

Reconfiguration in Rapidly Tunable Transmitter, Slowly Tunable Receiver Single-Hop WDM Networks *

Ilia Baldine George N. Rouskas

TR-96-10

June 4, 1996

Abstract

We consider broadcast WDM networks with nodes equipped with rapidly tunable transmitters and slowly tunable receivers. The rapidly tunable transmitters provide all-optical paths among the network nodes by creating logical connections that can be changed on a packet-by-packet basis. The ability of receivers to tune, albeit slowly, is invoked only for reallocating the bandwidth in response to changes in the overall traffic pattern. Since this variation in traffic is expected to take place over larger time scales, receiver retuning will be a relatively infrequent event, making slowly tunable devices a cost effective solution. Assuming an existing assignment of receive wavelengths and some information regarding the new traffic demands, we present two approaches to obtaining a new wavelength assignment such that (a) the new traffic load is balanced across the channels, and (b) the number of receivers that need to be retuned is minimized. One of our contributions is an approximation algorithm for the load balancing problem that provides for tradeoff selection, using a single parameter κ , between these conflicting goals. This algorithm leads to a scalable approach to reconfiguring the network since, in addition to providing guarantees in terms of load balancing, for certain values of parameter κ , the expected number of retunings scales with the number of channels, *not* the number of nodes in the network.

Department of Computer Science
North Carolina State University
Raleigh, NC 27695

*This work was supported by a grant from the Center of Advanced Computing and Communication, North Carolina State University.

1 Introduction

Single-hop lightwave networks have been proposed for Local and Metropolitan Area Networks (LANs and MANs) [1, 2]. The single-hop architecture employs Wavelength Division Multiplexing (WDM) to provide connectivity among the network nodes. The WDM channels are dynamically shared by the attached nodes, and the logical connections change on a packet-by-packet basis creating all-optical paths between sources and destinations. Single-hop networks require the use of rapidly tunable optical lasers and/or filters that can switch between channels at high speeds. Such devices do exist today [3]; however, they have to be custom-built and they tend to be extremely expensive, accounting for a significant fraction of the overall cost of building a lightwave network. Consequently, media access protocols such as HiPeR- ℓ [4], FatMAC [5], DT-WDMA [6], and Rainbow [7] that require tunability only at one end have the potential of keeping the overall cost at reasonable levels, leading to network architectures that can be realized cost effectively.

When tunability only at one end, say, at the transmitters, is employed, each fixed receiver is permanently assigned to one of the wavelengths used for packet transmissions. In a typical near-term WDM environment, the number of channels that will be supported within the optical medium is expected to be smaller than the number of attached nodes. As a result, each channel will have to be shared by multiple receivers, and the problem of assigning receive wavelengths arises. Intuitively, this assignment must be somehow based on the prevailing traffic conditions. But with fixed receivers, the assignment of receive wavelengths is permanent and cannot be updated in response to changes in the traffic pattern.

Alternatively, one can use *slowly tunable*, rather than fixed, receivers. In general, we will distinguish between rapidly and slowly tunable devices on the basis of two parameters: tunability speed and cost. We will say that an optical laser or filter is rapidly tunable if the time it takes to switch between channels is comparable to a packet transmission time at Gigabit per second rates. Slowly tunable devices can be significantly less expensive than rapidly tunable ones, but their tuning times can also be significantly longer (up to several orders of magnitude). As a result, these devices cannot be assumed “tunable” at the media access level (i.e., for the purposes of scheduling packet transmissions), as this requires fast tunability. However, use of slowly tunable receivers makes it possible to modify the assignment of receive wavelengths over time to accommodate varying traffic demands.

The issues that arise in reconfiguring a lightwave network by retuning a set of slowly tunable transmitters or receivers have been studied in the context of multihop WDM networks in [8, 9]. The work in [9] considered the problem of constructing a sequence of branch-exchange operations

of minimum length to take the network from an initial to a target connection diagram. The focus in [8] was on the design of dynamic policies for determining when and how to reconfigure the network. To the best of our knowledge, reconfiguration has not been studied in the context of single-hop networks.

In this paper we consider the problem of reconfiguring a single-hop network by retuning a subset of the slowly tunable receivers to obtain a new wavelength assignment. Assuming an existing assignment of receive wavelengths and some information about the new traffic demands, our objective is to obtain a new wavelength assignment such that (a) the traffic load is balanced across the various channels, and (b) the number of receivers that need to be retuned is minimum. We present two approaches for determining the new assignment. One approach first partitions the receivers into subsets by running an approximation algorithm for the load balancing problem on the new traffic matrix. It then assigns these receiver subsets to wavelengths so that the number of retunings, starting from the previous wavelength assignment, is minimized. We show that this straightforward approach requires a large number of retunings, and does not scale well with the size of the network. We then present a new approximation algorithm for the load balancing problem that provides for tradeoff selection, using a single parameter κ , between the two conflicting goals. This algorithm leads to a scalable approach to reconfiguring the network since, in addition to providing guarantees in terms of load balancing, for certain values of parameter κ , the expected number of retunings scales with the number of channels, *not* the number of nodes in the network.

The next section introduces the network model, and discusses the issues arising during the reconfiguration phase. In Section 3 we describe two approaches for determining the new wavelength assignment. In Section 4 we present some numerical results to compare the two approaches, and we conclude the paper in Section 5.

2 System Model

2.1 Network Model and Operation

We consider a packet-switched single-hop lightwave network with N nodes, and one transmitter-receiver pair per node. The nodes are physically connected to, and communicate over, a passive broadcast optical medium. The medium can support $C < N$ wavelengths, $\lambda_1, \dots, \lambda_C$. Both the transmitter and the receiver at each node are tunable over the entire range of available wavelengths. However, the transmitters are *rapidly tunable*, while the receivers are *slowly tunable*. We will refer to this tunability configuration as *rapidly tunable transmitter, slowly tunable receiver (RTT-*

STR). (We note that all our results can be easily adapted to the dual configuration, STT-RTR.)

Let Δ_t (Δ_r) denote the normalized tuning latency of transmitters (receivers), expressed in units of packet transmission time. In the RTT-STR system under consideration, we have that:

$$\Delta_r \gg \Delta_t = k \geq 1 \quad (1)$$

In (1), k is a small integer, while Δ_r takes values that may be several orders of magnitude greater than Δ_t . The main motivation for employing slowly tunable receivers is the significant savings in cost that can be realized. Given that with today's technology the cost of rapidly tunable optical devices can be extremely high, designing architectures that require fast tunability only at one end is a first step towards keeping the overall cost of building a lightwave network at reasonable levels.

We distinguish two levels of network operation, differing mainly in the time scales at which they take place. At the *packet transmission* level, connectivity among the network nodes is provided by a media access protocol that requires tunability only at the transmitting end. The protocol schedules packets for transmission by employing algorithms that can effectively mask the tuning latency of tunable transmitters [10, 11, 12, 13]. (For instance, the scheduling algorithms in [10] have been shown to work well with values of Δ_t up to 16-20.) One such protocol is HiPeR- ℓ [4]. Since the receiver latency Δ_r is significantly long and cannot be overlapped with packet transmissions, at this level the receivers are considered to be fixed to a particular wavelength. Let $\lambda(j) \in \{\lambda_1, \dots, \lambda_C\}$ be the wavelength currently assigned to receiver j . An assignment of wavelengths to receivers is a partition $\mathcal{R} = \{R_c, c = 1, \dots, C\}$ of the set $\mathcal{N} = \{1, \dots, N\}$ of nodes, such that R_c is the subset of nodes currently receiving on wavelength λ_c :

$$R_c = \{j \mid \lambda(j) = \lambda_c\} \quad c = 1, \dots, C \quad (2)$$

This assignment of receive wavelengths to the network nodes is known to the media access protocol, and it is used to determine the target channel for a packet given the packet's destination.

