{cases} \tag{6.15}$$

In the next section we discuss the effect of each change individually on the lower bound using Figures 6.3 – 6.6 and 6.8 – 6.9. We plot in these figures the average difference between the lower bound $\hat{F}(\mathcal{V}^{(k)})$ produced by each approach and $\hat{F}$. The 95% confidence intervals are shown for each point. We also discuss the effect of combining the changes illustrated in Figures 6.7, 6.10 and 6.11. Then we proceed to discuss the cost of the approaches.

6.4.1 Discussion of Results

Joining or leaving multicast groups does not affect the schedule length much in any of the three approaches. However, when multicast groups are created, the length of the schedules increase especially when using the initial virtual receiver assignment. In this case we are adding a new column in the traffic matrix and several virtual receivers could be affected by the change.

We observe a decrease in Figure 6.6 when multicast groups are eliminated. In this case a column in the traffic matrix is eliminated as well. This decrease results from the fact that the lower bound does not accurately characterizes the optimal length. Also, a change in the traffic is not necessarily reflected on the lower bound while it makes a difference on the virtual receivers. By eliminating some traffic, the approaches are able to balance better the remaining traffic among the virtual receivers and could be closer to the lower bound. In Figure 6.7 we notice that allowing the group composition change in any of the four ways (with equal probability) results in a combined effect from all of the changes.

When changing bandwidth, we notice that a change in the channels affects the performance significantly. As noted earlier, a change in the bandwidth of a channel affects not only that particular channel but also many of the virtual receivers. On the other hand, changing the bandwidth of a multicast group does not affect the lower bound much because the heuristics for forming the virtual receiver sets attempt to cluster nodes belonging to the same multicast group into the same virtual receiver. We again note in Figure 6.10 that changing the bandwidth of both channels and multicast groups results in a combined effect. That is, the approaches do not follow the lower bound as closely as when there is a change

