ABSTRACT

THAKER, DHAVAL V. Multicasting in a partially tunable broadcast WDM network..
(Under the direction of Dr. George N Rouskas.)

We consider the problem of scheduling multicast packet transmissions in a broadcast single hop WDM network. Tunablity is provided only at the one end, namely at the transmitter. Our objective is to schedule multicast transmission in a tunable transmitter and a fixed receiver broadcast WDM network. In a Single-hop WDM network having fixed receivers, the unicast and multicast traffic can be scheduled by a single scheduling algorithm. If so, the problem of scheduling multicast traffic, reduces to a Wavelength Assignment problem as to assign wavelengths to the fixed receivers before scheduling multicast packet transmission. A receiver-to-channel assignment has to meet two conflicting requirements. The first requirement is to minimize the number of retransmissions. The retransmissions are caused when members of a multicast group are assigned to different wavelengths and the group traffic is transmitted on each of these wavelengths. The second requirement is to maximize the channel utilization, to balance the incoming traffic optimally on all the available wavelengths. We address a fairly general version of the problem as we allow arbitrary traffic demands and arbitrary multicast group membership distribution. First, we define the Wavelength Assignment problem formally and prove it to be $\mathcal{NP}$-Hard problem. Since the problem is intractable in nature, next we develop different heuristics. The heuristics are evaluated based on their success in achieving the tradeoff between lower running time requirements and the accuracy of the obtained result to the optimum solution. Finally, we vary the different system parameters such as the number of nodes, channels and multicast groups and analyze their influence on the performance of the developed heuristics.

MULTICASTING IN A PARTIALLY TUNABLE BROADCAST WDM NETWORK.

by

Dhaval V Thaker

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

Computer Networking

Raleigh

2001

APPROVED BY:

H. G. Perros

Chair of Advisory Committee

BIOGRAPHY

Dhaval V Thaker was born in Bombay, India in 1975. He received his B. Engg degree in Electronics from Regional Engineering College at Surat in 1992. He worked as Software Engineer for Tata Infotech Ltd and Unisys Ltd in Pune, India and London UK, before joining Computer Networking department at North Carolina State University in 1999. He is currently working towards completion of his M.S in Computer Networking.

ACKNOWLEDGEMENTS

There are many people who have made the persuit of my Masters degree possible. I would first like to thank my parents and my sister for giving me the opportunity to achieve my goals and having stood by me in every decision I have made.

I would like to thank and express my deepest appreciation to my advisor Dr. George N Rouskas for his invaulable guidance and sincere help throughout my graduate study at North Carolina State University. I would also like to thank him for his continous support and advise in the completion of this thesis.

I would like to thank Dr. Harry Perros and Dr. Wenke Lee for their constructive comments and for serving on my thesis committee. A special thanks to friend Rudra Dutta who helped me review my work, encourage me and help me sort things out.

Contents

List of Figures **vi**

List of Tables **vii**

1 Introduction **1**
- 1.1 Optical Networks . . . 1
- 1.2 Single Hop Broadcast WDM Networks . . . 1
- 1.3 Motivation . . . 2
- 1.4 Thesis Organization . . . 3

2 Background and Related Work **5**
- 2.1 Multicasting in a WDM Network : Issues/Assumptions . . . 5
- 2.2 Classification of Multicast Scheduling Algorithms . . . 8
  - 2.2.1 Unicast Service . . . 10
  - 2.2.2 Multicast Service with No Fanout Splitting . . . 11
  - 2.2.3 Multicast with Fanout Splitting . . . 14
- 2.3 Combined Scheduling of Single- and Multi-Destination Traffic . . . 18

3 System Model **21**
- 3.1 Assumptions & Traffic Parameters . . . 21
- 3.2 Regions of Network operation . . . 24
  - 3.2.1 Transmission Phase . . . 24
  - 3.2.2 Reconfiguration Phase . . . 25

4 Problem Statement **27**
- 4.1 Formulation . . . 27
- 4.2 Discussion . . . 28
- 4.3 Solution Approaches . . . 30

5 Optimization Heuristics **32**
- 5.1 Greedy Join Heuristic . . . 32
- 5.2 Random Join Heuristic . . . 34
- 5.3 Greedy Split Heuristic . . . 35
- 5.4 Random Split Heuristic . . . 36

5.5 Multicast LPT Heuristic 37
5.6 Multicast LPT-Search Heuristic 38

6 Numerical Results **40**
6.1 Traffic generation 40
6.2 Discussion of Results 41

7 Summary and Future Research **50**
7.1 Summary 50
7.2 Future Research 50

Bibliography **52**

List of Figures

1.1 A Single-Hop Broadcast WDM Network 2

4.1 Monotonicity property of Channel Bound & Transmitter Bound 31

5.1 The *G-JOIN* heuristic 33
5.2 The *R-JOIN* heuristic 34
5.3 The *G-SPLIT* heuristic 35
5.4 The *MLPT* heuristic 37
5.5 The *MLPT-Search* heuristic 39

6.1 Heuristics Comparison for C=3 channels, G=5(Uniform Case) 42
6.2 Heuristics Comparison for C=3 channels, G=5(Uniform Case) 43
6.3 Heuristics Comparison for C=3 channels, G=5(Uniform Case) 43
6.4 Heuristics Comparison for C=10 channels, G=10(Uniform Case) 44
6.5 Heuristics Comparison for C=10 channels, G=10(Uniform Case) 45
6.6 Heuristics Comparison for C=10 channels, G=10(Uniform Case) 45
6.7 Heuristics Comparison for C=10 channels, G=30(Uniform Case) 47
6.8 Heuristics Comparison for C=10 channels, G=50(Uniform Case) 47
6.9 Heuristics Comparison for C=10 channels, G=80(Uniform Case) 48
6.10 Heuristics Comparison for C=3 channels, G=5(Non Uniform Case) 48
6.11 Heuristics Comparison for C=3 channels, G=5(Non Uniform Case) 49
6.12 Heuristics Comparison for C=3 channels, G=5(Non Uniform Case) 49

List of Tables

2.1 Classification of MSAs (CC: control channel, FT: fixed transmitter, TT: tunable transmitter, TR: tunable receiver) 10

6.1 Comparison of Heuristics,N=8,G=5,C=3 46

Chapter 1

Introduction

1.1 Optical Networks

Optical networks employing wavelength division multiplexing (WDM) are now a viable technology for implementing a next-generation network infrastructure that will support a diverse set of existing, emerging, and future applications [11]. WDM technology bridges the gap between the low electronic switching speeds and the ultra high transmission speeds achievable within the optical medium. WDM divides the enormous information carrying capacity of a single mode fiber into a number of channels, each on a different wavelength and operating at the peak electronic speed, making it possible to deliver an aggregate throughput in the order of Terabits per second. While WDM technology initially was deployed in point-to-point links, and has also been extensively studied, both theoretically and experimentally, in wide area or metropolitan area distances [10], a number of WDM local area testbeds have also been implemented [9] or are currently under development [8, 1]. To realize WDM local area networks, a passive star coupler is employed as a broadcast medium to connect all nodes in the network. Since the entire path between source and destination in such a network is entirely optical, and no electro-optic conversion of the signal is necessary, these networks are also known as *single-hop* WDM networks [19].

1.2 Single Hop Broadcast WDM Networks

In this thesis, we have assumed a Single-Hop Broadcast WDM network as our underlying architecture [6]. As seen in the Figure 1.1, an optical broadcast WDM network

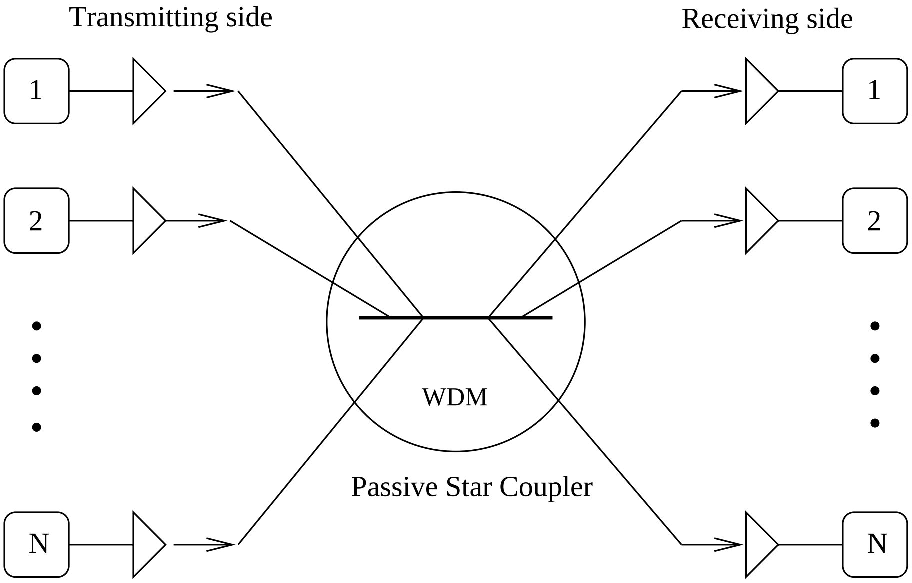

Figure 1.1: A Single-Hop Broadcast WDM Network

consists of a set $\mathcal{N} = \{1, 2, \ldots, N\}$ of nodes interconnected by a passive star coupler. This passive star coupler supports a set $\mathcal{C} = \{\lambda_1, \lambda_2, \ldots, \lambda_C\}$ of wavelengths. In a typical network, the number of channels C is at most equal to the number of nodes N, $C \leq N$. When the number of channels is strictly less than the number of nodes, we will say that the network is *wavelength (or bandwidth) limited.* Usually, each node is equipped with a number of either fixed tuned or tunable transmitters and receivers that can be used for data communication. For simplicity, we assume that the tunable components (either transmitters or receivers) can tune to, and transmit/listen on any of the C wavelengths. If the operation of the network relies on the presence of a control channel, then a separate pair of transceivers is required for every node. In general, this pair of transmitter and receiver is fixed tuned to the predetermined wavelength of the control channel, and cannot be used for data communication.

1.3 Motivation

Multicasting, the ability to transmit a message from a single source node to multiple destination nodes, has emerged as one of the essential features of current and future networks [2]. With the development of computer and communication applications such as distributed computing, audio and video conferencing, software and video distribution, and database replication, support for multicasting must be an an integral part of network design, rather than an afterthought, regardless of the network's underlying technology, data rates, or geographical reach. In a point-to-point network, a transmission by a node is received only by the node at the other end of the link. In a single-channel broadcast network, on the other hand, a transmission by a node is received by all the nodes attached to the channel. WDM broadcast networks occupy the center of the spectrum between these two extremes by providing a unique one-to-many transmission. Specifically, a transmission by a node in a broadcast WDM network on a given channel (wavelength) is received by *all* nodes listening on that channel at that point in time. This feature makes it possible to implement a number of different approaches for carrying multi-destination traffic in such a network, ranging from separately transmitting a copy of a message to each of its destination nodes, to transmitting multiple copies of the message with each copy received by a subset of the destination nodes, to transmitting a single copy of the message to all destinations at once.

The main challenge in the design of efficient multicast scheduling algorithms (MSA) for broadcast WDM networks is in exploiting the one-to-many transmission feature to provide a balance between two conflicting goals, namely, minimizing the number of copies of a message that need to be transmitted (a measure of the bandwidth efficiency of the algorithm) and maximizing concurrency (a measure of the ability of the algorithm to efficiently utilize the available wavelengths). Providing such a balance is important in order to achieve the maximum possible utilization of the channel, receiver, and transmitter resources within a network with multi-destination traffic [21, 20]. In the all of the literature surveyed in Chapter 2, the fundamental underlying assumption is presence of **tunable** receivers. The tunable receivers for a multi-destination communication appears to be a intuitive choice. In this thesis, we consider issues and problems faced when multidestination communication is done in the presence of **fixed** receivers.

1.4 Thesis Organization

The thesis is organized as follows. In Chapter 2 we cover background material related to scheduling of multi-destination traffic in a broadcast WDM network. Next, we survey the research done in the area and classify it. In Chapter 3 we first present a model of the broadcast WDM network with multi-destination traffic. We also define the performance measures relevant for a multidestination communication. We discuss various network parameters and assumptions that can affect the design of multicast scheduling algorithms in single-hop broadcast WDM network architecture. In Chapter 4 we formulate the problem of assigning fixed wavelength receivers to individual wavelengths, and show that the problem is $\mathcal{NP}$Hard. Next, we derive the lower bounds on a transmission schedule.The lower bound is dependent only on multidestination traffic and is independent of any receiver-to-wavelength assignment. The optimization heuristics are developed in Chapter 5 and in Chapter 6 we present numerical results. We then summarize our work, and point out directions for the future research in Chapter 7.

Chapter 2

Background and Related Work

In this chapter, we first discuss the design issues/assumptions, affecting scheduling of multidestination communication in broadcast WDM network. Next, we classify protocols and scheduling algorithms surveyed in the literature. The protocols are classified based on the underlying strategy used to transmit multicast packets, as well as on their assumptions regarding the network architecture. We discuss the advantages and disadvantages of each approach, and we also identify the regions of network operation for which each strategy is most appropriate. As multicast traffic has to co-exist with unicast traffic, we discuss schedule merging heuristics surveyed in the literature.

2.1 Multicasting in a WDM Network : Issues/Assumptions

In a broadcast WDM network, users contend for the resources including the data and control channels and the transmitters and receivers at the various nodes. Successful and efficient transmission of multicast packets requires careful coordination and scheduling of these resources. Some form of coordination is necessary because a transmitter and a receiver must both be tuned to the same channel for the duration of a packet's transmission. Also, the network must avoid or minimize packet loss due to *collisions*, which take place when two or more nodes simultaneously transmit on the same channel, and *destination conflicts*, which arise when two or more packets, each on a different channel, are addressed to a single receiver in the same slot. These issues become more difficult to tackle in the presence of multi-destination traffic in the network. Thus, at the heart of every media access control protocol for broadcast WDM environments is a scheduling algorithm responsible for coordinating access to the available channels.

The design of strategies or scheduling algorithms for carrying multi-destination traffic is strongly dependent on the underlying assumptions regarding the architecture and parameters of the broadcast WDM network. Differences in the issues ranging from the existence of a control channel to the Tunablity characteristics of each node to the tuning latency of the optical transceivers can result in radically different scheduling algorithms. For the rest of this section we take a closer look at the issues which can affect the design of algorithms for scheduling multi-destination packets in a broadcast WDM environment.