The ability of receivers to tune, albeit slowly, is invoked only at the *resource allocation* level; in this work, the shared resource of interest is bandwidth. We note that a partition $\mathcal{R} = \{R_c\}$ in (2) implies an allocation of the available bandwidth to the various receivers. Use of tunable receivers allows this allocation to be optimized to prevailing traffic conditions. As traffic varies, a new assignment of receive wavelengths may be sought that satisfies some optimality criteria; these criteria are the subject of the next subsection. We will use the term "reconfiguration" to refer to the reallocation of bandwidth to receivers. Since this variation in traffic will more likely take place over larger scales in time, reconfiguration is expected to be a relatively infrequent event, and each assignment of receive wavelengths will be long lived relative to the scheduling of packet transmissions

by the media access protocol. Consequently, receivers with a tuning time Δ_r significantly larger than the packet transmission time, will be acceptable at the resource allocation level as long as Δ_r is small compared to the mean time between successive reconfiguration events.

2.2 Assignment of Receive Wavelengths

Intuitively, receive wavelengths should be assigned so that the traffic load be balanced across the C channels. A recent study by one of the authors on the performance of HiPeR- ℓ [4], a new reservation protocol for single-hop networks, has confirmed this intuition. Specifically, it was shown in [4] that the maximum sustained throughput is directly affected by the degree of load balancing in the network: the higher the degree of load balancing, the higher the overall arrival rate that the network can accommodate, and vice versa. Although this result was derived specifically for HiPeR- ℓ , we believe that load balancing has a similar effect on the performance of any protocol for multichannel single-hop networks.

We represent the bandwidth requirements of source-destination pairs by a traffic demand matrix $\mathbf{T} = [t_{ij}]$. Quantity t_{ij} is a measure of the average traffic originating at node i and terminating at node j . Given matrix \mathbf{T} , we can compute the total bandwidth requirement b_j of receiver j as the sum of the elements of the j -th column of \mathbf{T} :

$$b_j = \sum_{i=1}^N t_{ij} \quad j = 1, \dots, N \quad (3)$$

Receive wavelengths are assigned on the basis of quantities b_j , $j = 1, \dots, N$. Based on our observations regarding load balancing, our objective is to assign the receivers to the available channels such that the total bandwidth used in each channel is approximately the same among different channels. This problem is equivalent to the multiprocessor scheduling problem [14], where given a set of tasks with *a priori* known processing times and a number of processing units, the objective is to allocate the tasks to the processors such that the overall finish time is minimized. (This implies that the total processing time of the various processors differs as little as possible.) In our case the channels take the place of the processors, the receivers replace the tasks and the bandwidth requirements b_j replace the processing times.

The multiprocessor scheduling problem is \mathcal{NP} -complete [15]. Two approximation algorithms for this problem are *MULTIFIT* [16], with an absolute performance ratio of 1.22, and *LPT* [17], with an absolute performance ratio of 1.33. Either of these two algorithms may be used to obtain an assignment of receive wavelengths based on the receiver bandwidth requirements b_j , $j = 1, \dots, N$, such that traffic is spread across the various channels as evenly as possible. We now proceed to

discuss what happens when, due to changes in the traffic pattern, the current wavelength assignment becomes suboptimal.

2.3 The Transition Phase

Let \mathcal{R} be an assignment of receive wavelengths based on traffic matrix \mathbf{T} and the corresponding bandwidth requirements $\{b_j\}$ in (3). As traffic varies over time, the elements of matrix \mathbf{T} , as well as the column sums $\{b_j\}$, will change. Let \mathbf{T}' be a new traffic matrix, and $\{b'_j\}$ be the new receiver bandwidth requirements. If, due to these traffic changes, assignment \mathcal{R} is no longer successful in balancing the load across the channels, two actions are taken:

- a new assignment \mathcal{R}' is obtained, optimized for the new bandwidth requirements $\{b'_j\}$, and
- a number of receivers are tuned to new wavelengths as specified by \mathcal{R}' .

In [9] it was assumed that the traffic pattern is slowly and predictably changing over time. In this case, an assignment of receive wavelengths may be precomputed for the expected new traffic conditions. If changes in the traffic pattern are not predictable, the network nodes (or a special node dedicated to managing the network) may monitor packet transmissions on the various channels, and apply statistical techniques to determine whether the overall conditions have changed in a way that significantly affects the optimality of the current wavelength assignment. The problem of determining *when* the wavelength assignment needs to be updated is beyond the scope of this paper; rather, we concentrate on the issues arising once a decision to reconfigure the network has been taken based on a new traffic matrix \mathbf{T}' .

The reconfiguration phase will take the network from the current assignment \mathcal{R} to some new assignment \mathcal{R}' . We define the distance \mathcal{D} between two wavelength assignments \mathcal{R} and \mathcal{R}' as follows:

$$\mathcal{D}(\mathcal{R}, \mathcal{R}') = N - \sum_{c=1}^C |R_c \cap R'_c| \quad (4)$$

The distance $\mathcal{D}(\mathcal{R}, \mathcal{R}')$ represents the number of receivers that would need to be retuned in order to take the network from wavelength assignment \mathcal{R} to the new assignment \mathcal{R}' .

There is a wide range of policies for reconfiguring the network, mainly differing in the tradeoff between the length of the transition period and the portion of the network that becomes unavailable during this period (see [9] for a discussion on similar issues arising in multihop networks). One extreme approach would be to simultaneously retune all the receivers that are assigned new channels under \mathcal{R}' . The duration of the transition period is minimized under this approach (it becomes equal

to Δ_r), but a significant fraction of the network may be unusable during this time. At the other extreme, an approach that retunes one receiver at a time minimizes the portion of the network unavailable at any given instant during the transition phase, but maximizes the length of this phase. Between these policies at the two ends of the spectrum lie a range of approaches in which two or more receivers are retuned simultaneously.

Let us define a *step* in the reconfiguration phase as an interval of length Δ_r during which one or more receivers are retuned. Let $k(p)$ be the number of steps required under policy p , and $x_n(p)$, $n = 1, \dots, k(p)$, be the number of receivers retuned during step n for the same policy. During the transition period, the network incurs some cost in terms of packet delay, packet loss, packet desequencing, and the control resources involved in receiver retuning. This cost is directly proportional to both the portion of the network that becomes unavailable and the length of the transition period. A measure of this cost that accounts for both these factors is the network unavailable fraction-unavailability length product, which can be obtained as the sum $\sum_{n=1}^{k(p)} \left(\Delta_r \frac{x_n(p)}{N} \right)$. But, for any reconfiguration policy p , this sum is equal to:

$$\sum_{n=1}^{k(p)} \left(\Delta_r \frac{x_n(p)}{N} \right) = \Delta_r \frac{\mathcal{D}(\mathcal{R}, \mathcal{R}')}{N} \quad \forall p \quad (5)$$

Thus, regardless of the policy used, the number of retuning operations emerges as an important parameter, one that determines the impact of the reconfiguration phase on the traffic carried by the network.

