University of Windsor
Scholarship at UWindsor

Electronic Theses and Dissertations | Theses, Dissertations, and Major Papers

2013

Resource Allocation for Periodic Traffic Demands in WDM Networks

Ying Chen
University of Windsor

Follow this and additional works at: https://scholar.uwindsor.ca/etd

Part of the Computer Sciences Commons

Recommended Citation

Chen, Ying, "Resource Allocation for Periodic Traffic Demands in WDM Networks" (2013). *Electronic Theses and Dissertations*. 4918.
https://scholar.uwindsor.ca/etd/4918

This online database contains the full-text of PhD dissertations and Masters' theses of University of Windsor students from 1954 forward. These documents are made available for personal study and research purposes only, in accordance with the Canadian Copyright Act and the Creative Commons license—CC BY-NC-ND (Attribution, Non-Commercial, No Derivative Works). Under this license, works must always be attributed to the copyright holder (original author), cannot be used for any commercial purposes, and may not be altered. Any other use would require the permission of the copyright holder. Students may inquire about withdrawing their dissertation and/or thesis from this database. For additional inquiries, please contact the repository administrator via email (scholarship@uwindsor.ca) or by telephone at 519-253-3000ext. 3208.

Resource Allocation for Periodic Traffic Demands in WDM Networks

By

Ying Chen

A Dissertation
Submitted to the Faculty of Graduate Studies
through the School of Computer Science
in Partial Fulfillment of the Requirements for
the Degree of Doctor of Philosophy at the
University of Windsor

Windsor, Ontario, Canada

2013

© 2013 Ying Chen

Resource Allocation for Periodic Traffic Demands in WDM Networks

By

Ying Chen

APPROVED BY:

Vinod Vokkarane, External Examiner
University of Massachusetts Dartmouth

Yunbi An
Odette School of Business

Jianguo Lu
School of Computer Science

Dan Wu
School of Computer Science

Arunita Jaekel, Advisor
School of Computer Science

May 14, 2013

Declaration of Co-Authorship / Previous Publication

I. Co-Authorship Declaration

I hereby declare that this thesis incorporates material that is a result of joint research, as follows:

This thesis incorporates the results of a joint research undertaken in collaboration with Dr. Ataul Bari under the supervision of Professor Arunita Jaekel. The collaboration is covered in Chapters 1-3 of the thesis. In all cases, the key ideas, primary contributions, experimental designs, data analysis and interpretation, were performed by myself working under my supervisor. The contribution of Dr. Ataul Bari was in data analysis and the literature review.

I am aware of the University of Windsor Senate Policy on Authorship and I certify that I have properly acknowledged the contribution of other researchers to my thesis, and have obtained written permission from each of the co-author(s) to include the above material(s) in my thesis.

I certify that, with the above qualification, this thesis, and the research to which it refers, is the product of my own work.

II. Declaration of Previous Publication

This thesis includes three original papers that have been previously published in peer reviewed journals as follows:

Thesis Chapter	Publication title/full citation	Publication status
Chapters 1, 2, 3	A. Jaekel, Y. Chen and A. Bari, Stable Logical Topologies for Survivable Traffic Grooming of Scheduled Demands, *Journal of Optical Communication and Networking*, 2 (10) (2010) 793-802.	published
Chapters 1, 2, 4	Y. Chen and A. Jaekel, Energy Aware Scheduling and Resource Allocation for Periodic Traffic Demands, *Journal of Optical Communications and Networking*, 5 (4) (2013) 261 - 270.	published
Chapters 1, 2, 5	Y. Chen, A. Jaekel, and A. Bari, A New Model for Allocating Resources to Scheduled Lightpath Demands, *Computer Networks*, 55 (13) (2011) 2821-2837.	published

I certify that I have obtained a written permission from the copyright owner(s) to include the above published material(s) in my thesis. I certify that the above material describes work completed during my registration as graduate student at the University of Windsor.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

Abstract

Recent research has clearly established that holding-time-aware routing and wavelength assignment (RWA) schemes lead to significant improvements in resource utilization for scheduled traffic. By exploiting the knowledge of the demand holding times, this thesis proposes new traffic grooming techniques to achieve more efficient resource utilization with the goal of minimizing resources such as bandwidth, wavelength channels, transceivers, and energy consumption. This thesis also introduces a new model, the segmented sliding window model, where a demand may be decomposed into two or more components and each component can be sent separately. This technique is suitable for applications where continuous data transmission is not strictly required such as large file transfers for grid computing. Integer linear program (ILP) formulations and an efficient heuristic are put forward for resource allocation under the proposed segmented sliding window model. It is shown that the proposed model can lead to significantly higher throughput, even over existing holding-time-aware models.

Dedication

To my parents, Wenzheng, GuiQin, my husband Ziyi and my daughter Christina.

Acknowledgements

I would like to express my deep appreciation to my supervisor, Dr. Arunita Jaekel. This work could not have been achieved without her guidance, generous support, and continuing encouragement.

I would like to thank Dr. Vinod Vokkarane for kindly serving as the External Examiner of my thesis defence.

I would like to thank Dr. Yubi An, Dr. Jianguo Lu, and Dr. Dan Wu for their valuable time and constructive comments.

My special thanks to my husband and my daughter for their endless love, care and understanding. They are always the source of support and encouragement.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Optical Networks

Wavelength division multiplexing (WDM) optical networks are widely used for high capacity backbone networks due to their ability to carry large volumes of data with a high degree of reliability and at a relatively low cost [1]. In such networks, the *end-to-end* optical communication channels called *lightpaths* [2] can be overlaid on top of a physical fiber network. A lightpath connects a transmitter at a source end-node to a receiver at a destination end-node with no opto-electronic conversion at any intermediate nodes. The set of lightpaths, established over the *physical* fiber links, constitute the edges in a *virtual* (or logical) topology. A logical edge, $e_i \rightarrow e_j$, is a directed link in the logical topology if there is a lightpath from end-node e_i to end-node e_j.

Figure 1.1 shows the physical topology of a small-size optical network with four end-nodes and four router nodes represented by circles and rectangles, respectively. Router nodes receive signal from a source node or other router nodes and forward them to the destination node or next router node in a route. Here, solid lines represent fiber links and directed dashed lines represent lightpaths established over the physical topology. For example, lightpath L_1 can be used to send data from end-node E_1 to E_3. It starts from source node E_1, passes through router nodes R_1, R_2, R_3, and finally reaches the destination node E_3. Figure 1.2 shows the logical topology corresponding to the lightpaths shown in Fig. 1.1. For example, logical edge *1 → 3* represents lightpath L_1.

Each lightpath must be assigned a route over the physical network and a specific channel on each fiber link it traverses. This is the standard *routing and wavelength assignment* (RWA) [3] problem. In wavelength convertible networks, a lightpath can be assigned a different channel on each fiber it traverses. However, full range all-optical wavelength conversion is generally not feasible due to both cost and technological restrictions. Therefore, most practical networks do not assume wavelength conversion capabilities. In the absence of wavelength converters, a lightpath must be assigned the same channel on all links in the route. This is called the *wavelength continuity constraint*. In Fig. 1.2, lightpath L_1 is routed over fiber links 1→0 and 0→3, and assigned channel 2 for data transmission.

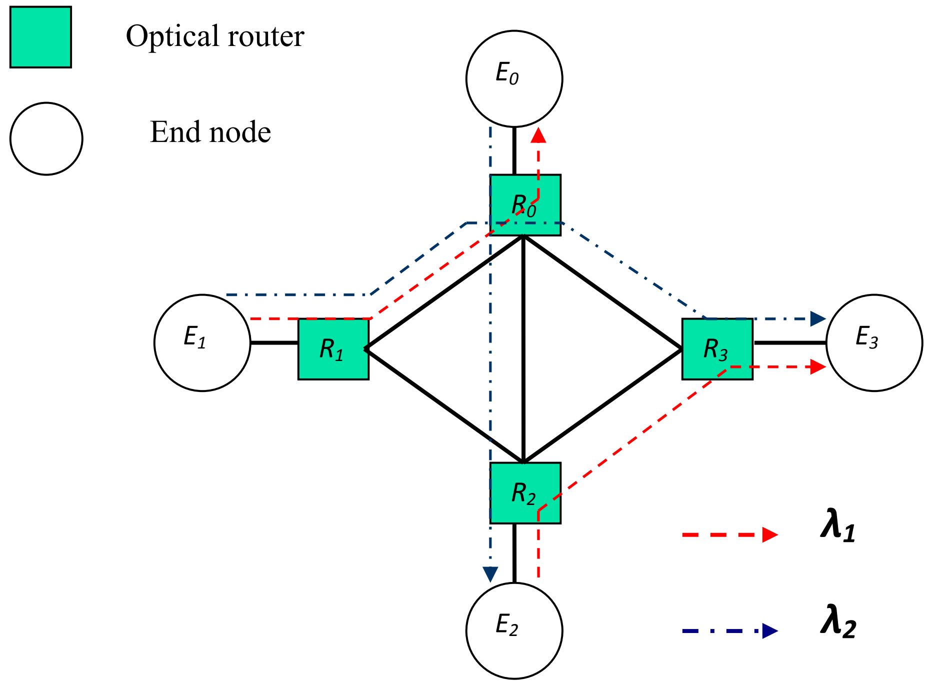

Figure 1.1: Some lightpaths on a physical topology.

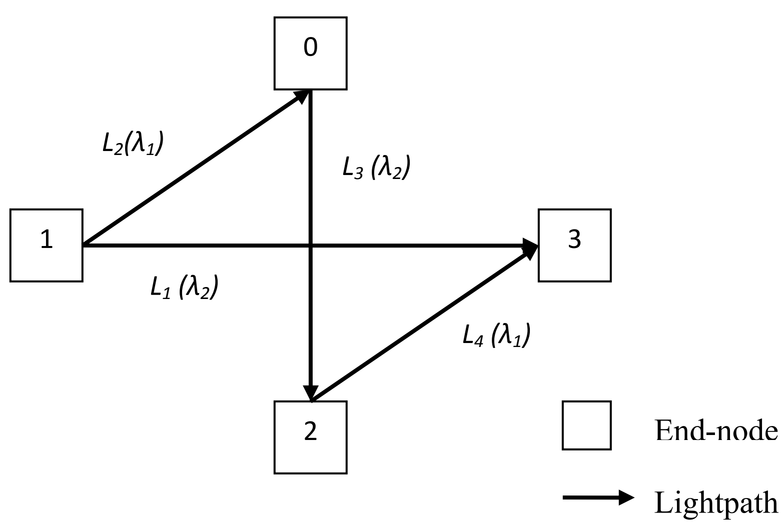

Figure 1.2: Logical topology corresponding to the lightpaths shown in Fig. 1.1.

Compared to the huge bandwidth of a lightpath (2.5 Gbps to over 10 Gbps), individual requests for connections are typically for data streams at a much lower data communication rate, of the order of mega-bits per second (Mbps). This tremendous mismatch between the capacity of individual lightpaths and the bandwidth requirements of individual traffic demands has led to the emergence of *traffic grooming* techniques. Traffic grooming techniques in WDM networks can be defined as a family of techniques for combining a number of low-speed data streams from users so that the high capacity of each lightpath may be used as efficiently as possible [4]. The complete logical topology design and traffic grooming problem has the following subproblems that have to be solved simultaneously for optimizing the network cost or the resource utilization [5]: i) topology design (TD): find the logical topology of the network, ii) route and wavelength assignment (RWA): ensure that a feasible RWA, to realize each logical edge, is possible, and iii) traffic routing (TR): decide which logical path(s) should be used for each data stream so that the total payload of each edge in the logical topology never exceeds the capacity of a lightpath.

1.2 Traffic Models

In general, traffic models can be classified into three categories including static [3], dynamic [6] – [8], and scheduled traffic models [9, 10]. In the *static* traffic model, the set of lightpaths to be established is known in advance and relatively stable over long periods of time. A single traffic matrix can be formed as an input for offline planning problems under this model. The *dynamic* traffic model considers online provisioning where the arrival time and duration of demands are randomly generated based on a specified distribution. A number of algorithms for dynamic allocation are available in the literature [6] - [8], and typically focus on decreasing the demand blocking probability. The *scheduled traffic model* is suitable for applications that require periodic use of bandwidth (e.g. on a daily or weekly basis) at predefined times such as database backup or on-line class. In this model, setup and teardown times of the demands may be known exactly, or may vary within a specific range, so resource allocation can be optimized in both space and time.

In the *fixed-window* scheduled traffic model [9], the setup and teardown times of each scheduled lightpath demand are predetermined and known in advance. An individual demand can be represented by a tuple (s, d, b, st, et), where s and d are the source and destination, b represents the requested bandwidth for the demand, and st, et are the setup and teardown times of the demand, respectively. The *sliding-window* scheduled traffic model [10] adds another degree of flexibility by allowing demands to be reserved within a larger time window. In this case, the demand setup and teardown times (st and et) are no longer fixed, but can vary within a specific range. So, each demand is represented as ($s, d, b, \alpha, \omega, \tau$), where s, d and b have the same meaning as before, α, ω are the start and end times of the larger window during which the demand is to be scheduled, and τ is the demand holding time.

1.3 Problem Statement and Solution Outline

By exploiting the knowledge of the demand holding times, this thesis proposes new traffic grooming techniques to achieve more efficient resource utilization with the goal of minimizing resources such as bandwidth, wavelength channels, transceivers, and energy consumption. This thesis also introduces a new model, the *segmented sliding window model* [11], where a scheduled demand may be decomposed into two or more components and each component can be sent separately. It adds another degree of flexibility to the existing sliding window model, which can lead to significantly higher throughput, even over existing *holding-time-aware* (HTA) models.

1.3.1 Virtual Topology Design under the Scheduled Traffic Model

In this thesis, we consider the *scheduled* traffic model, where there is a specified set of periodic sub-wavelength demands to be routed over the network. Unlike the static model, the demands are not permanent, but have a specified duration. However, since the demands are periodic and predictable, resource allocation decisions can be made for the entire demand set in an integrated manner to improve network performance. For such periodic traffic, one option that has been suggested in the literature is to establish lightpaths as needed. In this model, low-speed traffic demands can be routed over already established lightpaths, or additional new lightpath(s) may be created to accommodate a demand. As the demands change over time, a lightpath that is no longer needed can be taken down and its resources allocated elsewhere. In essence, this approach uses a *sequence* of logical topologies, which change over time based on the offered traffic pattern. This offers a degree of flexibility in terms of the re-use of transceivers at the end-nodes and wavelength links on fibers, which can lead to some potential savings in terms of these resources. However, this comes at the cost of requiring the use of more expensive reconfigurable switching equipment at the network nodes. Skorin-Kapov et al. [12] show that even though the use of reconfigurable switches might lead to a slight reduction in the number of transceivers, it does not appear to be cost-effective when the increased cost of the switching equipment is taken into consideration. A more cost-effective alternative is to design a stable or *static* virtual topology that does not change over time, but is capable of handling all traffic variations over time [13]. This approach has a number of important advantages as given below:

- it can be realized using less costly non-reconfigurable equipment,
- a stable topology avoids short (but potentially significant) disruptions to ongoing traffic, during topology changes, and
- it does not incur increased signaling complexity due to dynamic setup and teardown of lightpaths.

In this thesis we focus on the second alternative of designing a *static* virtual topology that can handle periodic, multi-hour traffic demands. In this context, it is extremely important that the final static logical topology is implemented as efficiently as possible, so it can accommodate all traffic demands with a minimum amount of resources. We address the complete problem of survivable traffic grooming under the scheduled traffic model in WDM networks and propose three integer linear program (ILP) formulations to solve this problem optimally [14]. It is well-known the complexity of an ILP is, in general, exponential and heavily dependent on the number of integer variables. As a result, most of the existing ILP formulations for traffic grooming and topology design can only handle very small networks with a small number of traffic demands. However, one of the important features of our proposed ILP formulations is that we employ novel techniques to develop efficient formulations that can generate optimal solutions to practical sized problems. For example, we have used special constraints that allow continuous variables to replace integer variables in certain cases. This reduces the number of integer variables and, hence, the overall complexity of the formulation. Furthermore, to the best of our knowledge this is the first approach that jointly considers survivable network design, traffic grooming and RWA problem for scheduled sub-wavelength demands.

1.3.2 Energy Efficient Design for Optical Networks

The tremendous growth in high-bandwidth applications and devices used in backbone networks has led to a corresponding increase in power consumption. In fact, a number of researchers predict that “energy consumption rather than the cost of the component equipment may eventually become the barrier to continued growth” [15].

Therefore, it is becoming increasingly important to design networks that consider effective use of the available power in addition to other resources such as bandwidth, wavelength channels, transceivers *etc*. For core networks, it has been established that optical processing is cheaper than electronic processing in terms of power consumption [16]. Therefore, WDM networks can play an important role in reducing energy consumption of core network nodes by allowing traffic to optically bypass electronic components whenever possible. Even considering its lower power cost (compared to electronic processing) it is important to develop robust energy-efficient design strategies for optical networks in order to meet the expected growth in demand and rising energy costs in the near future [15]. Many different approaches have been considered for reducing power consumption in WDM networks including switching off line cards [16, 17], or even entire links or nodes [18, 19].

The energy consumption for a group of demands routed over a given logical topology (a set of lightpaths) can be estimated using the model given in [20]. In this model, the total power consumption is expressed as:

$$P_{total} = \sum_{l \in L} (P_{static} + P_{dynamic} \cdot t_l) \tag{1.1}$$

where L is the set of active lightpaths, P_{static} corresponds to the static power consumption for a lightpath $l \in L$, $P_{dynamic}$ is the additional power needed for each traffic unit carried on a lightpath l, and t_l is the traffic on l. Static power consumption is reduced by routing sub-wavelength traffic demands in a way that minimizes the number of active router ports (corresponding to active lightpaths carrying some non-zero traffic). Dynamic power consumption is reduced by minimizing the amount of electronic switching required for each demand. It has been well established that the static consumption is the dominant component of the total power in Eq. (1.1) [20]. Therefore, power consumption can be greatly reduced by turning off the transponders/line cards corresponding to a lightpath, when it is not carrying any traffic.

A number of recent papers [19] – [22] have considered route and wavelength assignment (RWA) and traffic grooming with the goal of reducing energy consumption. Our approach differs from all these existing schemes, in that we consider applications with periodic bandwidth demands and show how consideration of demand holding times can play an important part in reducing the overall energy consumption of a network.

