Photonic Network Communications, 1, 49–64 (1999)

Reconfiguration and Dynamic Load Balancing in Broadcast WDM Networks*

Ilia Baldine[†]
MCNC, RTP, NC 27709, USA

George N. Rouskas
Department of Computer Science, North Carolina State University, Raleigh, NC 27695-7534, USA

Received September 10, 1998; Revised November 20, 1998

Abstract. In optical WDM networks, an assignment of transceivers to channels implies an allocation of the bandwidth to the various network nodes. Intuition suggests, and our recent study has confirmed, that if the traffic load is not well balanced across the available channels, the result is poor network performance. Hence, the time-varying conditions expected in this type of environment call for mechanisms that periodically adjust the bandwidth allocation to ensure that each channel carries an almost equal share of the corresponding offered load. In this paper we study the problem of dynamic load balancing in broadcast WDM networks by retuning a subset of transceivers in response to changes in the overall traffic pattern. Assuming an existing wavelength assignment 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 transceivers that need to be retuned is minimized. The latter objective is motivated by the fact that tunable transceivers take a non-negligible amount of time to switch between wavelengths during which parts of the network are unavailable for normal operation. Furthermore, this variation in traffic is expected to take place over larger time scales (i.e., retuning will be a relatively infrequent event), making slowly tunable devices a cost effective solution. Our main contribution is a new approximation algorithm for the load balancing problem that provides for tradeoff selection, using a single parameter, between two conflicting goals, namely, the degree of load balancing and the number of transceivers that need to be retuned. This algorithm leads to a scalable approach to reconfiguring the network since, in addition to providing guarantees in terms of load balancing, the expected number of retunings scales with the number of channels, *not* the number of nodes in the network.

Keywords: broadcast optical networks, wavelength division multiplexing (WDM), reconfiguration, dynamic load balancing

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

*This work was supported by the NSF under grant NCR-9701113.

[†]This work was performed while the author was with the Department of Computer Science, North Carolina State University.

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. 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. A comprehensive evaluation of reconfiguration policies and retuning strategies for single-hop networks has been conducted by the authors in [10], where we demonstrated the benefits of reconfiguration through both analytical and simulation results.

In this paper we consider the problem of reconfiguring a single-hop network by retuning a subset of the slowly tunable receivers in response to changing network traffic conditions. Our objective is to ensure that the traffic load remains balanced across the various channels, while minimizing the number of receivers that need to be retuned. We show that employing well-known load balancing algorithms leads to an approach that 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, between the two conflicting goals. Our algorithm is simple, fast, scalable, and tends to select the least utilized receivers for retuning, hence minimizing the impact of the reconfiguration phase on the carried traffic. Although our work is motivated by a problem in optical networks, our solution techniques are applicable to a generalized version of the classical multiprocessor scheduling problem [11], whereby it takes a non-negligible amount of time to transfer tasks among processors.

The next section introduces the network model, and discusses the issues arising during the reconfiguration phase. In Section 3 we describe two approaches for dynamically load balancing by retuning the slowly tunable receivers. 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 a passive broadcast optical medium that supports $C < N$ wavelengths, $\lambda_1, \ldots, \lambda_C$, as shown in Fig. 1. 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 $\Delta_t(\Delta_r)$ 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 $\Delta_r \gg \Delta_t \geq 1$, where Δ_t is a small integer, while Δ_r takes values that may be significantly greater than Δ_t. The main motivation for employing slowly tunable receivers vs. fast tunable ones is the significant savings in cost that can be realized.

We distinguish two levels of network operation, differing mainly in the time scales at which they take place. At the *packet scheduling* level, connectivity among the network nodes is provided by reservation protocol such as HiPeR-ℓ [4] that requires tunability only at the transmitting end. The protocol schedules packets for transmission by employing scheduling algorithms that can effectively mask the tuning latency of tunable transmitters [12–16]. Since the

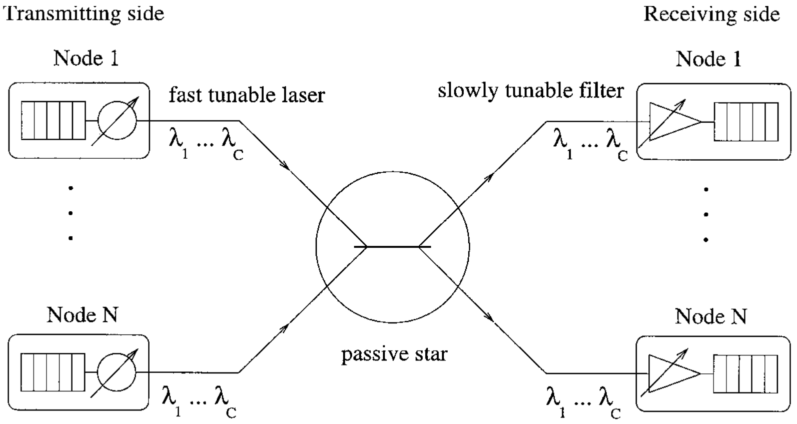

Transmitting side
Receiving side
Node 1
Node 1
fast tunable laser
slowly tunable filter
$\lambda_1 \dots \lambda_C$
$\lambda_1 \dots \lambda_C$
Node N
Node N
passive star
$\lambda_1 \dots \lambda_C$
$\lambda_1 \dots \lambda_C$

Fig. 1. A broadcast optical network with N nodes and C channels.

receiver latency Δ_r is significantly long and cannot be overlapped with packet transmissions, at this level of operation the receivers are considered to be fixed tuned 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:

$$R_c = \{j | \lambda(j) = \lambda_c\} \quad c = 1, \dots, C. \tag{1}$$

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 (1) implies an allocation of the available bandwidth to the various receivers. The availability 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. 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 on the performance of HiPeR-ℓ [4], a new reservation protocol for broadcast WDM networks, has confirmed this intuition. Let us define parameter ε_b such that no channel carries more than $\frac{(1+\varepsilon_b)}{C}$ times the total traffic offered to the network. In other words, ε_b is a measure of the *degree of load balancing* of the network; under perfect load balancing, $\varepsilon_b = 0$. It was shown in [4] that the maximum sustained throughput γ (i.e., the number of packets successfully transmitted per packet time) is directly affected by ε_b through the following stability condition:

$$\gamma < \frac{C}{(1+\varepsilon_b)(1+\varepsilon_s)}. \tag{2}$$

It can be seen from (2) that the higher the degree of load balancing (i.e., the lower the value of ε_b is), the higher the overall arrival rate γ that the network can accommodate, and vice versa. (Parameter ε_s is the guarantee on the schedule length and depends on the scheduling algorithm used, but for the purposes of this discussion it can be considered a constant; for more details, the reader is referred to [4].) Although the stability condition (2) 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 $\mathbf{T} = [t_{ij}]$. We emphasize that quantity t_{ij} is a measure of the long-term traffic originating at node i and terminating at node j, or the effective bandwidth [17] of the traffic from i to 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 $\mathbf{T}$:

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

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 [11], 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 [18], meaning that a polynomial-time algorithm is unlikely to be found. Two approximation algorithms for this problem are *MULTIFIT* [19], with an absolute performance ratio of 1.22, and *LPT* [20], 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 sub-optimal.