The rest of the paper considers the problem of minimizing the number of retuning operations given an initial assignment \mathcal{R} and a new traffic matrix \mathbf{T}' . As in [9], we also ignore network specific issues such as how to coordinate the individual steps of the transition phase and inform the nodes of which receivers to retune and when. Instead, we concentrate on an abstract model that hides the details of operation but is applicable to a wide range of network environments.

3 Minimizing the Number of Retuning Operations

Consider a network operating under wavelength assignment \mathcal{R} optimized for traffic matrix \mathbf{T} . As traffic varies over time, the matrix is updated to reflect the changes in the traffic pattern. Let \mathbf{T}' be the traffic matrix at the instant reconfiguration is triggered. Our objective is to obtain a new wavelength assignment \mathcal{R}' such that:

- the new traffic load, as specified by matrix \mathbf{T}' is evenly spread across the C channels, and

- the number of retunings required to take the network from assignment \mathcal{R} to assignment \mathcal{R}' is as small as possible.

We note that these requirements on \mathcal{R}' represent two conflicting objectives: minimizing the number of retunings alone would result in \mathcal{R}' being the same as \mathcal{R} , which may be suboptimal in terms of load balancing; while optimally balancing the load across the C channels might produce a new assignment such that the distance in (4) be large.

We distinguish two approaches in constructing a new assignment \mathcal{R}' , differing mainly in whether the optimization procedure attempts to satisfy both objectives simultaneously, or one at a time:

- The first approach consists of two steps. The first step is to partition the set of receivers by solving the load balancing problem on matrix \mathbf{T}' *independently* of the initial assignment \mathcal{R} . The second step assigns the new subsets of receivers to wavelengths so as to minimize the number of retunings required starting from \mathcal{R} . This gives rise to the *Channel Mapping* problem discussed in the next subsection.
- The second approach attempts to solve the load balancing problem on matrix \mathbf{T}' , while at the same time minimizing the number of retunings that have to be performed. We will call this the *Constrained Load Balancing* problem.

We now study the two problems, starting with the channel mapping problem.

3.1 The Channel Mapping Problem

We consider an initial wavelength assignment \mathcal{R} and a new traffic matrix \mathbf{T}' . The first step in the reconfiguration process is to run an approximation algorithm (such as *MULTIFIT* or *LPT*) to obtain a partition $\mathcal{S}' = \{S'_c\}$ of the set of receivers into C sets S'_c , $c = 1, \dots, C$. This partition \mathcal{S}' is such that the bandwidth requirements of the receivers in each set S'_c is approximately the same among the C sets. We note that the approximation algorithm does not distinguish among the various channels. Thus, the output of the algorithm is simply a partition \mathcal{S}' of the set of receivers, *not* a wavelength assignment as defined in (2); in other words, there is no association among the receiver subsets S'_c and the available wavelengths.

From \mathcal{S}' we may obtain a new wavelength assignment \mathcal{R}' by mapping each subset S'_c to one of the wavelengths, such that no two subsets map to the same wavelength. Since our objective is to minimize the number of retuning operations during the reconfiguration, the problem of selecting a mapping that results in the least number of retunings arises. We will call this the *Channel Mapping (CM)* problem; it can be formally stated as:

Problem 3.1 (CM) *Given an initial wavelength assignment $\mathcal{R} = \{R_c\}$, and a new partition $\mathcal{S}' = \{S'_c\}$ of the set of receivers, find a permutation $(\pi_1, \pi_2, \dots, \pi_C)$ of $\{1, \dots, C\}$ such that for the new wavelength assignment $\mathcal{R}' = \{R'_c\}$ with $R'_c = S'_{\pi_c}$, $c = 1, \dots, C$, the distance $\mathcal{D}(\mathcal{R}, \mathcal{R}')$ is minimum over all possible permutations.*

The following lemma emphasizes the importance of employing an optimal algorithm for the CM problem, by stating that using a simple minded scheme such as the identity permutation (i.e., letting $R'_c = S'_c$ for all c) may result in an unnecessarily large number of retunings.

Lemma 3.1 *Assume that the traffic matrix has changed so that at least one retuning is required under the optimal permutation. Then, the difference between the number of retunings required by the identity permutation and that required by the optimal permutation can be equal to the number of receivers minus one, in the worst case.*

Proof. We will construct an instance of the CM problem for which the difference in the number of retunings under the identity and optimal permutations is equal to $N - 1$. Consider a network with $C \geq 3$ and $N > C$. Let the initial wavelength assignment $\mathcal{R} = \{R_c\}$ be any arbitrary assignment such that $|R_c| \geq 2$ and let $j \in R_C$. Let the new partition $\mathcal{S}' = \{S'_c\}$ be such that

$$S'_c = \begin{cases} R_2 \cup \{j\}, & c = 1 \\ R_{c+1}, & c = 2, \dots, C - 2 \\ R_C - \{j\}, & c = C - 1 \\ R_1, & c = C \end{cases} \quad (6)$$

It is straightforward to verify that the identity permutation requires that all receivers retune to new wavelengths (N retunings), while the optimal permutation $(C, 1, 2, 3, \dots, C - 1)$ requires only one retuning, that of receiver j from wavelength λ_C to wavelength λ_2 . \square

The next lemma states that, even under an optimal solution to the CM problem, the number of retunings required may be very large.

Lemma 3.2 *Let \mathcal{R} and \mathcal{S}' be the initial wavelength assignment and new partition, respectively, of an arbitrary instance of the CM problem for a network with N nodes and C channels. If the optimal solution to this instance yields wavelength assignment \mathcal{R}' , it is possible that*

$$\mathcal{D}(\mathcal{R}, \mathcal{R}') \leq N - C - 1 \quad (7)$$

Proof. We will construct an instance of CM that requires exactly $N - C - 1$ retunings. Consider a network with $N = C^2$, and an initial wavelength assignment given by:

$$R_c = \{(c-1)C + 1, \dots, cC\} \quad c = 1, \dots, C \quad (8)$$

The new partition \mathcal{S}' is:

$$S'_c = \begin{cases} \{c\}, & c = 2, \dots, C \\ \{1, C + 1, \dots, C^2\}, & c = 1 \end{cases} \quad (9)$$

It is straightforward to verify that exactly $C^2 - C - 1 = N - C - 1$ retunings are required under the optimal permutation. \square

3.1.1 Matrix Representation

There is a natural way of representing all the important knowledge about an instance of the CM problem. Let \mathcal{R} and \mathcal{S}' be the initial wavelength assignment and the new partition, respectively, of an arbitrary instance of the CM problem. We first observe that the identities of the particular receivers in the various subsets R_c and S'_c have no relevance to the optimal solution to the CM problem. What is of interest is only the number of receivers each subset of \mathcal{R} has in common with each of the subsets of \mathcal{S}' (refer also to (4)). We now have the following definition:

Definition 3.1 *Two instances $(\mathcal{R}_1, \mathcal{S}'_1)$ and $(\mathcal{R}_2, \mathcal{S}'_2)$ of the CM problem are isomorphic if*

$$|R_{1,c} \cap S'_{1,c'}| = |R_{2,c} \cap S'_{2,c'}| \quad \forall c, c' \quad (10)$$

It is straightforward to show that an optimal solution to an instance $(\mathcal{R}, \mathcal{S}')$ of the CM problem is also an optimal solution to any instance isomorphic to $(\mathcal{R}, \mathcal{S}')$.