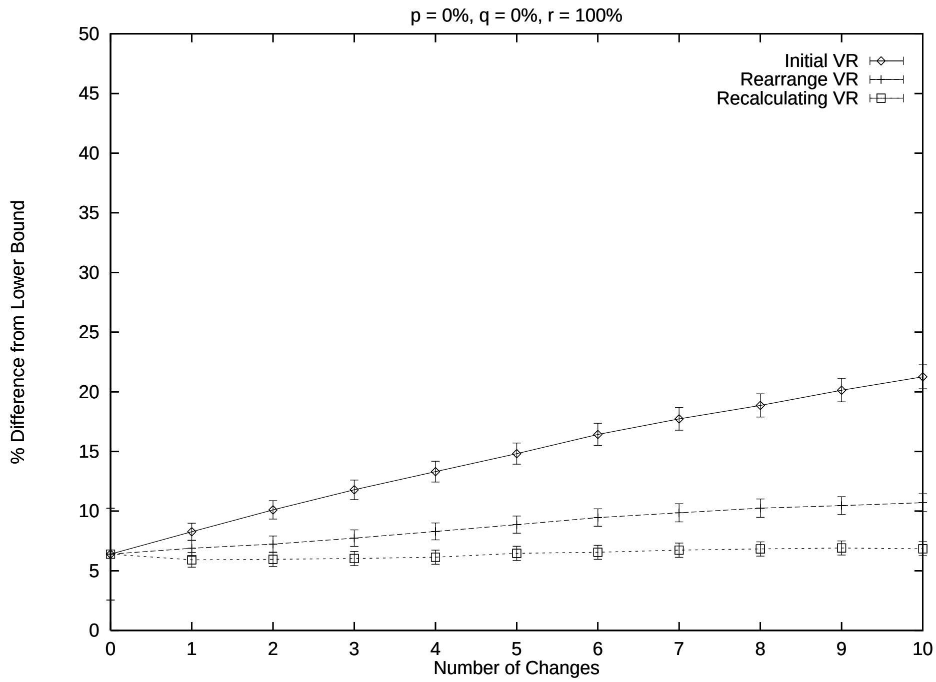

Figure 6.3: Comparison of Approaches when Nodes Join Multicast Groups

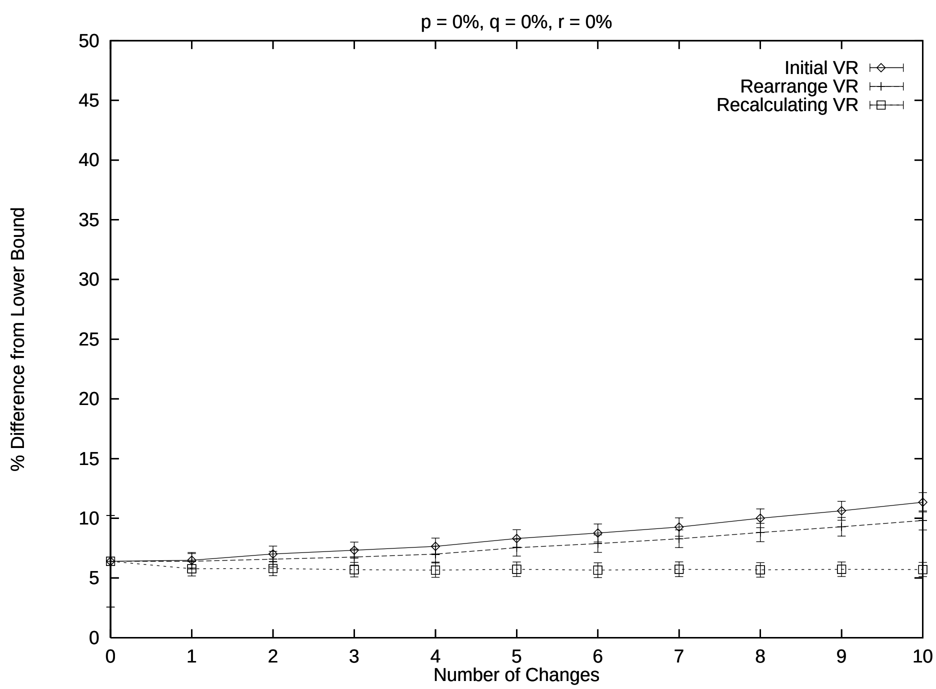

Figure 6.4: Comparison of Approaches when Nodes Leave Multicast Groups

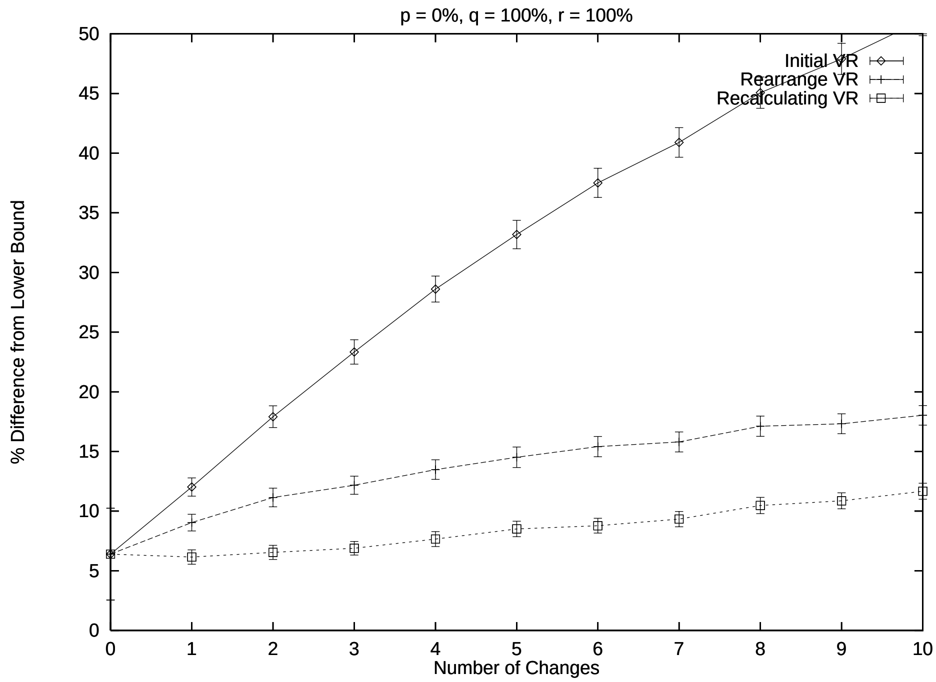

Figure 6.5: Comparison of Approaches when Multicast Groups are Created

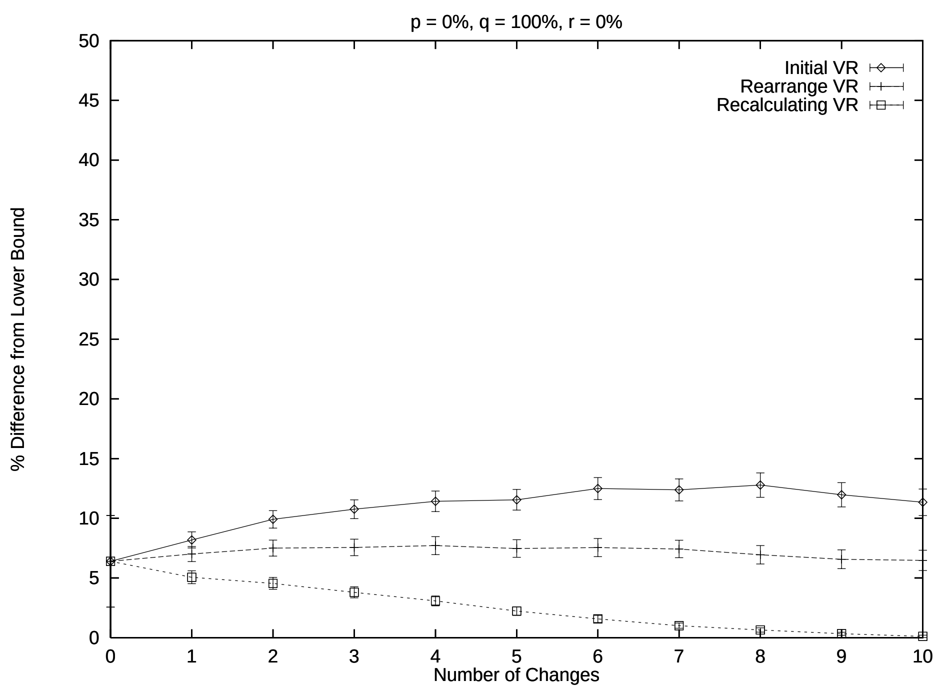

Figure 6.6: Comparison of Approaches when Multicast Groups are Eliminated

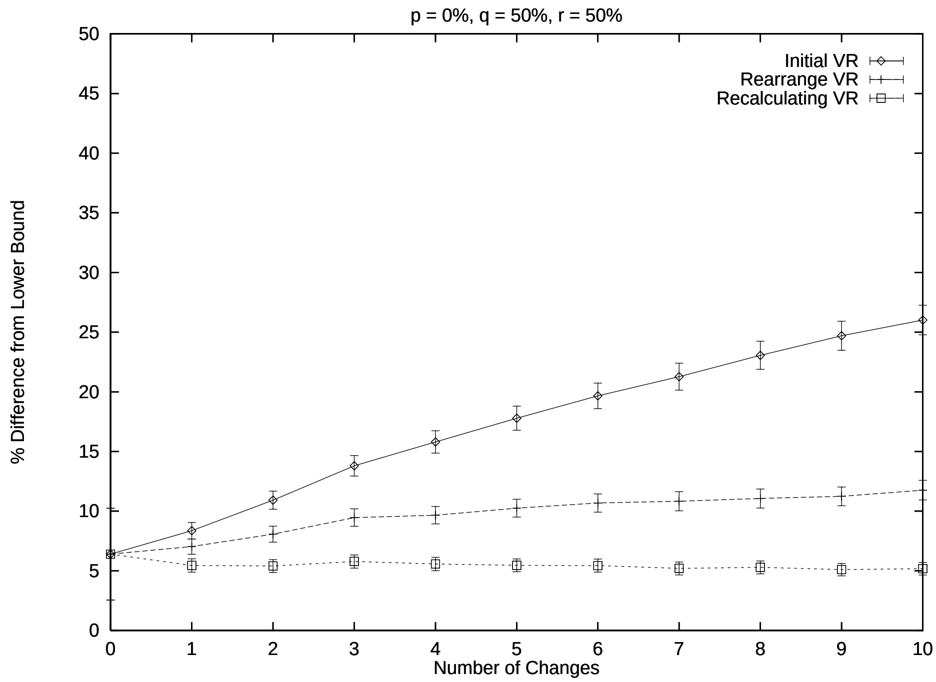

Figure 6.7: Comparison of Approaches when the Group Composition is Changed

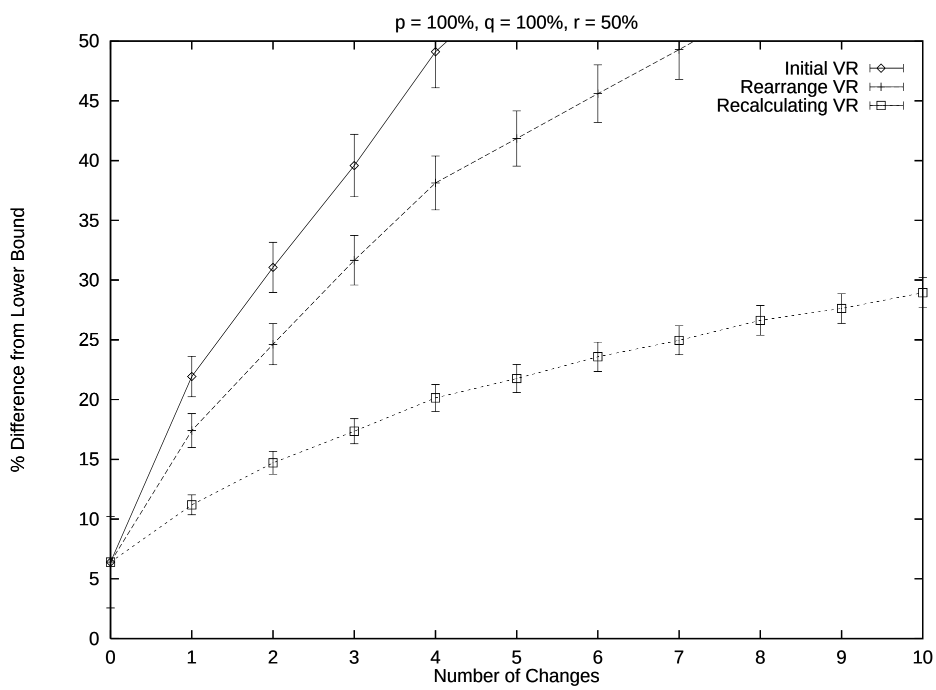

Figure 6.8: Comparison of Approaches when the Bandwidth of a Channel is Changed by 50%

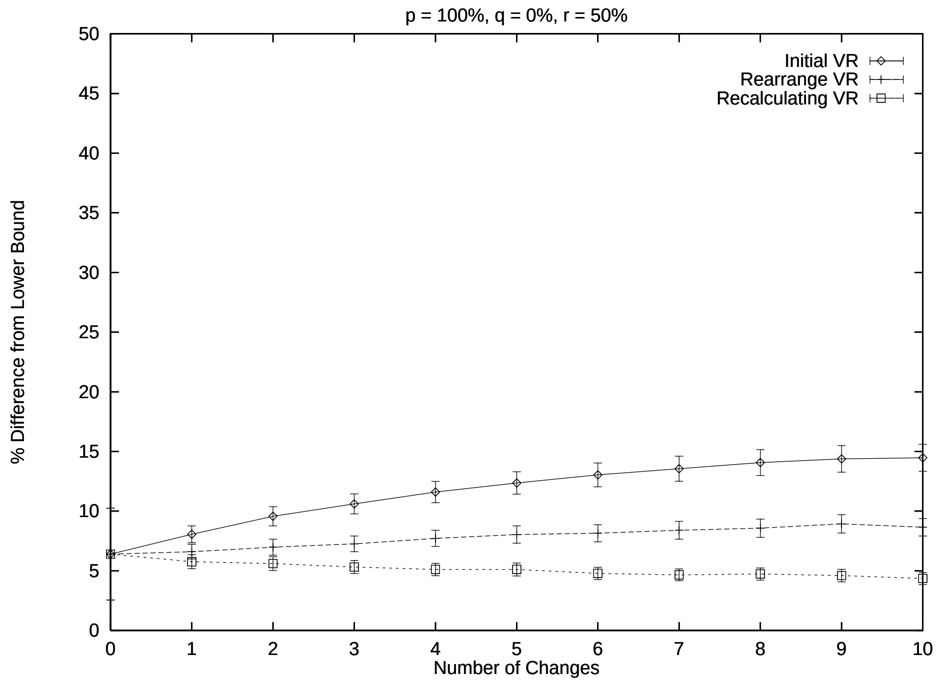

Figure 6.9: Comparison of Approaches when the Bandwidth of a Group is Changed by 50%

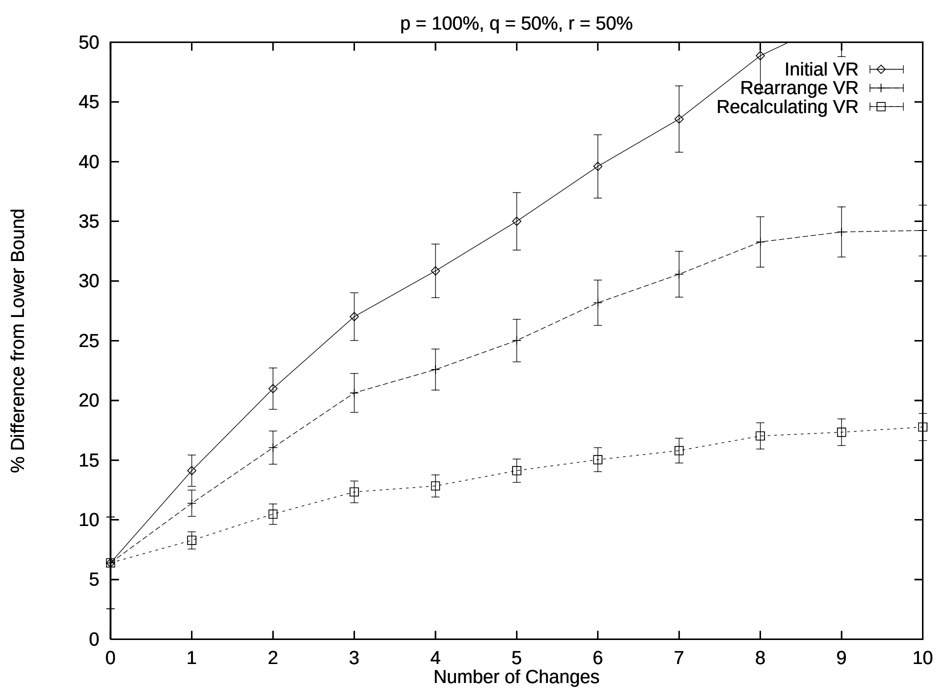

Figure 6.10: Comparison of Approaches when the Bandwidth of Channels and Groups is Changed by 50%

Table 6.1: Cost Comparison of the Approaches to Handling Multicast Traffic Changes

Approach	**Processing Time**	**Reconfiguration**
Use Initial Virtual Receiver Assignment	0	0
Recalculate Virtual Receiver Assignment	$\mathcal{O}(N^3)$	$\mathcal{O}(N)$
Rearrange Virtual Receiver Sets	$\mathcal{O}(N^2)$	$o(N/k)$

in the bandwidth of the multicast groups but they do not behave as badly as when changing bandwidth of the channels either.

Figure 6.11 shows the effect on the lower bound when multicast traffic is allowed to change in any of the different ways that we discussed. This figure shows 30 consecutive changes. We again notice a combination of effects in this graph. The growth due to the creation of multicast groups and the change in bandwidth of a channel is limited by the decrease due to elimination of groups.

6.4.2 Cost Comparison

Table 6.1 compares the three approaches discussed earlier in terms of processing time and reconfiguration. The cost associated with the schedule length is investigated numerically and discussed next.

The first thing that we can notice from Figures 6.3 – 6.11 is that the effect of the changes is cumulative. The deviation from the lower bound when conditions first change is added to the effect of the second change. When we recalculate the virtual receivers, the effect is not significantly noticeable; the lower bound in these cases grows much slower. The increase is in part attributed to the fact that the lower bound may not reflect accurately the change since it only takes into consideration the requirements of channels and physical receivers. Also, the $\epsilon^+(\cdot)$ extra slots that could be added may hide the change but when forming the virtual receiver sets the extra slots are difficult to accommodate.

We note from Figures 6.3 – 6.11 that when we use the initial virtual receiver assignment, the lower bound on the schedule length was higher than the bounds for the other approaches. With every new change in the system, the conditions in which the assignment was determined changes significantly and the solution provided is not the best for the network. In particular, when groups are created or the bandwidth of a channel is changed, this approach produces longer schedules. So, when those changes are introduced in the system this approach should not be used because low throughput and high delay will be seen in the system. In Figure 6.11 we can see that after several changes this approach results in schedules that are at least 50% longer than the lower bound, and it continues to produce longer schedules from then on.

As expected, recalculating the assignment produced the lowest bounds in every instance. Rearranging virtual receivers improved on the performance of the first approach but did not produce as lower bounds as for the second approach. Two factors related to the heuristic used contribute to its performance. First of all, we only perform one iteration of the heuristics to optimize the lower bound. Additionally, the use of the Greedy Split heuristic causes this approach to deviate from the best because it does not produce as good partitions as the Greedy Join heuristic. The behavior of this approach, though, is similar to the behavior of the second approach when the rearrangement is done after every change. Therefore, degradation is much slower.

6.5 Concluding Remarks

Changes in multicast traffic affect the formation of virtual receiver sets. Since traffic in a network is dynamic in nature, we should be able to adapt to these changes with the minimum amount of disruption. For optimal assignment of nodes to virtual receivers, we must recalculate the virtual receiver sets every time a change occurs. Recalculating, however, is expensive in terms of processing time and reconfiguration. If we decide not to recalculate for every change, the network will suffer because the schedules produced will be longer than necessary. We showed in this thesis that an alternative approach could be used to rearrange the nodes in the virtual receivers. This alternative approach produces a lower bound on the schedule length when compared to the bound produced by using the initial assignment while not requiring as much processing time and reconfigurations as the approach of recalculating the assignment. In an actual implementation, a combination of the approaches could be used.

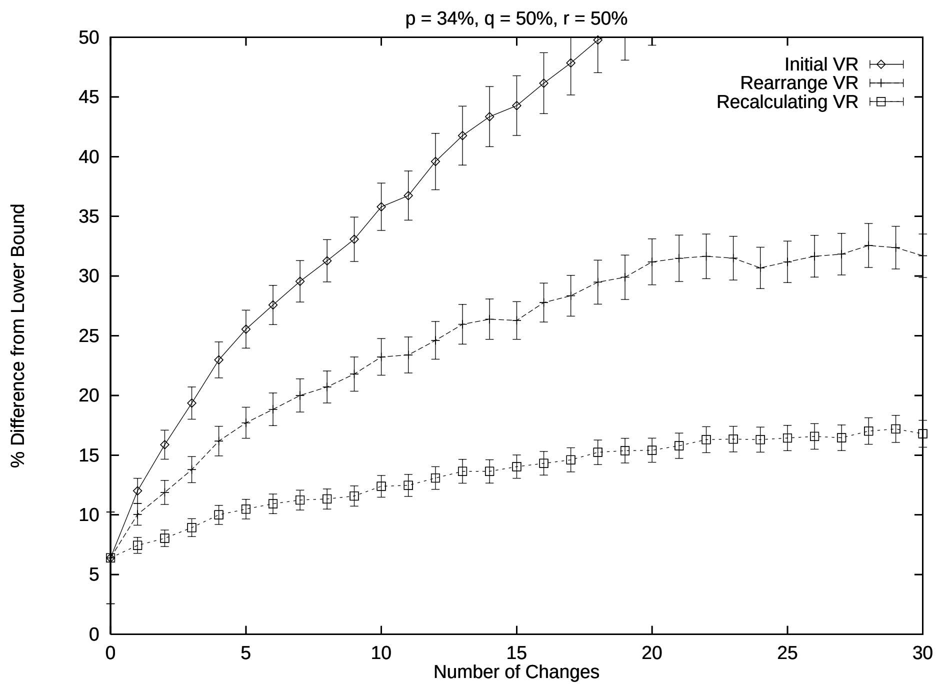

Figure 6.11: Comparison of Approaches when All Changes are Performed with Equal Probability

Chapter 7

Scheduling Combined Unicast and Multicast Traffic

Even though providing support for multipoint communication has been recognized as essential in current networks [2, 29], a mixed traffic scenario of unicast and multicast traffic is encountered in practice. When scheduling the combined traffic it is our goal to efficiently utilize network resources. In our case, efficiency is measured in terms of the length of the schedule produced: the shorter the schedule length, the higher the overall network throughput and the lower the average delay experienced by a packet. The challenge is then to determine how to treat the traffic such that we minimize schedule length. As discussed in Chapter 5, treating multicast messages as unicast uses a lot of bandwidth because a single packet is replicated for all its recipients and transmitted individually. A transmitter will have to do the replication and the requirement of the channels increase. To overcome this limitation, we partitioned the nodes into virtual receiver sets. With this technique all the nodes in a virtual receiver will tune to the same channel at the same time. Therefore, a node will have to filter out packets which are not addressed to it increasing the amount of traffic the node will receive. So, if we now include unicast traffic in the transmissions to a virtual receiver, all the nodes in the virtual receiver will get packets addressed to only one of them.