Number of transceivers per node and Tunablity characteristics. The number of transceivers per node used for data communication and their Tunablity characteristics can have a profound effect not only on the design but also on the performance of multicast scheduling algorithms (MSAs). In the literature surveyed, the transmitters can be either fixed tuned or tunable, but the receivers are always tunable. This node structure is not surprising given the fact that Tunablity at the receiving end can support multicasting in a natural and flexible manner by allowing a single packet transmission on a certain wavelength to be received by multiple destinations which have tuned their receivers to that wavelength. It has also been observed that the presence of multiple tunable receivers per node may significantly increase the maximum achievable throughput [4, 17, 18]. Employing additional tunable components allows the designer greater flexibility in the design of an MSA by alleviating the problem of destination conflicts (discussed below), which can be a severe one when multicast traffic is considered. On the other hand, having multiple tunable transceivers per node can increase the complexity of the MSA, especially when the tuning latency cannot be neglected. In this case, minimizing the effect of the tuning latency in the schedule involves careful coordination not only among the various nodes, but also among the various tunable transceivers at each node.

Tuning latency. While optical device technology has made great advances in the past few years, electronic speeds are also increasing to 10 Gigabits per second and beyond. Consequently, depending on the packet size and the data rate in the network, the value of the transceiver tuning latency relative to the packet transmission time can have a significant impact on the complexity of the MSA. If the tuning time is negligible compared to packet transmission time [27, 13, 3, 17, 5], it can be accounted for by including appropriate guard bands around the data packet within each slot. In this case, simpler preemptive scheduling algorithms can be employed, since, at the end of each slot, a transceiver can tune to a different wavelength without incurring any cost (i.e., without increasing the length of a schedule). If tuning latency is large, including it within each slot is inefficient and can lead to very long delays and low throughput. Thus, sophisticated MSAs that explicitly address the tuning latency are needed. Typically, non-preemptive algorithms are employed [21, 20] which prevent frequent retunings by having a transceiver tune to a wavelength and complete a number of packet transmissions/receptions before tuning to a different channel. The design of non-preemptive algorithms is more complex compared to preemptive ones. Non-preemptive algorithms also have higher running-time requirements, but they can effectively mask the tuning latency and thus significantly reduce the amount of time required to clear a set of traffic demands.

In-band vs. out-of-band signaling. Most broadcast WDM architectures that have appeared in the literature require the use of a control channel. The control channel is mainly used for the exchange of queue and traffic information among the nodes in the network, for slot reservation, as well as for other functions including network management and monitoring and global clock distribution. In general, one additional wavelength is required for the exchange of control information, and this wavelength cannot be used for data transmission. Systems with a centralized architecture, such as the ones in [13, 17], require two wavelengths for out-of-band signaling, one for sending and another for receiving control information from the scheduler. On the other hand, it is also possible to use in-band signaling that does not require a separate channel for control information. One example of an architecture which employs a distributed reservation protocol [25] to transmit the information needed by the MSA along with the transmission of data can be found in [21, 20].

Bandwidth allocation. Three different approaches have been proposed in the literature for allocating the bandwidth among the network nodes. In the *pre-allocation* approach [27], the channel bandwidth is divided into slots and slots are pre-allocated to each node. In each pre-allocated slot, two nodes tune their transmitter and receiver, respectively, to the same channel for communication. The slot pre-allocation can be fixed (i.e., independent of incoming traffic) or it could be dynamic to handle traffic variations. For dynamic pre-allocation, a schedule is computed by all the participating nodes in the network. Schedule computation can be distributed or centralized. This approach can generate large allocation tables and can be computationally intensive when there are many multicast groups and/or groups of large size.

In the *reservation-based* approach [5], nodes reserve the slots for transmission.

The reservation process is carried out on a separate control channel. Access to the control channel can be pre-allocated or random. The reservation-based approach may result in large packet delays for multicast packets with large destination sets. In the *random access* approach [17], nodes randomly access the channels to transmit packets. If there is a collision or destination conflict, nodes retransmit after a random or fixed interval, depending on the protocol. This approach is similar to conventional media access protocols such as Ethernet, and uses the broadcasting ability of the passive star coupler. This approach has the advantage of simpler scheduling compared to other approaches but it may lead to low throughput.

Centralized vs. distributed architecture. In a distributed architecture, schedule computation is performed by each node independently [3, 5, 23]. All the nodes share the necessary queue information and other traffic parameters using the control channel, and use the same algorithm to construct identical deterministic schedules. In a centralized architecture [13, 17], there is a single scheduler at the passive star coupler. This approach requires two control channels, one for sending control information to the scheduler and the other for receiving the schedule from it. In a centralized system, the scheduler knows the state of the network at any instant and it can schedule a retransmission immediately with out the overhead that is incurred in a distributed computation. The scheduler must continuously perform three tasks: receive a request, compute a schedule, and assign a slot for transmission to each node. Since these three tasks are performed by a single entity, as the number of channels and/or the data rate at which they operate increases, the load on the scheduler can be enormous. To reduce the processing requirement on the centralized scheduler, it is suggested in [13, 17] that very simple scheduling algorithm be employed. The centralized schedule computation approach is more suitable when the nodes are closely spaced (e.g., in a rack). For geographically distributed nodes, distributed schedule computation can reduce the control overhead and offers robustness against network failures.

2.2 Classification of Multicast Scheduling Algorithms

In this section we present and discuss the various approaches for scheduling multicast traffic in broadcast WDM networks that have appeared in the literature. We will use the framework introduced in [14] to classify the different Multicast Scheduling Algorithms (MSAs). This framework was developed in the context of an $N \times N$ multicast packet switch. We note that a packet-switched WDM network with N nodes and C wavelengths can be modeled as a bandwidth-limited $N \times N$ input-queued space-division switch operating in a time-slotted mode. The bandwidth limitation is due to the fact that the number of wavelengths available with tunable optical devices is smaller than the potential number of nodes ($C < N$). As a result, in each time slot, a maximum of C nodes may transmit their packets into the optical medium. On the other hand, a multicast switch with no bandwidth limitation ($C = N$) is potentially capable of switching packets from all N nodes (input ports) to their destinations (output ports) in one time slot. Using the terminology of [14], the strategies underlying the various MSAs can be classified in three broad categories. Note that the term *fanout* refers to the number of destinations of a multicast packet (i.e., the multicast group size).

1. **Unicast (sequential) service.** One copy of a multicast packet is separately transmitted to each of the destinations in the multicast group. Hence, the transmission of a packet takes at least as many slots as the number of destinations. This strategy results in high wavelength throughput but low multicast throughput (see Chapter 3), since essentially the same data packet is transmitted again and again.

2. **Multicast service with no fanout splitting.** Instead of transmitting a multicast packet to its destinations one at a time, another extreme is to insist that all destinations receive the packet in the same time slot. This strategy makes very efficient use of the bandwidth, since each multicast packet is transmitted exactly once. However, when the active multicast groups are not disjoint, this strategy can have poor performance in terms of both multicast throughput and delay.

3. **Multicast service with fanout splitting.** In between the above extreme strategies we have the multicast service discipline of fanout splitting, with better throughput and delay performance than either extreme. A multicast packet can be transmitted to more than one destination in a given time slot, depending on the availability of the destinations. The remaining destinations (if any) are served in later slots. In essence, the destination set of a multicast packet is partitioned into subgroups, and the packet is sequentially transmitted to each subgroup. A number of different fanout splitting strategies may be implemented based on the the manner in which the destination set is partitioned, and will be discussed later. We also note that this is the most

Schedule Computation	Tuning Latency	Multicast Service with No Fanout Splitting		Multicast Service with Fanout Splitting	
		Reference	Node Structure	Reference	Node Structure
Centralized	Zero	[13]	CC^2-FT^1-TR^1	[17]	CC^2-TT^1-TR^m
Distributed	Zero	[3]	CC^1-TT^1-TR^1	[15] [16]	CC^1-TT^1-TR^1
		[4]	CC^1-TT^n-TR^m	[18]	CC^1-TT^1-TR^m
	Arbitrary	[5] [26]	CC^1-TT^1-TR^1	[20] [21]	FT^1-TR^1

Table 2.1: Classification of MSAs (CC: control channel, FT: fixed transmitter, TT: tunable transmitter, TR: tunable receiver)

general service strategy, since unicast service and multicast service with no fanout splitting are special cases where the number of subgroups is equal to the fanout and one, respectively.

Finally, we note that the work in [7] also considered the problem of switching multicast traffic in a time-slotted switch with no fanout splitting. Specifically, it was shown that the problem of finding a conflict-free assignment of input queued packets to output slots so as to minimize the schedule length is $\mathcal{NP}$-hard. Consequently, it is not surprising that all the MSAs that have appeared in the literature are based on heuristics.

In Table 2.2 we classify the MSAs that have appeared in the literature according to the strategy they implement, and according to their assumptions regarding the underlying network environment. In the following subsections, we consider each strategy separately, and we discuss in detail the algorithms which appear in Table 2.2.

2.2.1 Unicast Service

With this strategy, one copy of a multicast packet is sent to each member of the packet's destination set, with each copy transmitted in a different time slot. The main advantage of this approach is that it makes it possible to employ unicast scheduling algorithms which have been extensively studied, are well-understood, and are significantly simpler and computationally more efficient than corresponding multicast scheduling ones. As pointed out in [14], an approximate analysis of this strategy can be carried out by analyzing the corresponding WDM network with unicast traffic and ignoring the batch arrivals of multicast packets. The main drawback is that this strategy does not take advantage of the one-to-many transmission feature of broadcast WDM networks. Consequently, unicast service may result in the transmission of a large number of copies leading to inefficient use of the available bandwidth, i.e., it achieves a low degree of efficiency, and thus low multicast throughput. It was shown in [23, 22] that unicast service is appropriate when the average multicast session is short and the average multicast group size is small relative to the number of nodes in the network. In such an environment, the total number of multicast packets in the network will be a small fraction of the unicast packets, and the overhead of implementing and running a specialized MSA may not be justified.

2.2.2 Multicast Service with No Fanout Splitting

The multicast protocols presented in [5, 3, 26, 13] all use MSAs that implement multicast service with no fanout splitting. Under this strategy, the source of a multicast packet insists on transmitting a single copy of the packet in a time slot which guarantees that all members of the packet's destination set will receive it. In other words, the algorithms require the simultaneous availability of all three network resources involved in the transmission of a packet, namely, one transmitter of the source node, one receiver at each of the destination nodes, and one channel. While achieving the highest possible degree of efficiency, usually these algorithms achieve low wavelength throughput, and thus low multicast throughput [21]. The performance of multicast with no fanout splitting was studied in [4]. By making a number of *protocol-free* assumptions, namely, a distributed transmission protocol with no control overhead, collision-less transmission, and no propagation delay on the control channel, an analytical model was developed to determine the performance limits of the network. For the model, tuning latency is assumed to be zero, packet arrivals are taken to be Poisson, and packet lengths are exponentially distributed (note that this work is the only one to assume variable size packets). Each node has a buffer that can hold exactly one packet, and packets that cannot be immediately transmitted to all nodes in their multicast group are dropped. With these assumptions, the network was modeled as a birth-death queuing system, and expressions for throughput and packet drop probability were obtained. It was shown in [4] that while wavelength throughput is low in such a network, *receiver throughput*, defined as the average number of busy receivers, can be higher. The latter result is due to the fact that multiple nodes are involved in (i.e., receive) each packet transmission. The results are in agreement with [23, 22] were it was shown that multicast with no fanout splitting works well only when the average multicast session is long and the average multicast group size is comparable to the number of nodes in the network (i.e., a broadcast or nearly broadcast scenario).

The four MSAs [5, 3, 26, 13] share a number of assumptions regarding the underlying network environment. Specifically, they all assume the presence of a control channel and the availability of tunable transmitters and receivers for data communication. The protocols differ in their assumptions regarding the tuning latency, the mechanism used to access the control channel and the details of operation of the MSA, as discussed in the following subsections.

TDMA Access to the Control Channel

The multicast protocol in [5] uses TDMA in the control channel. Each node accesses the control channel in a round-robin fashion and transmits a control packet. The control packet contains a multicast address identifying the multicast group nodes. Upon receiving a control packet, all nodes in the network simultaneously run the MSA in a distributed and deterministic fashion to determine the time slot and the channel on which the source of the control packet will transmit to the multicast group. Since all nodes have access to the same information and run the same algorithm, they will compute the same schedule, and both the source and the intended receivers will know when and in what wavelength to tune for the multicast packet transmission to be successful. The MSA employed is relatively simple, and it is based on the earliest availability of all necessary resources: channel, transmitter and receivers. First, the earliest time T_r at which all the receiver nodes in the group become free is determined. Next, the earliest time T_s at which both the source transmitter and the channel on which it is currently tuned are free is computed. If both are free then a new transmission can be scheduled on this channel, avoiding a tuning delay at the transmitter. If the channel is busy but the transmitter is free, then T_s is computed as the earliest time that another channel becomes free. At time $t = \max\{T_s, T_r\}$ all the receivers in the multicast group tune to the channel to receive the multicast transmission. Note that both T_r and T_s are computed so as to account for the tuning time at the transmitter and receivers, thus, this algorithm can accommodate arbitrary transceiver tuning latencies. While computing the earliest times T_r and T_s, the algorithm reserves the receivers of a multicast group as they become free until all receivers in the group become available. This feature can significantly limit the achievable throughput since reserved receivers cannot be used for other communication. To improve the performance, a modification was suggested in [26]. The modified MSA, known as *Backtrack* MSA improves the throughput by scheduling additional multicast transmissions to some of the free receivers which are waiting for other busy receivers to become free.

The *Backtrack* MSA works as follows. First, the MSA in [5] is run to obtain a schedule as before. Now consider a new multicast request with source s and multicast group g. Instead of running the MSA to find T_r and T_s for this request, the current schedule is first searched for slots in which a transmitter of s and a receiver for each node in g are free (possibly waiting for some busy receiver(s) to become free). If consecutive slots with this property are found that can accommodate the request, then the schedule is modified to include the multicast transmission from s to g in these slots. By satisfying this request without increasing the schedule length, the *Backtrack* MSA improves network performance in terms of both average packet delay and throughput. Overall, however, wavelength throughput can be very low for both protocols [5, 26].