A compact data structure that encompasses all the important knowledge about an instance $(\mathcal{R}, \mathcal{S}')$ of the CM problem is a $C \times C$ matrix $\mathbf{M} = [m_{c,c'}]$. Element $m_{c,c'}$ of the matrix represents the number of receivers that subset R_c of the initial assignment and subset $S'_{c'}$ of the new partition have in common:

$$m_{c,c'} = |R_c \cap S'_{c'}| \quad c, c' = 1, \dots, C \quad (11)$$

An example is given in Figure 1 for a network with $N = 8$ nodes and $C = 3$ wavelengths. By construction, the sum of the elements of the c -th row of \mathbf{M} equals the cardinality of subset R_c , $c = 1, \dots, C$. Similarly, the sum of the elements of the c -th column of \mathbf{M} equals the cardinality of set S'_c , $c = 1, \dots, C$.

$$\mathcal{R} : \begin{array}{l} R_1 = \{6, 7, 8\} \\ R_2 = \{3, 4, 5\} \\ R_3 = \{1, 2\} \end{array} \quad \mathcal{S}' : \begin{array}{l} S'_1 = \{2, 5\} \\ S'_2 = \{4, 6, 8\} \\ S'_3 = \{1, 3, 7\} \end{array} \quad \mathbf{M} = \begin{bmatrix} 0 & 2 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

Figure 1: An instance of the *CM* problem with $N = 8$, $C = 3$, and the corresponding matrix \mathbf{M}

Given a matrix \mathbf{M} derived from an instance $(\mathcal{R}, \mathcal{S}')$ of the *CM* problem, we may not be able to restore the original instance $(\mathcal{R}, \mathcal{S}')$, but we can easily create an instance that is isomorphic to $(\mathcal{R}, \mathcal{S}')$. Let us return to the matrix \mathbf{M} of Figure 1. After computing the row sums of \mathbf{M} , we know that subsets R_1 and R_2 of the initial assignment \mathcal{R} for this instance contain three receivers each, while subset R_3 contains two receivers. To create an assignment for an isomorphic instance of the problem, we may simply let

$$R_1 = \{1, 2, 3\}, \quad R_2 = \{4, 5, 6\}, \quad R_3 = \{7, 8\} \quad (12)$$

We can now create a new partition for the isomorphic instance. From \mathbf{M} we know that subset S'_1 contained two receivers, one receiver in common with R_2 , and one in common with R_3 . We can arbitrarily assign two receivers to S'_1 , one from R_1 and one from R_2 , say, $S'_1 = \{4, 7\}$. Proceeding in a similar manner, and making sure that at each step we do not consider receivers that have already been assigned to previous subsets, we can fill in S'_2 and S'_3 to obtain:

$$S'_1 = \{4, 7\}, \quad S'_2 = \{1, 2, 5\}, \quad S'_3 = \{3, 6, 8\} \quad (13)$$

Even though the new instance of the *CM* problem in (12) and (13) is very different from the original instance shown in Figure 1, both instances produce the same matrix \mathbf{M} . From the point of view of finding the optimal permutation, the two instances are equivalent, because in finding an optimal permutation we are interested in the number of receivers that subsets of \mathcal{R} have in common with subsets of \mathcal{S}' , not the identities of the receivers belonging to those subsets.

3.1.2 Weighted Bipartite Graph Representation

We can view matrix \mathbf{M} as the adjacency matrix of a bipartite graph $G = (V, E)$ with vertex set $V = \{v_1, \dots, v_C, u_1, \dots, u_C\}$ and edge set $E = \{(v_c, u_{c'}), c, c' = 1, \dots, C\}$. We assign a weight equal to $m_{c,c'}$ to edge $(v_c, u_{c'}) \in E, c, c' = 1, \dots, C$. For the example of Figure 1, the corresponding bipartite graph is shown in Figure 2.

We now observe that solving an instance $(\mathcal{R}, \mathcal{S}')$ of the *CM* problem, i.e., finding a permutation that minimizes the distance in (4), is equivalent to finding a matching of maximum weight in the

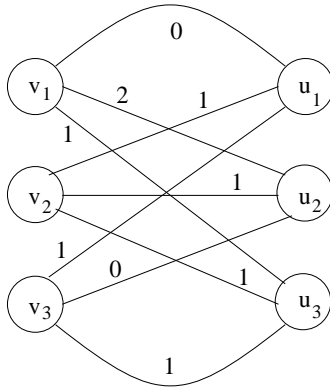


Figure 2: Bipartite graph corresponding to matrix \mathbf{M} of Figure 1

bipartite graph corresponding to $(\mathcal{R}, \mathcal{S}')$. Once such a matching has been found, if edge $(v_c, u_{c'})$ belongs to the matching, then subset $\mathcal{S}'_{c'}$ of receivers should be assigned to wavelength λ_c (that is, $\mathcal{R}'_c = \mathcal{S}'_{c'}$). The *Bipartite Weighed Matching* problem, also known as the *assignment* problem, can be solved in polynomial time using network flows [18]. Thus, there exists a polynomial time solution for the *CM* problem.

3.2 The Constrained Load Balancing Problem

We now consider a different approach for obtaining a new wavelength assignment \mathcal{R}' , given an initial assignment \mathcal{R} and a new traffic matrix \mathbf{T}' , one that attempts to simultaneously satisfy the two requirements for \mathcal{R}' discussed earlier in this section. This approach gives rise to the *Constrained Load Balancing (CLB)* problem, which can be formally stated as a decision problem:

Problem 3.2 (CLB) *Given an initial wavelength assignment \mathcal{R} , a traffic matrix \mathbf{T}' , and two positive integers K and L , is there a wavelength assignment \mathcal{R}' such that $\sum_{j \in \mathcal{R}'_c} b'_j \leq K \forall c$ and $\mathcal{D}(\mathcal{R}, \mathcal{R}') \leq L$?*

The *CLB* problem is \mathcal{NP} -complete because for $L \geq N$ it reduces to the multiprocessor scheduling problem which is \mathcal{NP} -complete [15]. We now present a heuristic for the *CLB* problem, which is based on *LPT* [17], an approximation algorithm for the multiprocessor scheduling problem. In describing the heuristic we will use the terminology of [17], i.e., we will refer to processors, tasks, and execution times instead of channels, receivers, and bandwidth requirements, respectively. This will be helpful in referring to the results of [17] to prove certain properties regarding the performance of our heuristic.

Recall that *LPT* first sorts the N tasks in a list $L = (\nu_1, \dots, \nu_N)$ in decreasing order of their execution times. Initially, each of the first C tasks in the list is assigned to a different processor to execute. Then, whenever a processor completes a task, it scans the list L for the first available task to execute, and this procedure repeats until all tasks have been executed. We modify *LPT* to take into account \mathcal{R} , the previous wavelength assignment (i.e., the previous assignment of tasks to processors), by introducing a *knob* κ , $1 \leq \kappa \leq N$. The new algorithm also orders the tasks in a list L in decreasing order of their execution times. However, when a processor searches for a new task to execute (initially, or after the completion of a task) it does not immediately select the first available task in the list. Instead, it considers the first κ available tasks in the list (if there are less than κ remaining tasks, then all of them are considered). If at least one of these tasks was assigned to the same processor under the previous assignment \mathcal{R} , then the processor starts executing the larger such task, even if it is not the first one in the list of available tasks. Otherwise, if no such task exists, the processor executes the first available task, as in *LPT*. There is one exception to this rule, namely, the first task in the list L (i.e., task ν_1) is always assigned to its processor under \mathcal{R} .