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 has been studied by the authors in [21]; in this paper, 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:

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

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 (which now becomes $\mathcal{D}(\mathcal{R}, \mathcal{R}')\Delta_r$). Between these policies at the two ends of the spectrum lie a range of approaches in which two or more receivers are retuned simultaneously.

During the transition period, the network incurs some cost in terms of packet delay, packet loss, packet resequencing, 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. Thus, regardless of the policy used, the number of retuning operations $\mathscr{D}(\mathscr{R}, \mathscr{R}')$ emerges as an important parameter. The significance of $\mathscr{D}(\mathscr{R}, \mathscr{R}')$ has been studied in [10], where it was shown that the number of retunings is one of the factors that determine 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 $\mathscr{R}$ 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 Determining the New Wavelength Assignment

Consider a network operating under wavelength assignment $\mathscr{R}$ 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 $\mathscr{R}'$ such that:

1. the new traffic load, as specified by matrix $\mathbf{T}'$ is evenly spread across the C channels, and
2. the number of retunings required to take the network from assignment $\mathscr{R}$ to assignment $\mathscr{R}'$ is as small as possible.

We note that these requirements on $\mathscr{R}'$ represent two conflicting objectives: minimizing the number of retunings alone would result in $\mathscr{R}'$ being the same as $\mathscr{R}$, 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 $\mathscr{R}'$, 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 $\mathscr{R}$. The second step assigns the new subsets of receivers to wavelengths so as to minimize the number of retunings required starting from $\mathscr{R}$. This gives rise to the *Channel Assignment* 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 assignment problem.

3.1 The Channel Assignment Problem

We consider an initial wavelength assignment $\mathscr{R}$ 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 $\mathscr{S}' = \{S'_c\}$ of the set of receivers into C sets S'_c, $c = 1, \ldots, C$. This partition $\mathscr{S}'$ is such that the bandwidth requirements (as defined by matrix $\mathbf{T}'$) 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 $\mathscr{S}'$ of the set of receivers, *not* a wavelength assignment as defined in (1); in other words, there is no association among the receiver subsets S'_c and the available wavelengths.

From $\mathscr{S}'$ we may obtain a new wavelength assignment $\mathscr{R}'$ by mapping each subset S'_c to one of the wavelengths, such that no two subsets map to the same wavelength. It can be easily seen that using a simple 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. 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 Assignment (CA)* problem; it can be formally stated as:

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

Problem *CA* is an example of a *bipartite weighted matching* or *assignment* problem [22], when given a weighted bipartite network it is required to find a perfect matching of minimum weight. Several polynomial-time algorithms exist for the assignment problem [22]. Unfortunately, this approach to obtaining the new wavelength assignment does not scale well with the size of the network. Even though the *LPT* or *MULTIFIT* algorithms can successfully balance the traffic load across the C channels, this approach performs poorly in terms of the number of retunings required to take the network to the new wavelength assignment. The next lemma states that, even under an optimal solution to the *CA* problem, the number of retunings required may be very large.

Lemma 3.1: *Let $\mathscr{R}$ and $\mathscr{S}'$ be the initial wavelength assignment and new partition, respectively, of an arbitrary instance of the CA problem for a network with N nodes and C channels. If the optimal solution to this instance yields wavelength assignment $\mathscr{R}'$, $N - C$ is an upper bound on the number of retunings required, i.e.,*

$$\mathscr{D}(\mathscr{R}, \mathscr{R}') \leq N - C. \tag{5}$$

Proof: *See Appendix A.* □

The main disadvantage of this solution is that it always satisfies the load balancing objective at the expense of the number of retunings. Furthermore, all the algorithms for the assignment problem are computationally expensive [22], making it difficult to apply them in dynamic high-speed environments. What is needed is a fast algorithm that looks at both objectives at the same time, and which allows the designer to adjust the tradeoff among them in favor of one or the other.

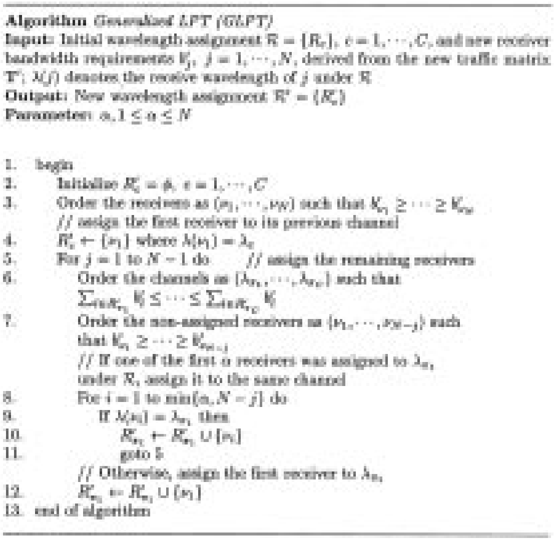
3.2 The Constrained Load Balancing Problem

We now consider a different approach to obtaining a new wavelength assignment $\mathscr{R}'$, given an initial assignment $\mathscr{R}$ and a new traffic matrix $\mathbf{T}'$, one that attempts to simultaneously satisfy the two requirements for $\mathscr{R}'$ 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 $\mathscr{R}$, a traffic matrix $\mathbf{T}'$, and two positive integers K and D, is there a wavelength assignment $\mathscr{R}'$ such that $\sum_{j \epsilon R'_c} b'_j \leq K \forall c$ and $\mathscr{D}(\mathscr{R}, \mathscr{R}') \leq D$?*

The *CLB* problem is $\mathscr{NP}$-complete because for $D \geq N$ it reduces to the multiprocessor scheduling problem which is $\mathscr{NP}$-complete [18]. We now present a heuristic for the *CLB* problem, which is based on *LPT* [20], an approximation algorithm for the multiprocessor scheduling problem. In describing the heuristic we will use the terminology of [20], 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 [20] to prove certain properties regarding the performance of our heuristic.