In this thesis, we address the problem of energy efficient routing and scheduling of periodic demands, which to our knowledge has not been considered previously, and propose a new technique for energy aware resource allocation of such demands [23]. We first present two new ILP formulations that can optimally route a single new demand under *fixed* and *sliding* window models in such a way as to minimize the incremental energy consumption due to the new demand, based on its holding time and bandwidth requirement. We then extend this concept and propose a simple and efficient heuristic algorithm that jointly allocates resources to a set of scheduled demands in such a way that the total static and dynamic power consumption, for the entire demand set, is reduced as much as possible.

1.3.3 Advantages of the Segmented Scheduled Traffic Model

For both the traditional fixed and sliding window models, once transmission of a demand starts, it continues uninterrupted until all data have been transmitted. We will refer to this model as the *continuous* scheduled traffic model. Such a model can be appropriate for applications such as a daily "virtual classroom", where bandwidth is required continuously for several hours as long as the class is "in session". However, there are many applications where such continuous data transmission is not strictly required. These include some file transfers for grid computing or an application where a bank has to transfer its data nightly to a central location, where the actual data transfer requires one scheduled lightpath for one hour and must be completed some time between 1am and 4am. In these cases it is not necessary to send the data continuously; instead the data may be divided into several smaller components and each component sent separately as long as the entire data are transferred within the specified time window between 1am and 4am. We will refer to this type of data transmission model as the *segmented* scheduled traffic model.

In this thesis, we propose a new traffic model called the *segmented scheduled traffic model*, which can lead to more efficient network design [11]. The advantages of the segmented scheduled traffic model are illustrated by the example in Fig. 1.3. We consider a single fiber link and for simplicity, we assume that the link can accommodate only one WDM channel. We also consider two demands q_1 and q_2, where q_1 (q_2) requires the entire WDM channel for 2 hours (3 hours), within time window 1 - 4 (0 -5). Clearly, under the *continuous* sliding window model, it will not be possible to accommodate both of these demands. However, as shown in Fig. 1.3(b), both demands can be easily handled using the *segmented* scheduled traffic model by dividing q_2 into two components q_{2-1} and q_{2-2} respectively.

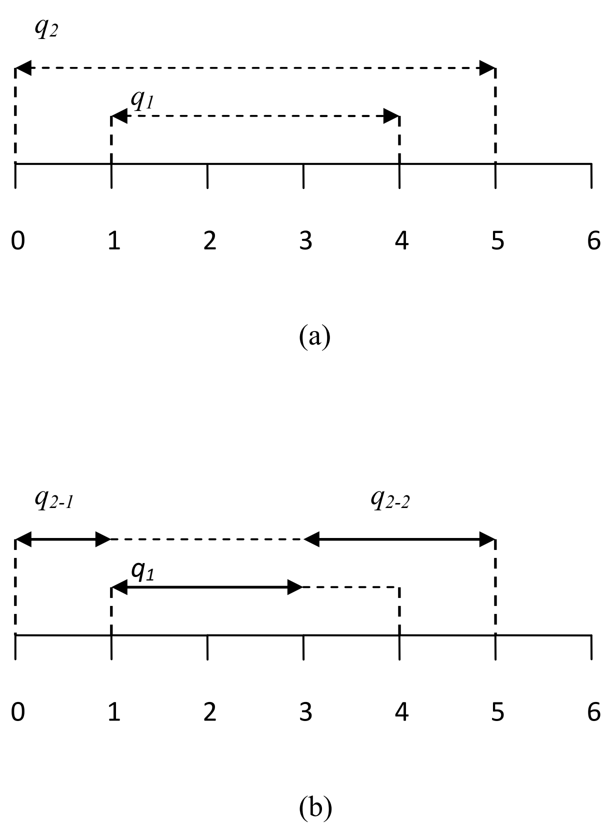

Figure 1.3: An example of demand scheduling under (a) the sliding window and (b) the segmented scheduled traffic model.

In addition to the usual RWA issues involved in scheduling lightpath demands, design strategies under the segmented scheduled traffic model also need to take into consideration a number of other important factors such as: i) which demands (if any) should be divided into segments, ii) the number and sizes of the segments for each demand, and iii) how to schedule the individual segments to optimize resource utilization. Therefore, resource allocation under this new model can be viewed as a complex optimization problem.

To the best of our knowledge, such a model for scheduling and allocating resource to static advance reservation requests has not been used previously. We present an ILP formulation for resource allocation under the proposed segmented sliding window model and show that both the fixed and the traditional sliding window models can be treated as a special case of our generalized segmented sliding window model. We consider networks both with and without wavelength converters, and include the option of path protection for each lightpath. We also present an efficient heuristic that can be used for larger networks with many scheduled demands.

1.4 Thesis Organization

The remainder of this thesis is organized as follows. Chapter 2 reviews the scheduled traffic model and traffic grooming techniques as well as energy aware design strategies in optical networks. Chapter 3 presents our ILP formulations for optimal topology design with traffic grooming under the fixed-window scheduled traffic model. Chapter 4 presents ILP formulations and a heuristic for power efficient grooming of both fixed and sliding demands. In Chapter 5, ILP formulations and an efficient heuristic are put forward for resource allocation under the proposed segmented sliding window model. Finally, we conclude the thesis with a summary of the original contributions and directions of the future work in Chapter 6.

Chapter 2

Review of Literature

2.1 Scheduled Traffic Models

There has recently been significant work in resource allocation under the scheduled traffic model [9, 10, 12], [24] - [35], and it has been shown that connection holding-time-aware approaches consistently outperform traditional RWA algorithms for scheduled lightpath demands [24]. Kuri et al. [9] appear to be the first to propose a scheduled traffic model to handle fixed scheduled lightpath demands. They present a branch and bound algorithm and a tabu search based algorithm to solve the routing problem in fault-free networks. A generalized graph coloring approach is used to solve the wavelength assignment problem separately. Skorin-Kapov [25] improves the Tabu search based routing algorithm proposed in [9]. Instead of relying on randomized neighbourhood search, the author develops a neighbourhood reduction technique to reduce the search space significantly. Wang et al. [26] present ILP formulations under the fixed window model for fault-tolerant WDM networks that allow full wavelength conversion capabilities at the network nodes. Heuristic solutions for the same problem have been presented in [26] – [29]. For large-size networks, Wang et al. [26] design a two-step optimization approach to handle the routing and wavelength assignment problems separately. For each demand, Eppstein's *k*-shortest path algorithm is used to pre-compute a set of routes as its working path candidates. For each of the working paths, the algorithm is further employed to identify a set of link-disjoint protection paths. The routing information serves as the input to the wavelength assignment step. Jaekel and Chen [30] propose two levels of service, where idle backup resources can be used to carry low-priority traffic under fault-free conditions. When a fault occurs, and resources for a backup path need to be reclaimed, any low-priority traffic on the affected channels is dropped. Skorin-Kapov et al. [12] propose *mixed integer linear program* (MILP) formulations and a tabu search heuristic approach to solve the problem of the logical topology design in transparent optical networks under the fixed scheduled traffic model with reconfigurable and non-reconfigurable equipment. The objective is to minimize the network cost in terms of the number of transceivers needed.

The fixed window traffic model can be extended so that the setup and teardown times are no longer fixed, but slide within a larger window [10, 31]. This is referred to as the *sliding* scheduled traffic model. Figure 2.1 shows that the sliding scheduled traffic model can lead to additional savings over the fixed window model by reducing demand overlap and increasing reuse of resources by time-disjoint demands. Similar to the scheduling example given in Fig. 1.3, we consider a single fiber link and assume that the link can accommodate only one WDM channel. Demand q_1 (q_2) that requires the entire WDM channel for 3 hours (4 hours) is to be routed on this link starting from 2am (3am). Under the fixed window scheduled traffic model, these two demands cannot be handled due to the overlap between 3am and 5am as shown in Fig. 2.1(a). On the other hand, under the sliding scheduled traffic model, if demands q_1 and q_2 are allowed to be routed within a time window between 1am and 6am and between 2am and 8am, respectively, as shown in Fig. 2.1(b), both demands can be accommodated by setting actual start time st_1 (st_2) of the demand q_1 (q_2) to 1am (4am).

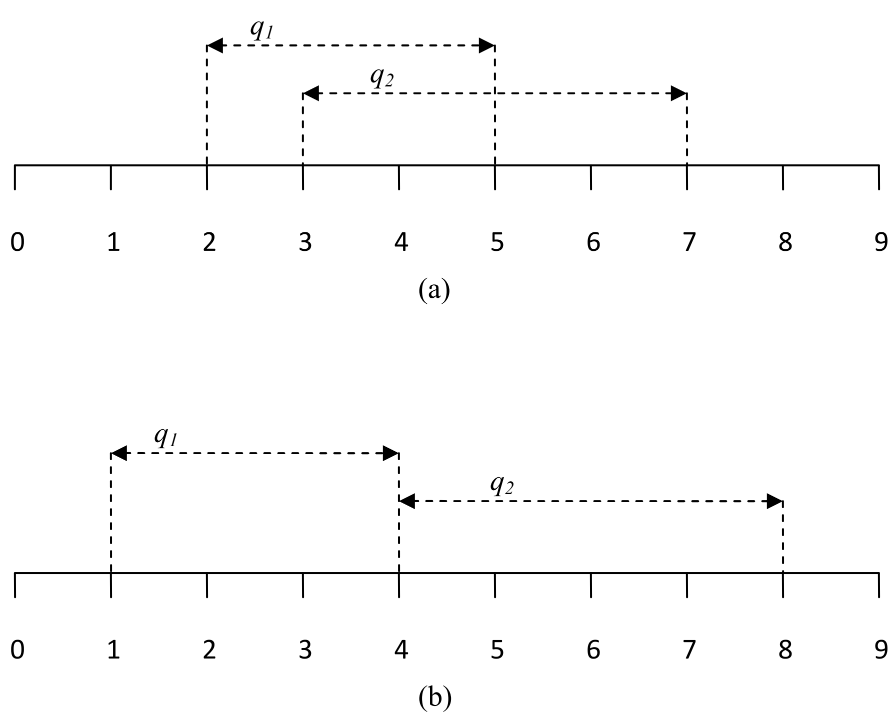

Figure 2.1: An example of demand scheduling under (a) the fixed window and (b) the sliding window scheduled traffic model.

The sliding scheduled traffic model provides us with more flexibility, but is also more complex. It simultaneously addresses scheduling demands in time to minimize demand overlap and allocating resources to lightpaths, and has typically been handled using heuristics in the literature [10, 31, 33]. Wang et al. [10] provide a heuristic algorithm for the demand scheduling and RWA problem with sliding demands in a fault-free network. The objective is to minimize the network resources in terms of the total number of wavelength-links needed for accommodating all scheduled lightpath demands. The authors concentrate on two sub-problems:

i) first, a demand time conflict reduction algorithm is used to schedule demands in such a way that the time overlapping among a set of demands is minimized,

ii) once this is completed, the problem is reduced to a fixed window scheduled traffic model. Two algorithms, a window based RWA algorithm and a traffic matrix based RWA algorithm, are used to solve the RWA problem separately.

Furthermore, the authors consider how to rearrange a blocked demand by setting a new start time with a minimal change to the current schedule.

Su et al. [31] investigate the relationship between the wavelength efficiency and the time flexibility of the scheduled demands. Andrei et al. [32] provide an integrated approach based on Lagrangean relaxation, which can jointly solve the problems of scheduling and RWA in fault-free WDM networks under the sliding scheduled traffic model. Saradhi et al. [33] propose a two-step heuristic to solve the RWA problem in survivable wavelength continuous networks under the sliding scheduled traffic model. First, a time conflict resolving window division algorithm is used to divide the demands into fixed-length time-independent windows. Then two RWA algorithms, Shortest Path Based RWA and Virtual Wavelength Graph Based RWA, are proposed to perform the RWA on the scheduled demand set. Jaekel and Chen [34] present ILP formulations for optimal design of survivable WDM networks, both with and without wavelength conversion capabilities, under the sliding scheduled traffic model. They consider both dedicated and shared path protection and jointly optimize the problem of scheduling the demands (in time) and allocating resources for the scheduled lightpaths. A two-step optimization process is also presented for larger networks. Andrei et al. [35] use light-trees to provision sub-wavelength multicast requests with flexible time scheduling over WDM optical networks.

Charbonneau and Vokkarane [36] provide a comprehensive survey on advance reservation (AR). Based on the knowledge of the start time and the duration of a demand, they classify AR into four categories, STSD (with the known start time and duration), UTSD (with uncertain start time and known duration and deadline), STUD (with known start time and uncertain duration), and UTUD (with uncertain start time and uncertain duration). STSD can be further divided into fixed and flexible window models. Our research falls within the scope of STSD. They also discuss various existing approaches, frameworks, and architectures that support AR applications.

In all of the above papers, once transmission of a demand starts, it continues until all data have been transmitted. Our approach differs from these existing techniques in that the data may be sent in non-consecutive intervals, if needed. In this thesis, we consider static advance reservation requests. Vokkarane et al. [37] – [41] focus on the dynamic traffic with a known duration and fixed or flexible start times. They solve the RWA problem using the lightpath segmentation and switching technique where a connection is allowed to be switched to another lightpath during different time slots.

The problem discussed in Chapter 5 has some similarities to the *task-scheduling* problem for parallel processing systems [42, 43], where a scheduled-lightpath-demand may be considered as a "task". Real time scheduling approaches commonly use the concept of *priority* to resolve contention for resources like processors and communication channels. Priorities are assigned to tasks by some policies. In *fixed-priority scheduling*, all jobs in a task have the same priority [43] – [45]. In *dynamic-priority scheduling*, priorities are assigned to individual jobs. The most commonly used algorithms for dynamic-priority scheduling are based on the *Earliest-Deadline-First* concept [43, 46]. Scheduling schemes can also be *preemptive* [42, 47] or *non-preemptive* [48] – [52]. A preemptive scheme allows a higher priority task to interrupt a lower priority task while a non-preemptive scheme does not permit this preemption.

Our approach focuses on the allocation of network resources, where the primary goal is to avoid contention, rather than resolving contention using priority. Furthermore, in the traditional task scheduling problem, the availability of the processors can usually be considered independently of each other. In other words, if a processor P1 is “busy” processing a job j_1, this normally does not affect the availability of another processor P2 for a different job j_2. In our model, however, the routing constraints on lightpaths (e.g. wavelength continuity constraint) typically result in complex dependencies for resource availability. For example, if a WDM channel λ_1 (considered as a resource to be allocated) is occupied on the incoming links to a node, the same channel on any outgoing link becomes “unavailable” for all other demands, even if the channel is actually free on those links.

We also note that there has been some recent research on optical grid networks that consider scheduling of both processors and network resources concurrently [53] – [55], but all these approaches consider the traditional continuous traffic model.

2.2 Traffic Grooming

Traffic grooming techniques are used to combine low-speed data streams onto high-speed lightpaths with the objective of minimizing the network cost, or maximizing the network throughput. Traffic grooming can use either the bifurcated model or the non-bifurcated model. In the non-bifurcated (bifurcated) model, each user data stream is communicated using a single (one or more) logical path(s) from the source of the data stream to its destination. The bifurcated model allows more efficient use of network resources, but the non-bifurcated model has a number of technological advantages [56]. In this thesis, we adopt the non-bifurcated model.

For *static* traffic grooming, it is assumed that the set of low-speed traffic demands is known beforehand and is persistent throughout the lifetime of the network. So, there is no opportunity of sharing resources among different demands. A number of ILP formulations for solving the complete traffic grooming problem have been proposed in the literature [56] - [58]. Such formulations quickly become computationally intractable, even for moderate sized networks, and the problem is usually solved by applying heuristics for practical networks.

For *dynamic* traffic grooming [59], on the other hand, the arrival time of requests are not known ahead of time. Each demand can then be accommodated by i) routing over the logical topology using available bandwidth on the existing lightpaths, or ii) establishing new lightpath(s) if the required resources are available. If an established lightpath is not carrying any traffic, it may be torn down. Since lightpaths are created and/or destroyed in response to the current demand set, the logical topology typically varies with time. Xin et al. [8] propose setting up a static topology a priori and then routing the dynamic requests to minimize blocking probability. This paper considers *single-hop* routing only.

Although there have been significant interests in traffic grooming techniques for both static and dynamic traffic models, relatively little work has been done in terms of traffic grooming of scheduled demands. Wang et al. [60] propose a number of *heuristic* algorithms including a customized tabu search scheme to schedule demands in time and allocate resources to lightpaths under the sliding scheduled traffic model. However, this approach does not build a stable logical topology, but requires a topology that changes based on the traffic demands active at any given time. Lightpaths are established and torn down based on the traffic demands active at any given time.

The *static* virtual topology design (VTD) problem for periodic low-speed demands has been considered in [61] – [65], and both ILP and heuristic solutions have been proposed. Pavon-Marino et al. [62] present a three-step algorithm to establish a static virtual topology in transparent optical networks for a given set of time-varying traffic demands. To reduce the complexity of the problem, they apply the concept of domination to replace a sequence of traffic matrices for each time interval with a single dominating matrix. Then the logical topology design and traffic routing are performed based on this unique traffic matrix with the objective of minimizing the number of transceivers needed. Agrawal and Medhi [63] solve the same problem and suggest a hybrid objective function that combines the benefit of minimizing average packet-hop distance, network congestion, and total number of lightpaths. These works typically focus on the topology design and grooming aspects (without RWA), and have not addressed survivability of the demands. Ricciato and Salsano [64] consider the offline planning at both packet and optical layers consisting of mapping a set of lightpaths onto the physical topology and mapping packet switched traffic over the logical topology with fixed flow routing. Aparicio-Pardo et al. [65] study the trade-offs between transceiver cost reduction and increased reconfiguration frequency in optical networks with/without reconfigurable switching equipment. Through the exhaustive experimental results, they claim that an advantageous tradeoff between the two costs can be achieved by allowing a limited amount of reconfiguration [65]. The aforementioned papers all focus on fault-free networks, and do not consider protection of lightpaths.

The survivable traffic grooming problem for WDM optical networks has been addressed in [7], [66] - [68]. Survivability can be achieved at the lightpath level by using standard dedicated path protection (DPP) or shared path protection (SPP) techniques. Path protection schemes establish two lightpaths, a primary (or working) path and an edge-disjoint backup (or protection) path, for each logical edge. If a link on a primary path fails, the traffic will be automatically redirected to the pre-assigned backup path. In SPP, resources allocated to a protection path can be shared with other protection paths if the corresponding primary paths are edge-disjoint. This is called *backup multiplexing*. The backup multiplexing is illustrated in Fig. 2.2. We can see that primary path 1 (P1: 1 → 2 → 3 → 6) and primary path 2 (P2: 4 → 2) do not have any common edge. Under the single link failure scenario, the data flow carried by P1 and P2 will not be interrupted by a failed link at the same time. So we can assign wavelength 1 (λ1) for both backup path 1 (B1: 1 → 4 → 5 → 6) and backup path 2 (B2: 4 → 5 → 3 →2) on their common link 4 → 5.