We will call the algorithm just presented the *Generalized LPT (GLPT)* algorithm; its detailed description can be found in Figure 3. It can be easily verified that, by implementing appropriate data structures, the complexity of *GLPT* is $O(N \max\{\log N, C, \kappa\})$. We note that *GLPT* reduces to pure *LPT* for $\kappa = 1$. For higher values of κ , it is more likely that receivers will be assigned to the same channels as before, and the new wavelength assignment \mathcal{R}' will be closer to \mathcal{R} ; this may be achieved at the expense of load balancing. By selecting a value for κ between 1 and N when implementing *GLPT*, the network designer can achieve the desired tradeoff between the two objectives: load balancing and number of retunings.

The following lemma provides an absolute performance ratio regarding the behavior of *GLPT* in terms of load balancing, *regardless* of the value of parameter κ .

Lemma 3.3 *Let ω denote the finish time of a multiprocessor schedule constructed by GLPT for any value of κ , and let ω^* denote the optimal finish time for the same set of tasks. Then,*

$$\frac{\omega}{\omega^*} \leq \frac{3}{2} - \frac{1}{2C} \quad (14)$$

Proof. Let us choose the m , $0 \leq m \leq N$ longest tasks of the set of tasks to be executed and arrange them in a list L which gives the *optimal* solution for these m tasks under this strategy: upon completion of a task, a processor scans the list and starts executing the next available task. Now let us extend L to include all the tasks by adjoining the remaining $N - m$ tasks arbitrarily

Algorithm *Generalized LPT (GLPT)*

Input: Initial wavelength assignment $\mathcal{R} = \{R_c\}$, and new receiver bandwidth requirements b'_j , $j = 1, \dots, N$, derived from the new traffic matrix \mathbf{T}' ; $\lambda(j)$ denotes the receive wavelength of j under \mathcal{R}

Output: New wavelength assignment $\mathcal{R}' = \{R'_c\}$

Parameter: $\kappa, 1 \leq \kappa \leq N$

1. begin
 1. Initialize $R'_c = \phi$, $c = 1, \dots, C$
 2. Order the receivers as (ν_1, \dots, ν_N) such that $b'_{\nu_1} \geq \dots \geq b'_{\nu_N}$
 3. $R'_c \leftarrow \{\nu_1\}$ where $\lambda(\nu_1) = \lambda_c$ // assign the first receiver to its previous channel
 4. For $j = 1$ to $N - 1$ do
 5. Order the channels as $(\lambda_{\pi_1}, \dots, \lambda_{\pi_C})$ such that $\sum_{l \in R'_{\pi_1}} b'_l \leq \dots \leq \sum_{l \in R'_{\pi_C}} b'_l$
 6. Order the non-assigned receivers as $(\nu_1, \dots, \nu_{N-j})$ such that $b'_{\nu_1} \geq \dots \geq b'_{\nu_{N-j}}$
// If one of the first κ receivers was assigned to λ_{π_1} under \mathcal{R} , assign it to the same channel
 7. For $i = 1$ to $\min\{\kappa, N - j\}$ do
 8. If $\lambda(\nu_i) = \lambda_{\pi_1}$ then
 9. $R'_{\pi_1} \leftarrow R'_{\pi_1} \cup \{\nu_i\}$
 10. goto 4
 - // Otherwise, assign the first receiver to λ_{π_1}
 11. $R'_{\pi_1} \leftarrow R'_{\pi_1} \cup \{\nu_1\}$
 12. // end of algorithm
-

Figure 3: The *Generalized LPT* algorithm for the *CLB* problem

to L , forming list $L(m)$. Let $\omega(m)$ denote the finish time for the N tasks when using the above strategy on $L(m)$, and let ω^* denote the optimal finish time for all N tasks. From [17, Theorem 3] we have that:

$$\frac{\omega(m)}{\omega^*} \leq 1 + \frac{1 - \frac{1}{C}}{1 + \lceil \frac{m}{C} \rceil} \quad (15)$$

Let L' denote the corresponding list of tasks for *GLPT*. This list is not known *a priori*, instead, it is formed dynamically during the execution of the algorithm. However, by construction, the same strategy is followed on L' , namely, a processor that becomes idle is always assigned the next available task on L' . Then, the result in (14) follows immediately from (15) for $m = 1$, since, regardless of the value of the knob κ , list L' is formed by concatenating some list of $N - 1$ tasks (as formed by the algorithm) to the list that gives the optimal solution for the longest task ν_1 . \square

4 Numerical Results

We now compare the two approaches for obtaining a new wavelength assignment \mathcal{R}' , given an initial assignment \mathcal{R} and a new traffic matrix \mathbf{T}' :

- The first approach is to run *LPT* [17] on the new receiver bandwidth requirements $\{b'_j\}$ derived from matrix \mathbf{T}' to obtain a partition \mathcal{S}' of the set of receivers into C subsets S'_c ; we then run the Hungarian algorithm [18] to obtain a solution to the *CM* problem, i.e., to map the subsets S'_c to the actual channels. The running time requirements of this approach are $O(N \log N + ???)$.
- The second approach is to run algorithm *GLPT*(κ), shown in Figure 3, with \mathcal{R} and \mathbf{T}' as input, to directly obtain the new assignment \mathcal{R}' ; in our experiments, we have used various values for parameter κ .

The two performance measures of interest are load balancing and the number of receiver retunings required. Since we do not have a polynomial time solution for the load balancing problem, we compare the two approaches against the lower bound, obtained from matrix \mathbf{T}' as $\frac{\sum_{i,j} t_{ij}}{C}$; we note that, in general, this lower bound may not be achievable.

Figures 4 and 5 show the performance of the two approaches in terms of load balancing and number of retunings, respectively, as we vary the number N of nodes in the network; the number of channels remains constant, $C = 10$. Figures 6 and 7 show results for the same performance measures as the number of channels varies while the number of nodes is kept constant at $N = 100$. To obtain the results shown in Figures 4 – 7 we constructed random traffic matrices whose elements were integers uniformly distributed in the range 0 through 20. Each point plotted corresponds to the average of 100 random instances for the stated values of N and C ; 95% confidence intervals are also plotted in the figures.