In this chapter, we focus on the problem of scheduling both unicast and multicast traffic. We present three different strategies for handling the combined traffic. These strategies are: scheduling unicast and multicast traffic separately, treating multicast traffic as unicast, and treating unicast traffic as multicast traffic with a group of size 1. The lower bound on the schedule length for each of the strategies is derived. We then compare these three strategies through extensive numerical experiments in order to determine which one yields the shortest schedule. In the comparisons, we discuss the conditions that make one strategy more effective than the others.

In the following discussion we use the unicast channel bound, $\hat{H}_{ch}$ (4.1), unicast receiver bound, $\hat{H}_r$ (4.2), and the overall lower bound, $\hat{H}$ (4.3), for clearing matrix $\mathbf{U}$ defined in Section 4.1. We also use the multicast channel bound, $\hat{F}_{ch}(\mathcal{V}^{(k)})$ (5.5), multicast receiver bound, $\hat{F}_r(\mathcal{V}^{(k)})$ (5.6), and the overall lower bound, $\hat{F}(\mathcal{V}^{(k)})$ (5.7) for clearing matrix $\mathbf{M}$ as derived in Section 5.2.

7.1 Transmission Strategies for Combined Unicast and Multicast Traffic

In this section we present three different strategies for scheduling and transmitting an offered load of combined unicast and multicast traffic. These are: separate scheduling, treating multicast as unicast traffic, and treating unicast as multicast traffic. These strategies were selected because they provide an intuitive solution to handling unicast and multicast traffic. We assume that the unicast and multicast traffic demands are given by matrices $\mathbf{U}$ and $\mathbf{M}$ respectively. Based on the results of the previous chapters, we derive lower bounds on the schedule length for each strategy. All three strategies use the algorithms in [26] to schedule packet transmissions. Since the lower bound accurately characterizes the scheduling efficiency of the algorithms in [26], the lower bounds will provide insight into the relative merits of each strategy. In the following, we will use $L_{ch}^{(i)}$, $L_r^{(i)}$, and $L^{(i)}$ to denote the channel, receiver, and overall lower bound, respectively, of strategy $i, i = 1, 2, 3$.

7.1.1 Strategy 1: Separate Scheduling

Our first strategy for transmitting the combined traffic offered to the network is to separately schedule the unicast and multicast matrices. That is, each traffic matrix is considered in isolation, and the appropriate scheduling techniques from [26, 21] are applied to each traffic component. The two schedules are then used in sequence. This is a straightforward approach and involves the following operations: *Sched*($\mathbf{U}$, Δ) and *MSched*($\mathbf{M}$, Δ).

Since at the end of the first schedule (say, the one for unicast traffic) the receivers may not be tuned to the channels required to start the next schedule (say, the one for multicast traffic), a sufficient number of slots for receiver retuning must be added between the two schedules [1]. Thus, we get a lower bound on the length of time it takes to clear matrices $\mathbf{U}$ and $\mathbf{M}$ under this approach as ($\mathcal{V}^{(k^\star)}$ is the near-optimal virtual receiver set for matrix $\mathbf{M}$):

$$L^{(1)} \quad = \quad \hat{H} \ + \ \hat{F}(\mathcal{V}^{(k^\star)}) \ + \ \Delta \tag{7.1}$$

We note that the separate scheduling strategy achieves a lower bound which is equal to the sum of the best lower bounds for each traffic component in isolation (plus Δ slots to account for the retuning between the schedules). However, it could be possible to obtain a schedule of smaller length by mixing together both traffic types. First, using a single schedule would eliminate the need for the Δ slots between the two schedules. We also note that a schedule may include some empty slots or gaps. These empty slots could be used to carry traffic of the other type, thus reducing the overall schedule length. However, the new schedule must still ensure that there are no channel or destination conflicts. The next two strategies combine both traffic types to produce a single schedule.

7.1.2 Strategy 2: Multicast Traffic Treated as Unicast Traffic

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

$$u^{(2)}_{cj} \quad = \quad u_{cj} \ + \sum_{g:j\in g} m_{cg} \tag{7.2}$$

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

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

$$L^{(2)}_{ch} \quad = \quad \max_{c=1,\cdots,C} \left\{ \sum_{j=1}^{N} u^{(2)}_{cj} \right\} \quad = \quad \max_{c=1,\cdots,C} \left\{ \sum_{j=1}^{N} u_{cj} \ + \ \sum_{j=1}^{N} \sum_{g:j\in g} m_{cg} \right\}$$

[1] A number Δ of slots is also required at the very beginning of transmission to ensure that receivers are tuned to the channels as required by the first schedule. But these Δ initial slots are needed for all four strategies and do not affect their relative performance. Hence, we will ignore these Δ initial slots in the expressions for the various bounds presented here.

$$
\begin{aligned}
L_{ch}^{(2)} &\leq \max_{c=1,\cdots,C} \left\{ \sum_{j=1}^{N} u_{cj} \right\} + \max_{c=1,\cdots,C} \left\{ \sum_{j=1}^{N} \sum_{g:j\in g} m_{cg} \right\} \\
&= \hat{H}_{ch} + \hat{F}_{ch}(\mathcal{V}^{(N)})
\end{aligned} \tag{7.3}
$$

Similarly, we can obtain the receiver bound by accounting for the traffic plus tuning requirements of each (physical) receiver. Let us define $T_j^{(2)}$ as the number of channels to which (physical) receiver j must tune during the schedule according to the new unicast matrix $\mathbf{U}^{(2)}$. Recall that T_j (respectively T'_j) is the number of channels to which receiver j must tune based on the requirements of traffic $\mathbf{U}$ (respectively $\mathbf{M}$). Obviously, we have that $T_j^{(2)} = T_j + T'_j - x_j$, where x_j is the number of channels in common between the tuning requirements of j for $\mathbf{U}$ and $\mathbf{M}$. We have:

$$
\begin{aligned}
L_r^{(2)} &= \max_{j=1,\cdots,N} \left\{ \sum_{c=1}^{C} z_{cj}^{(2)} + T_j^{(2)} \Delta \right\} \\
&= \max_{j=1,\cdots,N} \left\{ \sum_{c=1}^{C} u_{cj} + \sum_{c=1}^{C} \sum_{g:j\in g} m_{cg} + \left(T_j + T'_j - x_j \right) \Delta \right\} \\
&= \max_{j=1,\cdots,N} \left\{ \left[\sum_{c=1}^{C} u_{cj} + T_j \Delta \right] + \left[\sum_{c=1}^{C} \sum_{g:j\in g} m_{cg} + T'_j \Delta \right] - x_j \Delta \right\} \\
&\leq \hat{H}_r + \hat{F}_r(\mathcal{V}^{(N)}) - \min_{j=1,\cdots,N} \{ x_j \Delta \} \\
&\leq \hat{H}_r + \hat{F}_r(\mathcal{V}^{(N)})
\end{aligned} \tag{7.4}
$$

In (7.3) and (7.4) above, $\hat{F}_{ch}(\mathcal{V}^{(N)})$ and $\hat{F}_r(\mathcal{V}^{(N)})$ are the channel and receiver bounds, respectively, on clearing matrix $\mathbf{M}$ when the virtual receiver set is $\mathcal{V}^{(N)} = \{\{1\}, \cdots, \{N\}\}$, i.e., when there are N virtual receivers, each consisting of exactly one physical receiver. These bounds can be obtained from (5.5) and (5.6), respectively, by letting $\mathcal{V}^{(k)} = \mathcal{V}^{(N)}$.

From (7.3) and (7.4) we may obtain a lower bound for Strategy 2:

$$
L^{(2)} \leq \max \left\{ \hat{H}_{ch} + \hat{F}_{ch}(\mathcal{V}^{(N)}), \hat{H}_r + \hat{F}_r(\mathcal{V}^{(N)}) \right\} \leq \hat{H} + \max \left\{ \hat{F}_{ch}(\mathcal{V}^{(N)}), \hat{F}_r(\mathcal{V}^{(N)}) \right\} \tag{7.5}
$$

This strategy may result in a lower bound that is lower than $L^{(1)}$ when the unicast traffic is tuning limited (i.e., $\hat{H} = \hat{H}_r > \hat{H}_{ch}$). In this case, $L^{(1)} > L_r^{(2)} > L_{ch}^{(2)}$ must hold. We know that $L^{(1)} > L_r^{(2)}$ for tuning limited networks because $\hat{F}(\mathcal{V}^{(k^\star)}) + \Delta > \hat{F}_r(\mathcal{V}^{(N)})$ is true for any $\Delta > 0$ (see [21]). To satisfy the condition that $L_r^{(2)} > L_{ch}^{(2)}$, we must have a network where $k = N$ or where $\hat{F}_{ch}(\mathcal{V}^{(N)}) - \hat{F}_r(\mathcal{V}^{(N)}) < \hat{H}_r - \hat{H}_{ch}$. In the latter case, the difference between the unicast bounds must compensate for the difference between the multicast bounds to obtain a better lower bound in $L^{(2)}$. We may also obtain a better lower bound in a bandwidth limited network whenever $\hat{F}(\mathcal{V}^{(k^\star)}) = \hat{F}(\mathcal{V}^{(N)})$. The Δ additional slots required for the first strategy will make $L^{(1)} > L^{(2)}$.

7.1.3 Strategy 3: Unicast Traffic Treated as Multicast Traffic

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

$$m_{cg}^{(3)} = \begin{cases} m_{cg}, & g = 1, \cdots, G \\ u_{cj}, & g = G + j, j = 1, \cdots, N \end{cases} \tag{7.6}$$

We can then use the new matrix $\mathbf{M}^{(3)}$ to obtain a schedule for the combined unicast and multicast traffic: *MSched*($\mathbf{M}$(3), Δ). The near-optimal $k^{(3)}$-virtual receiver set obtained from matrix $\mathbf{M}^{(3)}$, however, will in general be quite different from the $k^\star$-virtual receiver set obtained from matrix $\mathbf{M}$. Consequently, we cannot express the channel and receiver bounds for this strategy as a function of the channel and receiver bounds for matrix $\mathbf{M}$ as we did with Strategy 2 in (7.3) and (7.4).

We could still express the lower bound for this strategy in terms of $k^{(3)}$ using equations (5.5), (5.6), and (5.7). These equations, however, will not allow us to compare the lower bound for the different strategies. Therefore, we now obtain a lower bound for Strategy 3 as (see Lemma 5.2.1):

$$L^{(3)} = \max\left\{\hat{F}_r^{(3)}(\mathcal{V}^{(N)}), \hat{F}_{ch}^{(3)}(\mathcal{V}^{(1)})\right\} \tag{7.7}$$

where $\hat{F}_r^{(3)}$ and $\hat{F}_{ch}^{(3)}$ represent the corresponding bounds on matrix $\mathbf{A}^{(3)}$. Expanding on these terms we obtain:

$$\hat{F}_r^{(3)}(\mathcal{V}^{(N)}) = \max_{j=1,\cdots,N}\left\{\left[\sum_{c=1}^{C}\sum_{g:j\in g} m_{cg}^{(3)}\right] + T_j^{(3)}\Delta\right\}$$

$$= \max_{j=1,\cdots,N} \left\{ \left[\sum_{c=1}^{C} \sum_{g:j\in g} m_{cg} \right] + \left[\sum_{c=1}^{C} u_{cj} \right] + T_j^{(3)}\Delta \right\}$$
$$\leq \hat{F}_r(\mathcal{V}^{(N)}) + \hat{H}_r \qquad (7.8)$$

$$\hat{F}_{ch}^{(3)}(\mathcal{V}^{(1)}) = \max_{c=1,\cdots,C} \left\{ \sum_{g} m_{cg}^{(3)} \right\}$$
$$= \max_{c=1,\cdots,C} \left\{ \left[\sum_{g} m_{cg} \right] + \left[\sum_{j=1}^{N} u_{cj} \right] \right\}$$
$$\leq \hat{F}_{ch}(\mathcal{V}^{(1)}) + \hat{H}_{ch} \qquad (7.9)$$

We thus have:

$$L^{(3)} \leq \hat{H} + \max\left\{ \hat{F}_r(\mathcal{V}^{(N)}), \hat{F}_{ch}(\mathcal{V}^{(1)}) \right\} \qquad (7.10)$$

We note, however, that, unlike the other bounds presented in this section, the bound in (7.10) is not tight and may not be achievable.