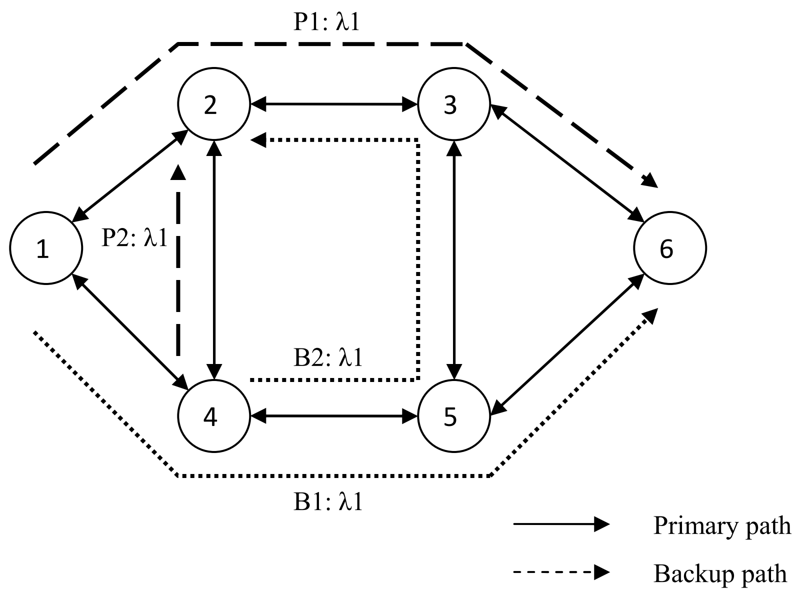

Figure 2.2: Shared path protection.

In dedicated protection, such sharing is not allowed. This leads to a simpler implementation at the cost of using additional resources. B1 and B2 have to be assigned to different channels since they both include link 4 → 5 in their route. As shown in Fig. 2.3, wavelength 1 (λ1) is reserved for B1 and wavelength 2 (λ2) is used for B2. Clearly, backup resource sharing achieves better network utilization than dedicated path protection.

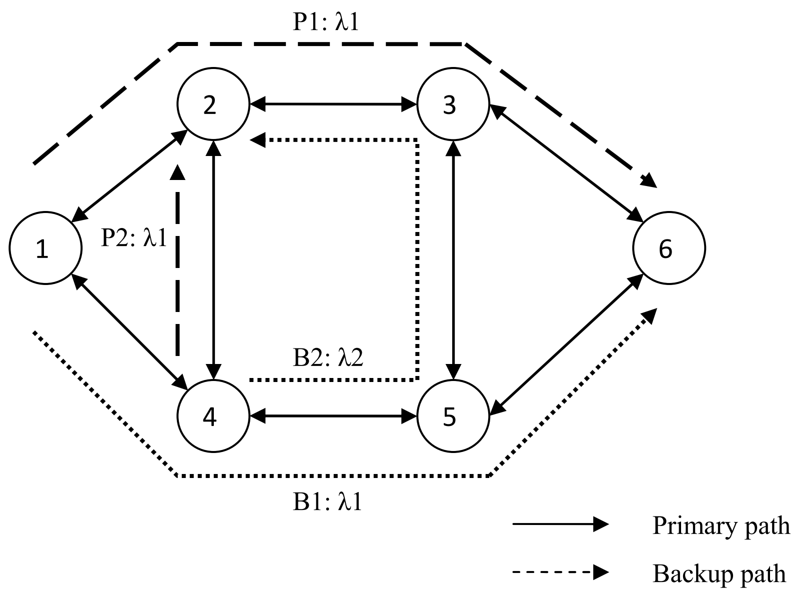

Figure 2.3: Dedicated path protection.

Heuristic grooming algorithms for solving the problem of survivable connections under various failures, such as fiber cut and duct cut, have been studied in [66, 67], under the general shared risk link group diverse routing constraints. Yao and Ramamurthy [66] consider static traffic and focus on the problem with protection at the sub-wavelength connection level, whereas path protection at the lightpath level has been considered in [67]. They also outline ILP formulations to generate optimal solutions for very small networks with a limited number of requests, which were mainly used to validate the results of the proposed heuristics. Jaekel et al. [68] propose an efficient ILP formulation for the complete survivable traffic grooming problem including topology design, traffic routing, and routing and wavelength assignment using both dedicated and shared protection at the lightpath level. However, this approach is for static grooming and does not consider scheduled demands. Ou et al. [7] focus on different frameworks for protecting low-speed connections against single link failures in WDM grooming networks. In this study, they propose three approaches for the protection of connections, namely, the protection-at-lightpath (PAL), mixed protection-at-connection (MPAC), and separate protection-at-connection (SPAC) levels. They provide a qualitative comparison among these methods and shown that, for both shared and dedicated protection, SPAC performs better with a sufficient number of grooming ports, and PAL performs better with a small to moderate number of such ports. In this thesis, we use PAL because the primary goal is to build a “stable” logical topology, i.e. we do not want the topology to keep changing as the short-lived sub-wavelength connections are established and torn down.

2.3 Energy Aware Resource Allocation

Energy aware design for wireless communication is a well established discipline, but research on energy efficient optical network design is only starting to receive increased research attention. In the past decade, the rapid growth in high-bandwidth applications, such as Video-on-Demand and online media sharing, has given rise to a corresponding increase in energy consumption of the network equipment [69]. Researchers have realized the importance of designing energy-efficient green networks to better utilize the available power and consequently reduce the network operational cost. For example, for a mid-size country, a 1% improvement of the total energy consumption can lead to a reduction of 5 billion US dollars per year in electricity cost [15]. Therefore, it is necessary to develop robust optimization strategies for the design of energy-efficient core networks. The typical approach is to switch off some network components during low traffic periods [16] – [18].

Some recent works have proposed a new cost model to estimate energy consumption [70], as well as new architectures for minimizing energy consumption [71]. Detailed power models for energy consumption in optical networks are proposed in [72] and [73]. Musumeci et al. [72] consider power consumption of IP routes, transponders, and MEMS-based optical switching for IP-over-WDM networks. Heddeghem et al. [73] propose an analytical power consumption model for various network layers. These comprehensive power models take into account different factors such as power consumption of inline amplifiers, signal regeneration, and optical switching, which become relevant when performing RWA of lightpaths over a physical fiber network. In our proposed approach, we assume that the logical topology (and corresponding RWA) is known, and we simply switch off the main power consumption components at the interfaces of the (inactive) lightpaths. For this scenario, the simplified model in [20] is more appropriate, and we use this model in our formulations.

Gupta and Singh [69] present the idea of putting network interfaces and components to sleep in order to reduce the energy consumption in Internet systems. This concept is also used in [17] to reduce energy consumption by rerouting IP traffic.

Muhammad et al. [74] advocate that only network devices deployed for protection paths can be put in the sleep mode. Cavdar et al. [75] study the trade-off between energy-efficiency and survivability. They implement shared path protection against single-link failures and assume that only the devices reserved for backup paths can be switched off in fault-free scenarios. When a failure occurrs, the corresponding backup resources need to be reactivated in a short time. They propose a hybrid objective function that minimizes both the total power consumption and the capacity consumption in terms of the number of wavelength-links. They observe that up to 40% saving in energy consumption can be achieved with a small increase of capacity consumption [75].

Chabarek et al. [16] formulate a generic model for router power consumption to exploit the power-awareness in network design and dynamic routing. Traffic flows on the low utilization links are allowed to be rerouted to an alternate path in order to minimize the number of active line cards. Orgerie et al. [76] develop an energy-efficient framework for Bulk Data transfers in dedicated networks with advance reservation. In order to save energy, the proposed framework allows unused network components to be switched off during certain period of time.

Bathula and Elmirghani [19] investigate the relationship between energy and Bit Error Rate (BER) in optical networks, and also propose a power-aware *anycasting* routing technique and sleep cycle protocols to meet the goal of energy saving. In Split-Incapable optical networks where an incoming signal cannot be splited into multiple output ports, a simple way for supporting *manycasting* demands is to establish a lighptah between the source node and each selected destination node. However, this approach might lead to inefficient use of network resources. To address this problem, Gadkar et al. [77] create a set of multi-hop logical paths for each manycast request. This routing technique reduces the total number of wavelengths (transponders) required to accommodate a set of static manycast demands and, hence, reduces the total power consumption.

Physical layer impairments in wavelength-routed networks limit the maximum distance a signal can travel in the optical domain without significant distortion. Therefore signal regeneration is required at some intermediate nodes for long-haul lightpaths. Typically, such regeneration involves reshaping, retiming, and reamplification of the signal (known as 3R regeneration), and requires O-E-O conversion at the regeneration site(s). In translucent WDM networks, sparsely located regenerators at certain nodes can be used to offset the impact of physical layer impairments. Xie et al. [78] address the regenerator placement (RP) problem with the objective of minimizing the total energy consumption of the transponders and 3R regenerators. They consider networks with mixed line rates. ILP formulations and energy aware RWA heuristics are proposed for determinining the number of regenerators and their locations to support lightpaths that exceed the reachability limit.

Recently, a number of energy aware RWA and traffic grooming options including minimizing the number of active router ports and/or line cards have been proposed [18], [20] - [22], [79, 80]. Huang et al. [21] propose both flow based and interface based formulations for network node power consumption. An ILP formulation and heuristics are provided to solve the traffic grooming problem with the objective of minimizing the power consumption of router ports. Coiro et al. [18] try to switch off optical links that do not carry any traffic demands as many as possible. If the connections on a link can be rerouted using some alterative paths, the link is powered off and removed from the network topology. Coiro et al. [79] propose an energy aware routing algorithm to minimize the total number of active optical amplifiers in a multi-fiber optical transport network. Chen and Jaekel [22] present an ILP formulation that minimizes both the static and dynamic components of power consumption. However, the ILP formulation becomes computationally intractable for practical networks and is not able to handle larger networks.

Yetginer and Rouskas [20] solve the complete static traffic grooming problem from a power consumption perspective, and show that significant energy saving can be achieved with power efficient grooming. Coiro et al. [80] consider the dynamic traffic scenario, and propose an energy aware routing scheme to improve the energy efficiency by minimizing the number of active optical amplifiers in the network. Scaraficci et al. [81] also investigate the dynamic traffic grooming problem and present an energy aware algorithm for lightpath provisioning in WDM optical networks. When a new traffic request arrives, an auxiliary graph $G(U, L)$ is generated, where U is a subset of end nodes and L represents existing and allocable lightpaths. The power consumption for data transmission is calculated and assigned to each edge as a weight. The goal is to route a new request while minimizing the total power consumption of the network. In order to reduce the search space, they adopt a Zone

Based With Neighbor Expansion algorithm [82] to find available resources for a new request without disrupting the existing data flows. Initially, a searching area, called "zone", is only a small fraction of the network. If sufficient network resources cannot be found, adjacent nodes are added to the auxiliary graph and the searching process is repeated. The procedure stops until a logical path is found for the new request, or the searching fails after K attempts and the connection is blocked.

The work closest to ours is [83], where Zhang et al. consider energy-efficient holding-time aware traffic grooming techniques. They propose an ILP and time-aware grooming heuristic for the fixed window model only. The approach presented in this thesis, on the other hand, can be applied to both fixed and sliding demands, and jointly performs resource allocation with scheduling (in time).

Chapter 3

Stable Logical Topology Design

In this chapter, we present our ILP formulations for traffic grooming of scheduled demands with fixed setup/teardown times. These approaches can also be used in conjunction with standard *dynamic* traffic grooming schemes. A stable logical topology capable of supporting the scheduled demands is set up first. Subsequently, if unscheduled demands are presented to the network, they can be accommodated using existing dynamic traffic grooming techniques. The *static* traffic grooming problem can be treated as a special case of our formulations, where the duration of each demand spans the entire network lifetime. This means that all the demands overlap in time so that resource sharing, among individual demands, is not possible. A preliminary version of our approach is presented in [84], however it can only handle networks with full wavelength conversion capabilities, and does not consider shared path protection.

We assume that a set of scheduled, low-speed, periodic traffic demands with fixed setup and teardown times are specified as input. The demands are arranged in an increasing order of their start times, and used to partition the entire time period into disjoint intervals. In our scheme, it is quite possible to have multiple demands with the same setup and/or teardown times. Our formulations solve the following sub-problems:

i) Designing a stable logical topology that does not change over time,

ii) Routing and wavelength assignment for each lightpath included in the logical topology,

iii) Combining low-speed traffic demands on to high capacity lightpaths, and

iv) Sharing resources among time-disjoint demands.

The resources being minimized may be at the optical level (e.g. number of transceivers) or at the electronic level (e.g. the amount of electronic switching needed for each demand). The actual resource being minimized depends on the particular objective function used by the ILP (as discussed in Sec. 3.5). Our formulations also take into account resource limitations, such as the number of transmitters and receivers at each node.

3.1 Notation

In our ILP formulations, we will use the following notation for input data:

- *G[N, E]*: A physical fiber network where *N* is the set of nodes, and *E* is the set of links.
- *K*: A set of channels that each fiber can accommodate.
- *L*: A set of potential logical edges that may be included in the logical topology.
- *o(l) (e(l))*: Originating (terminating) node for lightpath $l \in L$.
- T_n (R_n): Number of transmitters (receivers) available at node *n*.
- *C*: Capacity of a *single* lightpath in Optical Carrier level notation (*OC-n*). The base rate (OC-1) is 51.84Mbps. In this thesis, we have used *C* = *OC-160.*
- *Q*: A set of scheduled sub-wavelength traffic demands. Each element $q \in Q$ is represented as $(s_q, d_q, b_{q,} st_q, et_q)$.
- s_q (d_q): Source (destination) node of demand *q*.
- b_q: Bandwidth requirement of demand *q* in *OC-n* notation. We assume that the data rate of the individual demands varies between *OC-3* to *OC-24*, and is always less than the capacity of a lightpath.
- st_q (et_q): Setup (teardown) time of demand *q*.
- *R*: A set of pre-computed edge-disjoint routes, over the physical topology to be considered for RWA, between each ordered pair of end-nodes.

- $J_e^{sd,r}$ = 1, if and only if the r^{th} route between source s and destination d uses fiber link e.
- i_{max}: The total number of disjoint time intervals in the network.
- I: The set $\{i_j, 1 \leq j \leq i_{\max}\}$ of time intervals.
- $X_{q,i}$: A parameter given as input to the ILP and set to 1 if demand q is active during interval i.

The *binary* variables required for the ILP are defined as below:

- p_l = 1, if and only if logical edge l is included in the logical topology, 0 otherwise.
- $f_{q,l}$ = 1, if and only if demand q is routed over logical edge l, 0 otherwise.
- $g_{r,l}$ ($h_{r,l}$) = 1, if and only if logical edge l uses the r^{th} route to establish the primary (backup) lightpath, from its source s to destination d, 0 otherwise.
- $w_{k,l}$ ($u_{k,l}$) = 1, if and only if channel k is assigned to the primary (backup) lightpath corresponding to logical edge l, 0 otherwise.
- y_q = 1, if and only if demand q is accommodated, 0 otherwise.

We also define the following *continuous* variables. These continuous variables function as binary variables in the sense that they can only take on values of 0 or 1. This is accomplished by adding appropriate constraints, which prevent the continuous variable from taking on any values other than 0 or 1. This approach reduces the number of integer variables and, hence, the overall complexity of the formulation.

- $s_{e,l}$ ($t_{e,l}$) = 1, if and only if logical edge l uses link e on its primary (backup) route, 0 otherwise.
- $\alpha_{k,l}^{e}$ ($\beta_{k,l}^{e}$) = 1, if and only if channel k on physical link e is assigned to the primary (backup) lightpath corresponding to logical edge l, 0 otherwise.
- β_{k}^{e} = 1, if and only if channel k on physical link e is used by one or more backup lightpaths, 0 otherwise.

3.2 An ILP for Scheduled Traffic Grooming without Wavelength Conversion Capability (ILP1)

Minimize $$\sum_{l \in L} \sum_{q \in Q} f_{q,l} \cdot b_q \qquad (3.1)$$

Subject to:

a) Flow and capacity constraints:

$$\sum_{l:o(l)=n} f_{q,l} - \sum_{l:e(l)=n} f_{q,l} = \begin{cases} 1, if\ n = s_q, \\ -1, if\ n = d_q, \\ 0, otherwise \end{cases} \qquad \forall q \in Q, \forall n \in N \qquad (3.2)$$

$$\sum_{q \in Q} f_{q,l} \cdot b_q \cdot X_{q,i} \leq C \cdot p_l \quad \forall i \in I, \forall l \in L \tag{3.3}$$

b) Transceiver constraints:

$$\sum_{l:o(l)=n} p_l \leq T_n, \qquad \forall n \in N \tag{3.4}$$

$$\sum_{l:e(l)=n} p_l \leq R_n, \qquad \forall n \in N \tag{3.5}$$

c) RWA constraints:

$$\sum_{r} g_{r,l} = p_l \qquad \forall l \in L \tag{3.6}$$

$$s_{e,l} = \sum_{r} g_{r,l} \cdot J_e^{s_l d_l, r}, \forall l \in L, \forall e \in E \tag{3.7}$$

$$\sum_{k \in K} w_{k,l} = p_l \qquad \forall l \in L \tag{3.8}$$

$$w_{k,l} + s_{e,l} - \alpha_{k,l}^e \leq 1 \qquad \forall k \in K, \forall e \in E, \forall l \in L \tag{3.9}$$

$$w_{k,l} \geq \alpha_{k,l}^e \qquad \forall k \in K, \forall e \in E, \forall l \in L \tag{3.10}$$

$$s_{e,l} \geq \alpha_{k,l}^e \qquad \forall k \in K, \forall e \in E, \forall l \in L \tag{3.11}$$

$$\sum_{l \in L} \alpha_{k,l}^e \leq 1 \qquad \forall k \in K, \forall e \in E \tag{3.12}$$

The objective function in Eq. (3.1) minimizes the total amount of electronic switching required for all the demands, by minimizing the weighted hop count. The ILP uses the bandwidth requirement of each individual demand as the weight. The weighted hop count refers to the weighted sum of the number of logical edges or lightpaths traversed by each traffic demand. Our formulation designs a static logical topology to accommodate all the demands that are active during each time interval without exceeding the capacity of the lightpaths. Even though the bandwidth requirements vary from one interval to another, our logical topology remains stable for the entire duration. Any available spare capacity can be used to route unscheduled demands arriving at the network using existing dynamic traffic grooming techniques. Clearly, by minimizing the amount of resources (lightpaths) used by each demand, it aims to maximize the total amount of spare capacity available on all the lightpaths.