A different protocol and scheduling algorithm for the same problem is presented in [13]. Each node sends its multicast (and unicast) transmission requests to a central controller via a control channel, the controller computes the schedule and broadcasts it to all nodes, again on the control channel. The controller uses a slot decomposition technique, similar to that used in satellite-switched TDMA (SS/TDMA) systems [12] to construct a slot matrix (schedule) which defines how transmissions should take place within the slots. For purposes of scheduling, unicast packets are assigned a weight of 1, while multicast packets to a group of size k are assigned a weight of $1/k$. The slot matrix is constrained to have elements with values of at most 1. The slot decomposition algorithm constructs a slot matrix free of any conflicts, and such that a multicast packet is transmitted in a single slot (i.e., no fanout splitting).

Random Access to The Control Channel

The multicast protocol presented in [3] also employs an MSA that insists on transmitting a packet to all destinations in its multicast group, but it uses a different access method for the control channel. Time on the control channel is divided into two phases, a "contention" phase and a "contention-less" phase. During the contention phase, nodes use the slotted Aloha protocol to transmit reservations for multicast transmissions. A reservation is considered successful if all three conditions hold true: (1) the reservation does not collide with other requests in the contention phase of the control channel (no control channel collision), (2) the multicast group specified in the reservation does not have any nodes in common with the groups specified by reservations transmitted in previous slots within this contention phase (no destination conflict), and (3) the total number of previously successful reservations is less than the number of available data channels (no data channel collision).

If a node fails to reserve a transmission slot due to any of the above conflicts, it wins a slot in the "contention-less" phase of the *next* cycle of the control channel. Again, before allocating a mini-slot in the "contention-less" part in the next cycle, all of the above three conflicts should be resolved. Every node in the network monitors the control channel and is aware of the reservations that have been successful at any given time. Once a successful reservation is made, all the receivers in the multicast group tune to the transmitter node's wavelength; the algorithm assumes that tunable devices take a negligible time to tune to a different wavelength. Since it is assumed that the multicast transmission is completed in one slot, the control channel can become a bottleneck as it is necessary to incur the reservation overhead for each and every multicast packet.

2.2.3 Multicast with Fanout Splitting

When fanout splitting is used, the multicast group of a packet is partitioned in subgroups, and the packet is sequentially transmitted to each subgroup. This strategy can result in a dramatic improvement in network performance, since packet transmissions can take place whenever a transmitter of the source node and a receiver at one or more destination nodes are available, without having to wait for all receivers to become free. Two issues arise in this case: (1) how to split (partition) groups with common receivers, and (2) how to coordinate (schedule) the tuning of subgroups of receivers across the various channels. In the following subsections we consider three approaches that have appeared in the literature to address these issues.

Greedy Scheduling Algorithms

The work in [15] is based on the same architecture as in [5], with the exception that tuning latencies for the receivers are considered negligible. To improve the channel utilization of the network, [15] employs fanout splitting, and arranges the multicast transmissions in the schedule with the objective of minimizing the average receiver waiting time. The problem of scheduling multicast transmissions to subgroups of receivers is defined and referred to as the Multicast Partition Problem. Two greedy heuristics are then developed to solve the problem. The first heuristic, called the Earliest Available Receiver (EAR), schedules a transmission by the source to the first receiver which becomes free. If additional receivers become available during this transmission, a transmission by the source to these receivers is scheduled immediately after the completion of the first one. The second greedy approach, called the Latest Available Receiver (LAR), first schedules a transmission at the time the last receiver in a group becomes available. Next, LAR attempts to schedule earlier transmissions to other members of the group without creating any channel or receiver conflicts. A third variant, called the Best Available Receiver (BAR), combines EAR and LAR to obtain schedules that minimize the receiver waiting time. Though BAR constructs better schedules than either EAR or LAR, its running time is higher than the other two heuristics. All three heuristic MSAs make the assumption that receivers take negligible time to tune across channels.

A different approach was presented in [16], where the problem of partitioning the destination set of each packet and scheduling the transmissions so as to minimize the packet delay is studied. The problem is shown to be $\mathcal{NP}$-hard, and a heuristic is presented and compared to an algorithm with no fanout splitting. The main idea behind the heuristic is to schedule as many destinations as possible to receive the packet in the same slot. Simulation results indicate that partitioning the multicast group performs well when the network is not bandwidth limited. Otherwise (i.e., when the number C of channels is small compared to the number N of nodes), the no fanout splitting strategy can perform better in terms of packet delay.

Random Scheduling Algorithms

The work in [18] models a broadcast WDM network with N nodes and C wavelengths as a bandwidth limited time-slotted $N \times N$ switch, and extends the analysis first presented in [14] to obtain the saturation throughput when the nodes have one or more tunable receivers. It is assumed that tuning latency is negligible, and that a separate channel is used to carry relevant control information. The analysis considers a random selection policy at both the transmitting and the receiving ends. Specifically, at each time slot, a random set of C nodes is selected from among the N nodes and are allowed to transmit their multicast packets (note that since the performance parameter studied is saturation throughput, all the nodes are assumed to be constantly back-logged). Destination conflicts are also resolved in a random manner. In particular, if two or more nodes in a slot have packets for the same receiver, then the receiver selects one of the multicast packets destined for it with equal probability. By making the assumption that, when a node is selected to transmit, each of the nodes in its multicast group receives the packet with a constant probability independently of other receivers, a queuing model is developed from which the saturation throughput and average packet delay are obtained. The analysis is also extended to the case when each node has multiple receivers. It is shown that, if the number C of channels is small, then network performance is limited by insufficient bandwidth. However, if the number of channels is relatively large, performance is limited by the occurrence of destination conflicts, and thus, employing multiple receivers per node can significantly increase the throughput and decrease the average delay. The work in [17] also employs fanout splitting, and, as in [18], tuning latencies are taken to be negligible and nodes are constantly back-logged and may have multiple receivers. Unlike in [18], however, the algorithms are designed for a centralized architecture in which a master scheduler maintains complete information about the state of the network, and instructs transmitters and receivers to tune to the appropriate channels. At the transmitting end, the algorithm uses a random selection policy such that, when a node completes the transmission of a multicast packet (i.e., as soon as the packet is received by all members of its multicast group), another node, not currently in the middle of a multicast transmission, is randomly selected to transmit on this channel in the next slot. (Note that, since $C < N$, C nodes are involved in a multicast transmission during any given slot, while $N - C$ nodes are back-logged and waiting to start a multicast transmission.) Two transmission policies are considered and analyzed. In the first, a node repeatedly transmits a packet until it is received by all nodes in the multicast group. This policy has an important drawback. When several nodes have packets for the same receiver, a likely situation at high loads and for large multicast group sizes, the destination conflicts will persist over long periods of time, aggravating the head-of-line blocking effect and resulting in poor performance. To improve the situation, another transmission policy is proposed in [17]. Instead of continuously transmitting a packet, a node waits for a random delay between retransmissions. Since other nodes may access the channel between retransmissions, this policy alleviates the head-of-line blocking problem and achieves higher throughput.

The resolution of destination conflicts, an important issue in a multicast setting, is also considered in [17]. If two or more transmitters have packets for the same node, that node must decide which packet to receive. The conflict resolution algorithm may base its decision on traffic priorities, the time of arrival or the fanout size of the multicast packets, the amount of delay accumulated by the contending packets, etc. In [17] three conflict resolution policies are compared: one that randomly selects a packet, one that selects the packet with the earliest arrival time (FCFS), and one that selects the packet with the smallest number of (remaining) destinations. The intuition behind the last policy is that it maximizes the probability that a message will be released (received by all its destinations), thereby making way for a new message. Analytical and simulation results indicate that this policy performs better than either the FCFS or the random policies. As in earlier works, it is also shown that an improvement in performance is achieved when nodes have multiple receivers.

The Virtual Receiver Concept

The MSAs discussed in the previous two sections attempt to simultaneously solve the two issues that arise in fanout splitting, namely, the partitioning of the multicast groups and the scheduling of transmissions. Both issues are difficult to deal with, especially in the presence of non-negligible tuning latencies (note that all algorithms discussed so far ignore tuning latencies) and when receivers may belong to multiple multicast groups. Furthermore, all algorithms attempt to partition the destination set of each packet into subgroups, an approach that has two drawbacks. First, since each packet is considered independently of others, the algorithms may not achieve good performance for the network overall. Second, significant overhead is incurred when a partitioning and scheduling decision has to be made for each packet.

The virtual receiver concept was developed in [20, 21] as a novel way to perform fanout splitting that overcomes these problems. A *virtual* receiver $V \subset \mathcal{N}$ is a set of *physical* receivers that behave identically in terms of tuning. Thus, from the point of view of coordinating the tuning of receivers to the various channels, all physical receivers in V can be logically thought of as a single receiver. A virtual receiver set $\mathcal{V}$ is defined as a partition of the set $\mathcal{N}$ of physical receivers into a number of virtual receivers. Given a virtual receiver set $\mathcal{V}$, a multicast packet with destination set g must be transmitted to all virtual receivers $V \in \mathcal{V}$ such that V contains a destination of the packet (i.e., $g \cap V \neq \emptyset$). All receivers in V have to filter out packets addressed to multicast groups for which they are not a member, but they are guaranteed to receive the packets to all groups of which they are members.

In effect, a virtual receiver set transforms the original network with N transmitters, N receivers, and multicast traffic, to an equivalent network with N transmitters, $\mid \mathcal{V} \mid$ receivers, and unicast traffic. Thus, the virtual receiver concept decouples the problem of determining how many times a multicast packet should be transmitted (i.e., of partitioning the multicast groups) from the problem of scheduling the packet transmissions. As a result, one can take advantage of a wide range of algorithms that have been designed for unicast traffic, have well-understood properties, and which can handle arbitrary tuning latencies. For instance, [21] uses the algorithms developed in [24].

The work in [21] concentrates on the problem of optimally obtaining a virtual receiver set that maximizes multicast throughput, which is shown to be $\mathcal{NP}$-hard. Four heuristics of varying degree of complexity are then presented for selecting the virtual receivers so as to provide near-optimal performance. Since the virtual receiver set is selected by considering the total traffic demand to the network, this approach achieves better performance than is possible when each packet is considered independently of others.

2.3 Combined Scheduling of Single- and Multi-Destination Traffic

In any realistic environment, the network load will consist of a mix of single- and multi-destination traffic. This problem was specifically studied in [22, 27], and it has also been addressed by several other authors. This section discusses strategies for scheduling such a combined traffic load.

In [23], the observation was made that the scheduling algorithm to be used will depend on the relative amount of each type of traffic offered to the network. Three types of multicast traffic were identified, and it was suggested that different algorithms be used to schedule this traffic along with unicast traffic.

- **Type 1 multicast traffic** is such that the typical multicast session lasts for a short time, but the average multicast group size is large (a broadcast or nearly broadcast scenario). For this type of traffic it was suggested in [23] that all nodes in the network periodically tune their receivers to the same channel in the same slots (called *broadcast* slots) to receive multicast transmissions (nodes must then filter out transmissions not intended for them).

- **Type 2 multicast traffic** is such that both the typical multicast session and the average group size is small. In this case, it is suggested that multicast packets be replicated and transmitted to each destination separately.

- **Type 3 multicast traffic** is such that the duration of the average multicast session is long (the average group can be of any size). Since multicast traffic can be a significant component of the overall traffic, it was suggested that multicast packets be transmitted in special slots, called *multicast* slots. Multicast slots are defined for each group, and all nodes in a group tune their receivers to the same channel in the group's multicast slots to receive transmissions from a certain source. Adaptive multicast protocols were designed in [23] to dynamically allocate multicast slots so as to keep channel utilization at high levels.

The work in [27] defines a two-dimensional multicast threshold which is a function of the session duration and the group size, and which quantifies some of the ideas developed in [23]. The main conclusions regarding scheduling of a combined load of single- and multi-destination traffic are very similar to those in [23].

The following strategies have appeared in the literature for scheduling both single- and multi-destination traffic.

1. **Unicast Traffic as Special Case of Multicast Traffic.** Many protocols that have appeared in the literature, including [17, 18, 5], account for unicast traffic by allowing multicast groups of size one. By appropriately selecting a distribution of multicast group sizes, it is possible to study (analytically or via simulation) the performance of the network under a wide range of traffic scenarios. One of the advantages of this approach is that a single scheduling algorithm is used in the network. The strategy was extensively studied using simulation in [22], and it was found that it produces good schedules under a wide range of traffic scenarios and network parameters.

2. **Multicast Traffic Treated as Unicast Traffic.** As we mentioned above, replicating each multicast packet and separately transmitting it to each destination works well only for Type 2 multicast traffic [23]. This result was confirmed by the study in [22] where it was shown through comprehensive simulation results that this strategy is appropriate only under limited circumstances, namely, when there are few and short multicast sessions and the group sizes are small.

3. **Separate Scheduling of Unicast and Multicast Traffic.** With this strategy, two schedules are obtained, one for unicast traffic and one for multicast traffic. Each schedule is constructed by employing an appropriate unicast or multicast scheduling algorithm, respectively. A schedule for the overall traffic is then obtained by concatenating the two schedules. The main disadvantage of this approach is that two different algorithms must be run in order to compute the overall schedule. However, it was shown in [22] that this strategy produces short schedules, and thus, has good performance in terms of multicast throughput [21] in most network environments regardless of the specific mix of single- and multi-destination traffic.

4. **Schedule Merging Heuristics.** An alternative to separate scheduling, this strategy first constructs two schedules, one for unicast and one for multicast traffic, and then merges the two to obtain a schedule for the overall traffic. As shown in [23], careful merging of the two schedules can result in a schedule that can have good performance in terms of average packet delay and channel utilization. A somewhat similar approach [27] starts with a single schedule for unicast traffic, and appropriately modifies it to include multicast transmissions, by having several receivers tune to the same channel in a slot that was previously designated for unicast transmission.

In the next chapter, we discuss the system model discussed in the thesis.

Chapter 3

System Model

This chapter discusses the system model for a tunable transmitter, fixed received WDM broadcast network. It first introduces various traffic matrices and system parameters. Next it explains need for different network regions of operation and explains operations to be performed in each of them.