7.2 Numerical Results

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

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

The multicast traffic matrix was constructed as follows. Let p_{cg} be the probability that channel λ_c will have traffic for multicast group g. Then, with probability p_{cg}, m_{cg}

was set equal to a randomly selected value from the uniform distribution [1, 20], and with probability $1 - p_{cg}$ it was set equal to zero. The probability p_{cg} was calculated as follows:

$$p_{cg} = \begin{cases} \frac{C+c-\lfloor \frac{g}{\lceil G/C \rceil} \rfloor + 1}{C}, & c < \lfloor \frac{g}{\lceil G/C \rceil} \rfloor \\ \frac{c-\lfloor \frac{g}{\lceil G/C \rceil} \rfloor + 1}{C}, & \text{otherwise} \end{cases} \tag{7.11}$$

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

$$s = \frac{C\bar{g}G\bar{m}}{C\bar{g}G\bar{m} + CN\bar{u}} 100\% \tag{7.12}$$

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

$$q_{cj} = \begin{cases} \frac{C+c-\lfloor \frac{j}{\lceil N/C \rceil} \rfloor + 1}{C}, & c < \lfloor \frac{j}{\lceil N/C \rceil} \rfloor \\ \frac{c-\lfloor \frac{j}{\lceil N/C \rceil} \rfloor + 1}{C}, & \text{otherwise} \end{cases} \tag{7.13}$$

Then, with probability q_{cj} the corresponding entry of the unicast traffic matrix u_{cj} was set to a randomly selected number from the uniform distribution [1, 2$\bar{u}$-1], and with probability 1-q_{cj} it was set equal to zero. Probabilities p_{cg} and q_{cj} are formulated such that multicast groups and individual nodes do not receive packets from all channels. Also, the channels from which they are receiving packets are different for the different groups and nodes.

We also investigated the effects of hot-spots by introducing hot nodes which receive a larger amount of traffic compared to non-hot nodes. Specifically, we let the first five nodes of the network be the hot nodes. The average number of unicast packets received by these nodes was set to 1.5$\bar{u}$. Therefore, with probability q_{cj}, given by (7.13), the entry $u_{cj}, j = 1, \cdots, 5$, was set to a randomly selected number from the uniform distribution [1, 2(1.5$\bar{u}$)-1], and with probability 1-q_{cj} it was set to zero. The remaining $N - 5$ nodes receive an average number of unicast packets equal to $(\frac{N-7.5}{N-5})\bar{u}$. For these nodes with probability q_{cj}, the entry $u_{cj}, j = 6, \cdots, N$, was set to a randomly selected value from the uniform distribution [1, $2(\frac{N-7.5}{N-5})\bar{u} - 1)$], and with probability 1-q_{cj} it was set equal to zero. Note that the overall average number of unicast packets remains equal to $\bar{u}$, as in the non-hot-spot case.

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

The results presented below are organized as follows. In Section 7.2.1 we give some representative detailed comparisons of the three strategies obtained by varying only one of the parameters $s, \bar{g}, \Delta, C, G$, and N at a time. In Section 7.2.2 we summarize our findings, and we discuss under which conditions each strategy gives the shortest schedule.

7.2.1 Detailed Comparisons

The results are presented in Figures 7.1–7.12. In each figure, we plot $D^{(i)}, i = 1, 2, 3$, against s indicated as "% Multicast Traffic". In other words, the figures present the performance of the various strategies relative to each other. Confidence intervals are also shown in each figure. For presentation purposes, we use the following abbreviations for the names of the three strategies in the figures and tables:. Strategy 1 is referred to as "Separate"; Strategy 2, where multicast traffic is treated as unicast traffic, is referred to as "Unicast"; and Strategy 3, where unicast traffic is treated as multicast traffic is referred to as "Multicast".

Figure 7.1 gives the results for the case where $N = 20, G = 30, C = 10, \Delta = 4$, and $\bar{g} = 0.25N$. We note that Strategy 2 is the best strategy for $s < 50\%$, but that Strategy 3 becomes the best one for $s \geq 50\%$. This figure represents our base case. Figures 7.2 to 7.12 give results in which only one of the parameters has been changed while the remaining

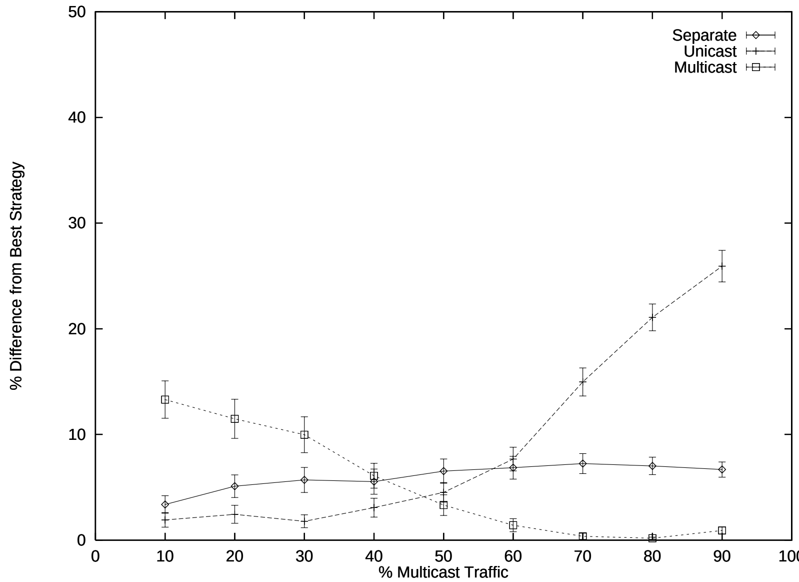

Figure 7.1: Comparison of strategies for $N = 20$, $G = 30$, $C = 10$, $\Delta = 4$, and $\bar{g} = 0.25N$ (base case)

parameters are the same as those in Figure 7.1. Specifically, Figures 7.2 and 7.3 show the cases in which we vary $\bar{g}$. In Figures 7.4 and 7.5 we varied Δ. The number of channels is varied in Figures 7.6 and 7.7, while the number of multicast groups is changed in Figures 7.8 and 7.9. The next three figures, namely 7.10, 7.11, and 7.12, show results when the number of nodes is increased.

Below, we discuss the results presented in Figures 7.1–7.12 for each strategy separately.

Separate Scheduling. Even though the behavior of Strategy 1 (relative to the others) appear to be unaffected by the different parameters, we noticed changes related to the tuning latency, as expected. When Δ was increased, $\bar{D}^{(1)}$ had a tendency to increase. From the expression (7.1) for $L^{(1)}$, we note that Δ slots are added to the optimal bounds for unicast and multicast traffic, while the lower bounds for the other two strategies did not have this component. It is thus expected for $\bar{D}^{(1)}$ to be sensitive to this parameter. Increasing s or C did not change the behavior of $\bar{D}^{(1)}$, except for large values of Δ ($\Delta = 16$). In these cases, the increase observed can be attributed to the large Δ.

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

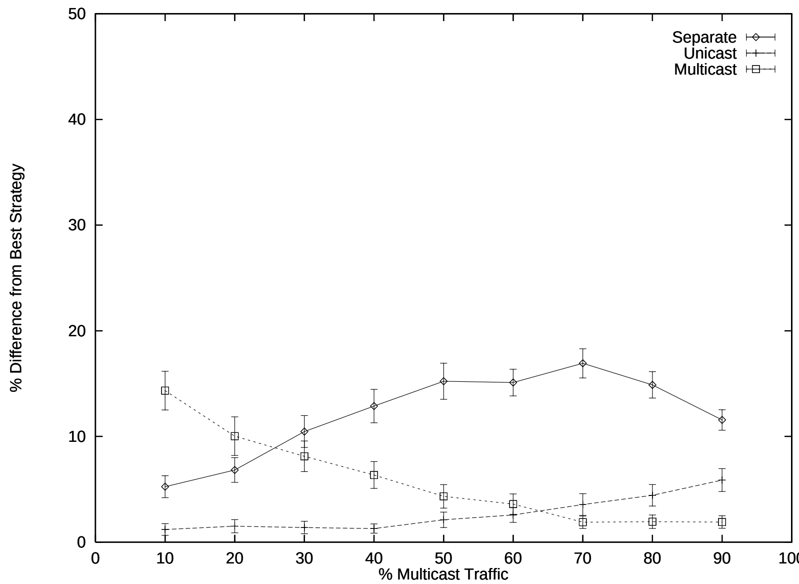

Figure 7.2: Comparison of strategies for $N = 20, G = 30, C = 10, \Delta = 4, \bar{g} = 0.10N$

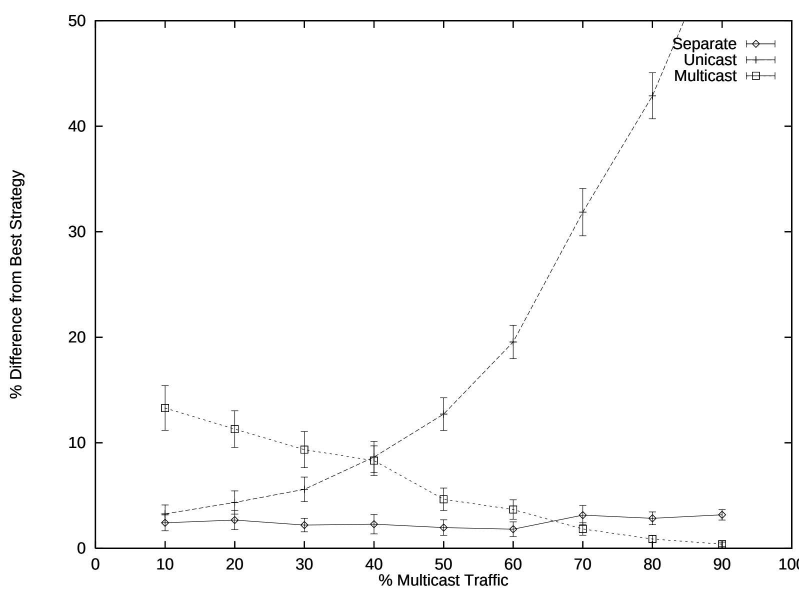

Figure 7.3: Comparison of strategies for $N = 20, G = 30, C = 10, \Delta = 4, \bar{g} = 0.50N$

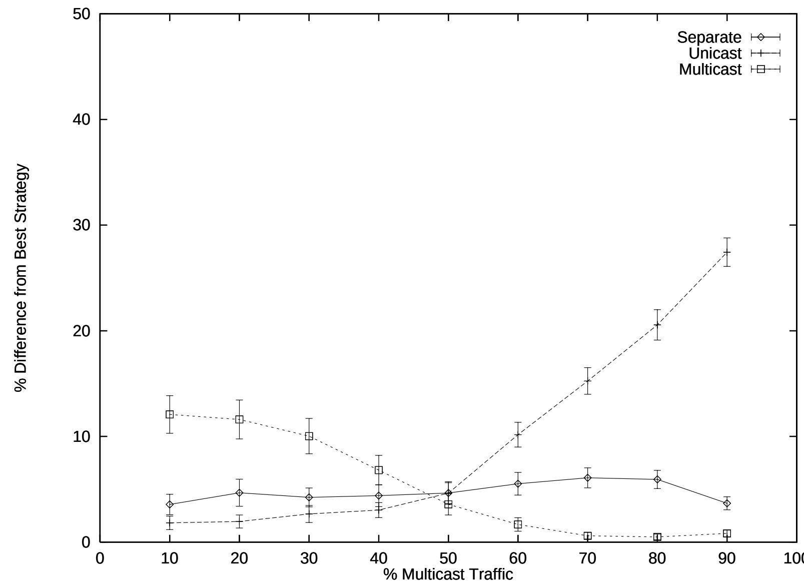

Figure 7.4: Comparison of strategies for $N = 20, G = 30, C = 10, \Delta = 1, \bar{g} = 0.25N$