Recall that *LPT* first sorts the N tasks in a list $L = (\nu_1, \ldots, \nu_N)$ 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 $\mathscr{R}$, the previous wavelength assignment (i.e., the previous assignment of tasks to processors), by introducing a parameter α, $1 \leq \alpha \leq N$. The new algorithm also orders the tasks in a list L in decreasing order of their execution times. However, when a processor i 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 α available tasks in the list (if there are less than α remaining tasks, then all of them are considered). If at least one of these tasks was assigned to the same processor i under the previous assignment $\mathscr{R}$, 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 $\mathscr{R}$.



```
Algorithm Generalized LPT (GLPT)
Input: Initial wavelength assignment $\mathcal{R} = \{R_c\}$, $c = 1, \cdots, C$, and new receiver
bandwidth requirements $t'_j$, $j = 1, \cdots, N$, 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: $\alpha, 1 \leq \alpha \leq N$

1.  begin
2.      Initialize $R'_c = \phi$, $c = 1, \cdots, C$
3.      Order the receivers as $(\nu_1, \cdots, \nu_N)$ such that $t'_{\nu_1} \geq \cdots \geq t'_{\nu_N}$
        // assign the first receiver to its previous channel
4.      $R'_c \leftarrow \{\nu_1\}$ where $\lambda(\nu_1) = \lambda_c$
5.      For $j = 1$ to $N - 1$ do        // assign the remaining receivers
6.          Order the channels as $(\lambda_{c_1}, \cdots, \lambda_{c_C})$ such that
            $\sum_{k \in R'_{c_1}} t'_k \leq \cdots \leq \sum_{k \in R'_{c_C}} t'_k$
7.          Order the non-assigned receivers as $(\nu_1, \cdots, \nu_{N-j})$ such
            that $t'_{\nu_1} \geq \cdots \geq t'_{\nu_{N-j}}$
            // If one of the first $\alpha$ receivers was assigned to $\lambda_{c_1}$,
            under $\mathcal{R}$, assign it to the same channel
8.          For $i = 1$ to $\min\{\alpha, N - j\}$ do
9.              If $\lambda(\nu_i) = \lambda_{c_1}$ then
10.                 $R'_{c_1} \leftarrow R'_{c_1} \cup \{\nu_i\}$
11.                 goto 5
            // Otherwise, assign the first receiver to $\lambda_{c_1}$
12.         $R'_{c_1} \leftarrow R'_{c_1} \cup \{\nu_1\}$
13. end of algorithm
```

Fig. 2. The *Generalized LPT* algorithm for the *CLB* problem.

We will call the algorithm just presented the *Generalized LPT (GLPT)* algorithm; its detailed description can be found in Fig. 2. We note that *GLPT* reduces to pure *LPT* for $\alpha = 1$. For higher values of α, 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 α 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. It can also be easily verified that, by implementing appropriate data structures, the complexity of *GLPT* is $O(N \max\{\log N, C, \alpha\})$. Note that the algorithm needs to be executed only prior to reconfiguration instants, and that reconfiguration is expected to take place at larger time scales. Furthermore, the algorithm can be executed in parallel with normal network operation, and its results (i.e., the new WLA) can be used once the algorithm terminates. Thus, we do not expect the processing time of the algorithm to present a bottleneck, even at very high speed local area networks.

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

Lemma 3.2: *Let ω denote the finish time of a multiprocessor schedule constructed by GLPT for any value of α, and let ω^* denote the optimal finish time for the same set of tasks. Then,*

$$\frac{\omega}{\omega^*} \leq 2 - \frac{1}{C}. \tag{6}$$

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 the following 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 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 [20, Theorem 3] we have that:*

$$\frac{\omega(m)}{\omega^*} \leq 1 + \frac{1 - \frac{1}{C}}{1 + \left\lfloor \frac{m}{C} \right\rfloor}. \tag{7}$$

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 (6) follows immediately from (7) for $m = 1$, since, regardless of the value of the parameter α, 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.* □

Finally, we note that the *CLB* problem is a generalized version of the classical multiprocessor scheduling problem [11], whereby it is necessary to transfer tasks between processors for load-balancing, but this transfer takes a non-negligible amount of time. Because of Lemma 3.2, *GLPT* is an approximation algorithm for this new problem.

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* [20] on the new receiver bandwidth requirements $\{b'_j\}$ derived from matrix $\mathbf{T}'$ to obtain a partition $\mathscr{S}'$ of the set of receivers into C subsets S'_c we then run the *Shortest Augmenting Paths* algorithm [22] to obtain a solution to the *CA* problem, i.e., to map the subsets S'_c to the actual channels. The running time requirements of this approach are $O(N \log N + N^4)$.
- The second approach is to run algorithm $GLPT(\alpha)$, shown in Fig. 2, with $\mathscr{R}$ and $\mathbf{T}'$ as input, to directly obtain the new assignment $\mathscr{R}'$; in our experiments, we have used various values for parameter α.

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.

Figs. 3 and 4 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, $C = 10$. Figs. 5 and 6 show results for the same performance measures as the number of channels varies while the number of nodes is kept constant at $N = 120$. To obtain the results shown in Figs. 3–6 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 have also been computed, but they are so narrow that they are not plotted in the figures.

Our first observation from Figs. 3 and 5 is that the first approach (i.e., employing *LPT* for load balancing and then solving the channel assignment problem),

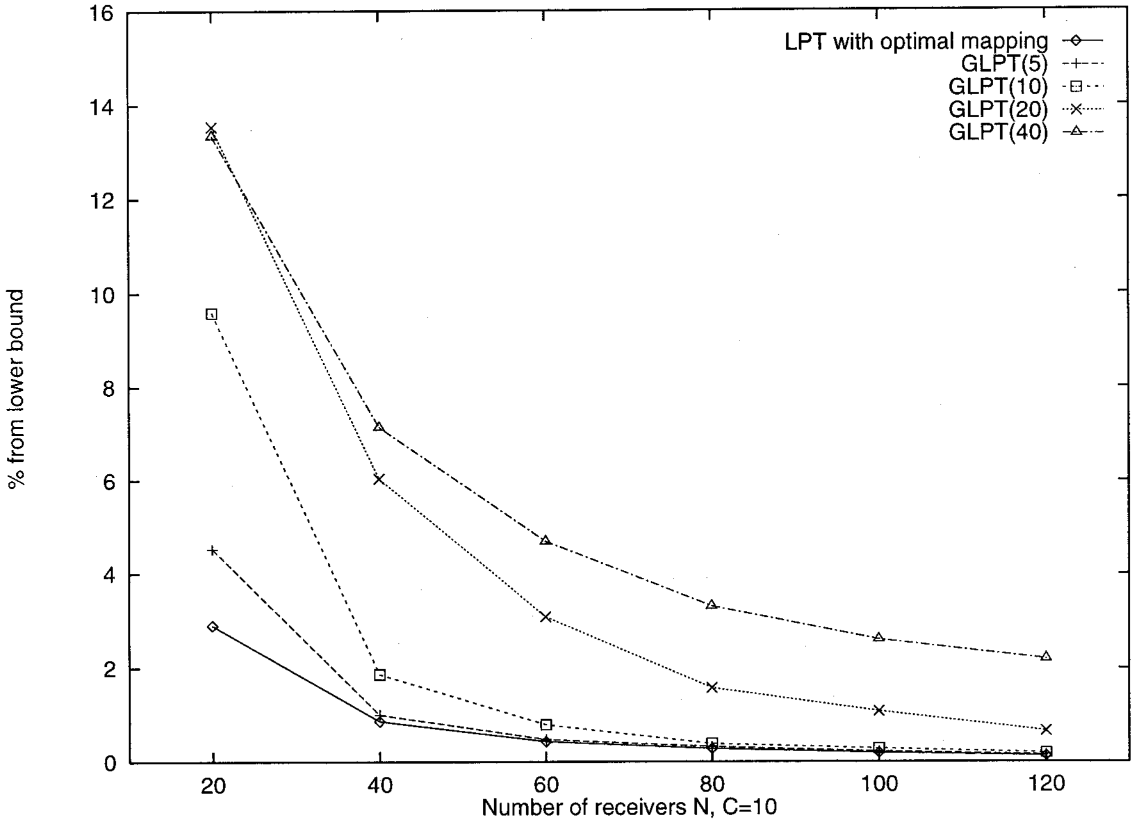


Fig. 3. Algorithm comparison on load balancing ($C = 10$ channels, random traffic matrices).