3.1 Assumptions & Traffic Parameters

We consider a passive star physical topology for an all-optical single-hop WDM network. There are N nodes in the network and each of the N nodes employs one transmitter and one receiver. The Broadcast and Select passive star coupler can support C wavelengths. As we assume the network to be *Wavelength-Limited*, $N \gg C$. The number of currently active multicast groups are represented by G. At any given time, G will be considerably smaller than the possible number of multicast groups which is equal to 2^N. As we consider multicast traffic in this chapter, we let $g \subseteq \mathcal{N} = \{1,2,\ldots,N\}$ represent the destination set (multicast group) of a packet.

All the network nodes are synchronized at the slot boundaries. The synchronization among nodes can be achieved by using Network Time Protocol. The network is packet switched, with fixed size packets. Time is slotted, with the slot time equal to the packet transmission time. The tuning latency is defined as the time taken by transceivers to tune from one wavelength to another. The tuning latency is not considered as a part of slot time as it is assumed to be larger then packet transmission time. Each Node has one rapidly tunable transmitter and one slowly tunable Receiver. A optical laser or filter is considered to be slowly tunable if the time taken to switch between the wavelengths is comparable to the packet transmission time. Slowly tunable optical component allows the network to be accommodate varying traffic demands unlike fixed optical components. It also offers significant cost savings compared to the rapidly tunable components. As Slowly tunable components can not be tuned during packet transmission, they can be taken off-line for tuning. As seen in Chapter 3.2, presence of the slowly tunable components can introduce a need for different regions of network operation.

Under the packet transmission scenario, we have matrix $\mathbf{M}[N \times G]$ as multicast traffic demand matrix. $\mathbf{M} = [m_{i,j}]$, where $m_{i,j}$ is the number of slots to be allocated for the transmission from the source transmitter N_i to the multicast group G_j. $m_{i,j} \geq 0 \forall i, j$, as it is not necessary for each transmitter to have a multicast traffic for each multicast group. The Total multicast traffic in the network can be given as,

$$M_t = \sum_{i=1}^{N} \sum_{j=1}^{G} m_{i,j},\ i = 1, \cdots, N,\ j = 1, \cdots, G \tag{3.1}$$

The multicast group membership matrix $[G \times N]$ is represented by $\mathbf{B} = [b_{i,j}]$ where $b_{i,j}$ represents the membership status of the node N_j to the multicast group G_i. Matrix $\mathbf{B}$ is a binary integer matrix, with $0 \leq b_{ij} \leq 1$. If $b_{ij} = 1$, implies node N_j is member of the multicast group G_i. If $b_{ij} = 0$, N_j is not member of the multicast group G_i.

The Node-to-Wavelength assignment matrix $[N \times C]$ is represented by $\mathbf{A} = [a_{i,j}]$. Matrix $\mathbf{A}$ is a binary integer matrix, with $0 \leq a_{ij} \leq 1$. If $a_{ij} = 1$, implies that the node N_i is tuned to wavelength C_j. For a valid wavelength assignment to the receivers, generated matrix $\mathbf{A}$ should meet the *Node constraint* and the *Channel Constraint* as defined in equations (4.8) and (4.9) in the Chapter 4.

The Transmission matrix $[N \times C]$ is represented by $\mathbf{T} = [t_{i,j}]$. It represents the number of transmission slots available to the source transmitter N_i to transmit on the wavelength C_j. Note that the transmission matrix $\mathbf{T}$ represents only multicast traffic. The combined transmission matrix can be obtained by adding unicast and multicast transmission matrices.

Once the transmission matrix is ready, a unicast scheduling algorithm such as [24] can be used to compute the transmission schedule. Thus problems of scheduling the packet transmission and wavelength assignment are separated. One can choose any scheduling algorithm,based on the various trade-offs such as schedule computation time, achieved throughput etc. The $[N \times C]$ transmission matrix $\mathbf{T}$ is computed by multiplication of the three matrices, i.e,

$$\mathbf{T}[N \times C] = \mathbf{M}[N \times G] \cdot \mathbf{B}[G \times N] \cdot \mathbf{A}[N \times C]$$

If receivers belonging to the same multicast group g_i, are assigned different wavelengths, than the transmitter has to retransmit the traffic to the group g_i on each of these wavelengths. The total multicast traffic due to retransmission is given by M_r as,

$$M_r = \sum_{i=1}^{N} \sum_{j=1}^{N} t_{i,j}, \qquad i = 1, \cdots, N,\ j = 1, \cdots, N \tag{3.2}$$

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

Let a *multicast completion* denote the completion of a multicast transmission of a packet to all receivers in its multicast group. We define the *multicast throughput* [21] D of the system as the average number of multicast completions per slot. This definition of throughput is independent of how multicast is actually performed (i.e., by performing a single or multiple transmissions), and thus is applicable to any network with multicast traffic. The multicast throughput [21] is related to wavelength throughput and the degree of efficiency through the expression: $D = S/\bar{l}$. As we can see, the multicast throughput [21] D combines both parameters S and $\bar{l}$ in a meaningful way, and it naturally arises as the performance measure of interest in a WDM network with multicast traffic. In the next section, we discuss the impact of using a slowly tunable components on the network operation and the need for reconfigurable networks. We also discuss the multicast traffic and its scheduling issues in such a reconfigurable lightwave network.

3.2 Regions of Network operation

With fast tunable transmitters and slowly tunable receivers, the broadcast WDM network operates in two distinct regions which we will refer to as the *Reconfiguration Phase* and the *Transmission Phase* of the network. The need for these two distinct regions can be explained as follows.

Initially we tune the slowly tunable receivers to the set of wavelengths which gives us optimal performance from the network for the expected traffic demand. Since the tuning time of the receiver is significantly large compared to the packet transmission time, the receiver can be assumed to a fixed tuned to the wavelength. This phase of network operation is identified as the *Transmission phase*. If receivers are fixed permanently to the wavelengths, the network would eventually give sub-optimal performance with variation in the multicast traffic. The fixed wavelength assignment might result in excessive retransmissions, lowering the *multicast throughput* [21] or possible under utilization of the wavelength. Also note that in TT-FR network, each retransmission is expensive as the transmitter has to tune to the different wavelength and transmit. Thus, to offer optimal network performance for varying traffic demands, we assume slowly tunable receivers instead of fixed receivers. The receivers are assumed to fixed for the packet transmission phase but are assumed to be tunable in the reconfiguration phase. In the reconfiguration phase the receiver-to-wavelength assignment is changed to meet the varying multicast traffic demands.

Thus we have reconfigurable single-hop networks with two distinct regions of operation as the reconfiguration phase and the transmission phase. The reconfiguration issues in a multi-hop lightwave WDM network due to slowly tunable components are discussed in [30]. The work in [31, 32] studies reconfigurable single-hop WDM network with unicast traffic demand. In the next sub-section, we study the issues that arises in the single-hop reconfigurable WDM network for the multicast traffic in both transmission phase and the reconfiguration phase.

3.2.1 Transmission Phase

In the transmission phase, the receivers are fixed tuned to a set of wavelengths. The assignment of the wavelengths to the receivers is done in the reconfiguration phase. As network is assumed to be *wavelength limited*, more than one receivers are fixed tuned to a single wavelength. The packet transmission on any given wavelength will be received by all the receivers tuned to that wavelength. If a unicast packet is transmitted, it will be accepted only by one node and will be rejected by rest of the nodes tuned to the same wavelength. If the multicast packet is transmitted, it will be accepted only by those nodes which are the member of the multicast group and the other nodes will reject the packet. Thus to receive the multicast traffic, receiver only needs to perform *multicast address filtering.* For the combined scheduling of the unicast and the multicast traffic, receiver processing is significantly reduced, compared to the schedule merging strategies discussed in the Chapter 2.3.

Similarly for a tunable transmitter in the transmission phase, to transmit the multicast traffic to a particular multicast group, it has to tune to $\bar{c}$ wavelengths. $\bar{c}$ is given as the total number of distinct wavelengths allocated to a multicast group, also $\bar{c} \leq C$. On each of the $\bar{c}$ wavelengths, transmitter will retransmit the multicast traffic for a particular multicast group. With $[N \times C]$ transmission matrix $\mathbf{T}$ for the multicast traffic matrix, it can be considered as a unicast traffic matrix and can be added to obtain a combined transmission matrix. The only possible difference could be that multicast transmission matrix will have more correlation compared to independent unicast transmission matrix. Any efficient unicast scheduling algorithm as [24] which can compensate for non-negligible tuning latency of the can be used to schedule combined unicast and multicast traffic.

Thus as seen above, with fixed transmitters and slowly tunable receivers, the multicast traffic does not require any functional change in the transmission phase. This allows the time-critical transmission phase of the network to be unaffected by the arbitrary multicast traffic variations. In the next subsection, we look at the issues to be considered in the reconfiguration phase.

3.2.2 Reconfiguration Phase

In the reconfiguration phase, the slowly tunable receivers are taken offline and are assigned to different wavelengths. The receivers can be retuned all together or a subset of the total can be taken offline for retuning. The new wavelength assignment should met following requirements as

1. The incoming traffic should be balanced among all the available wavelengths, i.e. to maximize the wavelength utilization for the given traffic demand.

2. The members of the same multicast group should be preferably assigned the same

wavelength i.e. to minimize the retransmissions caused by splitting of the multicast group on more than one wavelengths.

3. The number of retunings of the receivers from the old to the new wavelength assignment should be minimized.

In [31], it is proved that the problem of obtaining optimal wavelength assignment satisfying requirements (1) and (3) for the unicast traffic is $\mathcal{NP}$-Complete. Therefore, the problem to find optimal wavelength assignment satisfying all of the above three requirements for a combined unicast and multicast traffic will also be a $\mathcal{NP}$-Complete problem. The reader is referred to [31], for details on the heuristics meeting requirements (1) and (3) for the unicast traffic. For the rest of this thesis, we concentrate on the problem of meeting requirements (1) and (2) for the multicast traffic.

Thus, as seen in this section, the network with tunable transmitter and slowly tunable receiver offers us distinct advantages compared to other network architecture surveyed in the Chapter 2.2. First, in the transmission phase, unicast and multicast traffic can be merged in $O(1)$. No explicit schedule merging heuristics are required. Secondly, the processing done to accommodate varying multicast traffic is performed in the reconfiguration phase instead of in the transmission phase. This allows us to use computationally expensive heuristic in the reconfiguration phase, otherwise impractical to use in a time-critical transmission phase. Finally, this approach separates the problem of scheduling the multicast traffic to partitioning of multicast group [21]. This also provides the freedom of choosing any unicast scheduling algorithm in the transmission phase.

Chapter 4

Problem Statement

In this chapter, we define the wavelength assignment problem for the the multicast traffic in a tunable transmitter, slowly tunable receiver WDM network.

4.1 Formulation

The wavelength assignment problem for the multicast traffic can be defined as an Optimization problem. The terminology and symbols are as defined in Chapter 3.1.

Bounds on Schedule Length

Channel Bound

$$\mathcal{F}_{ch}^{(C)} = \max_{c=1,\ldots,C} \left\{ \sum_{i=1}^{N} t_{i,c} \right\} \tag{4.1}$$

Transmitter Bound

$$\mathcal{F}_{tr}^{(C)} = \max_{i=1,\ldots,N} \left\{ \sum_{c=1}^{C} t_{i,c} \right\} \tag{4.2}$$

Absolute Schedule Bound

$$\mathcal{F}_{min} = \max \left\{ \mathcal{F}_{tr}^{(1)}, \mathcal{F}_{ch}^{(N)} \right\} \tag{4.3}$$

Traffic Characteristic

Degree of Commonality

$$\epsilon_c = \max_{i=1,\ldots,G j=1,\ldots,G} \left\{ \sum_{k=1}^{N} (b_{ik} \oplus b_{jk}) \right\} \forall (i,j) \qquad (1 \leq \epsilon_p \leq N) \tag{4.4}$$

Degree of Participation

$$\epsilon_p = \max_{j=1,\ldots,N} \left\{ \sum_{i=1}^{G} b_{ij} \right\} \forall (i,j) \qquad (1 \leq \epsilon_p \leq G) \tag{4.5}$$

Schedule Characteristic

Degree of Retransmission

$$\epsilon_r = \left(\frac{M_r}{M_t}\right) - 1 \qquad (\epsilon_r \geq 0) \tag{4.6}$$

Degree of Load Balancing

$$\epsilon_l = \left(\frac{\mathcal{F}_{ch}^{(C)} \times C}{M_t}\right) - 1 \qquad (0 \leq \epsilon_l \leq C - 1) \tag{4.7}$$

Objective

A wavelength assignment matrix $\mathbf{A}[N \times C]$ which minimizes the schedule bound of the traffic, i.e. ,

$$\min \mathcal{F}_{sh}^{(C)}, \text{ where } \mathcal{F}_{sh}^{(C)} = \max\left\{\mathcal{F}_{ch}^{(C)}, \mathcal{F}_{tr}^{(C)}\right\}$$

Subject to:

Node Constraint

$$\left\{\sum_{c=1}^{C} a_{nc}\right\} \equiv 1,\ \forall(n, c) \tag{4.8}$$

Channel Constraint

$$\left\{\sum_{n=1}^{N} a_{nc}\right\} \geq 1,\ \forall(n, c) \tag{4.9}$$

4.2 Discussion

The input to the formulation are number of nodes N, number of channels C , number of multicast groups G, multicast traffic matrix $\mathbf{M}$, and multicast group membership matrix $\mathbf{B}$. The solution consists of the wavelength assignment matrix $\mathbf{A}$, which gives the lowest bound on the schedule length. Schedule bounds are computed from the transmission matrix $\mathbf{T}$. Two schedule length bounds are defined similarly as in [21] .The *Channel Bound* $\mathcal{F}_{ch}^{C}$ denotes the maximum number of packets transmitted on any single channel. If multicast group is split across more than one channel, $\mathcal{F}_{ch}^{C}$ includes the retransmitted slots. The *Transmitter Bound* $\mathcal{F}_{tr}$ identifies the transmitter, transmitting maximum multicast traffic. It indicates the transmission slots required by the transmitter.

In traffic characteristics, we attempt to classify the multicast traffic. *Degree of Commonality* ϵ_c is measure of correlation of members within multicast groups. As ϵ_c increases, the shared groups members between multicast group increases, eventullay increasing the partitioning of multicast groups and retransmissions. We expect lower bound on the schedule length to increase as ϵ_c is increased. *Degree of Participation* ϵ_p identifies the *hot-spot* receivers in group membership matrix $\mathbf{B}$. With increase in ϵ_c, the wavelength utilization, i.e. load balancing becomes difficult to achieve. This is because the wavelength assigned to the hot-spot receiver, will be carrying most of the multicast traffic M_t.This also results in *Channel Bound* remaining constant and not affected by number of channels C.