Our first observation from Figures 4 and 6 is that the first approach (i.e., employing *LPT* for load balancing and then solving the channel mapping problem), provides the best performance in terms of load balancing, as expected. However, algorithm *GLPT* with $\kappa = 5$ (*GLPT*(5)) performs almost identical to *LPT*, while *GLPT*(10) is also very close to *LPT*. As κ increases, *GLPT* starts behaving sub-optimally in terms of load balancing, as expected. However, even when κ is as large as 40, *GLPT* is never more than 14% away from the lower bound, and in some cases it is as close as 3%. In fact, because of Lemma 3.3, *GLPT* is guaranteed to always be within 50% from the optimal, regardless of the value of parameter κ . (The behavior of all algorithms for $C = 15$ in Figure 6 can be explained by noting that, unlike the values $C = 5, 10, 20, 25$, the value $C = 15$ does not divide the number of nodes $N = 100$ exactly. As a result, some channels will be assigned

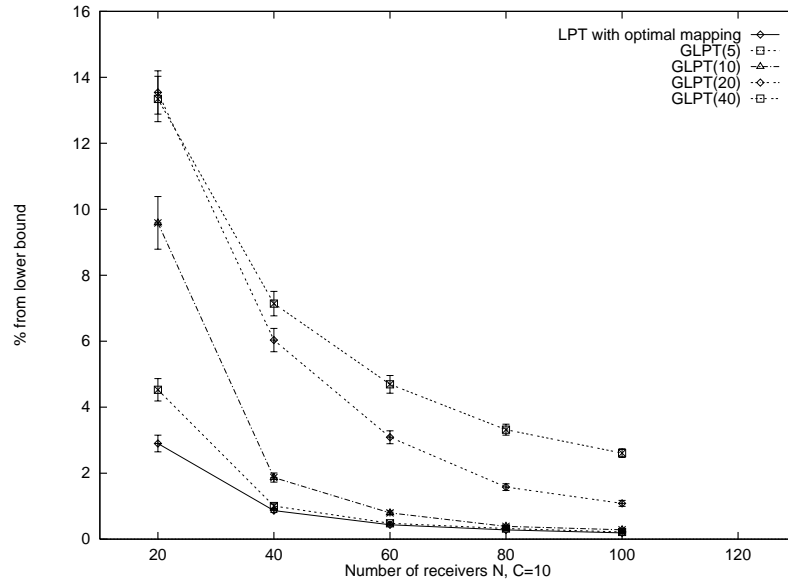


Figure 4: Algorithm comparison on load balancing ($C = 10$ channels, random traffic matrices)

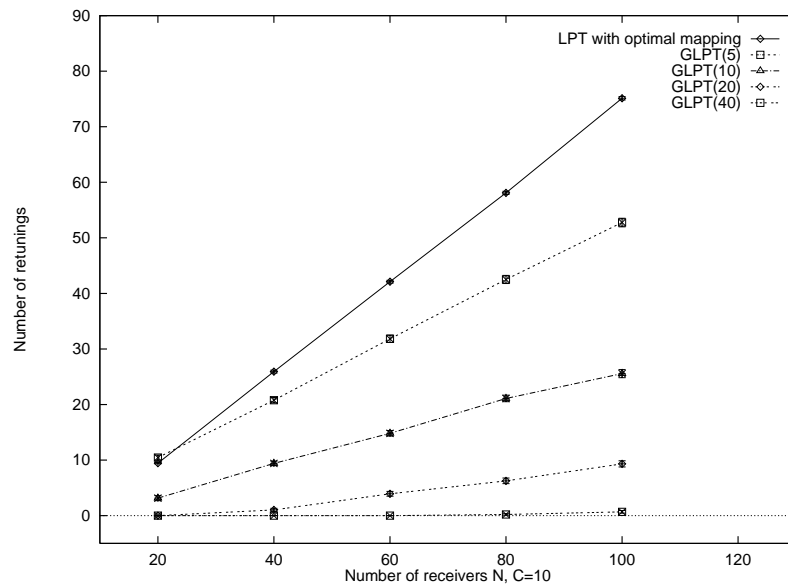


Figure 5: Algorithm comparison on number of retunings ($C = 10$ channels, random traffic matrices)

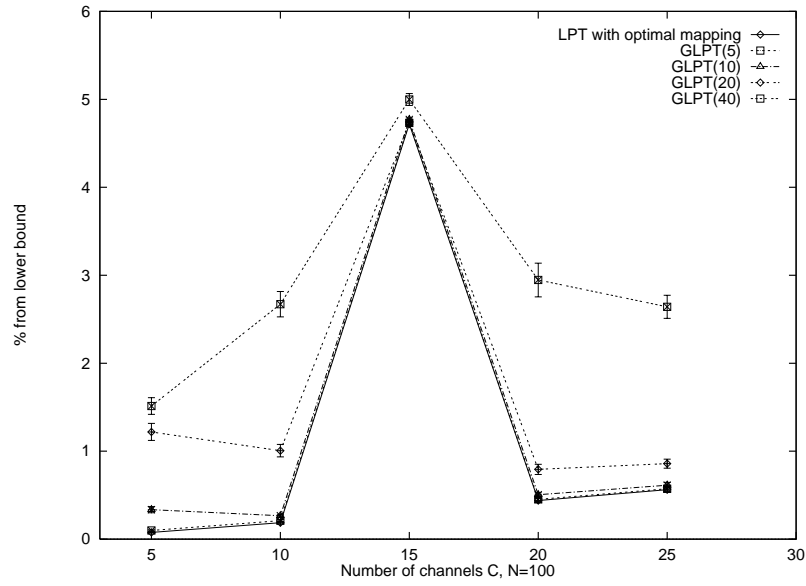


Figure 6: Algorithm comparison on load balancing ($N = 100$ nodes, random traffic matrices)

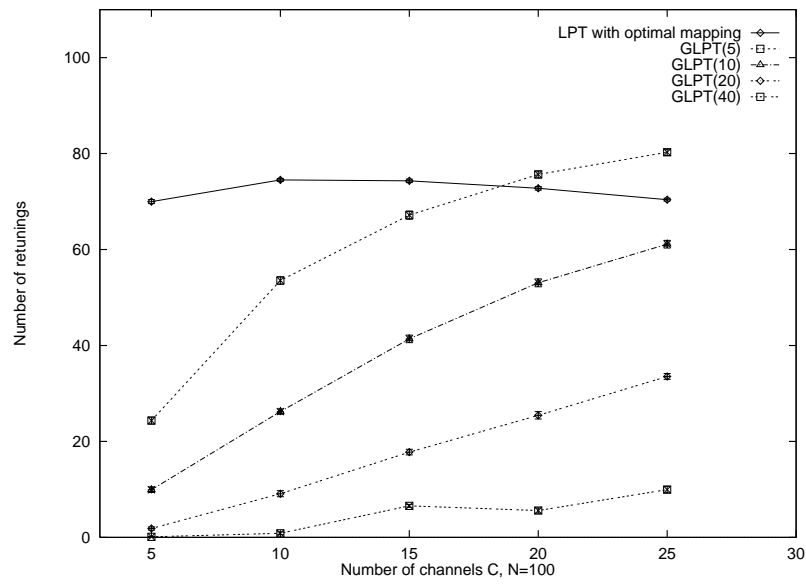


Figure 7: Algorithm comparison on number of retunings ($N = 100$ nodes, random traffic matrices)

more receivers than others, making it difficult to achieve the lower bound. In other words, there is nothing inherently wrong with the algorithms, it is just that the lower bound does not accurately characterize the optimal solution when $C = 15$.)

Let us now turn our attention to Figure 5 which plots the average number of retunings as a function of the size N of the network. We observe that the first approach always requires the most number of retunings, and that its retuning requirements increase linearly with the size of the network. Furthermore, the expected fraction of receivers that need to be retuned increases with the number of nodes, from 50% when $N = 20$, to 75% when $N = 100$. This means that this approach is not scalable, since, for large N , either the duration of the reconfiguration phase, or the fraction of the network that becomes unavailable, will be significant. The behavior of this approach in terms of number of retunings is in agreement with intuition: *LPT* is very successful in balancing the load of the network, but it does not take into account the previous wavelength assignment. As a result, the distance between the initial and target assignments tends to be large. We note also that, in many cases, the expected number of retunings is close to the number of retunings under the worst-case scenario of Lemma 3.2.