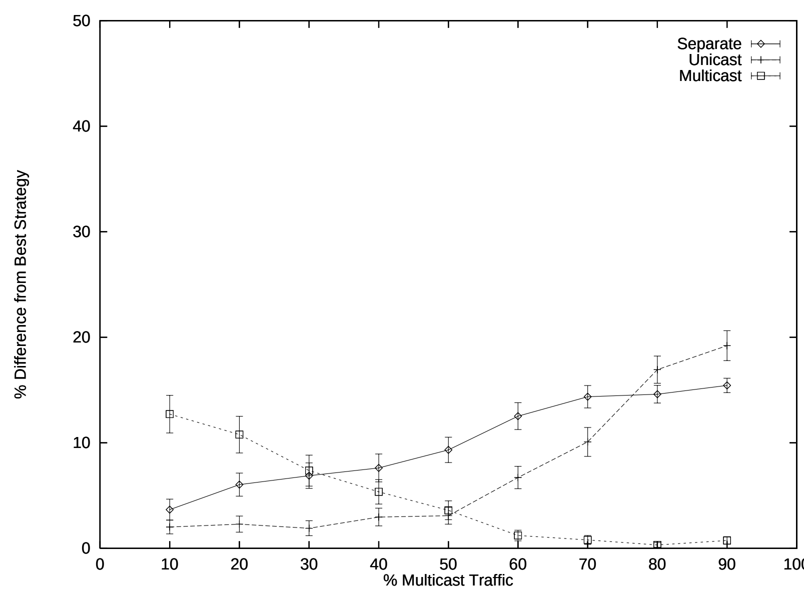

Figure 7.5: Comparison of strategies for $N = 20, G = 30, C = 10, \Delta = 16, \bar{g} = 0.25N$

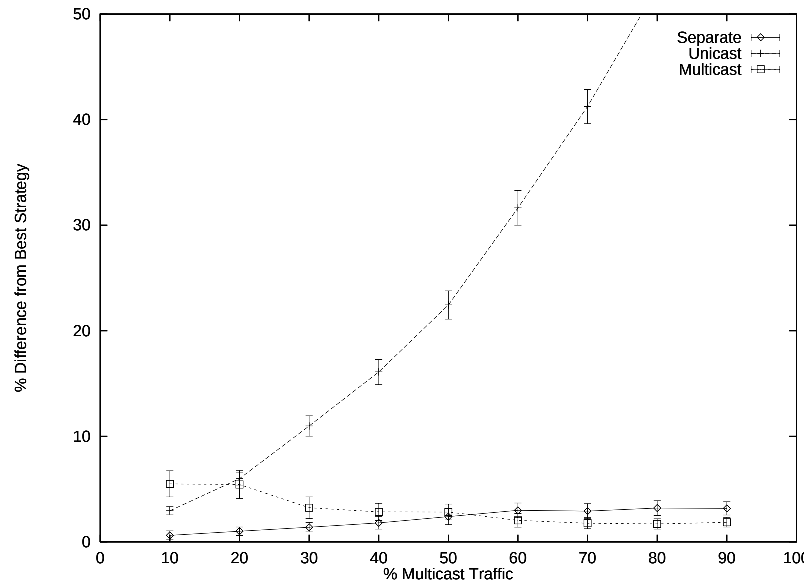

Figure 7.6: Comparison of strategies for $N = 20, G = 30, C = 5, \Delta = 4, \bar{g} = 0.25N$

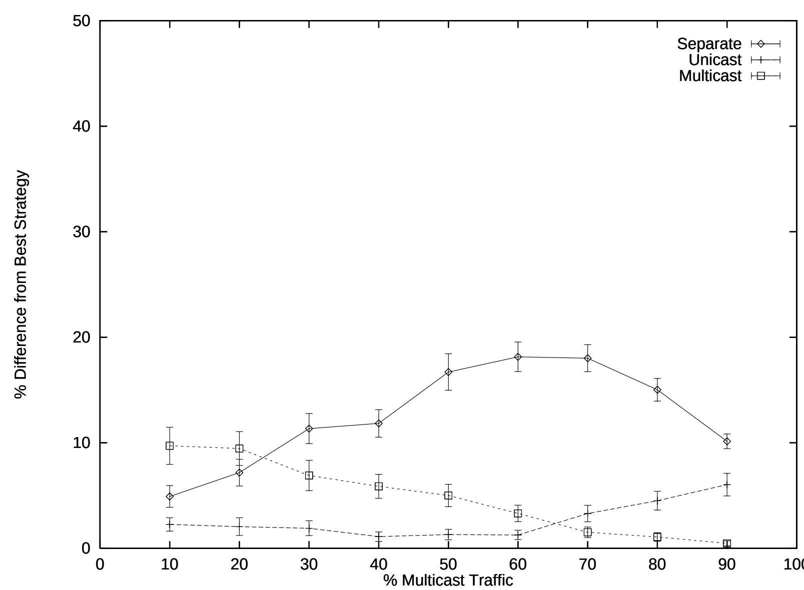

Figure 7.7: Comparison of strategies for $N = 20, G = 30, C = 15, \Delta = 4, \bar{g} = 0.25N$

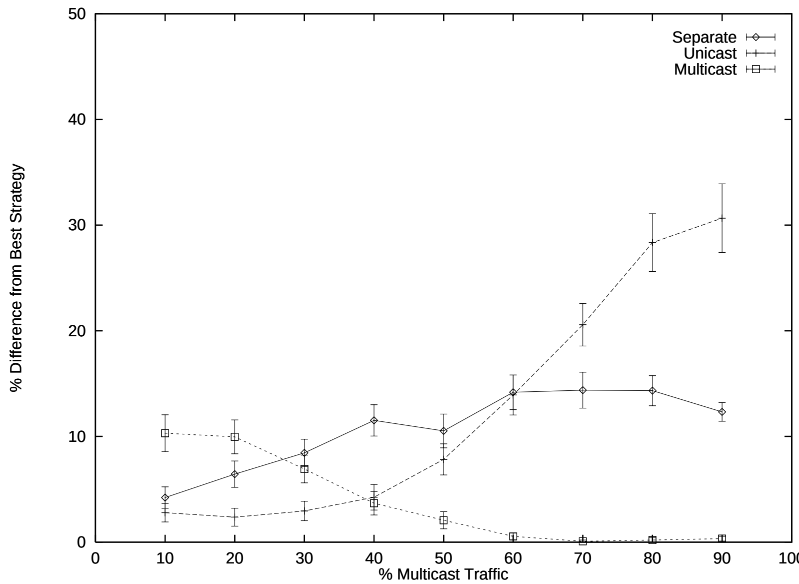

Figure 7.8: Comparison of strategies for $N = 20, G = 10, C = 10, \Delta = 4, \bar{g} = 0.25N$

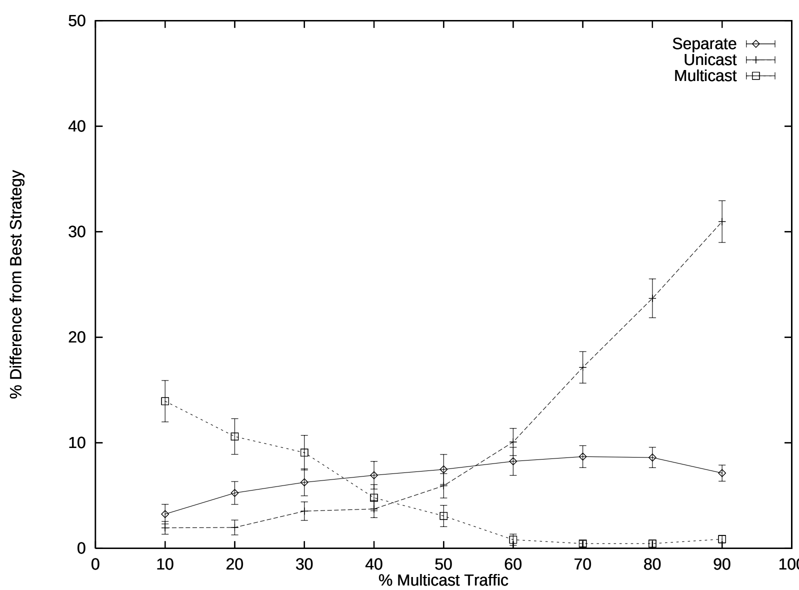

Figure 7.9: Comparison of strategies for $N = 20, G = 20, C = 10, \Delta = 4, \bar{g} = 0.25N$

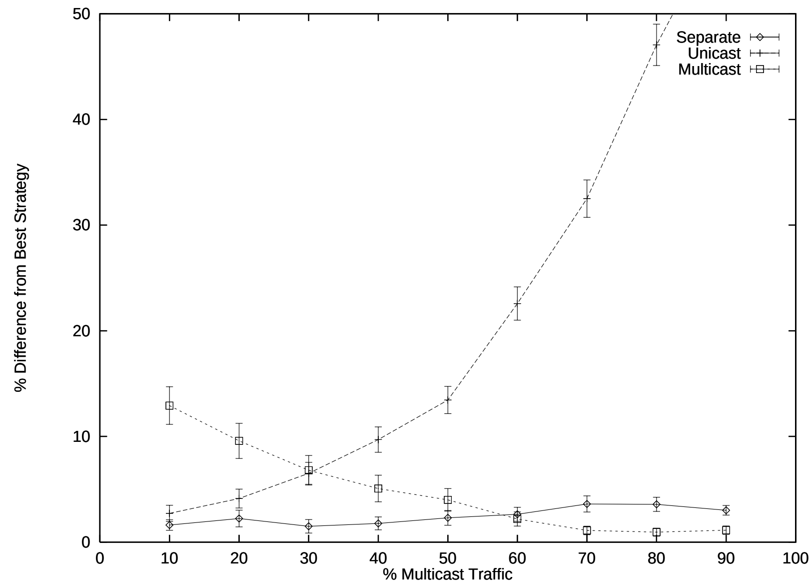

Figure 7.10: Comparison of strategies for $N = 30, G = 30, C = 10, \Delta = 4, \bar{g} = 0.25N$

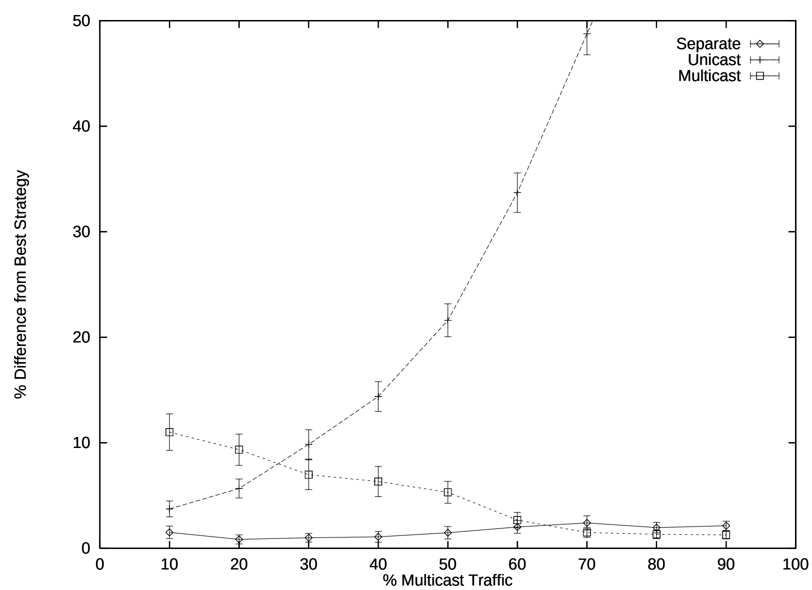

Figure 7.11: Comparison of strategies for $N = 40, G = 30, C = 10, \Delta = 4, \bar{g} = 0.25N$

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

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

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

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

Unicast Traffic Treated as Multicast Traffic. This strategy is not the best choice when we have a large amount of unicast traffic (compared to multicast traffic). For small values of s, it starts as the worst strategy, but it becomes the best one for larger values of s. As expected, where there is a significant amount of unicast traffic, most of the nodes in a virtual receiver set sit idle listening to transmissions that are not relevant to them. We note, however, that even for small amounts of multicast traffic (small s), its performance is not significantly worse than that of the best strategy. This behavior can be explained by the fact that the number k of virtual receiver sets formed is about the same as the number of channels C in the system. Maximum throughput in a system with unicast traffic only is bounded by the number of channels C (only C nodes can be receiving unicast transmissions simultaneously). Therefore, when $N > C$, receivers will have some idle time when unicast traffic is transmitted regardless of the use of virtual receivers. Changing any of the other parameters did not affect the performance of this strategy significantly.