As seen from above, the schedule length bounds indicates the dominant channel and the dominant transmitters. To get the overall schedule characteristics, we define, *Degree of Retransmission* ϵ_r as a fraction of a multicast traffic, retransmitted due to splitting of the multicast group on more than one channel. The value ϵ_r dependent on a wavelength assignment matrix $\mathbf{A}$. When ϵ_r is zero, there is no retransmission in the network. There is no upper bound on the value of ϵ_r. We also define *Degree of Load Balancing* ϵ_l as a measure of the wavelength utilization. ϵ_l is measured as a fraction of the multicast traffic carried by the dominant wavelength compared to other wavelengths. When ($\epsilon_l = 0$), the multicast traffic is perfectly balanced across C wavelengths. Similarly when $\epsilon_l = (C - 1)$, multicast traffic is completely unbalanced across wavelengths. Note that this may occur when a receiver is member of all the multicast group.

The generated wavelength assignment matrix $\mathbf{A}$ should meet the system constraints defined in equation (4.8) and (4.9). *Node Constraint* states that any node should not be assigned to more than one wavelength. This constraint is required as we have assumed a presence of single pair of transmitter and receiver at each node. Similarly, *Channel constraint* states that all wavelengths should have at least one receiver to them. This constraint ensures that all available wavelengths are utilized in a wavelength limited WDM network. Having discussed the wavelength assignment problem, and its constraints and characteristics, we study the theoretical complexity of the problem. The next lemma states that, the problem of wavelength assignment is $\mathcal{NP}$-Complete.

lemma 1 *The problem of assigning wavelengths to receivers which will minimize retransmissions and maximize the wavelength in a multicast traffic is* $\mathcal{NP}$*-Complete.*

Proof: 1 *The wavelength problem is* $\mathcal{NP}$*-Complete because it reduces to* **2-VRSP** *problem which is* $\mathcal{NP}$*-Complete [21]. The number of channels* C *in the wavelength assignment problem an be mapped to the* Virtual receivers *in the* **VRSP** *problem. The wavelength assignment problem adds an additional constraint to the* **VRSP** *problem. the number of* Virtual Receivers *in* **VRSP**, *can be greater than number of available wavelengths* C *in the* *network but in the wavelength assignment problem, we can not make number of receiver groups greater than C. By imposing additional constraint to the problem, complexity of the problem is increased further. As the* **VRSP** *is proven to $\mathcal{NP}$-Complete, the wavelength assignment problem is also $\mathcal{NP}$-Complete.*

4.3 Solution Approaches

As the wavelength assignment problem is proved to be $\mathcal{NP}$-Complete, the polynomial time solution for the problem does not exists as of now. In this section we study different approaches for obtaining the approximate solution in polynomial time.

First, we construct set of heuristics exploiting **monotonicity** property of two schedule bounds $\mathcal{F}_{ch}^{C}$ and $\mathcal{F}_{tr}^{C}$ with respect to the number of wavelengths C in the network. As the number of wavelengths are increased in a WDM network, retransmissions to a multicast group increases. With increase in number of channels, probability of a multicast group getting split on more than one channel also increases. Thus, the *Transmitter bound* of the network will increase with C due to increase in retransmissions and the *Channel Bound* will decrease as there are more wavelengths available than before to distribute the load. In Figure 4.1, we demonstrate this through experimental results. We can use this monotonicity property of schedule bounds to construct set of heuristics as done in [21]. We can reach to the available number of wavelengths C, either by **JOIN** operation or by **SPLIT** operation. With the JOIN approach, we begin with N channels. By following a pre-defined rule, We identify a pair of wavelengths to be merged, and assign receivers assigned to this pair of wavelengths to a new single wavelength. We continue the JOIN operation, till we reach the available wavelengths in the system, C.

In the SPLIT approach, we start from the opposite directions than the JOIN approach. we initially start with $C = 1$ and we repeatedly split a wavelength to create two different wavelengths. We continue till wavelengths are equal to C. As in [21], we can apply *Greedy* and *Random* approach to the JOIN and the SPLIT operations. In the next chapter, we look at the heuristics exploiting the monotonicity property of the schedule bounds. This heuristics are *Greedy-Join*, *Random-Join*, *Greedy-Split* and *Random-Split.*

The second approach to the problem uses the load balancing algorithm for the multicast traffic. Note that just using the LPT will not give us valid assignment matrix **A**. When the multicast traffic is considered as a batch arrival of the unicast traffic, load balanc-

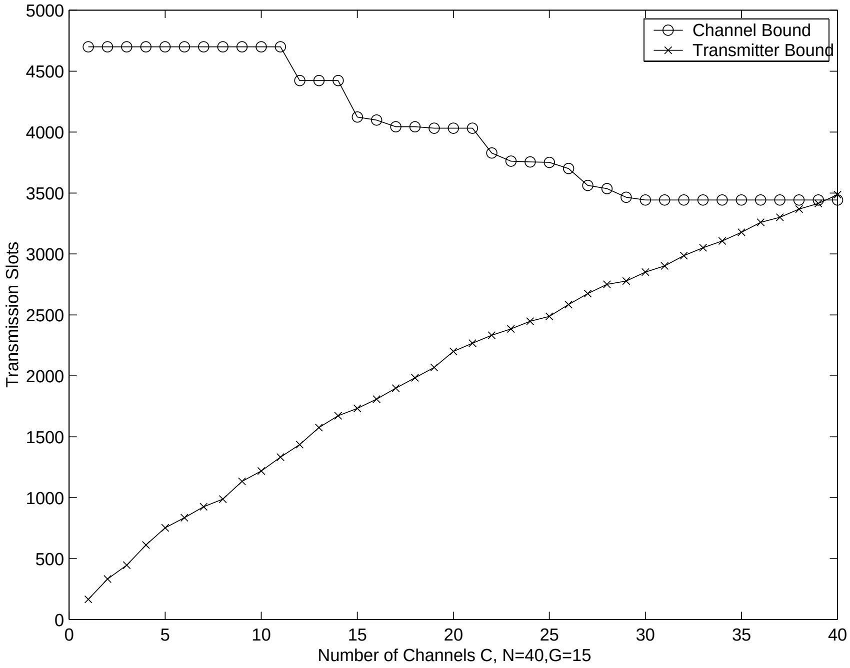

Figure 4.1: Monotonicity property of Channel Bound & Transmitter Bound

ing algorithm may assign members of the same multicast group to different wavelengths to distribute the load equally among all wavelengths. Such assignment would ignore the group membership among nodes and would not use the inherent capability of broadcast WDM network to support multicast communication. Thus, traditional load balancing algorithm needs to modified, when multicast traffic is to balanced across wavelengths. In the next chapter, we consider two heuristics *MLPT* and *MLPT-Search*, which modify the LPT [28] load balancing algorithm for multicast traffic.

In next chapter, we look at each solution approach and its set of heuristics in detail. We also do a time complexity analysis of each heuristic to evaluate the tradeoff between running time and the quality of the solution.

Chapter 5

Optimization Heuristics

In the previous chapter, we presented the formal definition of the wavelength assignment problem. As the wavelength assignment problem is $\mathcal{NP}$-Hard, an algorithm that can solve an arbitrary instance of the wavelength assignment problem in a polynomial time may not exist. The Heuristic provides a tradeoff between running time requirements and optimality of the solution. It is expected to perform well under general traffic matrices. In this chapter, we develop different heuristics, differing in their approach to the problem. Join and Split class of heuristics use monotonicity property of the channel bound and the transmitter bound to improve on an initial solution. MLPT heuristics uses LPT [28] load balancing algorithm to arrive at the approximate solution. In all of these approaches, we perform a preselected set of local operations, based on a certain rule, to improve on an initial solution. We continue this till no further local improvements can be made and "locally optimum" solution is found.

5.1 Greedy Join Heuristic

In the Greedy JOIN approach, Initially we assume N wavelength WDM network. Each receiver is placed on a one single wavelength, resulting in a $\mathbf{A}[N \times N]$ assignment matrix. This configuration resembles a point-to-point network and the Transmitter bound is expected to dominate the Channel bound, provided there is no hot-spot receiver. We then search for a pair of wavelengths which when merged, will offer the lowest transmitter bound among all other pair of wavelengths.

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

Input: N, C, G, Traffic matrix $\mathbf{M}[N \times G]$, Group membership matrix $\mathbf{B}[G \times N]$

Output: Wavelength assignment matrix $\mathbf{A}[N \times C]$

```
begin
    Set $W$ = Total number of nodes, say $N$;
    Assign a single wavelength to each receiver and
    construct $\mathbf{A}[N \times W]$ matrix;
    while $W \geq$ Total Number Of Wavelengths, say $C$;
        Apply Greedy-Join rule to the $\mathbf{A}[N \times W]$
        matrix to create a new $\mathbf{A}[N \times W - 1]$;
        $W \leftarrow W - 1$;
    end while
    Return the wavelength assignment matrix $\mathbf{A}[N \times C]$;
end
```

Figure 5.1: The *G-JOIN* heuristic

Greedy Join Rule :

Select a pair of wavelengths out of C wavelengths which when merged, gives the lowest transmitter bound $\mathcal{F}_{Tr}^{(C)}$among all other pairs of wavelengths.In case, there is more than one pair which achieves the minimum transmitter bound, select the pair which gives the minimum channel bound $\mathcal{F}_{Ch}^{(C)}$. If again there is a tie, break it arbirarly.

Time Complexity analysis :

First, considering the *Greedy Join* rule, to select a pair of channels out of C channels, would take $O(\frac{C \times (C-1)}{2}) \approx O(C^2)$ time. For each pair of wavelength selected, a matrix multiplication of complexity is $O(N^2.C)$ required, to compute $\mathcal{F}_{Tr}^{(C)}$. Assuming the worst case scenario i.e., when $W = 1$, the while loop in the Greedy-Join heuristic would be executed $(N-1)$ times. The overall running time of the heuristic is then $O(N^6)$. [1]

---

[1] This time-complexity analysis includes the binary matrix multiplication of the order of $O(N^3)$. Assuming the size of matrices $\leq$ the processor word size, it can take $O(1)$ time, reducing the effective running time of the heuristics to $O(N^3)$.

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

Input: N, C, G, Traffic matrix $\mathbf{M}[N \times G]$, Group membership matrix $\mathbf{B}[G \times N]$

Output: Wavelength assignment matrix $\mathbf{A}[N \times C]$

```
1.  begin
2.      Set $W$ = Total number of nodes, say $N$;
3.      Tune each receiver to a different wavelength and construct
        $[N \times N]$ matrix;
4.      while $W \geq$ Total Number Of Wavelengths, say $C$;
5.          Select a pair of wavelengths at random from the $\mathbf{A}[N \times W]$ matrix;
6.          Merge the selected channels to create new $\mathbf{A}[N \times W - 1]$
7.          $W \leftarrow W - 1$;
8.      end while
9.      Return the wavelength assignment matrix $\mathbf{A}[N \times C]$;
10. end
```

Figure 5.2: The *R-JOIN* heuristic

5.2 Random Join Heuristic

This heuristics is similar to G-JOIN. It also starts with assumption that N wavelengths are available in the network and that one channel assigned to each receiver. The main difference is that, the pair of wavelengths to be merged are selected randomly.

Time Complexity analysis :

Assuming it takes a constant time to generate two unique random numbers, step 5 of Random-Join heuristics, will take a constant amount of time. In the worst case, step 6 will take $\mathcal{O}(N \times (N-1))$ time to run, and while loop will be executed N times, giving us worst case upper bound of $\mathcal{O}(N^3)$.

This heuristics very well demonstrates the tradeoff involved between the running time complexity and the quality of the final solution. As seen from above, the Random-Join approach has low running time requirements compared to the Greedy-Join approach, but it gives inferior solution due to lack of intelligence in heuristics.

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

Input: N, C, G, Traffic matrix $\mathbf{M}[N \times G]$, Group membership matrix $\mathbf{B}[G \times N]$

Output: Wavelength assignment matrix $\mathbf{A}[N \times C]$

```
begin
    Set $W = 1$;
    Tune all the receivers to a single wavelength and construct
    $[N \times 1]$ matrix;
    while $W \leq$ Total Number Of Wavelengths, say $C$;
        Set $W^*$ to the wavelength carrying maximum traffic;
        Find two receivers say $(i, j)$ as having least number
        of common multicast groups;
        Apply the Greedy-Split rule to reassign nodes on
        $W^*$ to $W_1^*$ and $W_2^*$ to get the new $\mathbf{A}[N \times W + 1]$;
        $W \leftarrow W + 1$;
    end while
    Return the wavelegth assignment matrix $\mathbf{A}[N \times C]$;
end
```

Figure 5.3: The *G-SPLIT* heuristic

5.3 Greedy Split Heuristic

The Greedy-Split heuristic, takes a greedy approach to solve the wavelength assignment problem. It works from the opposite direction to that of G-JOIN. Initially, we assume single channel WDM network. All receivers are tuned to the single channel resulting in a $\mathbf{A}[N \times 1]$ wavelength assignment matrix. With a single channel, the WDM network is similar to a broadcast medium, and the Channel bound dominates over the Transmitter bound.

Once initialized, the G-SPLIT heuristic identifies the wavelength carying the maximum traffic. It splits the wavelength into two new different wavelength, and distributes the traffic from old wavelength to these new wavelengths. This effectively lowers the Channel Bound F_{Ch} for new wavelength assignment. With each splitting of the wavelength, we reassign the nodes from the old channels to two new channels in a way that reduces retransmission traffic, (degree of retransmission ϵ_r). Another aspect where G-SPLIT differs from the other heuristics is that it operates at the finer granularity of the multicast groups. The multicast group information helps in assigning the multicast group members to the same wavelength, resuling in increased multicast throughput.

Greedy Split Rule :

For each node previously assigned to the wavelength W^*, assign it to the new wavelength W_1^* if it has more multicast groups in common with the receiver i comapared to the receiver j. Assign a node to the wavelength W_2^* if the node has more multicast groups common with the node j as comapared to the node i. In case of a tie, assign a node to the wavelength, which will cause least retransmissions.