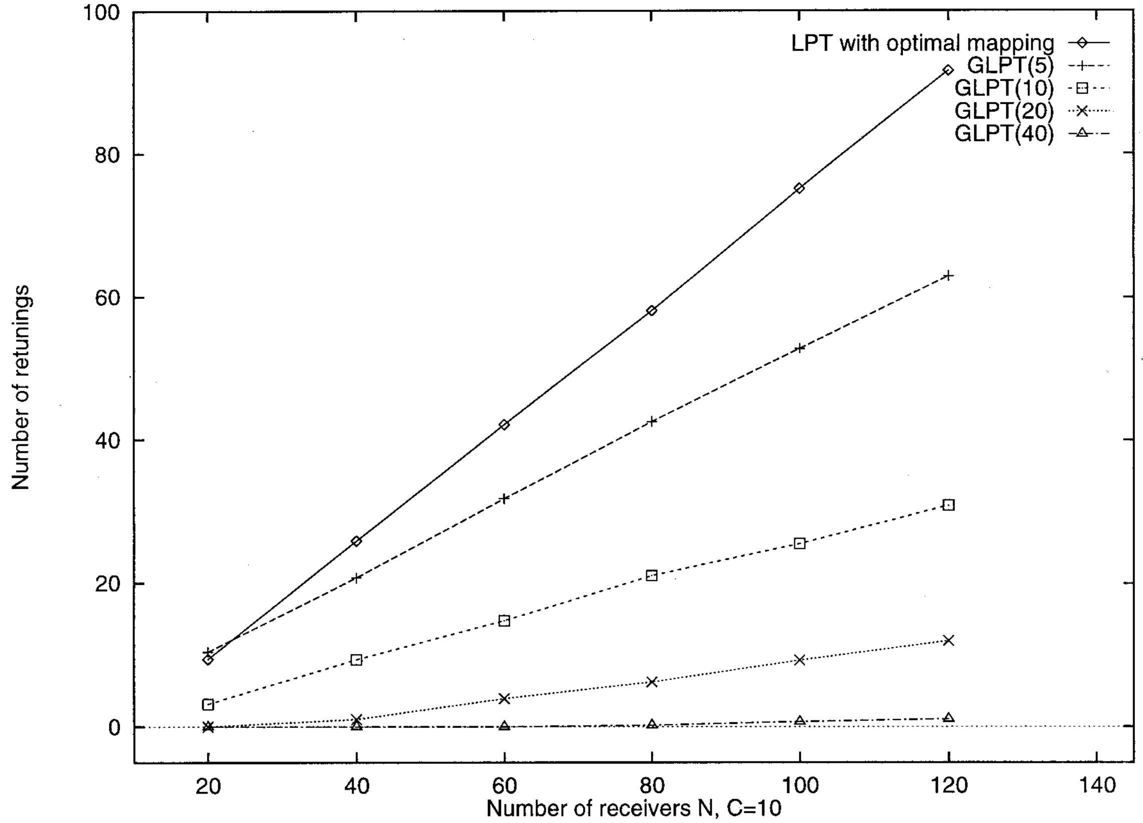


Fig. 4. Algorithm comparison on number of retunings ($C = 10$ channels, random traffic matrices).

provides the best performance in terms of load balancing, as expected. However, algorithm *GLPT* with $\alpha = 5$ (*GLPT(5)*) performs almost identical to *LPT*, while *GLPT(10)* is also very close to *LPT*. As α increases, *GLPT* starts behaving sub-optimally in terms of load balancing, as expected. However, even when α 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 3.2, *GLPT* is guaranteed to always be within 100% from the optimal, regardless of the value of parameter α.

Let us now turn our attention to Fig. 4 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 $N = 120$. This behavior suggests that the 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, for all values of N, the expected number of retunings is very close to the upper bound in Lemma 3.1.

From the same figure we see that, for small values of α, 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 $\alpha = 5$, about 50% of the receivers are retuned for all values of N, while when $\alpha = 10$,