Table 7.1 summarizes the results presented in Figures 7.1–7.12. The table shows the effect that increasing a parameter has on the length of the schedule obtained from each strategy.

Hotspots. Finally, in Figure 7.13 we show the behavior of the three strategies for the

Table 7.1: Behavior of strategies under varying parameters (↑: increase, ↓: decrease, —: no change)

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

hotspot pattern described earlier. Except for the unicast traffic matrix **U**, the remaining parameters are the same as those in Figure 7.1. We note that the results obtained in Figure 7.13 are not different from those in previous figures where all nodes were identical (no hotspots). This result was observed for a wide range of values for the various system parameters. We conclude that, although the existence of hotspots will certainly affect the schedule length, it does not affect the relative performance of the various strategies.

7.2.2 Summary

In table 7.2, we present the percentage of time that each strategy produced a schedule of length within 5% of the best schedule, for various values of $\bar{g}$ and s and for all values of the other parameters N, G, C, and Δ [2]. Tables 7.3 and 7.4 present similar results for different values of N, G, and N, C, respectively The strategy that produced the shortest schedules in each case corresponds to the one with the highest percentage shown. A strategy whose schedule length was within 5% of the best schedule length was also considered to be the best strategy. The 5% margin, though somewhat arbitrary, provides us with an insight into the performance of the strategies. When deciding which strategy to implement in an actual system, we may settle for one that produces the shortest schedules under most conditions while producing schedules within 5% of the best under other conditions. Below, we discuss under what conditions each of the three strategies is best.

Separate Scheduling. Overall, separate scheduling is effective in producing short schedules. Compared to Strategy 3, this strategy is better when there is a larger amount of unicast traffic, when there are many multicast groups (G is large), and when the number of channels is small compared to the number of nodes in the network.

Multicast Traffic Treated as Unicast Traffic. Strategy 2 is best when there is a small amount of multicast traffic in the network and the size of the multicast groups is small (see

[2] Even though the relative amount of multicast traffic in the network, s, is influenced by the size of the multicast groups, $\bar{g}$, we separate these two quantities to show that they affect the results independently.

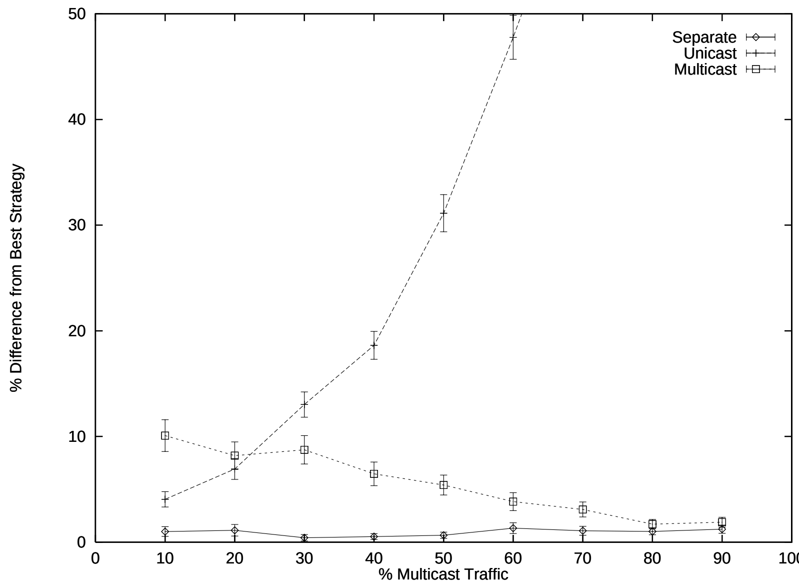

Figure 7.12: Comparison of strategies for $N = 50, G = 30, C = 10, \Delta = 4, \bar{g} = 0.25N$

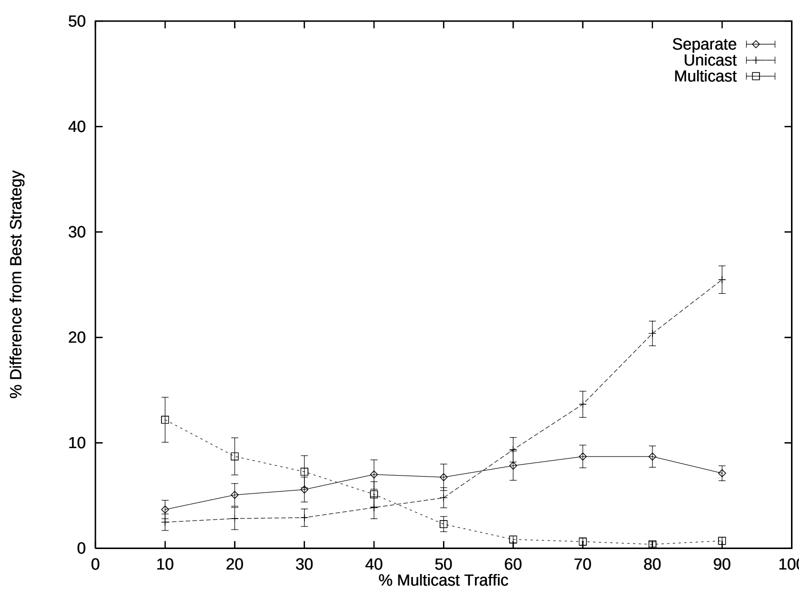

Figure 7.13: Comparison of strategies with hotspots for unicast traffic ($N = 20, G = 30, C = 10, \Delta = 4, \bar{g} = 0.25N$)

Table 7.2: Best strategy when $\bar{g}$ and s are varied

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

Table 7.2). This result is not surprising since replicating a multicast packet increases the requirements in the network and it can only be used efficiently in very limited situations. Also, this strategy is useful when the ratio of nodes to channels is small, i.e. N/C is close to 1 (see Table 7.3). In this case, the network operates in the tuning limited region.

Unicast Traffic Treated as Multicast. Strategy 3 produces schedules of short length in most situations. Even when the strategy does not produce the best schedule, the resulting schedule has a length no more than 20% larger than that of the best schedule (see Figures 7.1–7.13). Strategy 3 gives good results when G is small, i.e., $G \leq N/2$, when C is large, i.e., $C \geq N/2$, and when the amount of unicast traffic is small, i.e., $s \geq 40\%$.

7.3 Concluding Remarks

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

We presented three different strategies for scheduling a combined load of unicast and multicast traffic. These strategies are: separate scheduling, treating multicast traffic as unicast traffic, and treating unicast traffic as multicast traffic. As expected, multicast traffic should be treated as unicast traffic under very limited circumstances. More specifically, this strategy is useful only when there is a small amount of multicast traffic in the network and/or the multicast groups are small. On the other hand, if we treat unicast traffic as multicast traffic with a multicast group of size 1, the resulting schedule has a shorter length (when

Table 7.3: Best strategy when N and G are varied

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

Table 7.4: Best strategy when N and C are varied

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

compared with the schedules produced by the other strategies). This is the case especially when we have a large number of channels in the system, i.e. $C \geq N/2$ or when the number of multicast groups is small ($G \leq N/2$). Scheduling and transmitting each traffic separately also produces schedules of short length. Finally, one must also take into account memory and processing time limitations when considering which of the best two strategies to use. In particular, Strategy 3 requires storage for the $C \times (G+N)$ multicast traffic matrix when forming the virtual receiver sets, while for Strategy 1 the scheduling algorithms in [26] must be run twice, once for unicast traffic and once for multicast traffic.

Chapter 8

Conclusions and Future Research

The main contribution of this thesis consists of the examination of 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 utilization 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. Performance is measured in terms of the length of the schedule produced because it has direct effect on network throughput and average packet delay.

We also studied the sensitivity of the virtual receiver sets to changes in multicast traffic. Group membership changes do not add many slots to the schedule length. Eliminating a multicast group from the system or changing the bandwidth required by a multicast group does not have a profound impact on the lower bound of the schedule length either. So, using the original virtual receiver set calculated before the changes took place will not affect performance significantly. However, when a new multicast group is created or the bandwidth of a channel is changed, the schedule length increases dramatically. In these cases, recalculating the virtual receiver sets becomes necessary in order to improve performance. The virtual receiver set obtained will be tailored to the current network conditions but this approach requires processing and reconfiguration. Processing time, reconfiguration, and the length of the schedule produced contribute to the cost of handling multicast traffic changes. We presented an alternative approach that reduces the processing time of the recalculation and reconfiguration while minimizing the effect of the changes on the schedule length.

Finally, we studied strategies to schedule both unicast and multicast traffic since, in practice, we will find the combination in the network. Treating multicast traffic as unicast by replicating multicast packets to all recipients is not an effective strategy. The replication consumes many network resources causing the length of the schedule produced to be longer. Under very limited circumstances (when there is few multicast traffic compared to unicast traffic) this strategy produces a short schedule. Scheduling and transmitting unicast and multicast traffic separately produces short schedules under many network conditions. Obviously, applying optimization techniques to each component of the traffic we obtain an optimized scheduled. However, we showed that treating unicast traffic as a special case of multicast traffic also produces short schedules for a variety of configurations. Intuitively, using virtual receivers for unicast traffic will cause nodes to be idle at the times when packets are not addressed to them. However, since the network has more network nodes than transmission channels, some receivers will still be idle during unicast traffic transmissions if we use virtual receivers or not.

The study of multicast traffic in optical networks has just recently been addressed and there are many issues that remain to be examined. Below, we present several directions for future research in this area.

Multicasting in TT-FR systems: The focus of this thesis was on scheduling multicast traffic for FT-TR networks. In Tunable Transmitter - Fixed Receiver (TT-FR) systems, multicasting can only be accomplished by replicating the message to all the members of the group. In this environment, the problem is finding an allocation of wavelengths to receivers such that the number of transmissions of a multicast message is minimized. With the definition of virtual receivers introduced, we could extend this work to solve that problem. The number k of virtual receivers will determine the number of wavelengths needed in the network to support the multicast traffic. However, the number of wavelengths is limited by technology. The problem then becomes in finding the best partition of the receivers such that we have $k = C$ virtual receivers.

Admission Control: In this thesis we transform our original network with multicast traffic matrix **A** to an equivalent network with unicast traffic matrix **B**. This transformation increases the complexity of determining if nodes are complying with the bandwidth negotiation at connection setup or if the network can carry traffic for a new multicast group. Future work in this area requires the examination of the equivalent unicast traffic matrix **B**$= [b_{cl}]$ in Equation 5.2 to determine if it is still admissible. In addition, a modification of the approaches to handle dynamic multicast traffic may be necessary to allow new groups to be created.

Performance Analysis: The performance of the techniques presented in this thesis was evaluated in terms of the lower bound of the schedule length. However, the actual throughput depends on the unicast scheduling algorithm used. The algorithms of Rouskas and Sivaraman [25, 26] construct schedules of length equal to the lower bound when certain optimality conditions are satisfied. In other cases, the algorithms produce near-optimal schedules. So, the schedule produced does not degrade system performance significantly.

The performance of the techniques can be evaluated in terms of system throughput as defined by Modiano [18]. System throughput is the average number of multicast transmissions completed per slot per channel. This throughput is the inverse of the average number of transmissions required per successful multicast. The number of transmissions gives us an indication on the delay guarantees that could be provided with the techniques.

Another statistic of interest is the cell loss associated with replicating a packet for different virtual receivers. The use of a shared memory instead of independent finite capacity queues can be investigated in order to reduce loss.

Bibliography

[1] A. Aggarwal, A. Bar-Noy, D. Coppersmith, R. Ramaswami, and B. Schieber. Efficient routing and scheduling algorithms for optical networks. Technical Report RC 18967, IBM Research Report, 1994.

[2] 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.

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

[4] K. P. Birman and R. van Renesse. *Reliable Distributed Computing with the Isis Toolkit*. IEEE Computer Society Press, Los Alamitos, CA, 1994.

[5] 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.

[6] 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.

[7] E. Coffman, M. R. Garey, and D. S. Johnson. An application of bin-packing to multiprocessor scheduling. *SIAM Journal of Computing*, 7:1–17, Feb 1978.

[8] D. Dolev and D. Malki. The transis approach to high availability cluster communication. *Communications of the ACM*, 39(4):64–70, April 1996.

[9] N. R. Dono, P.E. Green Jr., K. Liu, R. Ramaswami, and F. F. Tong. A wavelength division multiple access network for computer communication. *IEEE Journal on Selected Areas in Communications*, 8(6):983–994, August 1990.