Time Complexity analysis :

Refering to step 5 of the Greedy-Split heuristic, to find a wavelength, carrying a maximum multicast traffic will require $\mathcal{O}(\lg C)$ time. As the multicast group membership matrix $B[G \times N]$ is a binary matrix, step 6 will take linear running time of $\mathcal{O}(G)$. Similarly execution of *Greedy-Split rule* will take $\mathcal{O}(N \times G)$ time. Assuming that the time taken to split a wavelength in step 7 is constant, *while* loop in step 4 would have the upper bound of $\mathcal{O}(N \times G^2 \times W \times \lg W)$. In the worst case as $W \approx N - 1$, the running time requirement for the G-SPLIT becomes $\mathcal{O}(N^2 G^2 \lg N)$.

5.4 Random Split Heuristic

The Random Split heuristics operates exactly like Greedy Split, except that it uses a different rule for splitting the wavelength W^* into the wavelength W_1^* and the wavelength W_2^*. A random number p is drawn at random from the set $(1, L-1)$, where L is the number of nodes assigned to the wavelength W^*. Next, we select p nodes at **random** from the wavelength W^*, and assign them to to the wavelength W_1^*. The remaining $(L - P)$ nodes are assigned to the wavelength W_2^*.

Time Complexity analysis :

We can assume constant computational time for the random number generation operations. In that case, finding a wavelength with the maximum mulitcast traffic load is the only time-consuming operation in R-SPLIT and it will take $\mathcal{O}(\lg W)$ to execute. In the worst

```
The Multicast LPT (MLPT) Heuristic
Input: $N, C, G$, Traffic matrix $\mathbf{M}[N \times G]$, Group membership matrix $\mathbf{B}[G \times N]$
Output: Wavelength assignment matrix $\mathbf{A}[N \times C]$

begin
    Assign the multicast groups to the wavelengths assuming multicast groups are disjoint groups;
    Using the Load Balancing algorithm LPT, distribute the multicast traffic evenly on all available wavelengths;
    for each node violating the Node Constraint
        Place the node on a wavelength, which will minimize the retransmission and satisfies the Channel Constraint;
        Increase the multiplier of the multicast groups split by the above wavelength assignment by 1;
    end for
    Return the wavelength assignment matrix $\mathbf{A}[N \times C]$;
end
```

Figure 5.4: The *MLPT* heuristic

case, when $W = N$, upper bound on the running time for the R-SPLIT heursitics will be $\mathcal{O}(N \lg N)$.

5.5 Multicast LPT Heuristic

Multicast LPT heuristics takes a different approach, compared to SPLIT and JOIN set of heurisitcs. It maintains the balance between two conflicting demands of load balancing and retransmissions by first performing the load balancing. It initally assumes that all multicast groups are disjoint and assigns the wavelength to the multicast groups, using LPT [28] load balancing algorithm . It also assigns a different *weight* to different multicast groups based on the traffic received by them.

Once the multicast groups are assigned to the wavelengths, there will be more than one node violating the *Node Constraint.* MLPT heuristic places each of these nodes to the wavelength, which will cause least retransmission in the network, i.e partitioning multicast groups with least *weight.* Before assigning a node to a wavelength, it checks whether the wavelength assignment matrix $\mathbf{A}[N \times C]$ meets the *Channel Constraint.* For all those multicast groups which are split on more than one wavelength due to node assignment, effective *weight* of these group is doubled. This will discourage the frequent partition of the smaller multicast groups and keeps members of same multicast group on the same wavelength as much as possible. Note that, transmitted multicast traffic remains the same as in $\mathbf{M}[N \times G]$ and is not affected by the weight assigned to the multicast groups in the heuristic.

This heuristic has one limitations as its performance is dependant on the initial load balancing algorithm. Using better alogithm like MULTIFIT [29] might give better results but would also increase the running time of the heuristics.

Time Complexity analysis :

The LPT [28] algorithm, first sorts the multicast groups according to their multicast traffic and then assigns them sequentially to wavelengths. Assuming the *insertion sort*, it will take $\mathcal{O}(G^2)$ time. We assume that the step 3 of the MLPT heuristic, of identifying nodes violating the *Node Constraint* will take constant time as the wavelength assignment matrix $\mathbf{A}[N \times C]$ is a binary matrix. The *for* loop in step 4, will be executed N times and each iteration of the loop will require $\lg G$ running time, as to find the multicast group with the maximum traffic. Summerizing, the worst case upper bound on running time for the MLPT will be $\mathcal{O}(NG^2 \lg G)$.

5.6 Multicast LPT-Search Heuristic

MLPT-Search heuristic is similar in approach to MLPT heuristics, discussed in Chapter 5.5. It also applies LPT [28] to assign multicast groups to channels, and assumes disjoing multicast group initially. The main difference is that once the nodes violating *Node Constraint* are identified, it generates all valid combinations of the wavelength assignment matrix $\mathbf{A}[N \times C]$ satisfying the *Node Constraint* and the *Channel Constraint.* For each valid assignment matrix, Channel bound $\mathcal{F}_{Ch}^{(C)}$ and Transmitter bound $\mathcal{F}_{Tr}^{(C)}$ are computed. The wavelength assignment matrix, with the minimum transmitter bound is returned. In case of a tie, the one with the minimum channel bound is selected.

The Multicast LPT Search (MLPT-Search) **Heuristic**

Input: N, C, G, Traffic matrix $\mathbf{M}[N \times G]$, Group membership matrix $\mathbf{B}[G \times N]$

Output: Wavelength assignment matrix $\mathbf{A}[N \times C]$

```
begin
    Assign the multicast groups to the wavelengths assuming multicast groups are
    disjoint groups;
    Use the Load Balancing algorithm LPT to distribute multicast traffic
    equally among all available wavelenths;
    Generate $\mathbf{A}[N \times C]$ satisfying the Node Constraint and the Channel Constraint;
    Select Assignment matrix $\mathbf{A}[N \times W]$, which gives $\mathbf{T}$ $[N \times N]$ with lowest $F_{Tr}^{(C)}$;
    In case of a tie, select $\mathbf{A}[N \times W]$ with lowest $F_{Ch}^{(C)}$;
    Return the wavelength assignment matrix $\mathbf{A}[N \times C]$;
end
```

Figure 5.5: The *MLPT-Search* heuristic

Time Complexity analysis :

As evident, the heuristics trades the running time and complexity for better and nearer to optimal, solution. In the worst case scenario, $N.W$ assignment matrices will generated. Each matrix involves computation of the transmitter bound $\mathcal{F}_{Tr}^{(C)}$, which is a matrix multiplication of $\mathcal{O}(N^2 \times W)$. With the LPT [28] running time of $\mathcal{O}(G^2)$ and with the worst-case scenario $W \approx N-1$, the loose upper bound for this heuristic will be $\mathcal{O}(N^5 G^2)$. Thus, although MLPT-Search, may give closest to the optimal results, it has the highest running time requirement compared to the other heuristics discussed in this chpater.

In the next chapter, we compare the performance of the various heuristics discussed and evaluate their relative performance. We vary some of the system parameters and analyze the performance of the heuristics with the change.

Chapter 6

Numerical Results

In this chapter, we study the relative performance of the six heuristics presented in the Chapter 5, namely, Greedy-Join, Greedy-Split, Random-Join, Random-Split, Multicast-LPT and Multicast-LPT-Search. We also look at the influence of the system parameters on the performance of the heuristics.

6.1 Traffic generation

We generate random instances of multicast traffic matrix $\mathbf{M}[N \times G]$ and multicast group membership matrix $\mathbf{B}[G \times N]$. The element of the matrix $\mathbf{M}$ indicates multicast traffic transmitted by a node to a multicast group. It is chosen with equal probability from the range $[0, 15]$. Matrix $\mathbf{B}$ is a binary integer matrix. Multicast group membership is determined as follows. For each multicast group, N random numbers corrosponding to N nodes are generated. These random numbers are uniformly distributed in the range $[0, 1]$. If the generated random number k is greater than 0.5, the node is part of multicast group, otherwise not. Due to uniform distribution, average multicast group size is $N/2$. For the non-uniform traffic as in case of hot-spot receivers, we modify the the probability such as average multicast group size remains the same $N/2$.

The generated group membership matrix is considered valid only if any node is part of atleast one multicast group and any multicast group has atleast one member. Multicast traffic has many variables, which can affect the performance of the heuristics. To create a level ground for a comparsion of the heuristics, we add two more constraints as *Degree of Commonality* ϵ_c and *Degree of Participitation* ϵ_p as defined in equations (4.4) and (4.5).

We consider only those group membership matrix which have ($\epsilon_c \leq 0.5$) and ($\epsilon_p \leq 0.5$). With low value of ϵ_c, correlation between different multicast group members is reduced. With lower ϵ_p the effect of the hot-spot node is lessened. With this constraints in place, more random uniform group membership matrix is generated. Note that above consraints are not forced for the non-uniform traffic generation.

In Figure 6.1, we plot the performance of six heuristics for a small number of nodes $N \leq 14$, and for $G = 5, C = 3$. We compare the heuristics against optimum lower bound $\mathcal{F}_{min}$ as in equation 4.3. In Figure 6.2 we evaluate different heusristics based on their ability to minimize the multicast retransmissions in the network. In Figure 6.3, the ability of a heuristic to balance the traffic across wavelength and maximize the channel utlization is evalauted by comparison. Figures 6.4 - 6.5, compare the heuristics for $G = 10, C = 10$ and number of nodes N are varied from 20 to 100. In Figures 6.7 - 6.9, we evaluate the effect of multicast group size on the performance of the heuristics. We vary the multicast group size $G = 30, 50, 100$ for the same value of $C = 10$. The behaviour of heuristics with non-uniform traffic can be observed in Figures 6.10- 6.12.

In next section, we discuss various obervations made from this plots and analyze the effect of varying the system parameters on hueristic performance.

6.2 Discussion of Results

In Figure 6.1, we compare different heuristics based on the schedule bound $\mathcal{F}_{sh}^{(C)}$. MLPT-Search offers the lowest schedule bound among all heuristics. Even though MLPT-Search performs exhaustive ansearch of the solution subspace, its schedule bound is up to 20% higher than the optimal value. This can be explained as follows. The absolute schedule bound $\mathcal{F}_{min}$ used in the heuristic comparison is defined in equation 4.3. It is the maximum of the multicast traffic received by the dominant receiver and multicast traffic transmitted by a dominant transmitter. Note that, when $N \geq C$, there could be more than one receiver assigned to the single wavelength. The traffic carried by the wavelength could be more than $\mathcal{F}_{ch}^{(N)}$ bound. Similarly, for the dominant transmitter bound to be realizable, all the recipients of the multicast transmission should be on a single wavelength. As multicast groups can not necessarily be disjoint, and so members of the same multicast group may be assigned to different wavelengths. This will result in the schedule bound, higher than $\mathcal{F}_{tr}^{(1)}$ bound. Thus although heuristics may appear sub-optimal compared to the absolute

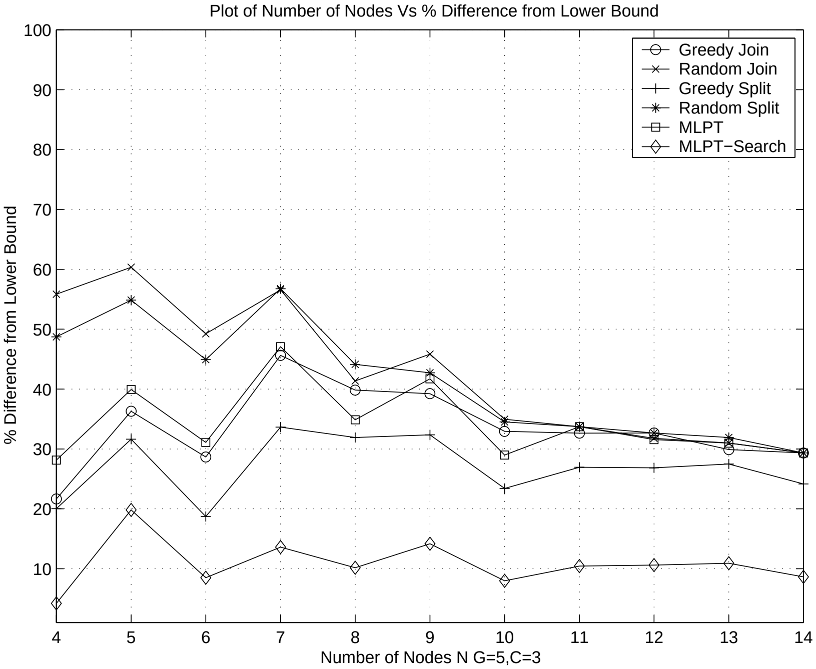

Figure 6.1: Heuristics Comparison for C=3 channels, G=5(Uniform Case)

schedule bound $\mathcal{F}_{min}$, it is possible that the absolute schedule bound $\mathcal{F}_{min}$ might not be achievable in practise. Evaluating other heuristics, Greedy-Split offers the lower schedule bound compared to Greedy-Join. G-SPLIT has a finer group-level granularity when it makes a decision to split a wavelength compared to the coarse wavelength-level granularity of G-JOIN. MLPT heuristic also performs almost the same as the G-SPLIT but the running time requirement of MLPT is more than the G-SPLIT. As we can expect, Random heuristics offer largest schedule bounds due to lack of intelligence at the expense of lowest running time.