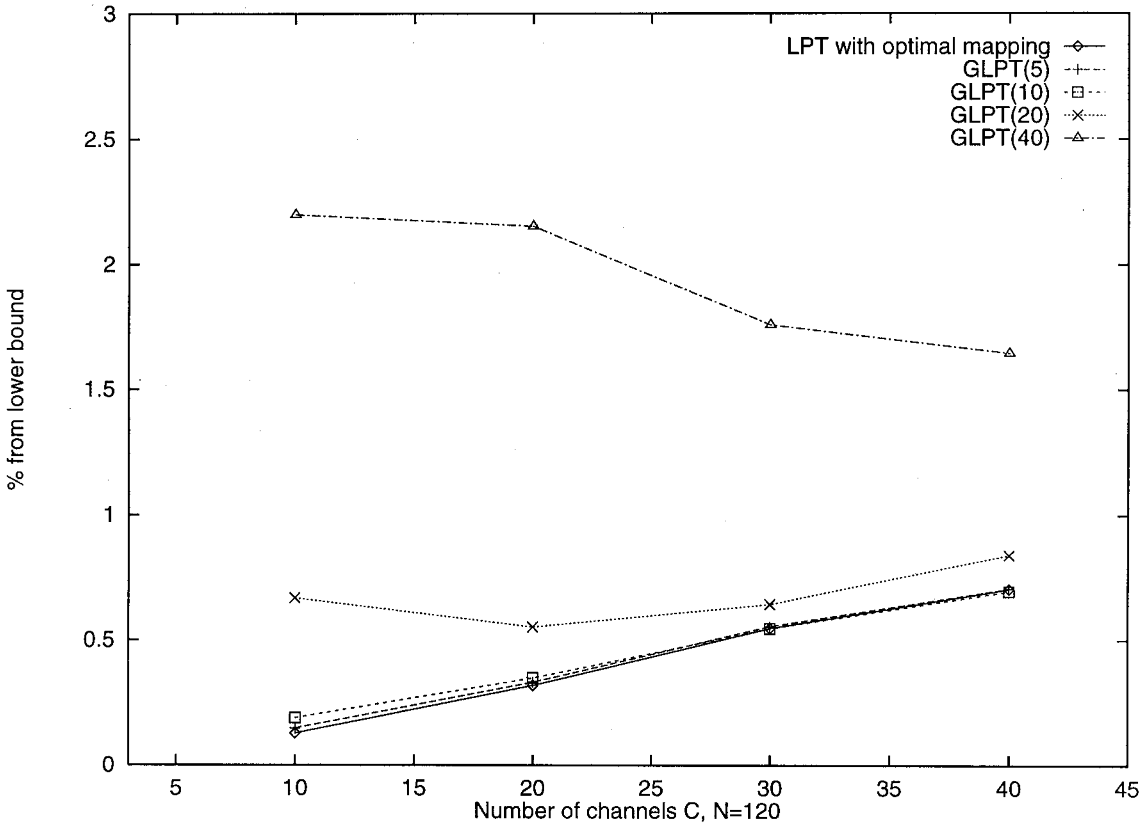


Fig. 5. Algorithm comparison on load balancing ($N = 120$ nodes, random traffic matrices).

about 20% of the receivers are retuned on average). As α increases, the behavior of *GLPT* improves dramatically. For $\alpha = 20$, the number of retunings does increase with *N*, but it is at most 12, while when $\alpha = 40$, only about one receiver needs to be retuned, *independently* of the number *N* of nodes for the range of *N* and *C* values considered. In fact, doubling the value of parameter α 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 α that is, in this case, larger than 40, since doing so may increase the running time requirements of algorithm *GLPT* without any significant effect on the number of retunings. This behavior of *GLPT* can be explained by noting that, for sufficiently large values of α, *GLPT* will assign most of the receivers to their previous channels. Only a few of the receivers with the smallest requirements will be assigned to new channels if it is necessary to do so in order to keep the channels balanced. This feature of *GLPT*, namely, that the receivers with the smallest requirements under the new traffic pattern are more likely to be retuned, is highly desirable. This is because it implies that the reconfiguration will affect the part of the network that is least utilized, minimizing the impact of the transition phase (in terms of packet loss, delay, etc.) on the overall traffic carried by the network.

In Fig. 6 we plot the number of retunings required against the number of channels, for $N = 120$. We note that the first approach always requires a number of receivers to be retuned which is very close to the upper bound $N - C$ of Lemma 3.1. On the other hand, for the range of *N* and *C* values considered, the number of retunings required by *GLPT* increases almost linearly with *C* for all values of α; also, larger values of α result in a smaller number of retunings, as expected. This result, combined with our previous observations, indicates that, for certain values of

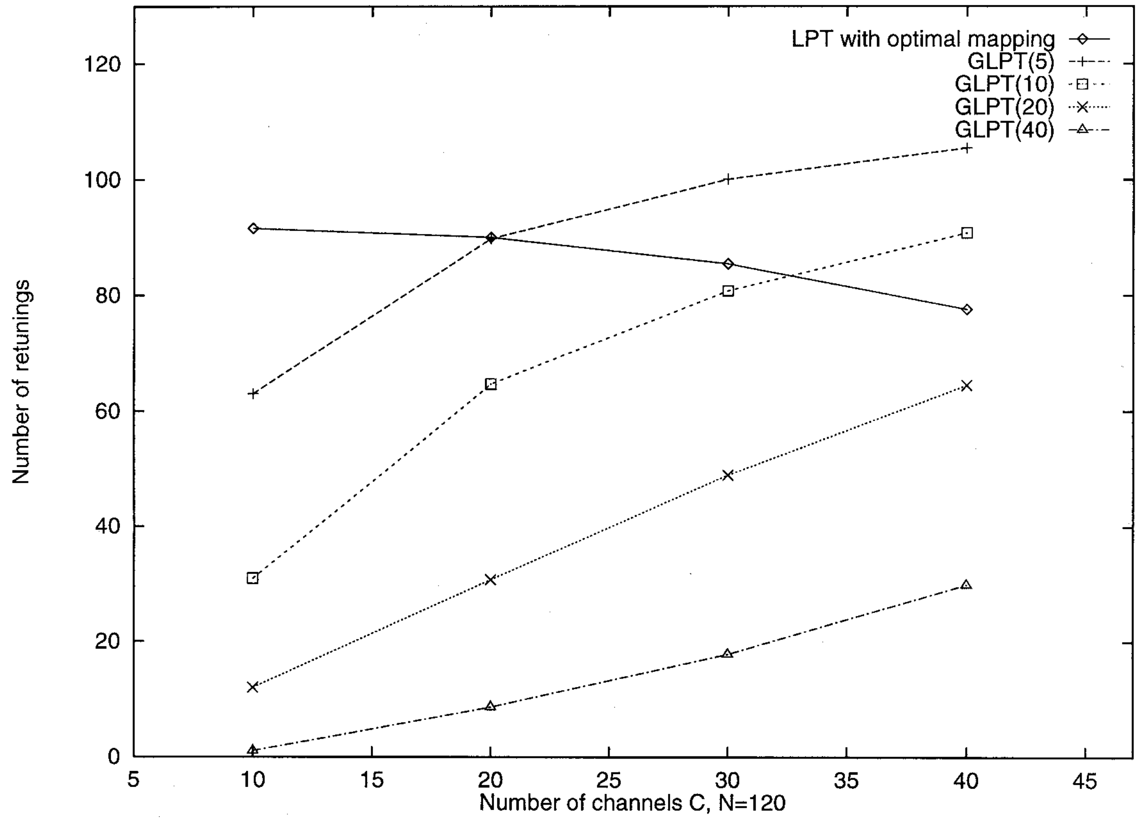


Fig. 6. Algorithm comparison on number of retunings ($N = 120$ nodes, random traffic matrices).

parameter α (in this case, for $20 \leq \alpha \leq 40$), *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 Figs. 3–6 were obtained by randomly selecting the initial traffic matrix **T**, and then randomly selecting the target matrix **T**′, independently of **T**. In practice, the new traffic matrix **T**′ will be related to the old matrix **T**, differing only by the changes in the traffic demands that have taken place in the time interval between successive reconfiguration instants. 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 **T** was constructed as before. This matrix was then evolved through a series of steps to obtain the target matrix **T**′. The Brownian motion was modeled by using two asymmetric probabilities: the probability of the particle 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 Figs. 3–6, 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 **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). 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