[10] E. Hall *et al.* The Rainbow-II gigabit optical network. *IEEE Journal Selected Areas in Communications*, 14(5):814–823, June 1996.

[11] I. P. Kaminow *et al.* A wideband all-optical WDM network. *IEEE Journal of Selected Areas in Communications*, 14(5):780–799, June 1996.

[12] L. E. Moser *et al.* Totem: A fault-tolerant multicast group communication system. *Communications of the ACM*, 39(4):54–63, April 1996.

[13] R. A. Barry *et al.* All-optical network consortium - ultrafast TDM networks. *IEEE Journal of Selected Areas in Communications*, 14(5):999–1013, June 1996.

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

[15] M. S. Goodman, H. Kobrinski, M. P. Vecchi, R. M. Bulley, and J. L. Gimlett. The LAMBDANET multiwavelength network: Architecture, applications, and demonstrations. *IEEE Journal on Selected Areas in Communications*, 8(6):995–1004, August 1990.

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

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

[18] E. Modiano. Unscheduled multicasts in WDM broadcast-and-select networks. In *Proceedings of INFOCOM '98*. IEEE, April 1998.

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

[20] B. Mukherjee. WDM-Based local lightwave networks Part II: Multihop systems. *IEEE Network Magazine*, pages 20–32, July 1992.

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

[22] G. R. Pieris and G. H. Sasaki. Scheduling transmissions in WDM broadcast-and-select networks. *IEEE/ACM Transactions on Networking*, 2(2):105–110, April 1994.

[23] M. Reiter. Distributing trust with the rampart toolkit. *Communications of the ACM*, 39(4):71–74, April 1996.

[24] 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.

[25] G. N. Rouskas and V. Sivaraman. On the design of optimal TDM schedules for broadcast WDM networks with arbitrary transceiver tuning latencies. In *Proceedings of INFOCOM '96*, pages 1217–1224. IEEE, March 1996.

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

[27] 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.

[28] A. S. Tanenbaum. *Modern Operating Systems*. Prentice Hall, Englewood Cliffs, NJ, 1992.

[29] J. Touch, K. Young, and J. Berthold. Bridging the gap between optical networks and the internet: A mini-workshop. In *http://www.isi.edu/~workshop/oi97*, October 1997.

[30] S. Tridandapani and J. S. Meditch. Efficient node placement for supporting multipoint connections in multihop networks. In *Proceedings of ICCCN '94*, pages 385–390, September 1994.

[31] S. Tridandapani and B. Mukherjee. Multicast traffic in multi-hop lightwave networks: Performance analysis and an argument for channel sharing. In *Proceedings of INFOCOM '96*. IEEE, March 1996.

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

[33] R. van Renesse, K. P. Birman, and S. Maffeis. Horus: A flexible group communication system. *Communications of the ACM*, 39(4):76–83, April 1996.

[34] B. W. Waxman. Routing of multipoint connections. *IEEE Journal on Selected Areas in Communications*, 6(9):1617–1622, December 1988.

Appendix A

A Branch-and-Bound Technique for VRSP

Searching through the state space of *VRSP* (i.e., the set of k-virtual receiver sets, $k = 1, \cdots, N$) to identify the receiver set with the lowest overall bound would require the examination of $\mathcal{O}(N!)$ combinations. We developed a branch-and-bound technique to prune the search tree to find the optimal solution of the problem. Although we do not expect the branch-and-bound technique to be useful in practice, we hope that it will be significantly faster than exhaustive enumeration, making it possible to compare heuristic solutions against the optimal solution (as opposed to the lower bound in (5.12) which is not tight) for reasonable network sizes. We are also motivated by a theoretical interest in the existence of such a technique.

Our branch-and-bound technique consists of two parts. First, we have developed a procedure to enumerate the state space starting with the N-virtual receiver set $\{\{1\}, \cdots, \{N\}\}$, and constructing disjoint subsets of the state space by applying only the $JOIN(\mathcal{V}^{(k)}, 1)$ operation. Secondly, we have a way of eliminating entire subsets of the state space from consideration, as follows. When we construct a new virtual receiver set, we compare its receiver bound to the lowest overall bound encountered so far. If the receiver bound is greater than or equal to the overall bound, we abandon further examination of the subset of virtual receiver sets that can be obtained from this set by applying the *JOIN* operation (because we know that the *JOIN* operation will not decrease the receiver bound), and backtrack to another virtual receiver set. Since this bounding technique is straightforward, in the following we describe in detail our procedure to enumerate all virtual receiver sets using only *JOIN* operations.

We represent a virtual receiver set by the tuple $(x_1, \cdots, x_N)$, where x_i represents the virtual receiver to which physical receiver i belongs. Hence, when there are k virtual receivers, we have that $x_i \leq k$, for all i. To enumerate all virtual receiver sets, we start from the N-virtual receiver set $(1, 2, \cdots, N)$, and repeatedly apply the $JOIN(\mathcal{V}^{(k)}, 1)$ operation. When joining two virtual receivers labeled l and $n > l$, the new virtual receiver is labeled l, and the labels of all virtual receivers labeled $p > n$ are decreased by one. Let us refer to Figure A.1 which shows the enumeration of virtual receiver sets when $N = 4$. When the virtual receiver set is (1234), we may join virtual receivers 1 and 2 to obtain virtual receiver set (1123); joining virtual receivers 1 and 2 once more, we obtain set (1112), and so on.

It is obvious from Figure A.1 that a certain k-virtual receiver set, $k < N - 1$, may be obtained by several $(k + 1)$-virtual receiver sets (e.g., (1112) can be obtained by either (1123) or (1213) after joining virtual receivers 1 and 2). To avoid examining a virtual receiver more than once, we adopt the following strategy. When exploring a k-virtual receiver set, we enumerate all $(k-1)$-virtual receiver sets that can be obtained from it using the *JOIN* operation *in increasing order of their labels.* For instance, in Figure A.1, the 3-virtual receiver sets obtained from (1234) are listed from left to right such that (1123) < (1213) < (1223) < (1231) < (1232) < (1233); similarly, the 2-virtual receiver sets obtained from (1123) are listed from left to right such that (1112) < (1121) < (1122). We continue the exploration of a k-virtual receiver set by recursively exploring each of its derived $(k - 1)$-virtual receiver sets, starting with the one with the smallest label and continuing in increasing order of labels. At each step, we keep a count of the highest label of any k-virtual receiver set, $k = 1, \cdots, N - 1$, we have already explored. If a k-virtual receiver with a label less than or equal to this highest label is encountered, we do not consider it any further, since we are guaranteed to have seen it before. Returning to Figure A.1, to explore (1234) we generate all 3-virtual receiver sets, and continue recursively with (1123) by generating the 2-virtual receiver sets (1112), (1121), and (1122). We then explore (1112), generate the 1-virtual receiver set (1111), and stop, since we have reached $k = 1$. We go back to explore (1121) and again generate (1111), but this time the receiver and channel bounds are not evaluated for (1111) since we know we have already encountered it. Similarly, after completing the exploration of (1123), the highest label of a 2-virtual receiver we have encountered is (1122). Therefore, when the 2-virtual receiver set (1112) is

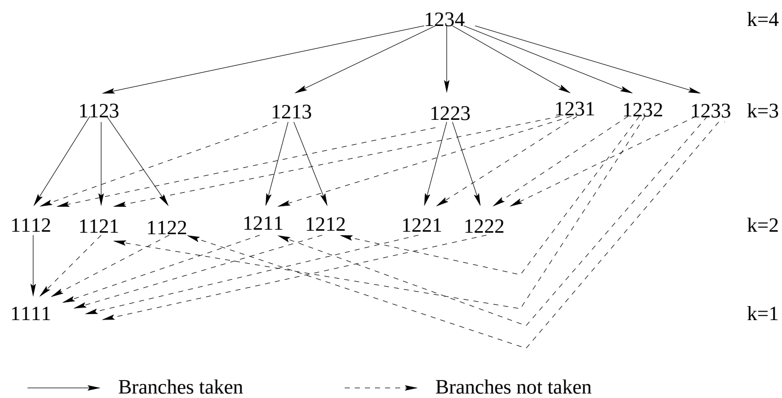

Figure A.1: Enumeration of the virtual receiver sets for $N = 4$

generated while exploring (1213), we also stop since (1112) < (1123).

The steps just described correspond to a preorder traversal of the tree defined by the solid arrows in Figure A.1, and completely enumerate all possible virtual receiver sets. The dotted arrows in the figure correspond to branches not followed due to the fact that the virtual receiver set to which they point has already been explored.

Appendix B

An Alternative Transmission Strategy for Combined Unicast and Multicast Traffic

We discussed in Chapter 7 three different strategies to schedule combined unicast and multicast traffic. We describe and analyze in this chapter an alternative transmission strategy where the virtual receiver sets are computed using only information from the multicast traffic matrix. Since multicast traffic can potentially require a lot of bandwidth, optimizing its transmission may seem more important than optimizing the transmission of unicast traffic. We explore this alternative to determine if ignoring unicast traffic will make a difference in the length of the schedule produced. The next section discusses in detail the strategy and presents the lower bounds similar to those developed for the other strategies. We then present numerical results with a discussion of the impact of the different parameters on the strategy.

B.1 Strategy 4: Adding Unicast Traffic to the Virtual Receivers

Our last approach to transmitting combined unicast and multicast traffic consists on a variation of Strategy 3. Specifically, this approach consists of three steps. First, a $k^{\star}$-virtual receiver set is obtained by applying the multicast optimization techniques to matrix

M: $\mathcal{V}^{(k^\star)} \leftarrow VR(\mathbf{M}, \Delta)$. Then, a new $C \times k^\star$ traffic matrix $\mathbf{B}^{(4)} = [b_{cl}^{(4)}]$ is constructed by adding all traffic (unicast and multicast) to each virtual receiver. That is, each element $b_{cl}^{(4)}$ of the new matrix is computed as:

$$b_{cl}^{(4)} = \sum_{j:j \in V_l^{(k^\star)}} a_{cj} + \sum_{g:g \cap V_l^{(k^\star)} \neq \phi} m_{cg} \tag{B.1}$$

Finally, the unicast scheduling algorithms are applied to matrix $\mathbf{B}^{(4)}$: *Sched*($\mathbf{B}^{(4)}$, Δ). This strategy differs from Strategy 3 in that the unicast traffic demands are not taken into account when constructing the $k^\star$-virtual receiver set $\mathcal{V}^{(k^\star)}$.

The channel bound for this strategy can be computed as follows (refer also to (5.5)):

$$\begin{aligned}
L_{ch}^{(4)} &= \max_{c=1,\cdots,C} \left\{ \sum_{l=1}^{k^\star} b_{cl}^{(4)} \right\} = \max_{c=1,\cdots,C} \left\{ \sum_{j=1}^{N} a_{cj} + \sum_{l=1}^{k^\star} \sum_{g:g \cap V_l^{(k^\star)} \neq \phi} m_{cg} \right\} \\
&\leq \max_{c=1,\cdots,C} \left\{ \sum_{j=1}^{N} a_{cj} \right\} + \max_{c=1,\cdots,C} \left\{ \sum_{l=1}^{k^\star} \sum_{g:g \cap V_l^{(k)} \neq \phi} m_{cg} \right\} \\
&= \hat{H}_{ch} + \hat{F}_{ch}(\mathcal{V}^{(k^\star)})
\end{aligned} \tag{B.2}$$

To obtain the receiver bound let us define, similar to Strategy 2, $T_l^{(4)}$ as the number of channels to which virtual receiver $V_l^{(k^\star)}$ has to tune to according to the new unicast traffic $\mathbf{B}^{(4)}$. We can then write $T_l^{(4)} = T_l' + y_l$, where y_l represents the additional tuning requirements of virtual receiver $V_l^{(k^\star)}$ due to the unicast traffic (matrix $\mathbf{B}^{(4)}$ differs from $\mathbf{B}$ in (5.2) by the amount of unicast traffic added to each element of the latter in (B.1)). The receiver bound for this strategy can then be written as:

$$\begin{aligned}
L_r^{(4)} &= \max_{l=1,\cdots,k^\star} \left\{ \sum_{c=1}^{C} b_{cl}^{(4)} + T_l^{(4)} \right\} \\
&= \max_{l=1,\cdots,k^\star} \left\{ \left[\sum_{c=1}^{C} \sum_{j:j \cap V_l^{(k^\star)} \neq \phi} a_{cj} \right] + \left[\sum_{c=1}^{C} \sum_{g:g \cap V_l^{(k^\star)} \neq \phi} m_{cg} \right] + T_l^{(4)} \Delta \right\} \\
&\leq \max_{l=1,\cdots,k^\star} \left\{ \left[\sum_{c=1}^{C} \sum_{j:j \cap V_l^{(k^\star)} \neq \phi} a_{cj} \right] + y_l \Delta \right\} + \max_{l=1,\cdots,k^\star} \left\{ \left[\sum_{c=1}^{C} \sum_{g:g \cap V_l^{(k^\star)} \neq \phi} m_{cg} \right] + T_l' \Delta \right\} \\
&= \max_{l=1,\cdots,k^\star} \left\{ \left[\sum_{c=1}^{C} \sum_{j:j \cap V_l^{(k^\star)} \neq \phi} a_{cj} \right] + y_l \Delta \right\} + \hat{F}_r(\mathcal{V}^{(k^\star)})
\end{aligned} \tag{B.3}$$