Constraint (3.2) is the standard flow equation [85], and is used to route each demand over the logical topology, using a single multi-hop logical path, in accordance with the non-bifurcation model used in this thesis.

Capacity constraint (3.3) ensures that there is no flow on a logical edge l if it is not selected for the logical topology (i.e., $p_l = 0$). If a logical edge is selected (i.e., $p_l = 1$), then Eq. (3.3) ensures that the total flow on l during any given time interval i does not exceed the capacity C of the lightpath. We note that the selected logical edges are obtained from L, which may contain multiple potential edges with the same source and destination nodes, but each with a distinct *id_number*. So, the ILP can handle multiple lightpaths between some (or all) node pairs.

Constraints (3.4) and (3.5) ensure that the total number of lightpaths originating and terminating at node n does not exceed the number of transmitters and receivers available at node n.

Constraint (3.6) is the routing constraint for the primary lightpath. It ensures that if the l^{th} logical edge is included in the logical topology, then the corresponding primary lightpath is allocated exactly one route over the physical topology from the R pre-computed routes for each node pair.

Constraint (3.7) is used to define variables $s_{e,l}$. The variable $s_{e,l}$ will have a value of 1 if the primary lightpath, from source s_l to destination d_l, uses the r^{th} route, i.e. $x_{r,l} = 1$, and edge e is on the r^{th} route i.e. $J_e^{s_l d_l, r} = 1$. In other words, $s_{e,l} = 1$ if and only if the primary lightpath corresponding to logical edge l uses physical link e.

Constraint (3.8) assigns a channel for each lightpath included in the logical topology. It also enforces the *wavelength continuity constraint* by allocating exactly one channel for each selected lightpath.

Constraints (3.9) – (3.11) are used to define the *continuous* variable $\alpha_{k,l}^{e}$ which is set to 1, if both $w_{k,l}$ and $s_{e,l}$ are 1, and 0 otherwise.

Finally, constraint (3.12) states that a channel k on a link e can be assigned to at most one lightpath.

3.3 An ILP for Survivable Scheduled Traffic Grooming (ILP2)

The initial formulation (ILP1) can be augmented in order to implement survivability. In this thesis we implement protection at the *lightpath* level, rather than at the connection level. Thus, for each primary lightpath, we also set up a backup lightpath using either DPP or SPP. In this section, we present an ILP for survivable logical topology design and traffic grooming of scheduled demands, by including additional constraints which handle resource allocation for the backup paths.

Minimize $$\sum_{l \in L} \sum_{q \in Q} f_{q,l} \cdot b_q \quad (3.13)$$

Subject to:

Constraints (3.2) – (3.11)

d) RWA constraints for backup lightpaths:

$$\sum_{r} h_{r,l} = p_l \qquad \forall l \in L \quad (3.14)$$

$$t_{e,l} = \sum_{r} h_{r,l} \cdot J_e^{s_l d_l, r}, \forall l \in L, \forall e \in E \quad (3.15)$$

$$\sum_{k \in K} u_{k,l} = p_l \qquad \forall l \in L \quad (3.16)$$

$$u_{k,l} + t_{e,l} - \beta_{k,l}^{e} \le 1 \qquad \forall k \in K, \forall e \in E, \forall l \in L \quad (3.17)$$

$$u_{k,l} \geq \beta_{k,l}^{e} \qquad \forall k \in K, \forall e \in E, \forall l \in L \tag{3.18}$$

$$t_{e,l} \geq \beta_{k,l}^{e} \qquad \forall k \in K, \forall e \in E, \forall l \in L \tag{3.19}$$

$$g_{r,l} + h_{r,l} \leq 1 \qquad \forall l \in L, \quad r = 0,1,2,\ldots, R-1 \tag{3.20}$$

$$\beta_{k}^{e} \geq \beta_{k,l}^{e} \qquad \forall k \in K, \forall e \in E, \forall l \in L \tag{3.21}$$

$$\beta_{k}^{e} \leq \sum_{l \in L} \beta_{k,l}^{e} \qquad \forall k \in K, \forall e \in E \tag{3.22}$$

$$\sum_{l \in L} \alpha_{k,l}^{e} + \beta_{k}^{e} \leq 1 \qquad \forall k \in K, \forall e \in E \tag{3.23}$$

e) Dedicated protection constraint:

$$\sum_{l \in L} \beta_{k,l}^{e} \leq 1 \qquad \forall k \in K, \forall e \in E \tag{3.24}$$

f) Shared protection constraint:

$$\beta_{k,l1}^{e1} + \beta_{k,l2}^{e1} + s_{e2,l1} + s_{e2,l2} \leq 3, \ \forall k \in K, \forall e1 \neq e2 \in E, \forall l1 < l2 \in L \tag{3.25}$$

Similar to Eq. (3.1), the objective function given in Eq. (3.13) also minimizes the amount of electronic switching required by all demands by minimizing the weighted hop count. We also use flow and capacity constraints (3.2) and (3.3) and transceiver constraints (3.4) and (3.5) as well as the RWA constraints for primary lightpaths (3.6) – (3.11) given in ILP1.

Constraints (3.14) - (3.16) are used for routing and wavelength assignment of backup lightpaths, and are analogous to (3.6) – (3.8). Constraints (3.17) – (3.19) are used to define the variable $\beta^{e}_{k,l}$ for backup lightpaths, and are similar to (3.9) – (3.11). Constraint (3.20) states that the same physical route cannot be selected for both the primary and backup lightpath corresponding to a given logical edge. Since all the *R* routes between each source-destination pair are pre-computed to be edge disjoint, this ensures that the primary and backup lightpaths are edge disjoint.

Constraints (3.21) and (3.22) are used to define the variable β^{e}_{k}, which is set to 1 if channel *k* on link *e* is assigned to at least one (possibly more) backup lightpath; otherwise, β^{e}_{k} is set to 0. This is needed since a channel may be assigned to more than one backup lightpath (for shared protection only). Even if multiple backup lightpaths are allocated to channel *k* on link *e*, the value of β^{e}_{k} does not exceed 1.

Constraint (3.23) replaces constraint (3.12) and ensures that if a channel *k* on link *e* is assigned to a primary lightpath, it cannot be assigned to any other primary or backup lightpath.

Constraint (3.24) is used for *dedicated* protection and ensures that two backup lightpaths are not allowed to share any resources, i.e. cannot be assigned the same channel on a common link. For shared protection, constraint (3.25) is used instead of

(3.24). This constraint states that if backup lightpaths for two logical edges, $l1$ and $l2$, share a channel k on a common link $e1$, then for any other link $e2$ it is not possible for both primary paths to share that link.

3.4 Handling Wavelength Convertible Networks (ILP3)

The formulation given in ILP2 can be easily modified to handle wavelength convertible networks as well. In the modified formulation, denoted as ILP3, the variables $\alpha_{k,l}^{e}$ and $\beta_{k,l}^{e}$ are defined as binary variables, rather than continuous variables. In wavelength convertible networks a lightpath may be assigned a different wavelength on each link it traverses. Therefore the constraints (3.8) and (3.16), each assigning a *single* wavelength for a primary and backup lightpath, respectively, are not appropriate for this case. Instead, these two constraints are replaced by constraints (3.26) and (3.27) respectively as shown below.

$$\sum_{k \in K} \alpha_{k,l}^{e} = s_{e,l} \qquad \forall l \in L, \forall e \in E \qquad (3.26)$$

$$\sum_{k \in K} \beta_{k,l}^{e} = t_{e,l} \qquad \forall l \in L, \forall e \in E \qquad (3.27)$$

Constraint (3.26) states that a channel is assigned on link e, to the primary lightpath for the l^{th} logical edge, if the lightpath traverses link e (i.e. $s_{e,l} = 1$); otherwise no channel is assigned. In a similar manner (3.27) assigns a channel on each traversed link for a backup lightpath. However, there is no need to restrict that a lightpath must be assigned the same channel on successive links. Finally, constraints (3.9) – (3.11) and (3.17) – (3.19) are no longer needed for ILP3. All other constraints remain unchanged.

3.5 Possible Modifications to the ILPs

The ILP formulations ILP1, ILP2, and ILP3 minimize the weighted sum of the number of *logical* edges or lightpaths traversed by each traffic demand. A number of different objective functions can also be easily implemented, by slightly modifying our ILPs. For example, one objective may be to minimize the number of lightpaths (i.e. minimize $\sum_{l \in L} p_l$), which reduces the total transceiver cost. Another possible objective is to reduce the optical resources at the physical level, in terms of the number of wavelength-links, i.e., the number of hops required to route the lightpaths over the *physical* topology (minimize $\sum_{e \in E} \sum_{l \in L} s_{e,l} + t_{e,l}$).

The objectives mentioned so far minimize the amount of resources required to accommodate a set of demands. However, if sufficient resources are not available in the network, it makes sense to maximize the amount of traffic that can be supported, i.e., maximize the throughput. This can be done by changing the objective function and the flow constraint. The objective function (3.1) or (3.13) should be changed to the following:

$$\textbf{\textit{Maximize}} \ \sum_{q \in Q} b_q \cdot y_q \tag{3.28}$$

Also, the flow constraint (3.2) should be replaced by the following constraint:

$$\sum_{l:o(l)=n} f_{q,l} - \sum_{l:e(l)=n} f_{q,l} = \begin{cases} y_q, if\ n = s_q, \\ -y_q, if\ n = d_q, \\ 0, otherwise \end{cases} \qquad \forall q \in Q, \forall n \in N \tag{3.29}$$

The other constraints remain the same. This modified formulation uses an integer variable y_q, which specifies whether traffic demand q can be accommodated in the network. If demand q is handled by the above formulation, $y_q = 1$. Thus $b_q \,.\, y_q$ gives the contribution of demand q to the network throughput. Equation (3.28) is the objective function that maximizes the weighted sum $\sum_{q \in Q} b_q \cdot y_q$, of traffic demands that can be handled by the network. This value depends on available resources such as the number of transceivers per node, the capacity of a lightpath, and the number of available channels per fiber.

Constraint (3.29) gives the standard flow conservation constraint as in constraint (3.2) except that it enforces the flow constraint only for the accommodated traffic in the network (i.e., $y_q = 1$).

3.6 Experimental Results

We have simulated our formulations with different demand sets on a number of networks ranging from a small 6-node network [56] to practical sized networks such as the 14-node NSFNET and 20–node ARPANET [86]. For each network, we ran our experiments with at least 5 demand sets, where the size of each demand set ranged from about 60 demands (for the 6- node network) to over 800 demands (for the 20-node network). Table 3.1 shows the simulation parameters used for different network sizes and the average number of lightpaths established for each network size. In this table, n indicates the network size, i.e., the number of nodes in the network. The column T_n/R_n indicates the number of transmitters and the number of receivers in each node. For our simulations, we assumed that each node has the same number of transmitters and receivers. However, such uniformity is not required in our formulations, and the proposed ILPs can easily handle different number of transceivers at each node. The source and the destination of each demand as well as its data rate, in *OC-n* notation, were randomly generated. The results were obtained using ILOG CPLEX [87]. For our experiments, we considered demands scheduled over a 24-hour period. However, any suitable scheduling period (such as hourly, daily, weekly, or even monthly scheduling) can be used with our formulation.

No. of nodes (n)	T_n/R_n	Number of demands					Number of lightpaths
		OC-3	OC-6	OC-12	OC-24	Total	
6	3	14	14	16	19	63	18
10	5	43	42	50	57	192	50
14	5	92	92	91	110	385	70
20	10	188	190	186	242	806	200

Table 3.1 Number of demands and number of lightpaths established for each network size.

3.6.1 Degree of Overlap of Traffic Demands

We have compared the performance of our approach to the conventional approach that does not consider the connection holding time of individual demands. We refer to such a model as a *full demand overlap* (*FDO*) model. The *FDO* model was simulated, in our experiments, by setting the duration of every demand to the full 24-hour period. The relative performance of the proposed ILP, over the FDO model, depends directly on the amount of time overlap among the individual sub-wavelength demands in the demand set Q. Therefore, we considered three different demand sets as follows:

i) Low demand overlap (*LDO*): duration of each demand is set randomly between 1 and 10 hours.

ii) Medium demand overlap (*MDO*): duration of each demand is set randomly between 1 and 24 hours.

iii) High demand overlap (*HDO*): duration of each demand is set randomly between 10 and 24 hours.

The start time and duration for each individual demand in each set was randomly generated. The main idea in generating the demand sets was that given the same overall time period and same number of demands, the amount of overlap between demands would in general increase as the length of the individual demands are increased. We have used the demand overlap factor (δ) as a metric to characterize the degree of overlap of a set of demands. The demand overlap factor is defined as:

$$\delta = \frac{\sum_{i}^{i_{\max}} \sum_{j=1}^{|N|-1} \sum_{k=j+1}^{|N|} T_{j,k}^{i}}{0.5 \cdot i_{\max} \cdot |N|(|N|-1)} \qquad (3.30)$$

Here, $T_{j,k}^{i} = 1$ if demands j and k overlap during interval i. The value of δ varies between 0 (there are no overlapping demands in any interval) and 1 (all demands overlap in all intervals). If the value of δ for a given set of demands is close to 0, it indicates low demand overlap, and its value increases as the amount of overlap increases. Obviously, a demand set with δ = 1 corresponds to the FDO model. The values of δ for the LDO, MDO, and HDO sets used in our simulations are 0.05, 0.28, and 0.5, respectively.

3.6.2 Results of Fault-Free Networks

In general, our simulations indicated that if a demand set can be accommodated with less number of channels (e.g., $|K|=16$), no additional advantage is gained by setting it with a larger value (e.g., $|K|=32$). However, if a demand set cannot be handled with $|K|=16$, it is possible to accommodate the demand set by increasing the number of available channels to 32. The time required to obtain an optimal solution ranged from less than 1*s* for the smaller networks to several thousand seconds for the 20-node network. These solution times are reasonable since this type of topology design is expected to be done offline.

We first present the results for networks without wavelength conversion using ILP1. Figure 3.1 shows the maximum amount of resources used during any time interval (in terms of the weighted hop count at the logical level) by the scheduled demand set. It is clear from Fig. 3.1 that consideration of connection holding times allows more efficient use of available resources. The amount of improvement increases as the demand overlap decreases. This means that considerable savings can be achieved by allowing resource sharing among time-disjoint demands. The average improvement varies from over 25% for a demand set with a high degree of overlap to about 70% for low demand overlap.

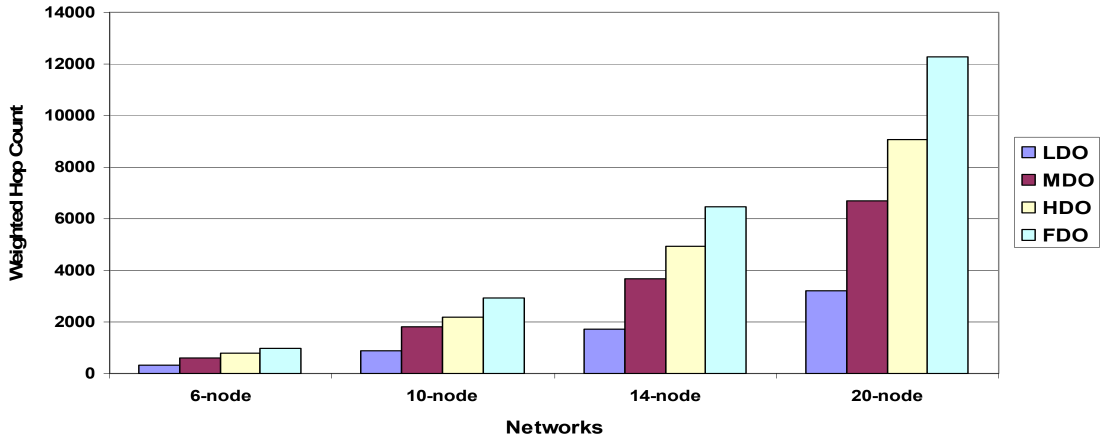

Figure 3.1: Resource requirements vs. network size in fault-free networks without wavelength conversion.

We note that since our proposed ILP generates an optimal solution for a given set of demands, it is guaranteed to perform as well as or better than a heuristic dynamic traffic grooming based approach for the same demand set. Figure 3.2 shows the percentage reduction in resource requirements with respect to the *FDO case* for different network sizes and different amounts of demand overlap.

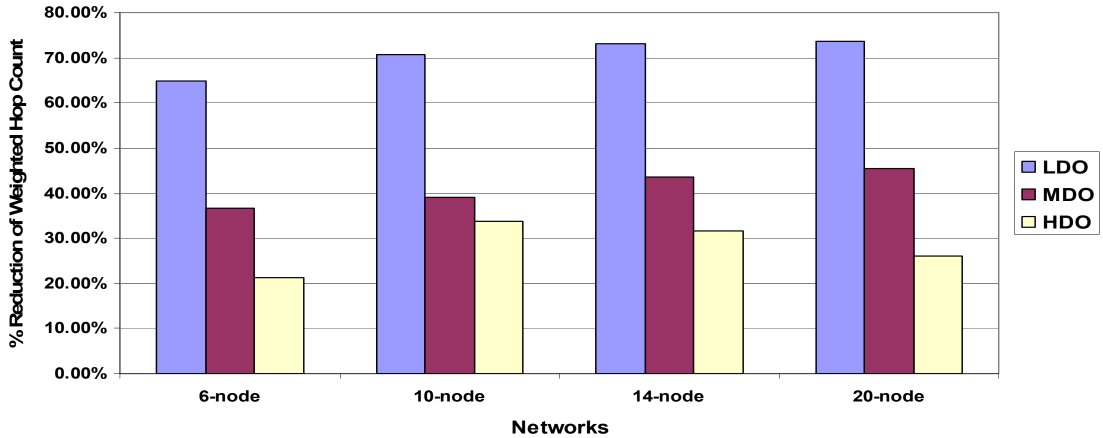

Figure 3.2: Percentage improvement in resource requirements compared to *FDO* case using ILP1.

We see that the amount of improvement is significant in all cases, even when demand overlap is high. Consideration of connection holding times seems to have a greater impact for larger networks (14-node and 20-node networks).

We next consider results for wavelength convertible networks. Figure 3.3 indicates network resources used (in terms of the weighted hop count at the logical level) on different network sizes under different demand overlap models.