From the same figure we see that, for small values of κ , algorithm *GLPT* requires a number of retunings which also increases linearly with the size of the network. However, the rate of increase is much slower (for instance, when $\kappa = 5$, about 50% of the receivers are retuned for all values of N , while when $\kappa = 10$, about 20% of the receivers are retuned on average). As κ increases, the behavior of *GLPT* improves dramatically. For $\kappa = 20$, the number of retunings does increase with N , but it is always less than 10, while when $\kappa = 40$, only about one receiver needs to be retuned, *independently* of the number N of nodes. In fact, doubling the value of parameter κ reduces the number of retunings to less than half its previous value. As a result, it does not make sense to use a value of κ that is larger than 40, since doing so may increase the running time requirements of algorithm *GLPT* without any significant effect on the number of retunings.

In Figure 7 we plot the number of retunings required against the number of channels, for $N = 100$. We note that the first approach always requires about 70% of the receivers to be retuned, regardless of the number of channels. On the other hand, the number of retunings required by *GLPT* increases almost linearly with C for all values of κ ; also, larger values of κ result in a smaller number of retunings, as expected. This result, combined with our previous observations, indicates that, for certain values of parameter κ (in this case, for $20 \leq \kappa \leq 40$), *GLPT* provides a scalable approach to reconfiguring the network since (a) it achieves a guaranteed level of performance in terms of load balancing, (b) its retuning requirements are low, and more importantly, (c) the number of retunings scales with the number of channels, *not* the number of nodes in the network.

The results plotted in Figures 4 – 7 were obtained by randomly selecting the initial traffic matrix \mathbf{T} , and then randomly selecting the target matrix \mathbf{T}' , independently of \mathbf{T} . In practice, the new traffic matrix \mathbf{T}' will be closely related to the old matrix \mathbf{T} , differing only by the changes in the traffic demands that have taken place. To study the relative performance of the two approaches under a model that more closely captures the characteristics of a realistic traffic scenario, we have run a set of experiments in which we have used Brownian motion to model the change in the source-destination traffic demands.

In the new model, the initial random matrix \mathbf{T} was constructed as before. This matrix was then evolved through a series of steps to obtain the target matrix \mathbf{T}' . The Brownian motion was modeled by using two asymmetric probabilities: the probability of moving towards the “wall” and the probability of moving away from the “wall”, the “wall” being either the lower or the upper limit on the source-destination traffic demand (to obtain results comparable to those in Figures 4 - 7, we used 0 and 20, respectively, for the lower and upper limit on the traffic demands). At every step of the evolution process, each element of the demand matrix \mathbf{T} is treated as a one-dimensional Brownian particle. In our model, the probability of moving away from the wall (set to 0.5) was higher than the probability of moving towards the wall (set to 0.2)¹. Thus, a particle always has a “direction of likely movement”. Based on these probabilities, a newly generated random number at each step determines whether the bandwidth demand for a certain source-destination pair will increase by an amount δ , decrease by δ , or remain the same, independently of the other elements; in our experiments we let $\delta = 1$. If an element hits the upper or lower limit, its “direction of likely movement” is reversed. After performing several steps ($\sim 10 - 20$) in this manner, the resulting matrix was taken as the new traffic matrix \mathbf{T}' .

The results from the Brownian model are shown in Figures 8 - 11. As we can see, the new model had little effect on the behavior of the various algorithms, confirming our conclusions regarding the relative performance of the two approaches.

5 Concluding Remarks

We considered the problem of updating the bandwidth allocation in single-hop WDM networks to accommodate varying traffic demands, by retuning a set of slowly tunable receivers. Our objective was to balance the traffic load across all channels, while keeping the number of retunings to a minimum. We studied a straightforward approach to obtaining a new wavelength assignment, one

¹Note that these probabilities need not sum up to unity; if they do not, the difference represents the probability that the particle will not change its position during a step.

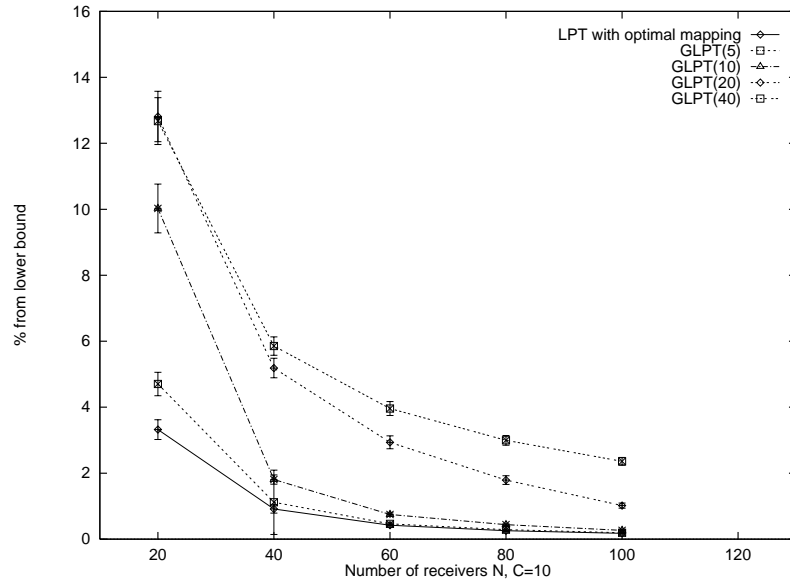


Figure 8: Algorithm comparison on load balancing ($C = 10$ channels, Brownian model)

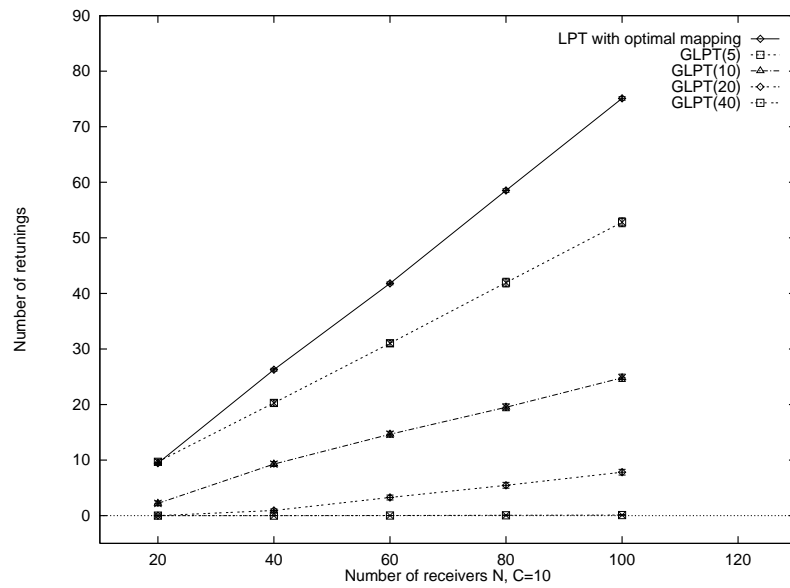


Figure 9: Algorithm comparison on number of retunings ($C = 10$ channels, Brownian model)