Figures 6.2 and 6.3 offers us additional insight to the problem. In Figure 6.2, we observe that Greedy-Join heuristic has the lowest *Degree of Retransmission* compared to other heuristics including MLPT-Search. G-JOIN heuristic very well illustrates the need to balance the conflicting requirements of minimizing retransmissions and maximizing wavelength utilization. It causes the least retransmissions in the network, but its schedule length is significantly higher than other heuristics as G-JOIN. In the JOIN operation, G-JOIN heuristic selects a pair of wavelengths, merging those will cause the least retransmissions. Load balancing i.e. the wavelength utilization is considered only in the case of a tie. Thus,

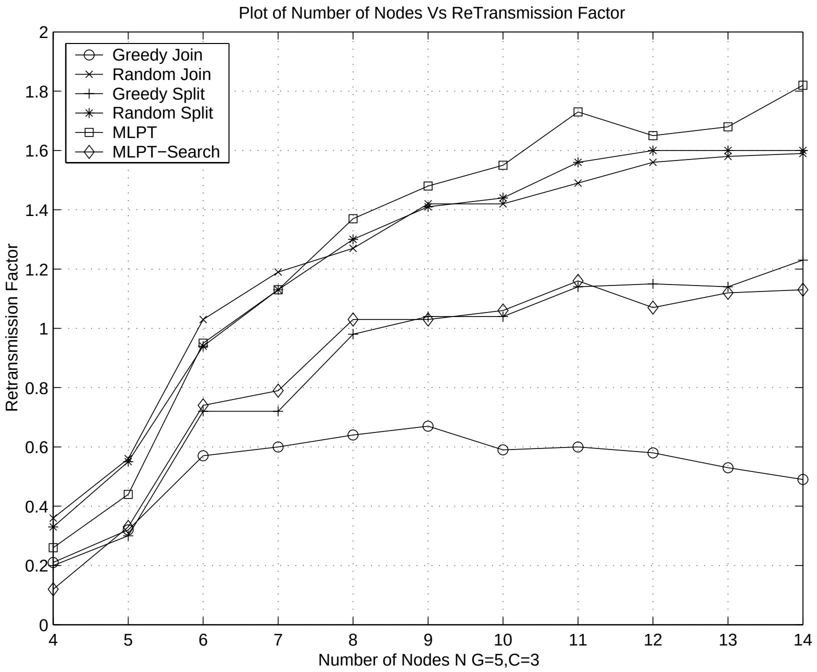

Figure 6.2: Heuristics Comparison for C=3 channels, G=5(Uniform Case)

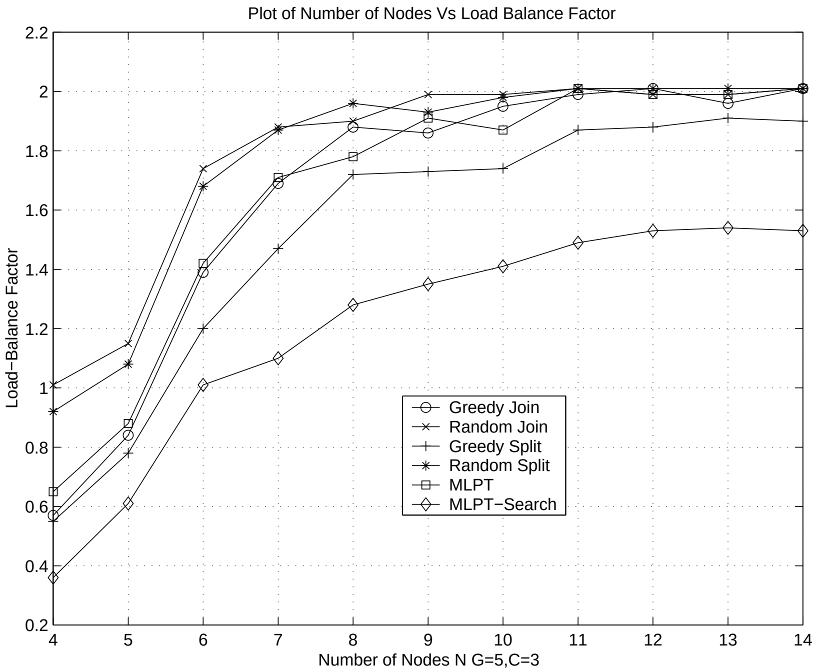

Figure 6.3: Heuristics Comparison for C=3 channels, G=5(Uniform Case)

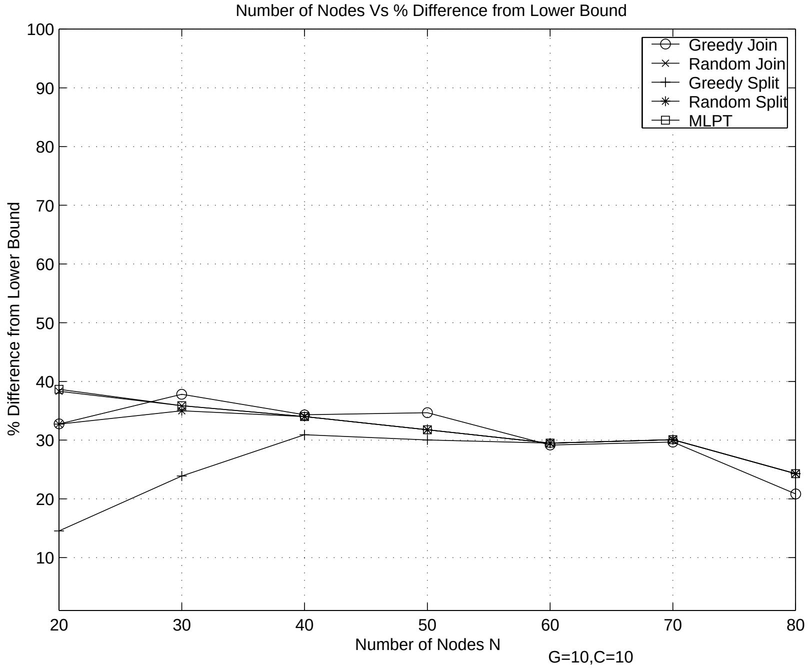

Figure 6.4: Heuristics Comparison for C=10 channels, G=10(Uniform Case)

to achieve schedule length closer to the optimal value, a careful balance must be maintained between both requirements. In Figure 6.3, we plot the *Degree of Load Balancing* achieved by different heuristics. We expect MLPT-Search to offer the lowest *Degree of Load Balancing* as initially it assigns wavelengths to the multicast group using load balancing algorithm. Figures 6.4 - 6.6, support above observations for $G = 10, C = 10$ and N ranging from 20 to 100. Note that MLPT-Search heuristic is not compared for large values of N due to its high running time requirements.

Next, we observe the behavior of the heuristic when number of multicast groups are increased. Figures 6.7 - 6.9 show the schedule bounds generated by the heuristics for $G = 30, 50, 80$. As seen in the plots, with increase in the number of multicast groups G, the performance of all heuristics degrade. As the number of multicast groups G increases relative to the number of nodes N, *Degree of Commonality* ϵ_c among multicast groups also increases. As average multicast group size remains the same $N/2$, larger ϵ_c would result in more partition of the multicast groups, and subsequently more retransmissions. This will result in increase in schedule bound as obtained by heuristics. The important thing to note is that absolute schedule bound $\mathcal{F}_{min}$ would decrease instead of increasing. When

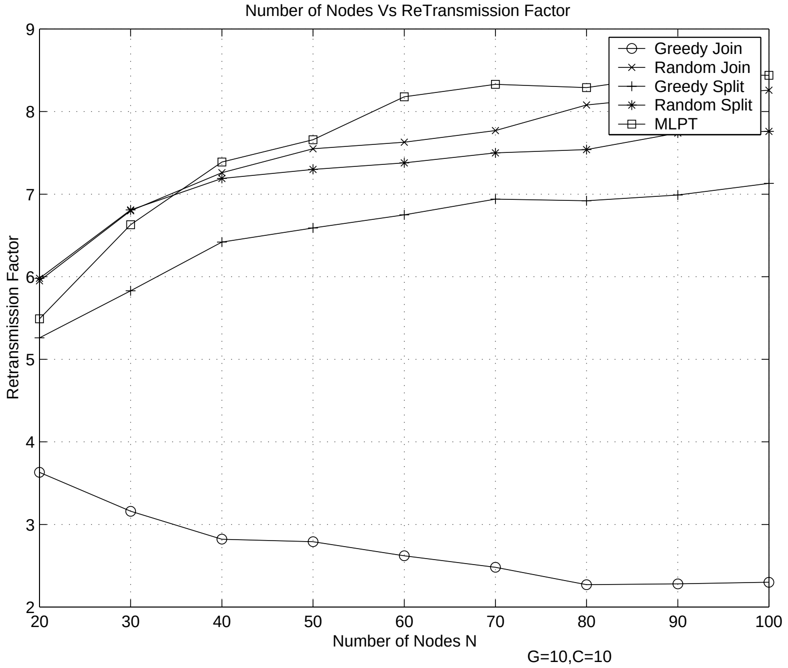

Figure 6.5: Heuristics Comparison for C=10 channels, G=10(Uniform Case)

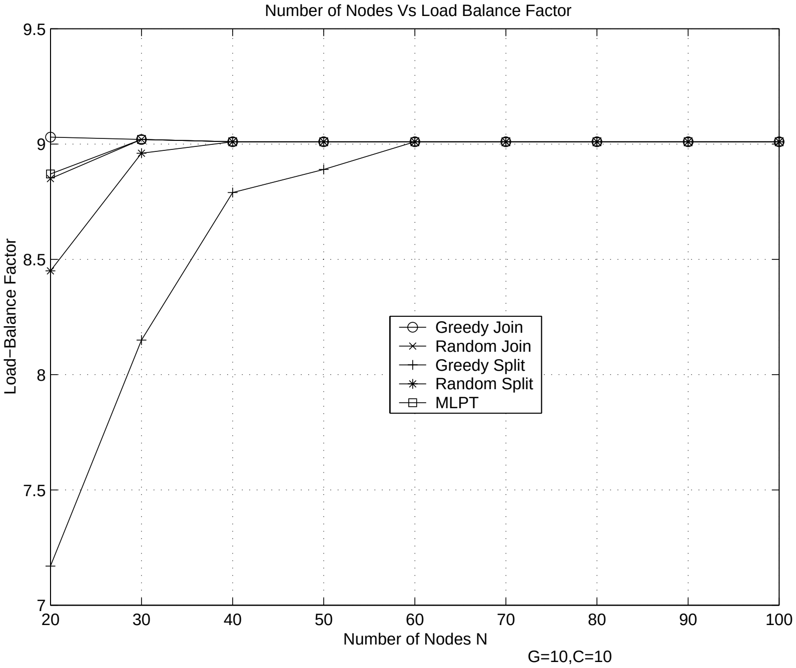

Figure 6.6: Heuristics Comparison for C=10 channels, G=10(Uniform Case)

Heurisitic	Time Complexity	% Difference from the Lower Bound
MLPT-Search	$\mathcal{O}(N^5.G^2)$	10.44
Greedy-Join	$O(N^6)$	39.84
Greedy-Split	$\mathcal{O}(N^2G^2\lg N)$	31.91
Multicast-LPT	$\mathcal{O}(NG^2\lg G)$	33.73
Random Join	$\mathcal{O}(N^3)$	41.35
Random Split	$\mathcal{O}(N\lg N)$	44.92

Table 6.1: Comparison of Heuristics,N=8,G=5,C=3

calculating $\mathcal{F}_{min}$, we consider extreme cases of $C = 1$ and $C = N$. In both of these scenarios, higher value of ϵ_c, reduces the schedule bounds, lowering $\mathcal{F}_{min}$ with increase in value of G.

Until now, we have considered uniform multicast group distribution. In Figures 6.10 - 6.12, we compare heuristics for the non-uniform multicast group distribution. We select some receivers out of N nodes which are more likely to be a part of multicast group. As observed in the plots, relative performance of the heuristics remains the same compared to uniform distribution scenario.

As seen from the numerical results and above discussion, all six heuristics produce schedules which are within 100% of the absolute schedule bound under wide range of system parameters. To pick up the winner out of six heuristics, we refer to Table 6.2. It very clearly demonstrates the trade off involved in any heuristic design, i.e running time complexity vs. quality of the solution. Quality of the solution of the heuristic can be improved at the expense of simplicity. We can rule out MLPT-Search and G-JOIN heuristics due to their high running time requirements. The choice of the best heuristic for a given system, among remaining four is dependent on the desired tradeoff between time to compute the solution and quality of the solution.

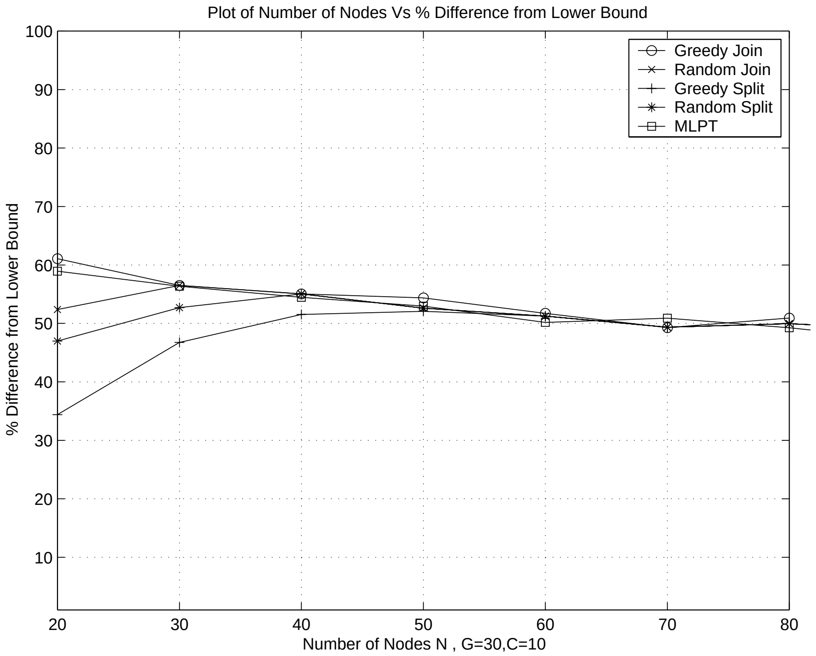

Figure 6.7: Heuristics Comparison for C=10 channels, G=30(Uniform Case)