Figure 3.4 shows the percent reduction of the weighted hop counts using LDO, MDO, and HDO models compared to the FDO model for networks with different sizes. The number of channels (|K|) for the networks with 6 nodes to 14 nodes was considered as 16, while for a 20-node network, |K| was 32. It is evident from Figs. 3.3 and 3.4 that significant savings can be achieved using our approach compared to full demand overlap models. As noted in Fig. 3.1, the amount of improvement increases as the demand overlap decreases.

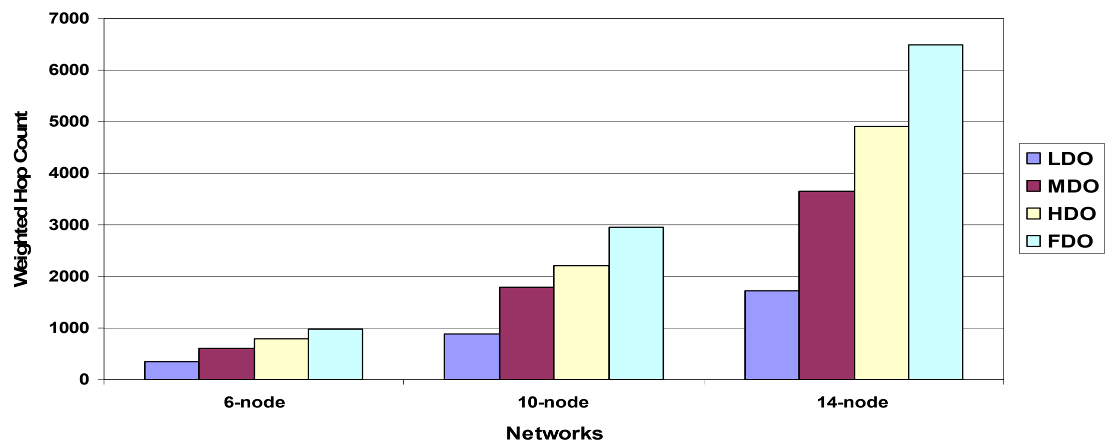

Figure 3.3: Resource requirements vs. network size with $|K| = 16$ in fault-free networks with wavelength conversion.

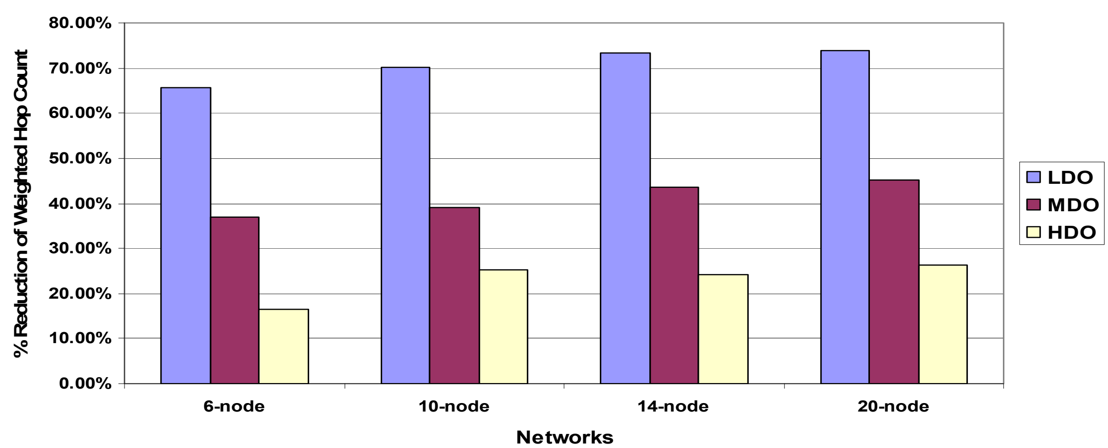

Figure 3.4: Percentage improvement in resource requirements compared to *FDO* case in fault-free networks with wavelength conversion.

Figure 3.5 shows that the resource requirements in wavelength continuous and wavelength convertible networks under LDO and MDO models are very close, regardless of the network size. The results for the HDO and FDO models follow a similar pattern. This seems to indicate that in most cases little or no improvements are realized, in terms of resource utilization at the *logical* level, by allowing wavelength conversion capabilities at the network nodes. Please note that these results are only for the objective of reducing the weighted hop count. However, if we consider different objective functions, such as minimizing the number of wavelength-links or maximizing the throughput, it is quite possible that wavelength convertible networks have noticeable advantages by allowing lightpaths to use shorter routes.

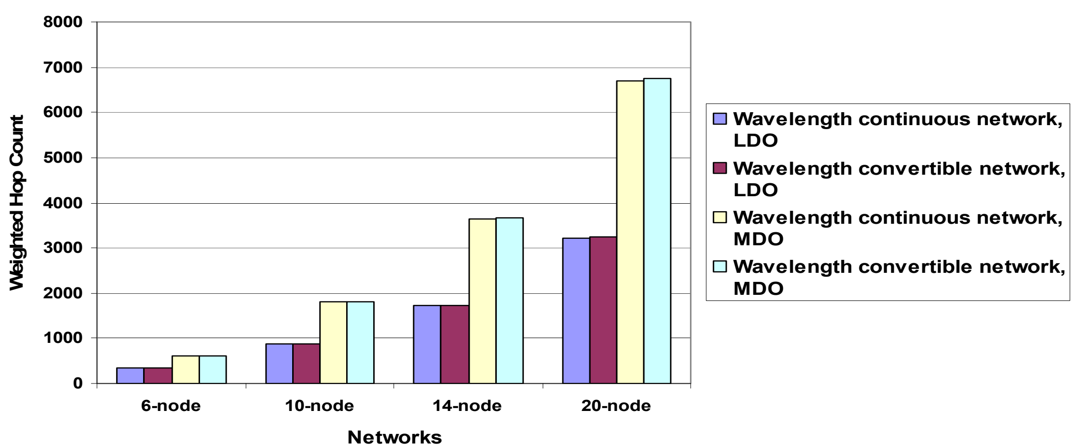

Figure 3.5: Comparison of resource requirements in wavelength continuous and wavelength convertible networks under LDO and MDO models.

In addition to evaluating the performance of our proposed formulations, we also studied the performance (after modifying ILP1 appropriately) for each of the objective functions on 6-node and 10-node networks. The results for the 6-node network are shown in Fig. 3.6, and the results for the 10-node network are similar and, hence, omitted here.

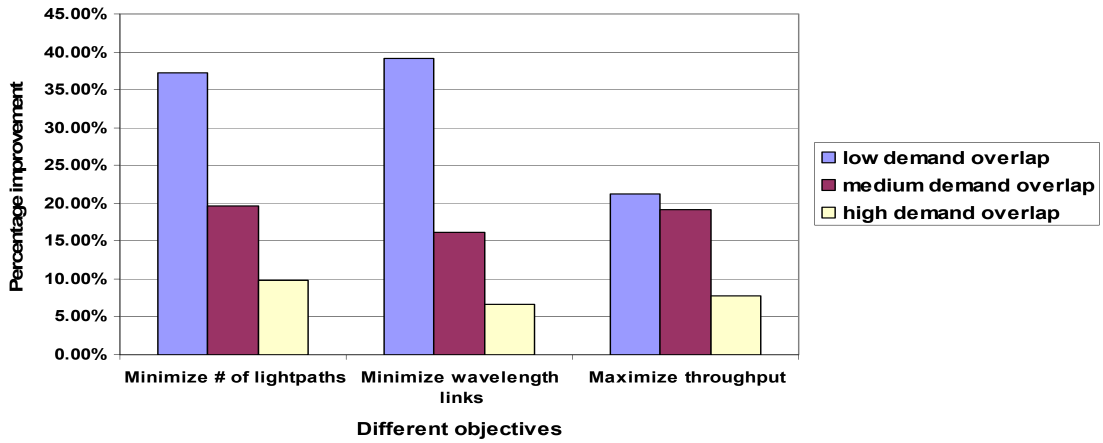

Figure 3.6: Relative improvement with different objective functions.

The above results are consistent with the results of our proposed formulations, and indicate that noticeable improvements can be expected, even for these alternative objectives. In order to include the results for different objectives on the same graph, we have expressed the values in terms of the relative improvement achieved (over FDO model) for each case. The first (second) set of bars shows the reduction in the number of lightpaths (wavelength-links) obtained by using min $\sum_{l \in L} p_l$ (min $\sum_{e \in E} \sum_{l \in L} s_{e,l} + t_{e,l}$) as the objective. The third set shows the relative increase in the number of demands that can be accommodated, using the objective max $\sum_{q \in Q} b_q \cdot y_q$.

3.6.3 Results of Survivable Networks

In this subsection, we consider results for survivable networks without wavelength conversion. Results for wavelength convertible networks followed a similar pattern and are not reported separately. We implement network survivability using both dedicated and shared path protections. As in the previous cases, our objective is to minimize the weighted hop count. However, the weighted hop count (at the logical level) remains the same whether or not backup lightpaths are implemented. Therefore, for comparison purposes, we have compared the number of wavelength-links required for fault-free (i.e., no backup paths), dedicated, and shared protection schemes. Figure 3.7 compares the average number of wavelength-links required for wavelength continuous networks for the case with no protection, shared protection, and dedicated protection under low demand overlap. As expected, shared path protection requires fewer resources compared with dedicated path protection. Results for the other cases follow a similar pattern.

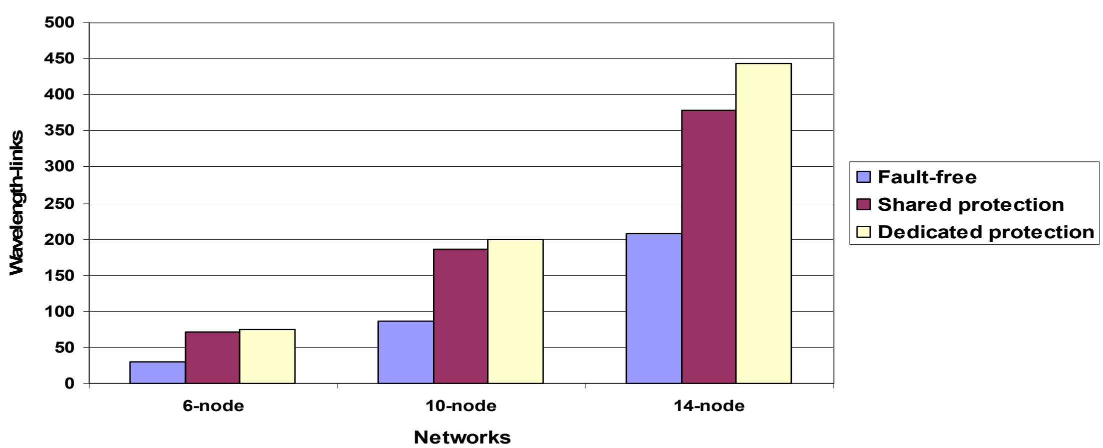

Figure 3.7: Resource requirements vs. network size with $|K| = 16$ in survivable wavelength continuous networks under LDO model.

In summary, we have experimented with a number of different networks; for each network we considered several demand sets and different amounts of overlap between demands. The significance of the results can be noted as follows:

i) The results not only show that knowledge of connection holding times can decrease the amount of resources required to accommodate a given set of demands (which was expected), they also indicate how the relative improvements are affected by the amount of demand overlap.

ii) The results clearly indicate (Table 3.1) that it is possible to accommodate a large number of low-speed demands using a stable logical topology with relatively few logical edges, when the data rates of the individual demands are relatively small compared to the capacity of a lightpath.

iii) The results indicate that wavelength conversion does not lead to significant benefits, in terms of the objective used in our simulations.

3.7 Summary

The main contributions of this chapter are:

- We propose a new approach for survivable traffic grooming and topology design that exploits the knowledge of connection holding times of low-speed scheduled traffic demands to devise a stable logical topology and maximize resource sharing.

- We present efficient ILP formulations that optimally solve the complete traffic grooming problem (including logical topology design and RWA) and provide protection against single link faults at the lightpath level. This allows us to avoid using suboptimal heuristics, which often have no specified performance bounds.
- We extend our proposed ILP to handle networks with and without wavelength converters.
- We demonstrate through simulations that, unlike most existing ILP formulations for traffic grooming, our formulation can be used for practical networks with hundreds of individual traffic demands. The problem sizes considered in this work are comparable to those used for existing heuristic approaches available in the literature [56, 64, 67].

Chapter 4

Energy Efficient Traffic Grooming

In this chapter, we address the problem of joint scheduling (in time) and traffic routing with the goal of minimizing energy consumption for sliding demands. We first present an efficient ILP formulation for scheduling and allocating resource to a single sub-wavelength traffic demand, such that the incremental energy consumption due to the new demand is minimized. Next, we extend the proposed approach, and present a new heuristic that jointly performs scheduling and resource allocation for a large set of periodic demands with the goal of reducing the overall energy consumption of the network for the entire demand set.

Before formally presenting the problem, we will first discuss, using an example, how knowledge of demand holding times can be exploited to obtain more power efficient grooming. We consider the network of Fig. 4.1, which shows the logical topology of a small network with four nodes and five logical edges (lightpaths) represented by circles and solid lines, respectively. We partition the entire time period of interest into a number of consecutive time intervals $i_1, i_2, i_3, \ldots i_{max}$ ($i_{max} = 7$ in our case) as shown in Fig. 4.2. We note that the duration of each interval can be made as long or short as required based on the demand start and end times. During interval i_3, four demands q_1, q_2, q_3, and q_4, shown as dashed lines in Fig. 4.1, are active and are routed over the four active lightpaths l_1, l_2, l_3, and l_4, respectively. Lightpath l_5 is *inactive* since it is not carrying any traffic at this time. The bandwidth requirement for each demand is expressed as a fraction of the lightpath capacity, and α_i (ω_i) represents the start (end) time for demand q_i. Now, suppose a new demand q_5 with bandwidth requirement of 0.4 needs to be routed from node 1 to node 4, starting in interval i_4, as shown in Fig. 4.2. There are three possible ways to accomplish this:

1) using only lightpath l_5,
2) using lightpaths l_1 and l_2, or
3) using lightpaths l_3 and l_4.

Obviously, option 1 requires the most energy since it increases the static power consumption by switching *on* lightpath l_5. Regarding option 2 and option 3, at first it appears that there is no difference between them, since all four lightpaths (l_1, l_2, l_3, and l_4) are already active during interval i_4, and the increase in dynamic power consumption is the same for both cases. However, when we consider the demand holding times, we see that significant energy savings can be achieved by selecting option 3, i.e., routing the new demand over lightpaths l_3 and l_4. If new demand is routed over l_3 and l_4, then l_1 and l_2 can both be turned *off* at the end of interval i_4. Lightpath l_4 remains *on* as scheduled, and only l_3 needs to remain *active* for one extra interval (up to end of i_6). On the other hand, if q_5 is routed over l_1 and l_2, then both l_1 and l_2 must remain *active* until the end of i_6 (two extra *active* intervals for each lightpath). Hence option 3 reduces the overall static power consumption compared to option 2.

In the above example, we consider only the *fixed* window model, where the start and end times of the demands are already known and cannot be changed. However, if there is some flexibility in how the demands are scheduled in time (*sliding* window model), then it is possible to reduce energy consumption even further. For example, let us suppose that demand q_5 cannot start before interval i_2 ends, and must end before interval i_7. In this case, if we schedule q_5 to start at the beginning of interval i_3 (instead of i_4), then lightpath l_3 can be turned *off* at the end of interval i_5, as originally intended. So, jointly performing both routing and scheduling poses an interesting design problem, which can lead to a further decrease in the overall energy consumption.

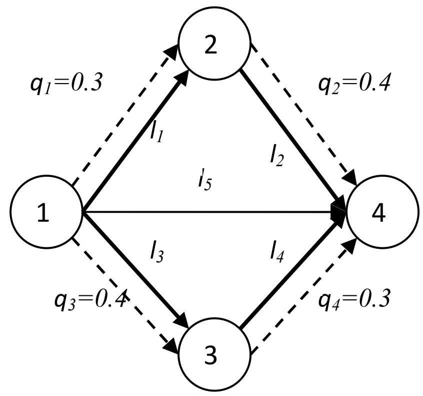

Figure 4.1: Logical topology and traffic routing.

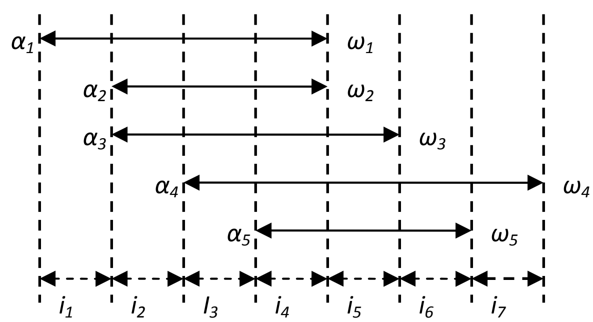

Figure 4.2: Overlapping demands under the fixed window model.

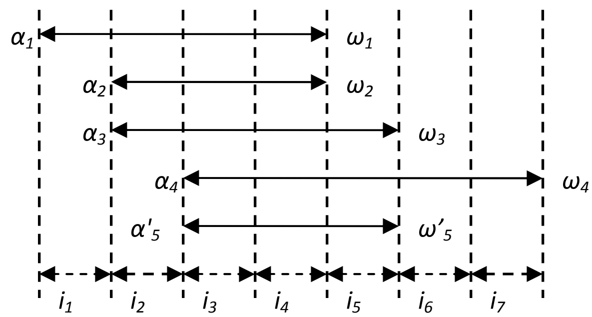

Figure 4.3: Demand scheduling under the sliding window model.

In general, by appropriately choosing the routes (and possibly the start times) of the scheduled demands, we can ensure that the number of active lightpaths at any given time is minimized, leading to considerably reduced energy requirements. We note that our approach does not require reconfiguring the optical switches *etc.*, since the lightpaths are not *rerouted*. The ports for an existing lightpath are simply turned off when there is no traffic on the lightpath.