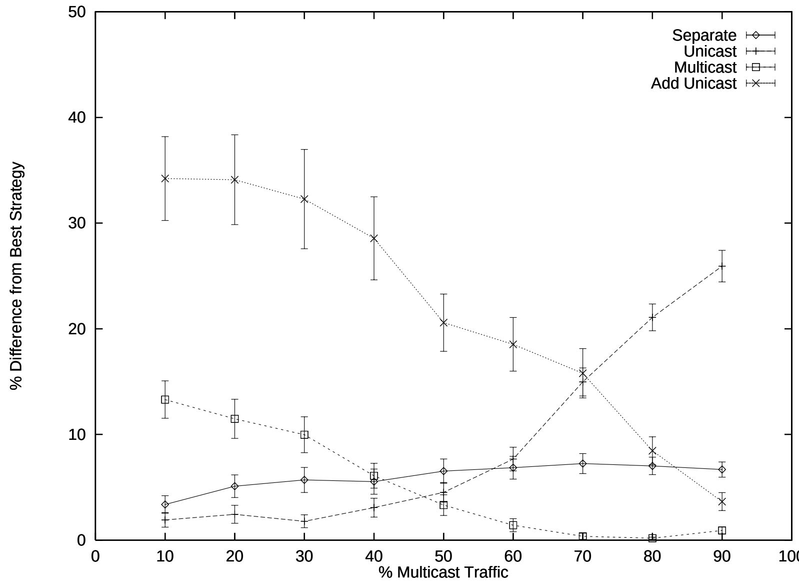

Figure B.1: Comparison of Strategy 4 for $N = 20, G = 30, C = 10, \Delta = 4, \bar{g} = 0.25N$

Unicast traffic is not taken into consideration when forming the virtual receiver sets in this strategy. Therefore, when there is a considerable amount of unicast traffic, the additional traffic for a virtual receiver will result in a schedule length larger than in the other strategies.

B.2 Numerical Results

We study the behavior of this strategy with the different network conditions used in Chapter 7 and described in Section 7.2. We also compute the quantity $D^{(4)} = \frac{S^{(4)} - S^{\star}}{S^{\star}} 100\%$, which indicates how far is the schedule length of this strategy, $S^{(4)}$, from the best one, $S^{\star}$. For the discussion of the results we plot again the results for the different strategies and vary some of the parameters in Figures B.1 – B.5. We refer to Strategy 4 in the figures as "Add Unicast".

For strategy 4 we note that as s is increased, $\bar{D}^{(4)}$ decreases. In fact, for very high values of s, this strategy produces schedules with short lengths that could be used for the transmissions. Since $\bar{a}$ decreases with increasing s, the unicast component of $L^{(4)}$ decreases as well. Also, since the length of the schedule for the multicast traffic component is optimal for that traffic, this strategy benefits from the decrease.

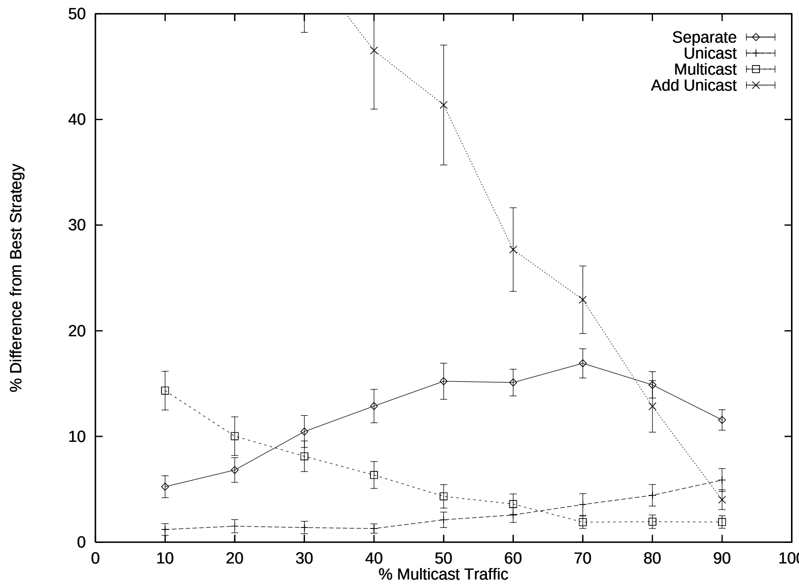

Figure B.2: Comparison of Strategy 4 for $N = 20, G = 30, C = 10, \Delta = 4, \bar{g} = 0.10N$

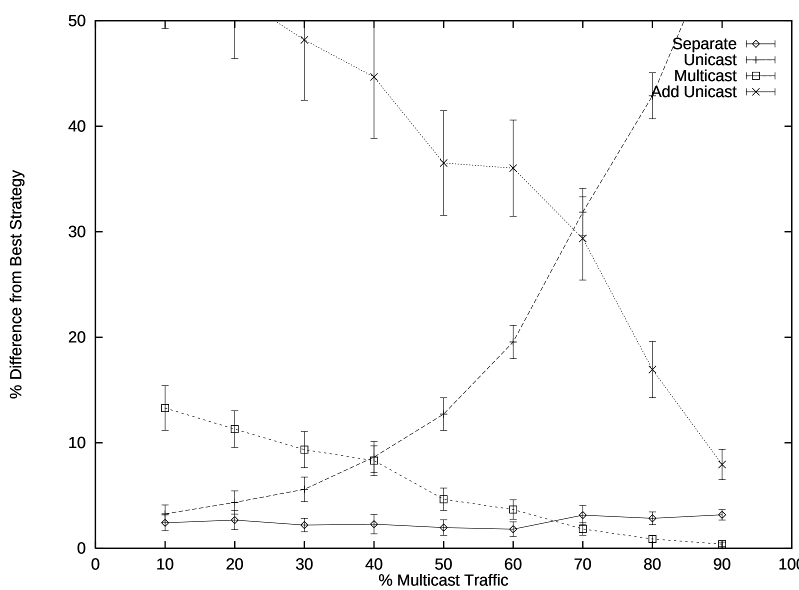

Figure B.3: Comparison of Strategy 4 for $N = 20, G = 30, C = 10, \Delta = 4, \bar{g} = 0.50N$

From Figures B.1 and B.3 we note that as $\bar{g}$ is increased, $\bar{D}^{(4)}$ increases. The increase in $\bar{D}^{(4)}$ is explained by the added unicast traffic to each virtual receiver set. Even when the virtual receiver sets were formed to balance the multicast traffic, the unicast traffic is not considered in this balance. With a bigger multicast group there is greater overlap. So, the $VR(\cdot)$ operation may try to keep groups together. This observation is valid for all the cases we observed except when $\bar{g}$ was increased from 2 to 5 nodes in average for $N = 20$ (Figures B.2 and B.1). In this case we observed a decrease in $\bar{D}^{(4)}$. This behavior is unique in the sense that there is a higher probability of having only one member in the multicast group. When the virtual receiver sets are formed, groups with one member are easily merged with other groups. However, the impact of the unicast component is not assessed at that point and another decision may work better. With an average of 5 members, there is more information to form the virtual receivers sets.

When G is increased (from $G = 10$ in Figure B.4 to $G = 20$ in Figure B.5 and then to $G = 30$ in Figure B.1), $\hat{H}$, $\hat{F}_{ch}(\mathcal{V}^{(N)})$, $\hat{F}_r(\mathcal{V}^{(1)})$ and $\hat{F}_{ch}(\mathcal{V}^{(1)})$ increase. Therefore, all the schedule lengths, $S^{(i)}$ are expected to increase. However, we note that $\bar{D}^{(4)}$ decreases in this case. We can explain this observation by understanding how are virtual receiver sets formed. Increasing the number of multicast groups, G, affects the formation of the virtual receivers in terms of the overlap. The greater G becomes, the greater the overlap between groups is. Therefore, when a node is added to a virtual receiver set, the added transmissions to the virtual receiver is less. Consequently, $\hat{F}_r(\mathcal{V}^{(k^\star)})$ is smaller and $\bar{D}^{(4)}$ decreases.

B.3 Concluding Remarks

Given the overhead required to carry multicast traffic, we may be tempted to focus on balancing multicast traffic through the use of virtual receivers. However, the fact that this strategy did not do well in any of the cases indicate that we can not ignore the unicast traffic in the network.

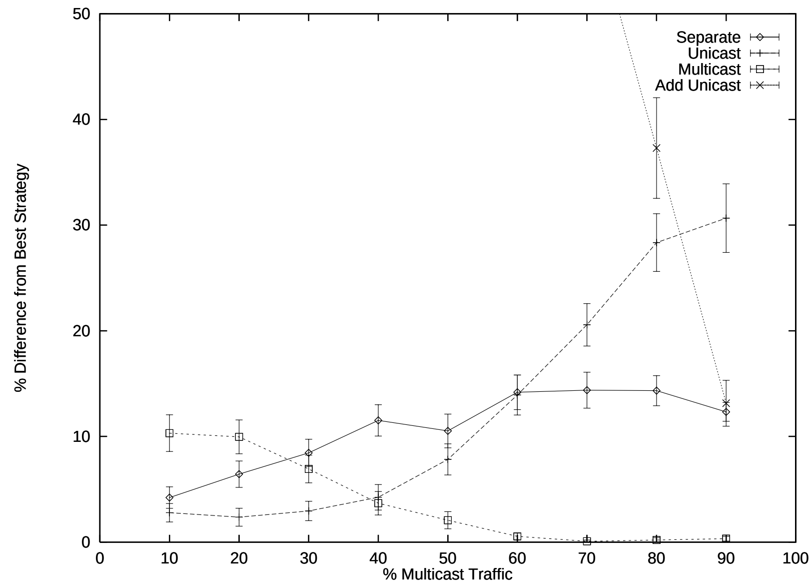

Figure B.4: Comparison of Strategy 4 for $N = 20, G = 10, C = 10, \Delta = 4, \bar{g} = 0.25N$

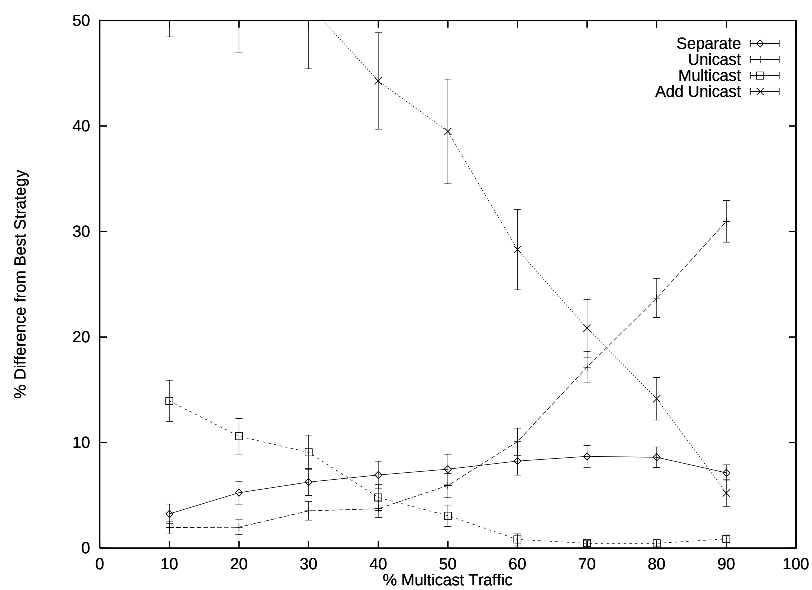

Figure B.5: Comparison of Strategy 4 for $N = 20, G = 20, C = 10, \Delta = 4, \bar{g} = 0.25N$