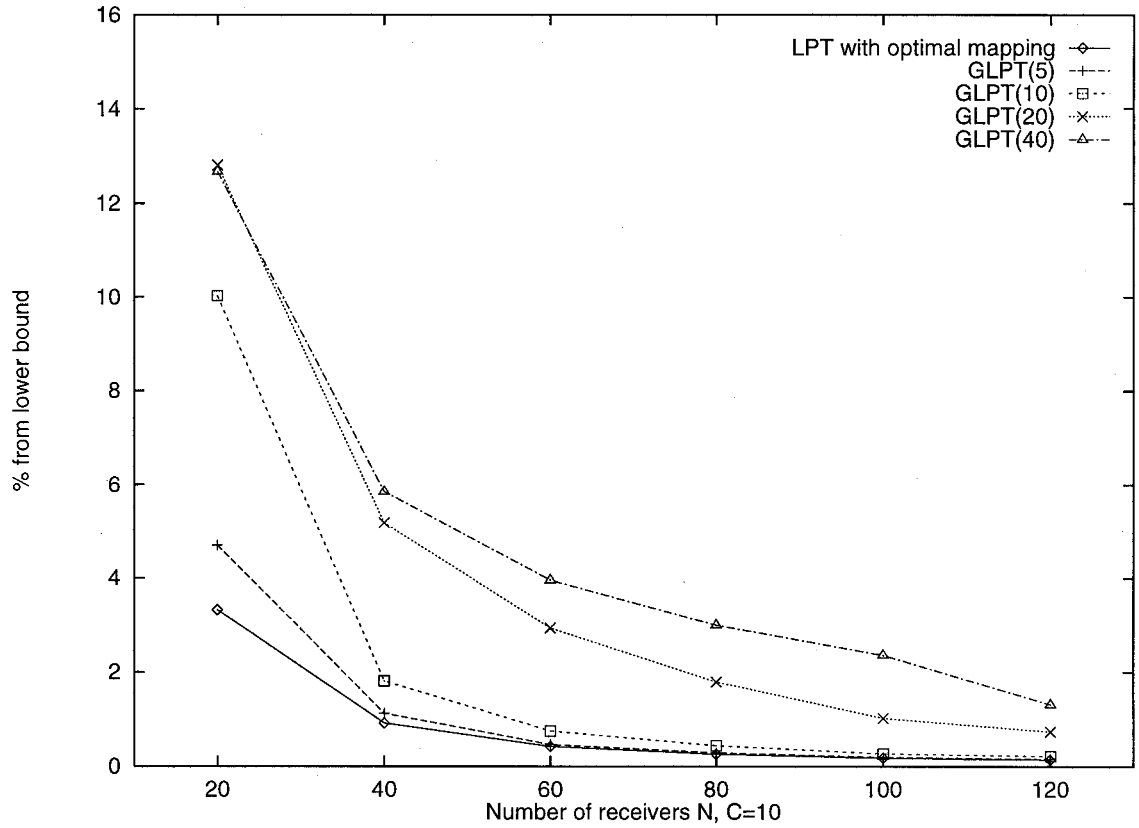


Fig. 7. Algorithm comparison on load balancing ($C = 10$ channels, Brownian model).

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 Figs. 7–10. 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 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.

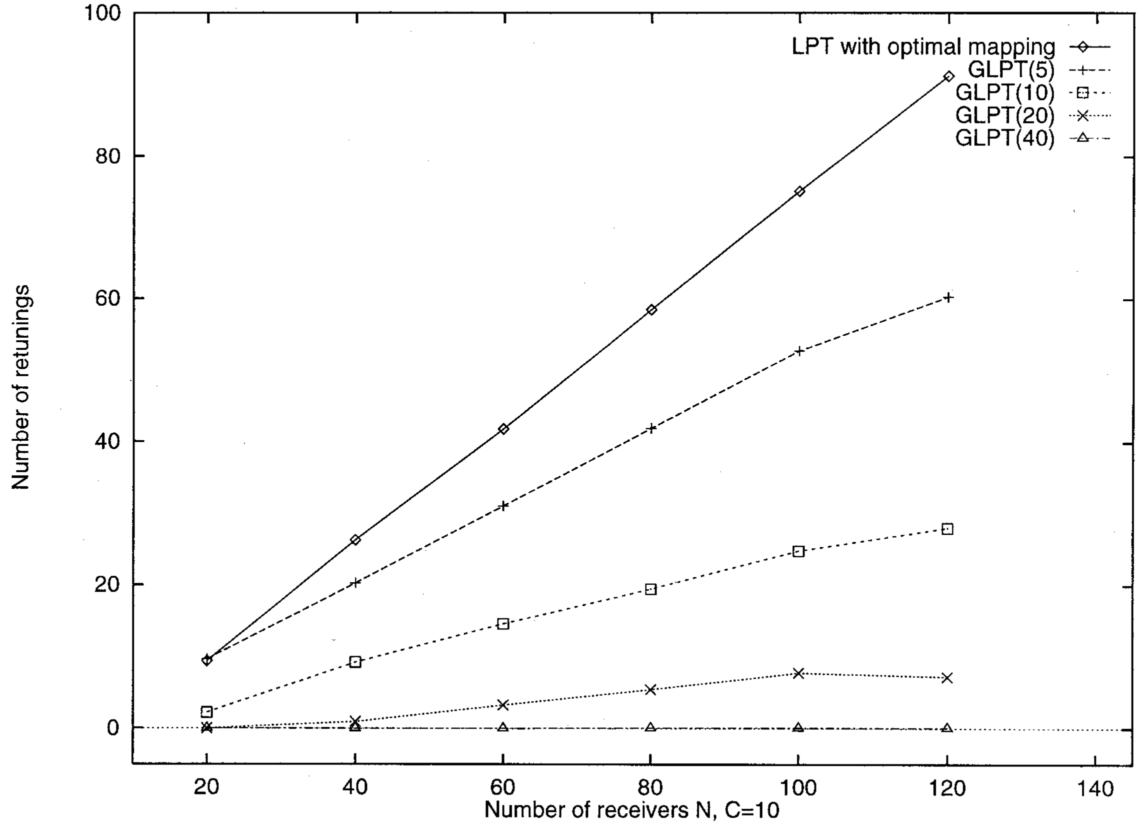


Fig. 8. Algorithm comparison on number of retunings ($C = 10$ channels, Brownian model).

Appendix

Proof of Lemma 3.1: *We will first prove that no more than $N - C$ retunings are needed under an optimal solution to the CA problem. We will then show that this is a tight bound by constructing instances of the CA problem that require a number of retunings equal to the upper bound.*

Consider a network with N nodes and $C \leq N$ channels. Let m be an integer such that, for any arbitrary instance $(\mathscr{R}(N), \mathscr{S}'(N))$ of the CA problem, there will be at least m (out of N) receivers that do not need to be retuned under the optimal solution (the reason why we express R and $\mathscr{S}'$ as functions of the number of nodes will become apparent shortly). In other words, if $\mathscr{R}'(N)$ is the optimal new wavelength assignment for instance $(\mathscr{R}(N), \mathscr{S}'(N))$, we have that:

$$\sum_{c=1}^{C} |R_c(N) \cap R'_c(N)| \geq m. \tag{8}$$

Now consider a network with $N' > N$ nodes and C wavelengths. We show by contradiction that, if $(\mathscr{R}(N'), \mathscr{S}'(N'))$ is an arbitrary instance of the CA problem for this network, and $\mathscr{R}'(N')$ is the optimal new wavelength assignment, then we also have:

$$\sum_{c=1}^{C} |R_c(N') \cap R'_c(N')| \geq m' = m, \quad N' > N. \tag{9}$$

Indeed, suppose that $m' < m$, and consider an instance of the CA problem for this network for which the left part of (9) holds with equality. Then, by removing from this instance $N' - N$ receivers that need to be