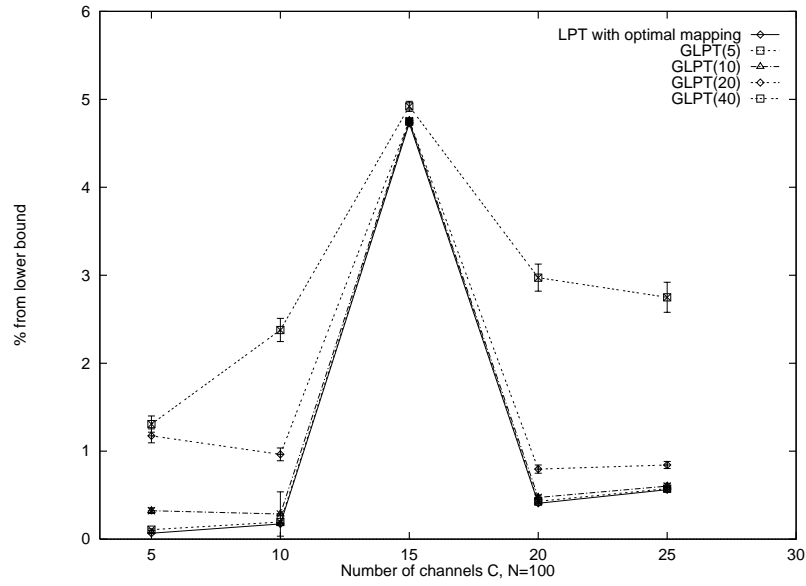


Figure 10: Algorithm comparison on load balancing ($N = 100$ nodes, Brownian model)

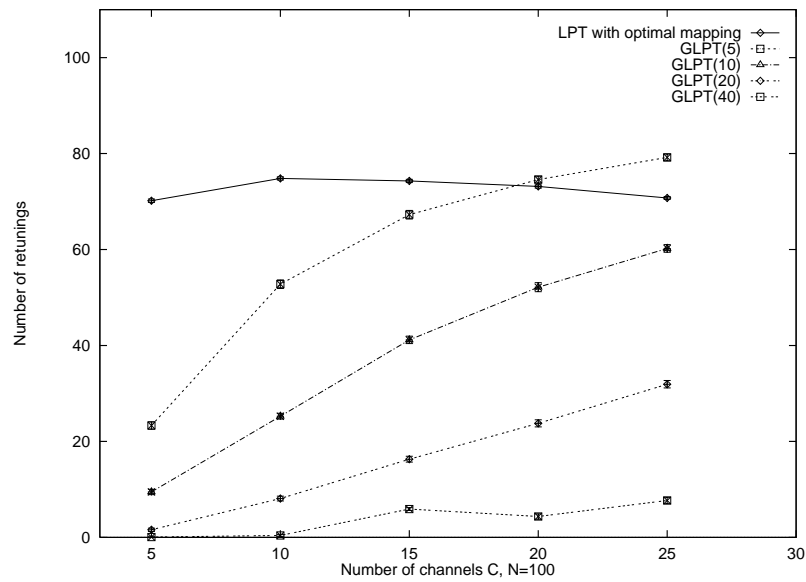


Figure 11: Algorithm comparison on number of retunings ($N = 100$ nodes, Brownian model)

that employs well-known algorithms to satisfy the two requirements independently of each other, and we have shown that it is not scalable. We then presented a new algorithm that attempts to construct the new wavelength assignment in a way that simultaneously achieves the stated objectives. The algorithm provides for tradeoff selection between the two requirements, and scales well with the size of the network. The main conclusion of our work is that it is possible to employ rapidly tunable optical devices only at one end of the network without making sacrifices in terms of performance, thus leading to lightwave architectures that can be realized cost effectively.

References

- [1] B. Mukherjee. WDM-Based local lightwave networks Part I: Single-hop systems. *IEEE Network Magazine*, pages 12–27, May 1992.
- [2] R. Ramaswami. Multiwavelength lightwave networks for computer communication. *IEEE Communications Magazine*, pages 78–88, February 1993.
- [3] P. E. Green. *Fiber Optic Networks*. Prentice-Hall, Englewood Cliffs, New Jersey, 1993.
- [4] V. Sivaraman and G. N. Rouskas. HiPeR- ℓ : A High Performance Reservation protocol with ℓ ook-ahead for broadcast WDM networks. Technical Report TR-96-06, North Carolina State University, Raleigh, NC, 1996. (Submitted to Infocom '97).
- [5] K. Sivalignam and P. Dowd. A multi-level WDM access protocol for an optical inter-connected multi-processor system. *IEEE/OSA Journal of Lightwave Technology*, 13(11):2152–2167, November 1995.
- [6] Mon-Song Chen, N. R. Dono, and R. Ramaswami. A media-access protocol for packet-switched wavelength division multiaccess metropolitan area networks. *IEEE Journal on Selected Areas in Communications*, 8(6):1048–1057, August 1990.
- [7] N. R. Dono, P.E. Green Jr., K. Liu, R. Ramaswami, and F. F. Tong. A wavelength division multiple access network for computer communication. *IEEE Journal on Selected Areas in Communications*, 8(6):983–994, August 1990.
- [8] G. N. Rouskas and M. H. Ammar. Dynamic reconfiguration in multihop WDM networks. *Journal of High Speed Networks*, 4(3):221–238, October 1995.
- [9] J-F. P. Labourdette, F. W. Hart, and A. S. Acampora. Branch-exchange sequences for reconfiguration of lightwave networks. *IEEE Transactions on Communications*, 42(10):2822–2832, October 1994.
- [10] G. N. Rouskas and V. Sivaraman. On the design of optimal TDM schedules for broadcast WDM networks with arbitrary transceiver tuning latencies. In *Proceedings of INFOCOM '96*, pages 1217–1224. IEEE, March 1996.
- [11] M. Azizoglu, R. A. Barry, and A. Mokhtar. The effects of tuning time in bandwidth-limited optical broadcast networks. In *Proceedings of INFOCOM '95*, pages 138–145. IEEE, April 1995.

- [12] M. S. Borella and B. Mukherjee. Efficient scheduling of nonuniform packet traffic in a WDM/TDM local lightwave network with arbitrary transceiver tuning latencies. In *Proceedings of INFOCOM '95*, pages 129–136. IEEE, April 1995.
- [13] G. R. Pieris and G. H. Sasaki. Scheduling transmissions in WDM broadcast-and-select networks. *IEEE/ACM Transactions on Networking*, 2(2):105–110, April 1994.
- [14] M. R. Garey, R. L. Graham, and D. S. Johnson. Performance guarantees for scheduling algorithms. *Operations Research*, 26:3–21, Jan 1978.
- [15] M. R. Garey and D. S. Johnson. *Computers and Intractability*. W. H. Freeman and Co., New York, 1979.
- [16] E. Coffman, M. R. Garey, and D. S. Johnson. An application of bin-packing to multiprocessor scheduling. *SIAM Journal of Computing*, 7:1–17, Feb 1978.
- [17] R. L. Graham. Bounds on multiprocessing timing anomalies. *SIAM Journal of Applied Mathematics*, 17(2), Mar 1969.
- [18] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, Englewood Cliffs, N.J., 1993.