4.1 Network Model and Problem Definition

In this section, we first consider the case of allocating resources for a *single* new demand, when there are a number of existing demands already routed over the network. We are given the routing scheme for the set of successfully allocated demands over a *logical* topology represented by a directed graph *G(N, L)*, where *N* is the set of nodes and *L* is the set of logical edges (lightpaths). The goal is to find a route for the new demand over the specified logical topology, such that the amount of additional energy consumption due to the new demand is minimized. A preliminary version of this approach for the fixed window model only has been introduced in [88]. At any given time a lightpath $l \in L$ is *active* if it is carrying some non-zero traffic. The router ports of an inactive lightpath may be switched off to reduce energy consumption. Therefore, when finding a route for the new demand, the primary goal is to reduce the number of lightpaths that were previously inactive and must become active to accommodate the new demand. We use the following notation in describing our ILPs:

- P_{static}: Traffic independent portion of power used for a lightpath.
- $P_{dynamic}$: Additional power consumed by a lightpath for each traffic unit carried.
- Q_e: The set of existing demands that have already been routed over the logical topology.
- D_i: Duration of interval i.
- i_{max}: The total number of disjoint time intervals in the network.
- L_i: A set of active lightpaths during interval i (before routing the new demand).
- $o(l)$ $(e(l))$: Originating (terminating) node for lightpath $l \in L$.
- C: Capacity of a *single* lightpath in *OC-n* notation. In this thesis, we have used $C = OC\text{-}160$.
- $X_{q,i}$: A parameter given as input to the ILP and set to 1 if demand q is active during interval i.

- $Y_{q,l}$: A parameter given as input to the ILP and set to 1 if an *existing* demand q is routed over lightpath l.
- Q: A set of scheduled sub-wavelength traffic demands whose setup and teardown times can vary within a larger specified time window. Each element $q \in Q$ is represented as $(s_q, d_q, b_{q,} \alpha_q, \omega_q, \tau_q)$. For the fixed window model, we assume the duration of the window is the same as the demand holding time.
- s_q (d_q): Source (destination) node of demand q.
- b_q: Bandwidth requirement of demand q in *OC-n* notation. We assume that the data rate of the individual demands varies between *OC-3* to *OC-24*, and is always less than the capacity of a lightpath.
- α_q (ω_q): The start (end) time of the larger window of demand q.

4.2 An ILP for Fixed-window Scheduled Demands (ILP-FSD)

In this section, we present an ILP for routing the new scheduled demand under the fixed window model, where the demand holding time and start time are known in advance and given as input to the ILP. We define the following binary variables:

- d_l = 1, if and only if new demand is routed over lightpath l, 0 otherwise.
- $v_{l,i}$ = 1, if and only if lightpath l is active during interval i, 0 otherwise.

The ILP formulation given below calculates the optimal routing for the new scheduled demand in such a way that the additional energy consumption during its active intervals is minimized.

$$\boldsymbol{Minimize} \sum_{i=\alpha_q}^{\omega_q} P_i \cdot D_i \tag{4.1}$$

Subject to:

$$\sum_{l:o(l)=n} d_l - \sum_{l:e(l)=n} d_l = \begin{cases} 1, if\ n = s_q \\ -1, if\ n = d_q \\ 0, otherwise \end{cases} \qquad \forall n \in N \tag{4.2}$$

$$v_{l,i} = 1 \qquad \forall l \in L_i, \forall i, \alpha_q \le i \le \omega_q \tag{4.3}$$

$$b_q \cdot d_l \cdot X_{q,i} + \sum_{q' \in Q_e} b_{q'} \cdot Y_{q',l} \cdot X_{q',i} \le C \cdot v_{l,i} \qquad \forall l \in L, \forall i, \alpha_q \le i \le \omega_q \tag{4.4}$$

$$d_l \le v_{l,i} \qquad \forall l \in L, \forall i, \alpha_q \le i \le \omega_q \tag{4.5}$$

$$P_i = \sum_{l \in L - L_i} v_{l,i} \cdot P_{static} + \sum_{l \in L} b_q \cdot X_{q,i} \cdot P_{dynamic} \cdot d_l \qquad \forall i, \alpha_q \le i \le \omega_q \tag{4.6}$$

The objective function (4.1) minimizes the weighted sum of the total energy consumption over all intervals. Since the demands are processed one at a time, we only consider the active intervals of the new demand currently being allocated. Clearly, it is more advantageous if a lightpath remains inactive during a longer interval compared to a shorter interval. Therefore, the ILP uses the duration (D_i) of an interval i as the *weight* of the energy consumption during that interval. Constraint (4.2) is the standard flow equation [85], and is used to route the new demand over the logical topology using a single multi-hop logical path. Constraint (4.3) states that any lightpath that was active during interval i, before considering the new demand, will continue to remain active during interval i. Constraint (4.4) ensures that the total flow on a lightpath, during any active interval, does not exceed its capacity C. The first term of constraint (4.4) represents any additional traffic load on lightpath l, if the new demand q is routed over l during its active interval i. The second term of constraint (4.4) is the existing load on the lightpath l during interval i due to the demands that have already been allocated. Constraint (4.5) states that if the new demand is routed over lightpath l, then l must remain active for the entire duration of the demand. Finally, (4.6) calculates the *additional* power consumption for each interval i during which the new demand is active. The first term represents the static power consumption by all new lightpaths that must be activated to accommodate the new demand during interval i. The second term calculates the additional dynamic power consumption on each lightpath (both newly activated and existing active lightpaths) due to the new demand.

4.3 An ILP for Sliding-window Scheduled Demands (ILP-SSD)

In this section, we present an ILP for jointly scheduling a demand in time and routing the newly scheduled demand. In the sliding window scheduled traffic model, the new demand is to be scheduled within the specified larger window and its actual start time must be calculated by the ILP. For ILP-SSD we assume that the total time interval is divided into a number of intervals of *equal duration*. This means that the value of D_i is the same for all intervals, so the holding time τ_q of a demand q can be expressed as the number of intervals for which the demand should be active, i.e., $0 < \tau_q \le \omega_q - \alpha_q + 1$ (fractional values are rounded up). As mentioned previously, the actual duration for each interval (i.e., the value of D_i) can be chosen as large or as small as needed for a particular problem.

In addition to the routing variables d_l and $v_{l,i}$, we introduce the following *binary* variables for demand scheduling:

- $st_i = 1$, if and only if a new demand starts during interval i, 0 otherwise.
- $m_i = 1$, if and only if a new demand is active during interval i, 0 otherwise.

We also define the following *continuous* variable. Similar to Section 3.1, a continuous variable is restricted to take on binary values only by the corresponding constraints.

- $\rho_{l,i}$ = 1, if and only if a new demand is routed over lightpath l during interval i, 0 otherwise.

$$\textbf{\textit{Minimize}} \sum_{i=\alpha_q}^{\omega_q} P_i \cdot D_i \tag{4.7}$$

Subject to:

$$\sum_{l:o(l)=n} d_l - \sum_{l:e(l)=n} d_l = \begin{cases} 1, if\ n = s_q \\ -1, if\ n = d_q \\ 0, otherwise \end{cases} \qquad \forall n \in N \tag{4.8}$$

$$b_q \cdot \rho_{l,i} + \sum_{q^{'} \in Q_e} b_{q^{'}} \cdot Y_{q^{'},l} \cdot X_{q^{'},i} \le C \cdot v_{l,i} \qquad \forall l \in L, \forall i, \alpha_q \le i \le \omega_q \tag{4.9}$$

$$\sum_{i=\alpha_q}^{\omega_q} st_i = 1 \tag{4.10}$$

$$m_{i+j} \ge st_i \qquad \forall i, 0 \le j < \tau_q \tag{4.11}$$

$$\sum_{i=\alpha_q}^{\omega_q} m_i = \tau_q \tag{4.12}$$

$$m_i + d_l - \rho_{l,i} \le 1 \qquad \forall l \in L_i, \forall i, \alpha_q \le i \le \omega_q \tag{4.13}$$

$$d_l \ge \rho_{l,i} \qquad \forall l \in L, \forall i, \alpha_q \le i \le \omega_q \tag{4.14}$$

$$m_i \ge \rho_{l,i} \qquad \forall l \in L, \forall i, \alpha_q \le i \le \omega_q \tag{4.15}$$

$$P_i = \sum_{l \in L} v_{l,i} \cdot P_{static} + \sum_{l \in L} b_q \cdot \rho_{l,i} \cdot P_{dynamic} + \sum_{l \in L} \sum_{q' \in Q_e} b_{q'} \cdot Y_{q',l} \cdot X_{q',i} \cdot P_{dynamic}$$

$$\forall i, \alpha_q \le i \le \omega_q \qquad (4.16)$$

The objective function (4.7) and flow constraint (4.8) for ILP-SSD are the same as those for ILP-FSD. Similar to constraint (4.4), constraint (4.9) is used to enforce the capacity constraint and makes sure that the total traffic load on a lightpath *l* during any active interval *i* is less than or equal to the capacity of a lightpath *C*. Constraints (4.10) - (4.12) are used to schedule the demand within the specific window. Constraint (4.10) indicates that exactly one interval can be designated as the "starting interval" for the new demand. Constraint (4.11) specifies that the new demand must be active during its starting interval and for the next τ_q-*1* consecutive intervals. Constraint (4.12) makes sure that the total number of active intervals for the new demand is equal to its demand holding time τ_q. The value of the continuous variable $\rho_{l,i}$ is defined by constraints (4.13) - (4.15). It is set to 1 if and only if the new demand is routed over lightpath *l* and is active during interval *i*. In other words, constraint (4.13) sets $\rho_{l,i}$ = 1 if and only if both d_l = 1 and m_i = 1. If either d_l = 0 or m_i = 0, then constraints (4.14) and (4.15) set $\rho_{l,i}$ = 0. Finally, constraint (4.16) calculates the total power consumption for each interval within the larger window. The first term represents the static power consumed by all active lightpaths during interval *i*. The second term calculates the additional dynamic power consumption due to the new demand. The third term denotes the dynamic power consumption for the existing demand(s), already ongoing during interval i.

4.4 Heuristic for Energy Efficient Routing

In this section, we outline our *heuristic for energy efficient routing* (H-EER) of a given set (Q) of scheduled sub-wavelength traffic demands. For the fixed window model, we assume that the duration of the window is the same as the demand holding time ($\tau_q = \omega_q - \alpha_q$). For the sliding window model, we increase the window size around this fixed window. We assume that the data rate of the individual demands varies between *OC-3* to *OC-24*, and is always less than the capacity of a lightpath.

The proposed heuristic attempts to route each individual low speed traffic demand onto high capacity lightpaths in such a way that both the static and dynamic components of power consumption during any interval are minimized. The demands are added one at a time using the formulations presented in Sections 4.2 and 4.3. Although each demand is allocated optimally based on the current information, the order in which the demands are processed can have a considerable effect on the performance of the heuristic. Therefore, the entire heuristic is divided into two phases; the first phase (lines 1-7) determines the order in which the demands are processed and the second phase (lines 8-12) finds an energy efficient routing for each demand one at a time. We assume that actual connection setup/teardown is implemented by appropriate control plane protocols, which are outside the scope of this thesis.

In PHASE 1, an initial ordering the demands (Q_{sort}) is obtained (line 2) by sorting the demands based on one of the following criteria:

- bandwidth requirement,
- source node ID,
- demand holding time,
- demand start time, or
- randomized order.

Next, we partition the entire time period into a number of consecutive time intervals, $i_1, i_2, i_3, \dots i_{max}$ as depicted in Fig. 4.2 and sort the intervals by their duration in a descending order (line 4). The idea is that the demands that are active during longer intervals should be processed first, since they will have a greater impact on the total energy consumption. The demands active during each interval (longest to shortest) are then selected for processing according to the specified criteria used to sort the demands (lines 6-7). Once the order for processing the demands has been determined, ILP-FSD (ILP-SSD) is used to find an optimal routing for the current demand q under the fixed (sliding) window model, based on the set of all demands that have been previously allocated Q_e. Once a successful routing is found (line 10), the current demand q is added to the set of allocated demands (line 11) and the set of active lightpaths for each interval (G_{min}) is updated (line 12)[1].

Heuristic: H-EER

Input: $G = (N, L)$ and a scheduled sub-wavelength traffic demands set Q

Value returned: Set of active lightpaths G_{min} in each interval and an appropriate traffic routing scheme for Q.

1. PHASE 1: Determine Ordering
2. $Q_{sort} \leftarrow$ demands in Q sorted by a selected criterion.
3. Partition the entire time period into a number of consecutive time intervals $i_1, i_2, i_3, \ldots i_{max}$..
4. $I_{sort} \leftarrow$ intervals sorted by their duration D_i in a descending order.
5. For each interval $i \in I_{sort}$
6.   For each demand $q \in Q_{sort}$ do
7. $If \quad q \notin Q_{order} \quad and \quad X_{q,i} = 1, \qquad then \quad Q_{order} = Q_{order} \bigcup \{q\}$.
8. PHASE 2: Determine Routes
9. For each demand $q \in Q_{order}$ do
10.   Use ILP-FSD (or ILP-SSD) to find optimal routing for q.

[1] We assume that the network has sufficient resources to accommodate all demands. If this is not the case the heuristic can be easily modified to accommodate as many demands as possible.

11. $Q_e = Q_e \bigcup \{q\}$.

12. Update $G_{min.}$

13. return $G_{min.}$

4.5 Experimental Results

We have tested our algorithm with different demand sets on a number of networks ranging from a small 6-node network [56] to practical sized networks including the 14-node NSFNET and 20-node ARPANET topologies [86]. The logical topologies were constructed based on the approach outlined in [58]. We calculated the average number of active lightpaths in each interval and the total energy consumption (both static and load dependent components) for different sets of scheduled demands with varying amounts of overlap among the demands. The computation time ranged from less than 1*s* for the smaller networks to several minutes for the 20-node network.

We classified the demand sets as having low, medium, and high demand overlap (referred to as LDO, MDO, and HDO, respectively), as defined by the demand overlap factor (δ) in Section 3.6.1 [14] for each demand set. The source and destination for each demand were randomly generated, with each node having an equal probability of being selected. The start time of each demand was also randomly generated. The end time (and consequently the duration) of a demand was determined based on δ. Since the start time for each individual demand in each set was randomly generated, so it is possible that several demands start simultaneously. For each case the results were compared with the full demand overlap (FDO) case, which corresponds to $\delta = 1$. The FDO model was simulated in our experiments by setting the duration of every demand to the full 24-hour period. For the higher values of δ, the results approach the FDO case. In our simulations, the traffic loads (set of demands) for the different values of δ are the same; the only difference is in the demand holding time.

4.5.1 Results for Fixed Window Model

The ILP formulation (ILP-FSD) could easily generate optimal solutions for a single incoming traffic demand in 14-node and 20-node networks with over 800 existing demands using ILOG CPLEX solver [87]. For a given set of demands, we have investigated three options for performing traffic routing:

i) Select a route that minimizes energy only for the current interval (energy-aware holding time unaware (HTU) case).

ii) Greedily select one demand at a time (based on arrival times) and use ILP-FSD to find the minimum-energy route over all intervals for each demand.

iii) Jointly route the set of demands to minimize the overall energy, using H-EER.

For all three options, resources allocated to a demand are released after the demand terminates. Our simulations indicate that additional improvements, over both energy-aware HTU and greedy selection, can be achieved by jointly allocating the set of scheduled demands using our proposed H-EER approach. Figure 4.4 compares the number of active lightpaths (and thus the power consumption, which closely follows the number of active lightpaths) for a 20-node network with over 800 demands. Holding time aware (HTA) approaches (i.e., options ii and iii) clearly outperform the HTU approach, particularly for LDO and MDO cases. As demand overlap increases, knowledge of holding times becomes less important, and there is no significant difference between options i) and ii). However, H-EER is able to reduce the number of active lightpaths by about 20% for the MDO and HDO cases. The results follow a similar pattern for other networks. This indicates that the order in which the demands are processed has a significant impact on the overall energy consumption, particularly for large demand overlaps, and knowledge of demand holding times plays a more significant role for low demand overlaps. Since joint allocation clearly reduces the overall energy consumption, we have reported those results in the rest of this section. Unless explicitly stated otherwise, the sorting criterion used in PHASE 1 of H-EER is based on the demand bandwidth.

Figure 4.5 shows the percentage reduction in total power consumption and the average number of active lightpaths compared with Energy-aware HTU case for a 10-node network, when sorting the demands according to the bandwidth requirement. As shown in Fig. 4.5, significant energy savings (10% to > 50%) can be achieved by utilizing knowledge of the demand holding times. The improvements over the HTU model increases steadily as the demand overlap factor decreases. We also note that the total power consumption is closely related to the number of active lightpaths. This makes sense because in our model the static component of the power consumption dominates the load dependent component [20].

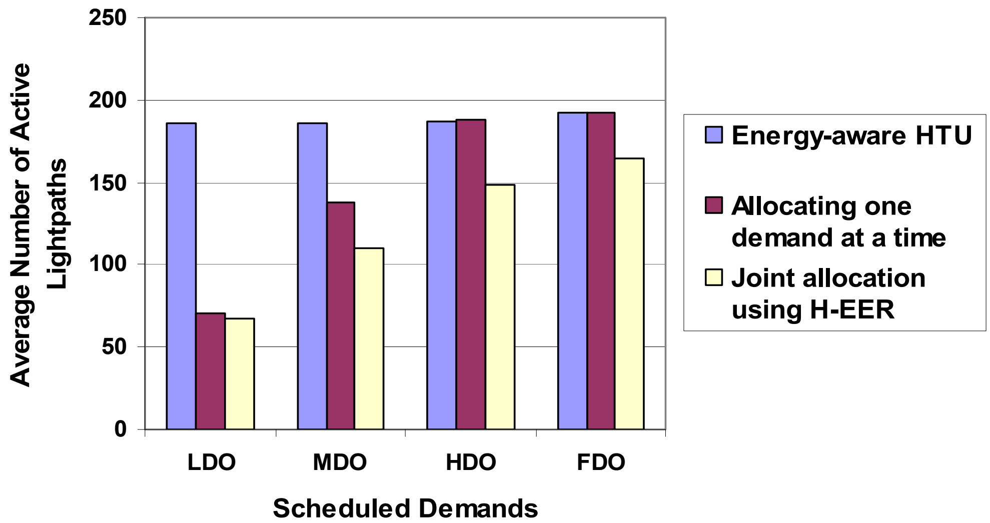

Figure 4.4: Comparison of the average number of active lightpaths used for single and joint demand allocation.