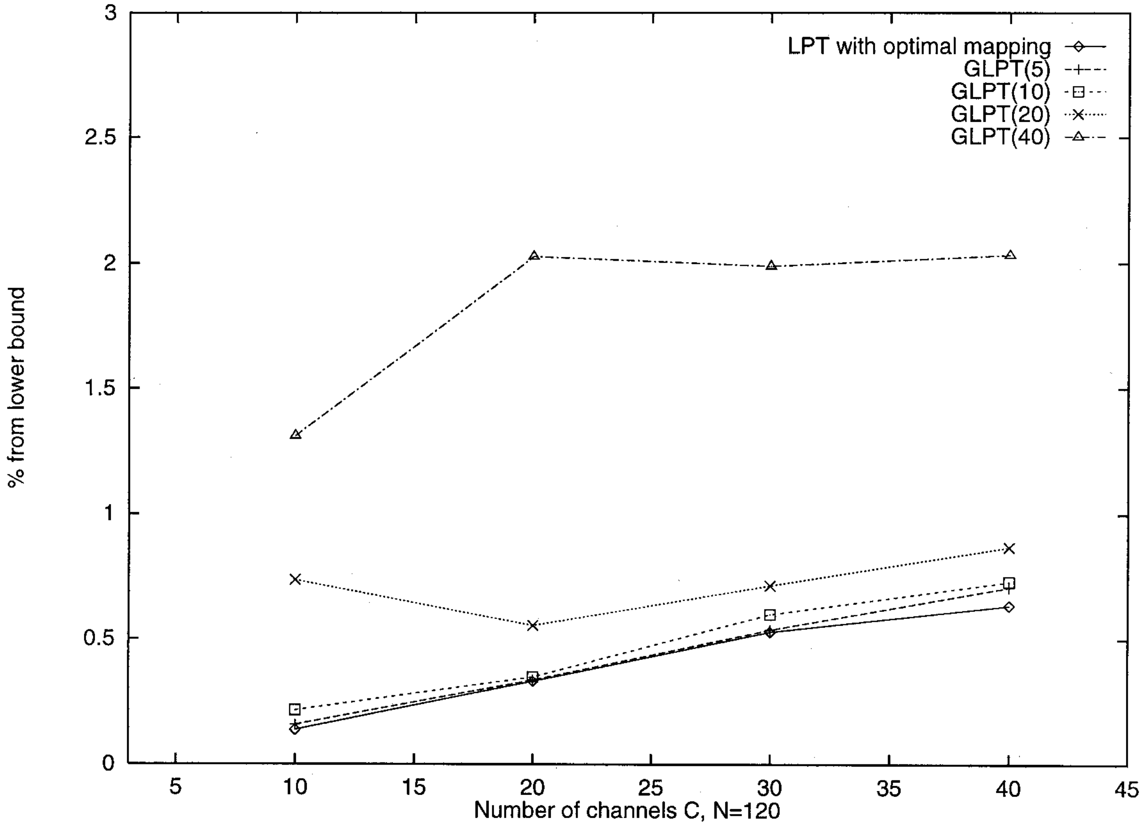


Fig. 9. Algorithm comparison on load balancing ($N = 120$ nodes, Brownian model).

retuned, we obtain an instance of the CA problem for a network with N nodes such that

$$\sum_{c=1}^{C} |R_c(N) \cap R'_c(N)| = m' < m. \qquad (10)$$

But, because of our hypothesis that (8) holds, (10) is impossible. Therefore, (9) must necessarily hold. The result in (5) now follows from (9) and the fact that, when $C = N$, each channel is assigned exactly one receiver, and, under optimal channel assignment, no receiver needs to be retuned (i.e., when $N = C, m = C$ in (8)).

A trivial instance for which the upper bound is achieved is for a network with $N = C + 1$ nodes where (a) in the initial assignment all receivers are assigned a unique channel, except i and j who share the same channel, and (b) in the new partition, i is in a subset by itself and j moves to a subset with, say, receiver k. Then, under optimal channel assignment, exactly $N - C = 1$ receiver must be retuned, receiver j, from its original channel to the channel of k. However, even for large N, the number of retunings may be very close to the upper bound $N - C$. Specifically, we now construct an instance of CA that requires exactly $N - C - 1$ retunings. Consider a network with $N = C^2$, and an initial wavelength assignment given by:

$$R_c = \{(c-1)C + 1, \ldots, cC\} \quad c = 1, \ldots, C. \qquad (11)$$

The new partition $\mathcal{S}'$ is:

$$\mathcal{S}'_c = \begin{cases} \{c\}, & c = 2, \ldots, C \\ \{1, C+1, \ldots, C^2\}, & c = 1. \end{cases} \qquad (12)$$

It is straightforward to verify that (a) a permutation is optimal if it assigns S'_1 to any of channels λ_2 through