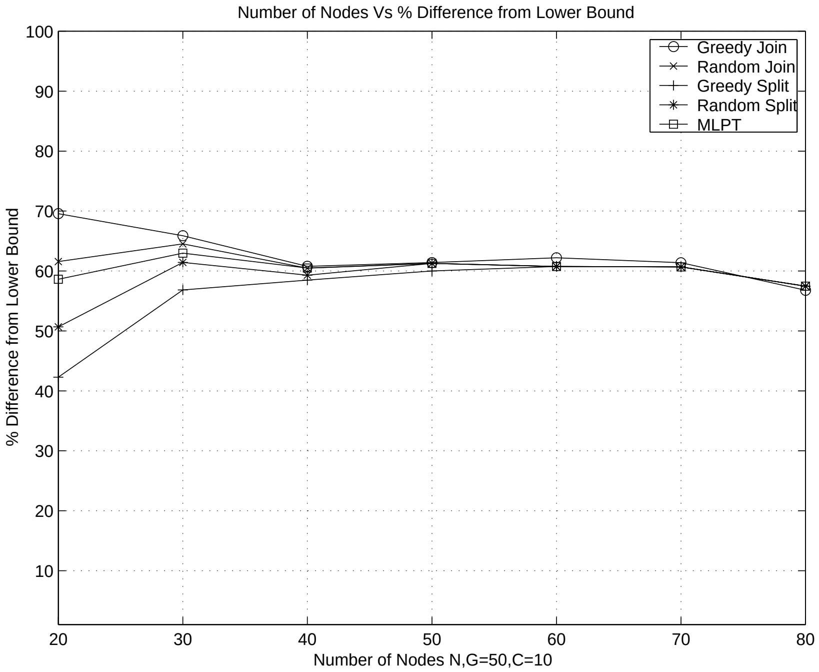

Figure 6.8: Heuristics Comparison for C=10 channels, G=50(Uniform Case)

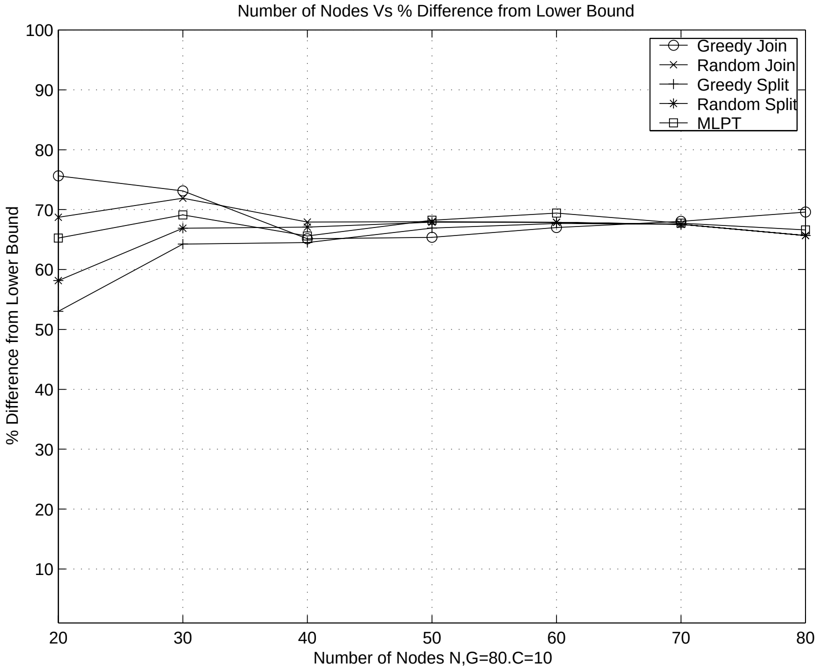

Figure 6.9: Heuristics Comparison for C=10 channels, G=80(Uniform Case)

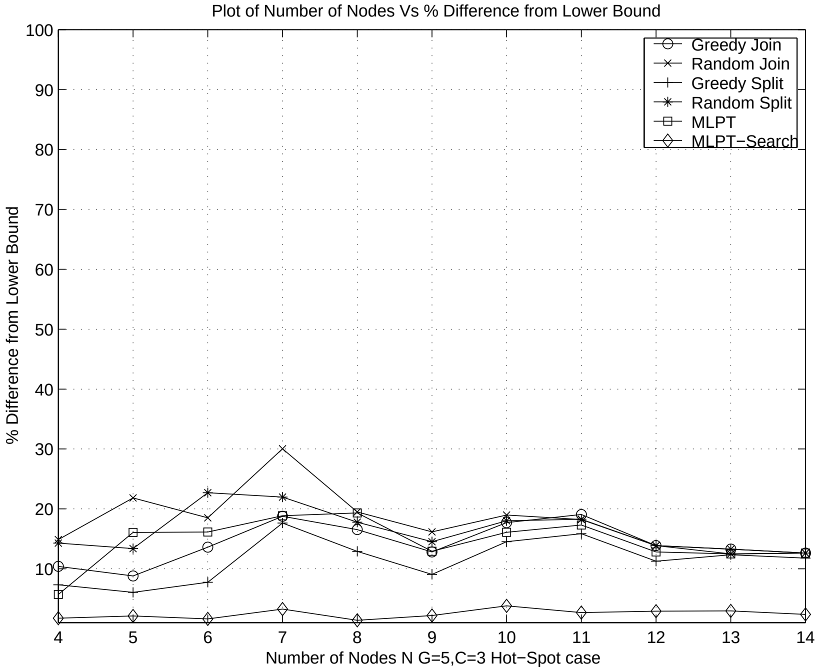

Figure 6.10: Heuristics Comparison for C=3 channels, G=5(Non Uniform Case)

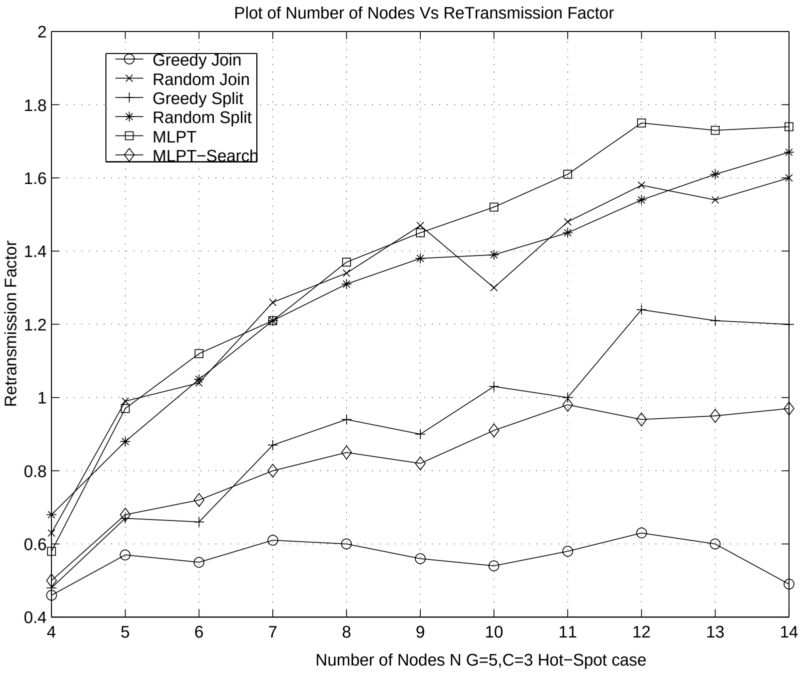

Figure 6.11: Heuristics Comparison for C=3 channels, G=5(Non Uniform Case)

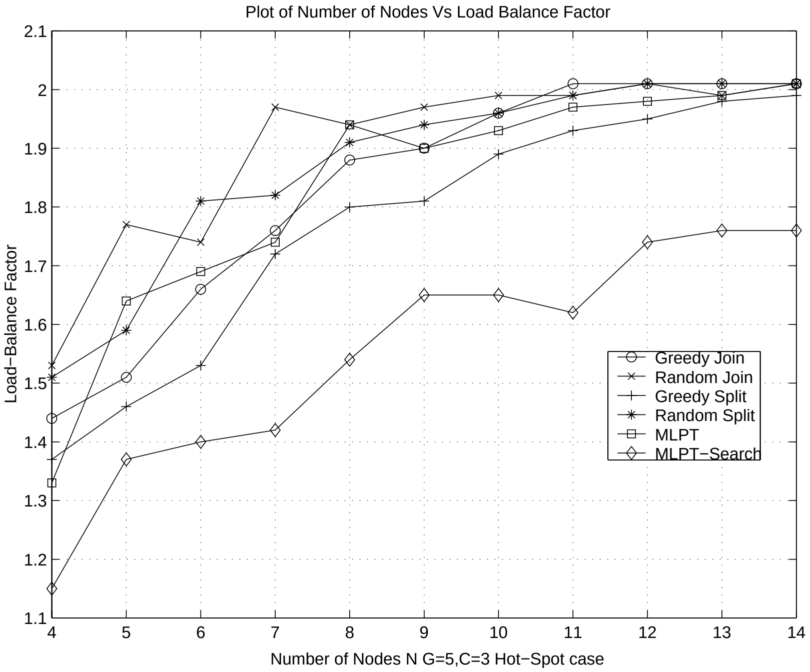

Figure 6.12: Heuristics Comparison for C=3 channels, G=5(Non Uniform Case)

Chapter 7

Summary and Future Research

7.1 Summary

We have considered the problem of scheduling multicast traffic in a tunable transmitter and slowly tunable receiver in a broadcast WDM network. The scheduling problem transforms into a wavelength assignment problem due to slow tunability of the receiver. It also introduces the reconfigurability in the network to accommodate varying traffic demands. As the wavelength assignment problem is proved to be intractable, we have presented heuristics which would assign wavelengths to the receivers. This assignment of the wavelength is done to minimize the transmission schedule length, to minimize the retransmission of the multicast traffic due to group-splitting and to maximize the wavelength utilization. The heuristics presented provides a design tradeoff of quality of solution and speed. The conclusion of our work is that it is possible to realize cost effective reconfigurable broadcast WDM networks which can support multicast traffic, by using slowly tunable transceivers without sacrificing the network performance.

7.2 Future Research

Due to extensive work in the last few years, this research area is now well-understood, and efforts are currently under way to put our knowledge into practice by implementing some of these techniques in testbed environments [8, 1].

Although we have presented efficient heuristics for scheduling multicast traffic in TT-STR network, there are other pending issues, as merging unicast and multicast traffic demands, and minimizing retunings of transceivers for the combined traffic in the reconfiguration phase of the network. Another area of research could be to support quality-of-service (QoS) for multi-destination traffic in a broadcast WDM network.

Bibliography

[1] The NGI Helios project. http://www.anr.mcnc.org/projects/Helios/Helios.html.

[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. Bandai, S. Shiokawa, and I. Sasane. Performance analysis of a multicasting protocol in a WDM based single-hop lightwave network. In *Proceedings of Globecom '97*, pages 561–565, 1997.

[4] M. Borella and B. Mukherjee. Limits of multicasting in a packet switched WDM network. *Journal of High Speed Networks*, 4(2):155–167, 1995.

[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] B. Mukherjee. WDM-Based local lightwave networks Part 1:Single-Hop systems In *IEEE Networks magazine*, pages 12–27. IEEE, May 1992.

[7] W-T. Chen, P-R. Sheu, and J-H. Yu. Time slot assignment in tdm multicast switching system. *IEEE Transactions on Communications*, 42(1):149–165, 1994.

[8] M. Kuznetsov et al. A next-generation optical regional access network. *IEEE Communications Magazine*, 38(1):66–72, January 2000.

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

[10] R. E. Wagner *et al.* MONET: Multiwavelength optical networking. *Journal of Lightwave Technology*, 14(6):1349–1355, June 1996.

[11] O. Gerstel, B. Li, A. McGuire, G. N. Rouskas, K. Sivalingam, and Z. Zhang (Eds.). Special issue on protocols and architectures for next generation optical WDM networks. *IEEE Journal Selected Areas in Communications*, 18(10), October 2000. (To appear).

[12] I. Gopal and C. Wong. Minimizing the number of switchings in an SS/TDMA system. *IEEE Transactions on Communications*, 33(6):497–501, June 1985.

[13] N-F. Huang, Y-J. Wu, C-S. Wu, and C-C. Chiou. A multicast model for WDM-based local lightwave networks with a passive star topology. In *Proceedings of TENCON '93*, pages 470–473, 1993.

[14] J. H. Hui and T. Renner. Queueing analysis for multicast packet switching. *IEEE Transactions on Communications*, 42(2/3/4):723–731, February 1994.

[15] J. Jue and B. Mukherjee. The advantages of partitioning multicast transmissions in a single-hop optical WDM network. In *Proceedings of ICC '97*, pages 427–431. IEEE, 1997.

[16] H-C. Lin and C-H. Wang. Minimizing the number of multicast transmissions in single-hop WDM networks. In *Proceedings of ICC 2000*, pages 1645–1649. IEEE, 2000.

[17] E. Modiano. Random algorithms for scheduling multicast traffic in WDM broadcast-and-select networks. *IEEE/ACM Transactions on Networking*, 7(3):425–434, June 1999.

[18] A. Mokhtar and M. Azizoglu. Packet switching performance of WDM broadcast networks with multicast traffic. In *Proceedings of SPIE '97*, pages 220–231, 1997.

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

[20] 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.

[21] Z. Ortiz, G. N. Rouskas, and H. G. Perros. Maximizing multicast throughput [21] in WDM networks with tuning latencies using the virtual receiver concept. *European Transactions on Telecommunications*, 11(1):63–72, January/February 2000.

[22] Z. Ortiz, G. N. Rouskas, and H. G. Perros. Scheduling of combined unicast and multicast traffic in broadcast WDM networks. *Photonic Network Communications*, 2(2):135–154, May 2000.

[23] 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.

[24] 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.

[25] G. N. Rouskas and V. Sivaraman. A reservation protocol for broadcast WDM networks and stability analysis. *Computer Networks*, 32(2):211–227, February 2000.

[26] S-T. Sheu and C-P. Huang. An efficient multicast protocol for WDM star-coupler networks. In *Proceedings of the IEEE International Symposium on Computers and Communications*, pages 579–583, 1999.

[27] Wen-Yu Tseng and Sy-Yen Kuo. A combinational media access protocol for multicast traffic in single-hop WDM LANs. In *Proceedings of Globecom '98*, pages 294–299. IEEE, 1998.

[28] R.L.Graham. Bounds on multiprocessing timing anomalies In *SIAM Journal of Applied Mathematics*, pages 416–429. March 1969.

[29] E.Coffman,M.R.Garey,D.S.Johnson, An application of bin-packing to multiprocessor scheduling In *SIAM Journal of Computing*, pages 1–17. February 1978.

[30] G. N. Rouskas, M.H. Ammar Dynamic reconfiguration in multihop WDM networks, In *Journal of High Speed Networks*, pages 221–238. 1995.

[31] I. Baldine, G. N. Rouskas On the design of dynamic reconfiguration policies for broadcast WDM networks. In *Proceedings of SPIE.* November 1998.

[32] I. Baldine, G. N. Rouskas Reconfiguration and Dynamic Load Balanacing In *Photonic Network Communications*, pages 49–64. 1999.