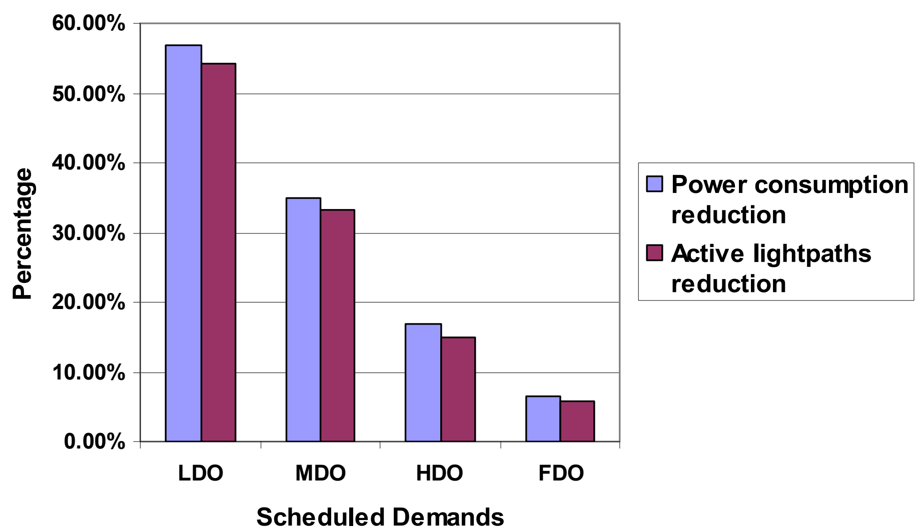

Figure 4.5: Percentage reduction in total power consumption and the average number of active lightpaths compared with Energy-aware HTU case for a 10-node network.

Figure 4.6 shows the overall energy consumption for a set of scheduled demands, routed over a 10-node network. We compare the energy consumption for our proposed H-EER algorithm for all four cases, i.e., LDO, MDO, HDO, and FDO, with

i) a traditional shortest path traffic grooming algorithm, which does not consider energy consumption, and

ii) an energy-unaware, holding time aware approach, which is obtained by modifying the objective function of our ILP to minimize the number of lightpaths, rather than the energy consumption.

The results are normalized to the FDO case. It is clear that H-EER significantly outperforms the other two approaches for all four cases, ranging from improvements of 15% for the FDO case to 40% for the LDO case. It is interesting to note that there is no significant difference between the simple shortest path routing and energy-unaware HTA approaches because typically the latter approach ends up selecting the shortest path over the logical topology. We also note that the total power consumption obtained using H-EER, even for the FDO case, is actually lower than the power consumption for both MDO and HDO cases using energy-unaware traffic grooming. This clearly illustrates the importance of energy-aware resource allocation schemes and highlights their significant impact on overall network power consumption.

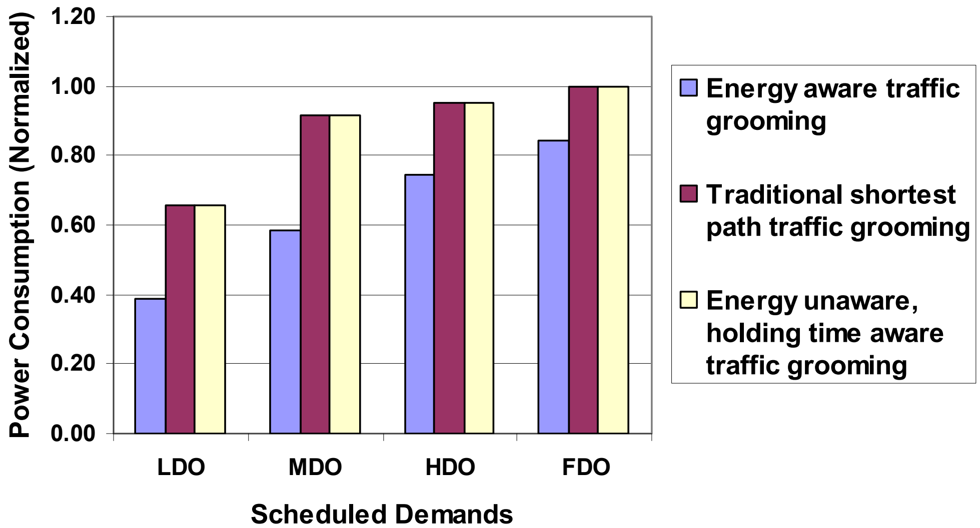

Figure 4.6: Comparison of total power consumption for FDO case, traditional traffic grooming and H-EER for a 10-node network.

Figure 4.7 shows the amount of resources needed to accommodate a given set of scheduled demands in terms of the average number of active lightpaths for different network sizes and different amounts of demand overlap when sorting the demands according to the source nodes. As expected, the amount of resources needed increases as the network size and the demand overlap factor increase. It is also interesting to note that consideration of connection holding times seems to have a greater impact for larger networks.

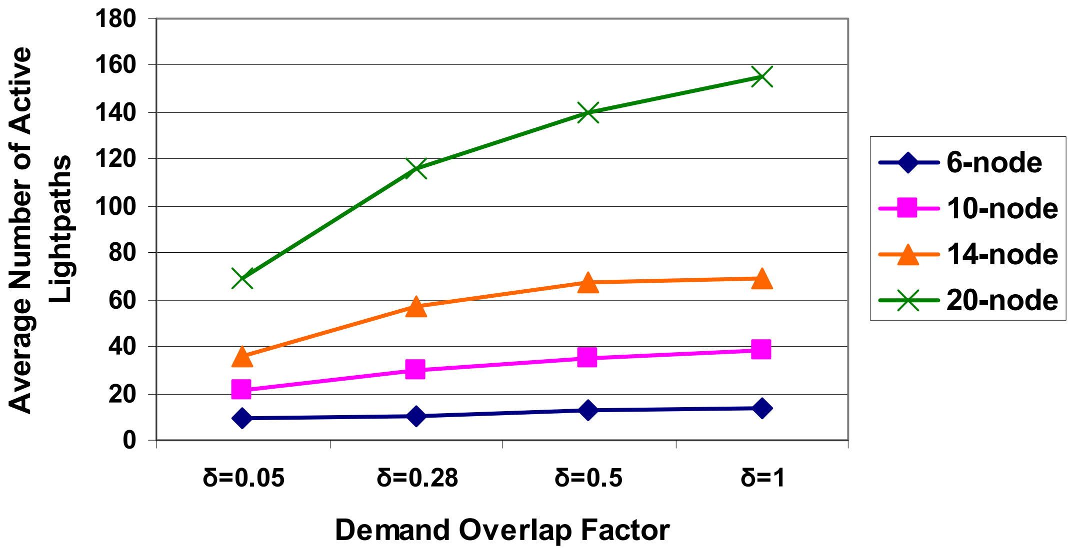

Figure 4.7: Comparison of the average number of active lightpaths used for different topologies.

Figure 4.8 shows a comparison of the average number of active lightpaths needed for 20-node ARPANET topology when using different sorting criteria (including random selection of demands) for ordering the demands. In general, we observed that the results for the different criteria were close for $\delta = 0.05$ (LDO case). However, as δ increases, the effect of ordering becomes more and more important. For $\delta = 0.5$ (HDO case), sorting by the source node performs better and results in almost 20% improvement compared to sorting by start time or random ordering.

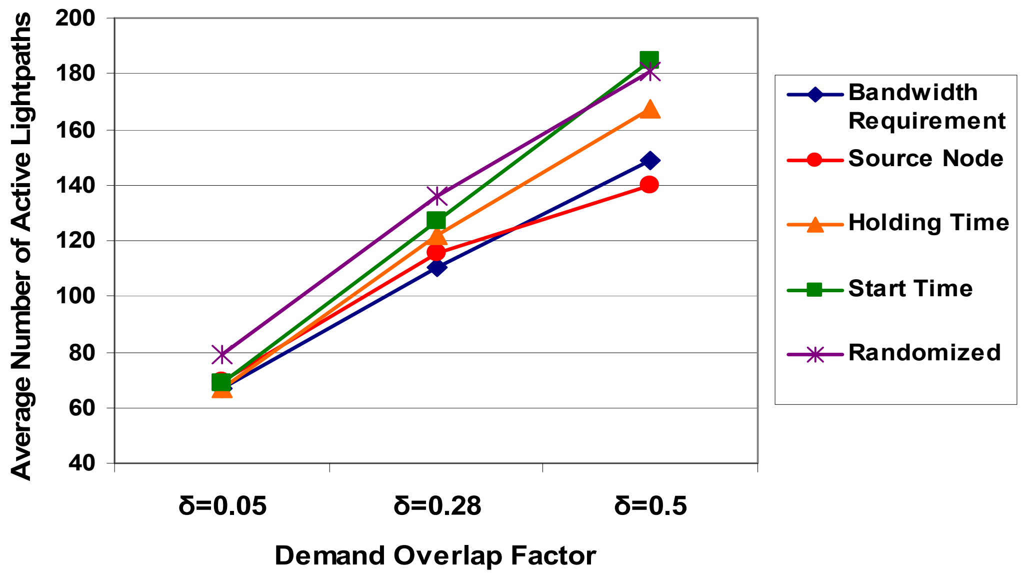

Figure 4.8: Comparison of the average number of active lightpaths used for different sorting criteria.

4.5.2 Results for Sliding Window Model

In this subsection, we show how adding flexibility to the demand setup/teardown times benefits the network utilization. In order to study the effect of demand scheduling, we extended the window for each demand by two, four, and six hours so that the demand could be scheduled any time within this larger window. Since the fixed window model clearly outperforms the FDO model for scheduled demands as shown in Section 4.5.1, from now on, we will compare our heuristic using ILP-SSD to the *fixed window* model.

Figure 4.9 shows the percentage reduction in total power consumption compared to the fixed window model for the LDO case when sorting the demands according to the bandwidth requirement. It is clear that the proposed approach can lead to additional improvements, depending on the window size, even over the holding-time-aware models.

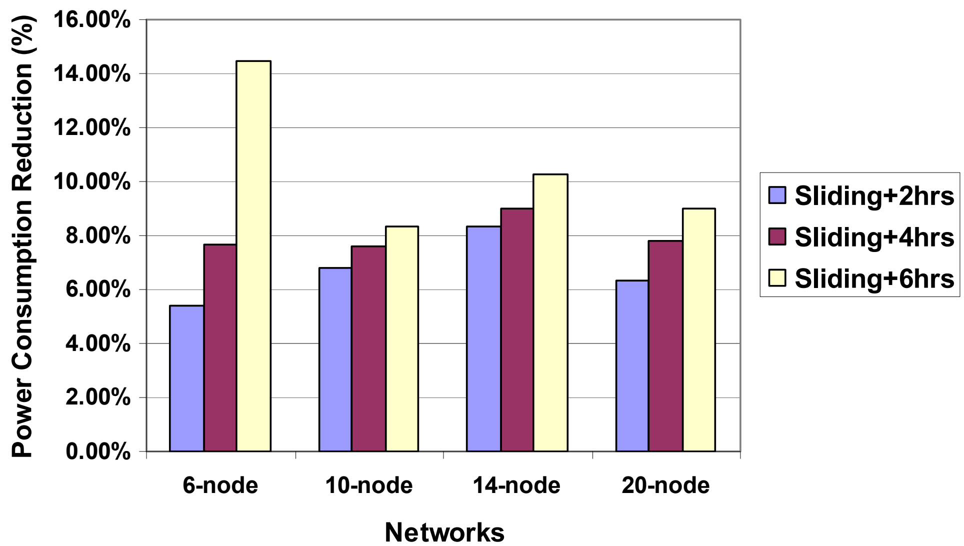

Figure 4.9: Percentage reduction in total power consumption compared with the fixed scheduled traffic model.

Figure 4.10 shows the effect of window size on resource utilization in terms of the average number of active lightpaths used for different topologies when the same sorting criterion is used as in Fig. 4.9. We see that as the window size increases, the amount of saving increases a little bit initially and then levels off. Additional saving over the fixed window model of about 9% to 15%, depending on the network size, can be obtained by extending the window size by six hours. This is because a demand has more flexibility when it is to be scheduled within a larger window.

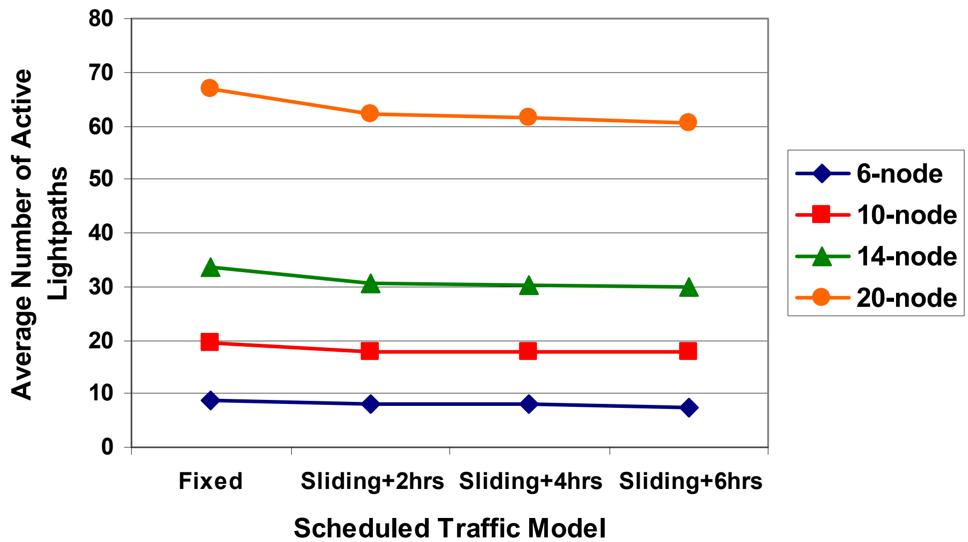

Figure 4.10: Effect of increasing window size for different topologies.

For the HDO demand set, most of the demands overlap in time, and it is difficult to move demands around. Hence, it is possible that some of the demands can not be accommodated by the given logical topologies. Figure 4.11 shows the comparison of the amount of dropped traffic in a 14-node network under the fixed and sliding scheduled traffic models. If we only consider the *number* of dropped demands, it appears that no improvement is achieved by using the sliding window model and increasing the duration of the allowable window for each demand. However, when we consider the total dropped bandwidth, we see that larger windows result in less dropped traffic and, hence, higher throughput. This means that the fixed window model tends to drop more high bandwidth demands compared to the sliding model. Thus, joint scheduling (in time) and routing improves overall throughput, even compared to energy aware approaches that only consider demand holding times.

Figure 4.11: Comparison of traffic dropped under the different scheduled traffic model.

4.6 Summary

The main contributions of this chapter are:

- We propose an efficient ILP formulation for energy aware grooming of a single new sub-wavelength demand under the *fixed* scheduled traffic model.

- We extend this formulation to solve the same problem under a *sliding* scheduled traffic model, and jointly perform scheduling and routing.

- We demonstrate through simulations that the proposed ILPs can be used for practical networks with hundreds of existing demands and lead to significant reductions in power consumption.

- We present an integrated heuristic that jointly allocates resources for a large set of scheduled demands and leads to significant *additional* improvements in power consumption over those already achieved using our initial formulations.

Chapter 5

A Segmented Scheduled Traffic Model

In this chapter, we introduce a new model, the *segmented sliding window model*, where a demand may be decomposed into two or more components and each component can be sent separately. The segmented scheduled traffic model adds another degree of flexibility to the existing sliding window model, which can be exploited to generate more resource efficient solutions to the network design problem, or to accommodate more traffic for a given set of resource constraints. We provide an ILP formulation as well as a heuristic for solving this problem.

We note that one significant aspect of the cost associated with dynamic setup and teardown of lightpaths is the need for reconfigurable optical switching equipment, which is more expensive compared to non-reconfigurable equipment. However, even traditional scheduled traffic models (fixed and continuous sliding window) require the use of reconfigurable switches. Therefore, we do not incur additional switching equipment cost by using the proposed model, as compared to existing fixed and sliding window models. In addition to switching equipment costs, the segmented model will likely require the use of a somewhat more complex control plane, which is capable of making intelligent segmentation decisions, scheduling each demand (or segment) in time and allocating network resources. In this context, it is important to keep in mind that such design decisions are performed *offline*, since the requirements for each demand is assumed to be known in advance. Therefore, it is quite feasible to use a centralized agent to compute the optimal allocation and then transmit the necessary information to the appropriate network nodes. Finally, reassembling of demands with multiple segments will also require some extra overhead such as additional buffering at the destination node. However, our main focus in this work is on efficient utilization of bandwidth resources in the fiber links, so we do not consider the additional costs at higher layers. It is also important to note that there are a number of applications where decomposing demands into multiple segments may not be acceptable. Our proposed model can easily accommodate such demands as a special case of our formulation, and the techniques for doing this are discussed in Section 5.5.2.

5.1 Network Model and Problem Definition

We consider a physical fiber network *G[N, E]*, where *N* is the set of nodes, *E* is the set of edges (fiber links) in the physical topology, and each edge $e \in E$ can accommodate a set *K* of distinct WDM channels. We are given, as input, the physical topology of the network, the capacity of each fiber link, and a set *Q* of demands that have to be routed over the physical topology. Each element $q \in Q$ is represented as $(s_q, d_q, b_{q,} \alpha_q, \omega_q, \tau_q)$. Based on this information, our goal is to generate a set of demands Q_{alloc} such that all demands in Q_{alloc} are accommodated in the network and the weighted sum of the demands in Q_{alloc} is maximized. Based on the objective under consideration, different policies can be used to assign weights to the demands, and is discussed in Section 5.3. In addition, a physical route and an assigned wavelength for each lightpath associated with an allocated demand must be determined. It has been shown in the literature that just the static RWA problem by itself, considering only a single interval, is NP-hard [89]. Since, our formulation performs both scheduling and RWA, it contains the RWA problem as a subset, hence it is also NP-Hard. Therefore, we present an ILP formulation for solving the above problem. A set of potential routes for each demand can also be given as input, in order to reduce the complexity of the ILP.

We note that in our formulations, we find a set of *R* edge-disjoint routes, if possible, over the physical topology for each pair of end-nodes. The idea is that if certain links of a path are congested, these links would be automatically avoided in an alternate path that is edge-disjoint. For the fault-free case, this is only a preference, rather than a requirement. However, for implementing path protectin (discussed in Section 5.5.1) it is necessary that the routes be edge-disjoint, otherwise a failure in a common link may cause both the primary and backup routes to become unusable. If for a given topology, we can only find r_{sd} ($r_{sd} < R$) edge-disjoint pahts for node pair *s*-*d*, then we only use these r_{sd} routes for that partivular node pair. Of course we must have at least $r_{sd} \geq 2$ in order to implement path protection. The rationale for pre-computing up to R routes is that it leads to a reduction in the complexity of the ILP, without any significant degradation of performance. This approach has been widely used in the literature for routing of lightpaths [9, 26, 32], and we have also adopted this approach. If a lightpath is to be established from a source node s to a destination node d, we select one of the R routes between s and d as a physical route for the lightpath. In our experiments, we have used R =3.