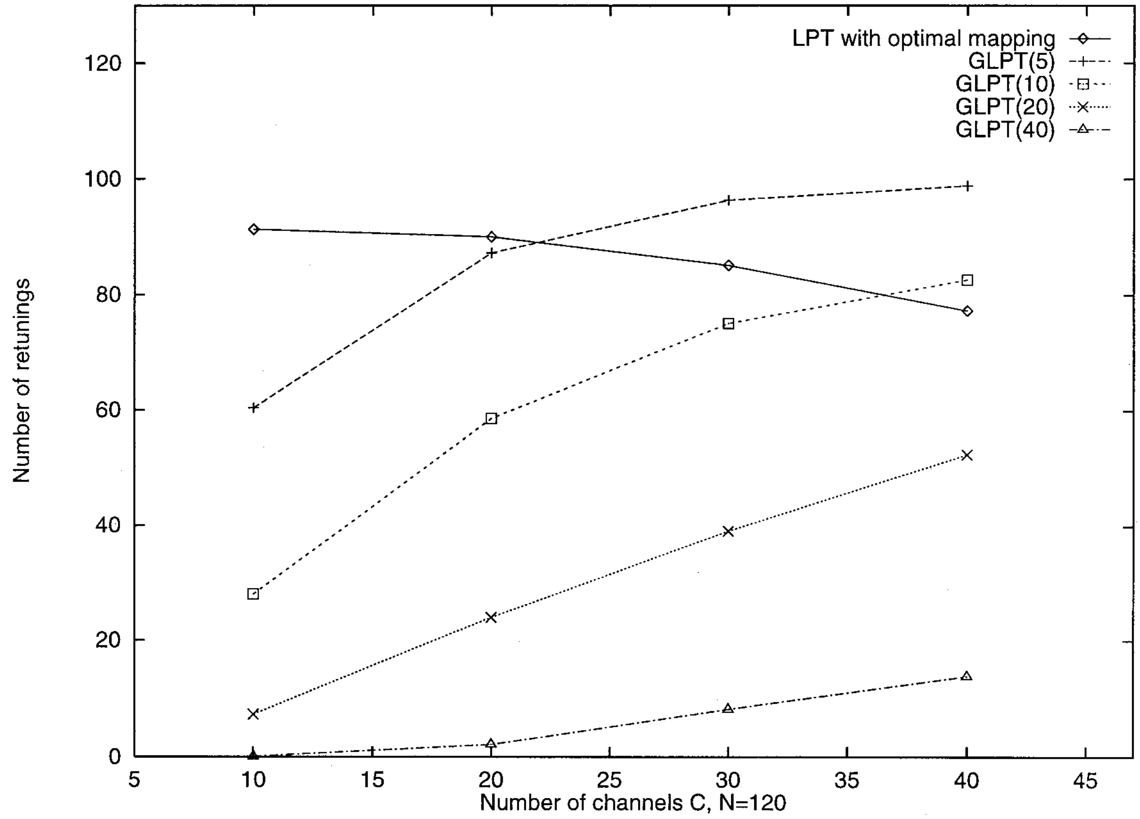


Fig. 10. Algorithm comparison on number of retunings ($N = 120$ nodes, Brownian model).

λ_C, and that (b) exactly $C^2 - C - 1 = N - C - 1$ retunings are required under an optimal permutation. □

References

[1] B. Mukherjee, WDM-Based local lightwave networks Part I: Single-hop systems, IEEE Network Magazine, (May 1992), pp. 12–27.

[2] R. Ramaswami, Multiwavelength lightwave networks for computer communication, IEEE Communications Magazine, (February 1993), pp. 78–88.

[3] P. E. Green, Fiber Optic Networks. Prentice-Hall, Englewood Cliffs, (New Jersey, 1993).

[4] V. Sivaraman, G. N. Rouskas, HiPeR-*l*: A High Performance Reservation protocol with *l*ook-ahead for broadcast WDM networks. In Proceedings of INFOCOM '97, IEEE, (April 1997), pp. 1272–1279.

[5] K. Sivalingam, P. Dowd, A multi-level WDM access protocol for an opticall interconnected multi-processor system, IEEE/OSA Journal of Lightwave Technology, vol. 13, no. 11, (November 1995), pp. 2152–2167.

[6] Mon-Song Chen, N. R. Dono, R. Ramaswami, A media-access protocol for packet-switched wavelength division multiaccess metropolitan area networks, IEEE Journal on Selected Areas in Communications, vol. 8, no. 6, (August 1990), pp. 1048–1057.

[7] E. Hall, et al., The Rainbow-II gigabit optical network, IEEE Journal Selected Areas in Communications, vol. 14, no. 5, (June 1996), pp. 814–823.

[8] G. N. Rouskas, M. H. Ammar, Dynamic reconfiguration in multihop WDM networks, Journal of High Speed Networks, vol. 4, no. 3, (1995), pp. 221–238.

[9] J-F. P. Labourdette, F. W. Hart, A. S. Acampora, Branch-exchange sequences for reconfiguration of lightwave networks, IEEE Transactions on Communications, vol. 42, no. 10, (October 1994), pp. 2822–2832.

[10] I. Baldine, G. N. Rouskas, On the design of dynamic reconfiguration policies for broadcast WDM networks. In Proceedings of SPIE '98, (November 1998). (To appear.)

[11] M. R. Garey, R. L. Graham, D. S. Johnson, Performance guarantees for scheduling algorithms, Operations Research, vol. 26, (January 1978), pp. 3–21.

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

[13] G. N. Rouskas, V. Sivaraman, Packet scheduling in broadcast WDM networks with arbitrary transceiver tuning latencies, IEEE/ACM Transactions on Networking, vol. 5, no. 3, (June 1997), pp. 359–370.

[14] M. Azizoglu, R. A. Barry, A. Mokhtar, Impact of tuning delay on the performance of bandwidth-limited optical broadcast networks with uniform traffic, IEEE Journal on Selected Areas in Communications, vol. 14, no. 5, (June 1996), pp. 935–944.

[15] M. S. Borella, B. Mukherjee, Efficient scheduling of nonuniform packet traffic in a WDM/TDM local lightwave network with arbitrary transceiver tuning latencies, IEEE Journal on Selected Areas in Communications, vol. 14, no. 5, (June 1996), pp. 923–934.

[16] G. R. Pieris, G. H. Sasaki, Scheduling transmissions in WDM broadcast-and-select networks, IEEE/ACM Transactions on Networking, vol. 2, no. 2, (April 1994), pp. 105–110.

[17] H. G. Perros, K. M. Elsayed, Call admission control schemes: A review, IEEE Communications Magazine, vol. 34, no. 11, (1996), pp. 82–91.

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

[19] E. Coffman, M. R. Garey, D. S. Johnson, An application of bin-packing to multiprocessor scheduling, SIAM Journal of Computing, vol. 7, (February 1978), pp. 1–17.

[20] R. L. Graham, Bounds on multiprocessing timing anomalies. SIAM Journal of Applied Mathematics, vol. 17, no. 2, (March 1969), pp. 416–429.

[21] I. Baldine, G. N. Rouskas, Dynamic reconfiguration policies for WDM networks. In Proceedings of INFOCOM '99, IEEE, (March 1999). (To appear).

[22] R. K. Ahuja, T. L. Magnanti, J. B. Orlin, Network Flows: Theory, Algorithms, and Applications (Prentice Hall, Englewood Cliffs, N.J., 1993).

Ilia Baldine was born in Dubna, Moscow Region, Russia in 1972. He received the B.S. degree in Computer Science from the Illinois Institute of Technology, Chicago, in 1993, and the M.S. and Ph.D. degrees in Computer Science from the North Carolina State University in 1995 and 1998, respectively. He is currently a research scientist with MCNC, RTP, USA.

His research interests include all-optical networks, ATM networks, network protocols and security.

George N. Rouskas received the Diploma in Computer Engineering from the National Technical University of Athens (NTUA), Athens, Greece, in 1989, and the M.S. and Ph.D. degrees in Computer Science from the College of Computing, Georgia Institute of Technology, Atlanta, GA, in 1991 and 1994, respectively. He joined the Department of Computer Science, North Carolina State University in August 1994, as an Assistant Professor. His research interests include high-speed and light-wave network architectures, multipoint-to-multipoint communication, and performance evaluation.

He is a recipient of a 1997 NSF Faculty Early Career Development (CAREER) Award. He also received the 1995 *Outstanding New Teacher* Award from the Department of Computer Science, North Carolina State University, and the 1994 *Graduate Research Assistant* Award from the College of Computing, Georgia Tech. He is a member of the IEEE, the ACM and of the Technical Chamber of Greece.