The objective of our ILP formulation for scheduling segmented demands is to accommodate as many demands as possible under specified network resource constraints. We divide the entire time period into a sequence of *intervals* (i) of equal duration. If a demand is active for any amount of time within an interval, it is assumed to be active during the entire interval. Such a slotted-time model can lead to somewhat decreased flexibility, particularly if the durations of the individual time-slots are much longer compared to the durations/intervals of the lightpath demands. However, we allow the duration of an interval to be selected by the designer, based on expected traffic patterns and made as coarse or as fine as desired. This helps to minimize any adverse effects on the quality of the solutions due to the slotted-time approach. Clearly, a finer granularity allows greater control at the expense of increased computation. For example, Fig. 1.3 shows 6-hour time period divided into 6 intervals, each of 1 hour duration. In our experiments (Section 5.7), we have set each interval to 15 minutes and the entire time period to be 12 hours or 24 hours. Since the reconfiguration times for the optical switches are insignificant (generally in the range of tens of milliseconds) compared to the interval durations and demand holding times, we have ignored the reconfiguration times in our simulations.

The formulation presented here allocates each demand within its time window, separating them into segments if necessary, and performs RWA for each demand that is routed over the network. The different segments are still treated as part of a single demand, rather than a series of independent demands. While it is possible to consider the smaller components of a demand to be completely independent, it will mean that there would be no mechanism to limit the number of segments, the routes taken by individual segments, or impose other restrictions such as ensuring segments are delivered in a proper order. This may indeed be acceptable for certain applications and in such cases, a single demand can be decomposed into an appropriate number of individual segments (an increased number of segments allow increased flexibility in scheduling, but may require more complex processing at the destination node) and given as separate inputs to the ILP. However, we assume that for most applications it is desirable to have some level of control over the number of segments, routes over the physical topology etc. In such cases, simply viewing a demand as a collection of independent components, under the continuous model, will not be sufficient and the proposed segmented model will be required. In this context, we also note that for certain applications, it may be possible to have concurrent transmission of smaller segments of a demand. However, this will exacerbate the problem of jitter and out-of-order delivery of segments, and more complex control mechanisms will be required at the destination node for reassembling the different segments. Therefore, we have not considered this option in this chapter.

5.2 Notation

In our ILP formulations, we will use the following notation for input data:

- E: The set of directed edges in the physical topology, each edge representing a fiber in the network.
- N: The set of end-nodes in the network.
- K: A set of channels that each fiber can accommodate.
- F_q : The priority of demand q.
- Q_c: The set of all *continuous* traffic demands, *i.e.* demands that cannot be divided into multiple segments.
- Q_n: The set of all *segmented* traffic demands, *i.e.* demands that may be divided into multiple segments, if necessary.

- Q: The set of all traffic demands. Each element $q \in Q$ is represented as $(s_q, d_q, b_{q,} \alpha_q, \omega_q, \tau_q)$. We note that $Q = Q_c \bigcup Q_n$.
- s_q (d_q): Source (destination) node of demand q.
- b_q: The number of lightpaths required by demand q. Here, we consider traffic demands with a coarse granularity (corresponding to the number of lightpaths to be routed over the network).
- α_q (ω_q): The start (end) time of the larger window of demand q during which the demand must be met, expressed in terms of the first (last) interval during which the demand may be active.
- τ_q: The holding time of demand q, expressed in terms of the number of intervals during which the demand is active (fractional values are rounded up) $0 < \tau_q \leq \omega_q - \alpha_q + 1$.
- R: A set of pre-computed edge-disjoint routes, over the physical topology to be considered for RWA, between each ordered pair of end-nodes.
- $H_{r,q}^{e}$: A matrix where each cell is set to 1 if the r^{th} physical route of demand q from s_q to d_q includes fiber link e, 0 otherwise.

We define a number of binary variables as follows:

- $a_{q,i} = 1$, if and only if demand q is active during the interval i, 0 otherwise.

- $x_{r,q}$ ($z_{r,q}$) =1, if and only if the r^{th} physical route is selected to route the primary (backup) lightpath for demand q, 0 otherwise.
- $\gamma^{i}_{k,e,q}$ ($\delta^{i}_{k,e,q}$) = 1, if and only if channel k on edge e is assigned to the primary (backup) lightpath for demand q during the interval i, 0 otherwise.
- $p^{i}_{k,q}$ ($b^{i}_{k,q}$) = 1, if and only if channel k is assigned to the primary (backup) lightpath for demand q during the interval i, 0 otherwise.
- y_q = 1, if and only if demand q is accommodated, 0 otherwise.

We also define the following *continuous* variables. These continuous variables will be constrained to take values of 0 or 1 only, as explained in Section 5.4.

- $\sigma^{i}_{e,q}$ ($\lambda^{i}_{e,q}$) = 1, if and only if the primary (backup) lightpath for demand q uses edge e during the interval i, 0 otherwise.

The variables $z_{r,q}$, $\delta^{i}_{k,e,q}$, $b^{i}_{k,q}$, and $\lambda^{i}_{e,q}$, associated with backup lightpaths, are not needed for the fault-free formulation (ILP-S). They are used in ILP-DP (Sec. 5.5.1) to implement path protection. We also note that in our formulations $\sigma^{i}_{e,q}$ and $\lambda^{i}_{e,q}$ are the only *continuous* variables, and all other variables are *binary* variables.

5.3 ILP Formulation for Segmented Demands (ILP-S)

The formulation presented here allocates each demand within its time window, separating them into segments if necessary, and performs RWA for each demand that is routed over the network. The different segments are still treated as part of a single demand, rather than a series of independent demands.

$$\textbf{\textit{Miximize}} \sum_{q \in Q} y_q \tag{5.1}$$

If a demand q is accommodated, then $\boldsymbol{y_q}=1$. Hence, the objective function (5.1) maximizes the the number of demands that can be accommodated in the network, by maximizing $\sum_{q \in Q} y_q$. It is also possible to assign a *weight* to each demand q, either in terms of its bandwidth requirement b_q, or a preassigned priority level F_q. In that case the objective function would be changed to Maximize $\sum_{q \in Q} b_{q} \cdot y_q$ or Maximize $\sum_{q \in Q} F_{q} \cdot y_q$.

Subject to:

a) Demand holding time constraints

$$\sum_{i, \alpha_q \le i \le \omega_q} \alpha_{q,i} = y_q \cdot \tau_q \qquad \forall q \in Q \tag{5.2}$$

$$\alpha_{q,i} = 0 \qquad \forall q \in Q, \forall i, i < \alpha_q, \forall i, i > \omega_q \tag{5.3}$$

Eq. (5.2) ensures that the sum of the *active* intervals for each demand q equals to its holding time. A demand is allocated resources only if it is accommodated (i.e.,

y_q=1). Eq. (5.3) guarantees that no interval is active for demand q outside its larger window.

b) Route assignment for primary lightpaths

$$\sum_{r,0\leq r<R} x_{r,q} = y_q \qquad \forall q \in Q \tag{5.4}$$

Eq. (5.4) ensures that an accommodated demand is assigned exactly one physical route for its primary lightpath. This means that bifurcated routing of demands will not be allowed, if the above constraint is enforced.

$$\sum_{r,0\leq r<R} x_{r,q} \cdot H^e_{r,q} + \alpha_{q,i} - \sigma^i_{e,q} \leq 1 \qquad \forall q \in Q, \forall e \in E, \forall i, \alpha_q \leq i \leq \omega_q \tag{5.5}$$

$$\sum_{r,0\leq r<R} x_{r,q} \cdot H^e_{r,q} \geq \sigma^i_{e,q} \qquad \forall q \in Q, \forall e \in E, \forall i, \alpha_q \leq i \leq \omega_q \tag{5.6}$$

$$\alpha_{q,i} \geq \sigma^i_{e,q} \qquad \forall q \in Q, \forall e \in E, \forall i, \alpha_q \leq i \leq \omega_q \tag{5.7}$$

The variable $\sigma^i_{e,q}$ indicates whether demand q is routed over link e during an interval i. Constraints (5.5) - (5.7) are used to set the value of $\sigma^i_{e,q}$ to either 0 or 1. The first term in (5.5) is set to 1 if the r^{th} route is selected for the primary lightpath(s) for demand q (*i.e.* $x_{r,q} = 1$) and this route includes edge e (i.e. $H^e_{r,q}$ =1). In other words $\sum_{r,0\leq r<R} x_{r,q} \cdot H^e_{r,q} = 1$ if the lightpath(s) for demand q use physical link e and demand q is

active during interval i (i.e., $\boldsymbol{\alpha_{q,i}}$ =1), then (5.5) forces $\sigma_{e,q}^{i}$ to be 1. Otherwise, constraints (5.6) and (5.7) set $\sigma_{e,q}^{i}$ to 0.

c) Wavelength assignment for primary lightpaths

$$\sum_{k \in K} \gamma_{k,e,q}^{i} = b_q \cdot \sigma_{e,q}^{i} \qquad \forall q \in Q, \forall e \in E, \forall i, \alpha_q \leq i \leq \omega_q \quad (5.8)$$

$$\sum_{q \in Q} \sum_{k \in K} \gamma_{k,e,q}^{i} \leq |K| \qquad \forall e \in E, \forall i \quad (5.9)$$

$$\sum_{q \in Q} \gamma_{k,e,q}^{i} \leq 1 \qquad \forall k \in K, \forall e \in E, \forall i \quad (5.10)$$

According to constraint (5.8), there are exactly b_q channels allocated to demand q on edge e during interval i, if the primary lightpath(s) of q are active and use edge e during interval i. Constraint (5.9) ensures that total channels used on a link must not exceed $|K|$. Constraint (5.10) states that a channel k on link e can be assigned to at most 1 demand during an interval i. We note that it is possible for different segments of the same demand to be assigned different wavelengths on the same link e at different intervals. This means that wavelength continuity along different slots (intervals) is not enforced in ILP-S. If it is desired to do this, we simply have to include constraints (5.25) and (5.26) from Section 5.5.2 in ILP-S.

d) Wavelength continuity constraint for primary lightpaths

$$\sum_{k \in K} p_{k,q}^{i} = b_q \cdot \alpha_{q,i} \qquad \forall q \in Q, \forall i, \alpha_q \leq i \leq \omega_q \quad (5.11)$$

$$p_{k,q}^{i} \geq \gamma_{k,e,q}^{i} \qquad \forall q \in Q, \forall k \in K, \forall e \in E, \forall i, \alpha_q \leq i \leq \omega_q \qquad (5.12)$$

Constraints (5.11) and (5.12) together are used to enforce the wavelength continuity constraint. Constraint (5.11) states that exactly b_q channels are allocated to demand q if it is active during interval i ($\alpha_{q,i}$ =1), and no resources are allocated if it is inactive during interval i ($\alpha_{q,i}$ =0). Constraint (5.12) states that if a channel k is assigned to demand q during interval i (i.e., $p_{k,q}^{i}$ =1), then the same channel k must be assigned to q on all physical links e traversed by q, thus enforcing the wavelength continuity constraint.

5.4 Complexity Analysis of the ILP

An ILP is characterized by the number of integer and continuous variables and the number of constraints. Table 5.1 gives the number of integer variables, continuous variables, and constraints in the formulation presented in section 5.3. It has been well established that the complexity of an ILP is exponential in the number of integer variables [90], and typically, this is the primary factor determining if the ILP will be computationally tractable. Therefore, in our formulation, we have tried to reduce the number of integer (binary) variables, by using a technique where we declare a variable (e.g. $\sigma_{e,q}^{i}$) as a continuous variable, but restricts its permissible values to 0 and 1. So, instead of defining $\sigma_{e,q}^{i}$ as a binary variable, we need to carefully formulate the constraints such that $\sigma_{e,q}^{i}$ is only allowed to take on integer values. This is exactly what we have done in constraints (5.5) - (5.7). Clearly, this technique cannot be applied to all binary variables, but where applicable it allows us to reduce the complexity of the ILP by reducing the number of binary variables at the expense of more continuous variables. In the table, we used i_t to indicate the total number of intervals over the entire time period during which the demands may be active.

Formulation	# Integer variables	# Continuous variables	# Constraints
ILP-S	$2\lvert\mathcal{Q}\rvert(\mathbb{R} + i_t\lvert\mathcal{K}\rvert\lvert E_P\rvert)$ $+i_t\lvert\mathcal{Q}\rvert(2\lvert\mathcal{K}\rvert + 1)$ $+\lvert\mathcal{Q}\rvert$	$2i_t\lvert\mathcal{Q}\rvert\lvert E_P\rvert$	$2\lvert\mathcal{Q}\rvert(i_t + 1)+$ $i_t\lvert\mathcal{K}\rvert\lvert E_P\rvert+$ $i_t\lvert\mathcal{Q}\rvert\lvert E_P\rvert(\lvert\mathcal{K}\rvert + 5)$

Table 5.1: Number of variables and constraints in the ILP formulations.

5.5 Extensions to the ILP Formulation

In this section, we discuss a number of useful extensions to the ILP formulation presented in section 5.3. These include constraints for implementing path protection, handling traditional fixed window and continuous sliding window models, as well as a number of other extensions.

5.5.1 Implementing Path Protection (ILP-DP)

The use of path protection techniques [91, 92] has emerged as a widely accepted strategy for providing guaranteed survivability of network traffic in WDM networks. In this approach, for each primary (or working) lightpath an edge-disjoint backup (or protection) lightpath must be established over the network. The resources for the backup lightpaths are reserved in advance during connection setup. Therefore, path protection guarantees that in case of a single link failure, the traffic on the failed lightpath(s) can be re-routed to their destinations over the corresponding backup lightpaths. In dedicated protection, the resources allocated to a backup path cannot be shared with any other primary or backup path. Shared path protection improves resource utilization by allowing resource sharing among two or more backup paths, if the corresponding primary paths are edge-disjoint. In this section, we present a formulation (ILP-DP) that shows how the original formulation (ILP-S) can be extended in order to implement dedicated path protection. Shared protection can also be implemented by adding a few additional constraints to the formulation for dedicated protection, as described in [30]. ILP-DP has the same objective function and RWA constraints for primary lightpaths as ILP-S. Therefore, in this section, we have only listed the *additional* constraints required for the backup lightpaths. The complete formulation for implementing path protection includes the objective function and constraints from ILP-S plus constraints (5.13) – (5.22) given below.

a) Primary and backup lightpaths must be edge-disjoint

$$\chi_{r,q} + z_{r,q} \le 1 \qquad \forall q \in Q, 0 \le r < R \tag{5.13}$$

b) Route assignment for backup lightpaths

$$\sum_{r,0\le r<R} z_{r,q} = y_q \qquad \forall q \in Q \tag{5.14}$$

$$\sum_{r,0\le r<R} z_{r,q} \cdot H^{e}_{r,q} + \alpha_{q,i} - \lambda^{i}_{e,q} \le 1 \quad \forall q \in Q, \forall e \in E, \forall i, \alpha_q \le i \le \omega_q \tag{5.15}$$

$$\sum_{r,0\le r<R} z_{r,q} \cdot H^{e}_{r,q} \ge \lambda^{i}_{e,q} \qquad \forall q \in Q, \forall e \in E, \forall i, \alpha_q \le i \le \omega_q \tag{5.16}$$

$$\alpha_{q,i} \ge \lambda^{i}_{e,q} \qquad \forall q \in Q, \forall e \in E, \forall i, \alpha_q \le i \le \omega_q \tag{5.17}$$

c) Wavelength assignment for backup ligthpaths

$$\sum_{k\in K} \delta^{i}_{k,e,q} = b_q \cdot \lambda^{i}_{e,q} \qquad \forall q \in Q, \forall e \in E, \forall i, \alpha_q \le i \le \omega_q \tag{5.18}$$

$$\sum_{q\in Q}\sum_{k\in K} (\gamma^{i}_{k,e,q} + \delta^{i}_{k,e,q}) \le |K| \qquad \forall e \in E, \forall i \tag{5.19}$$

$$\sum_{q\in Q} (\gamma^{i}_{k,e,q} + \delta^{i}_{k,e,q}) \le 1 \qquad \forall k \in K, \forall e \in E, \forall i \tag{5.20}$$

d) Wavelength continuity constraint for backup lightpaths

$$\sum_{k\in K} b^{i}_{k,q} = b_q \cdot a_{q,i} \qquad \forall q \in Q, \forall i, \alpha_q \le i \le \omega_q \tag{5.21}$$

$$b^{i}_{k,q} \ge \delta^{i}_{k,e,q} \qquad \forall q \in Q, \forall k \in K, \forall e \in E, \forall i, \alpha_q \le i \le \omega_q \tag{5.22}$$

In the above formulation, constraint (5.13) ensures that the routes for the primary and backup paths are edge-disjoint. The remaining constraints, (5.14) - (5.22), for RWA of backup lightpaths are analogous to the corresponding constraints for primary lightpaths, as explained in ILP-S. In particular, constraints (5.15) - (5.17) ensure that resources for backup paths of a demand *q* are allocated only during the active intervals of the demand (i.e., when $a_{q,i}$ =1). Similarly, primary paths are also allocated only during active intervals. Since the total number of active intervals is limited to τ_q, it means that a working path and its corresponding backup path must be created in the same (active) interval(s). Therefore, 100% restorability is achieved during each active interval. Finally, constraint (5.19) states that the total number of primary and backup lightpaths on edge *e* during interval *i* cannot exceed |K|, and constraint (5.20) states that at most one lightpath (either primary or backup) can be assigned to channel *k* on link *e* during interval *i*.

5.5.2 Handling Continuous Demands (ILP-C)

As we have noted in the introduction, although our proposed segmented model allows increased flexibility in scheduling demands, it may not be suitable for all applications. In these cases, either the fixed window or the continuous sliding window model may be more appropriate, and we refer to the corresponding demands as *continuous* demands. Continuous demands can be easily accommodated in our approach with a few additional variables and constraints, as discussed in this section.

For the continuous sliding window model, the start time is not fixed, but once transmission starts all segments must be sent in consecutive intervals, along the same route and set of channels. For this case, we define and add the following binary variables and constraints. In this section, we focus on the constraints for the primary lightpaths, but similar constraints for backup lightpaths can be easily added.

- $st_{q,i}$ = 1, if and only if transmission of demand q starts during the interval i, 0 otherwise.
- $\gamma_{k,e,q}$ = 1, if and only if channel k on link e is assigned to demand q, 0 otherwise.

The following constraints should be added to the ILP-S formulation:

$$\sum_{i,\alpha_q \le i \le \omega_q} st_{q,i} = y_q \qquad \forall q \in Q_c \tag{5.23}$$

$$a_{q,i+j} \ge st_{q,i} \qquad \forall q \in Q_c, \forall i, \alpha_q \le i \le \omega_q, \forall j, 0 \le j < \tau_q \tag{5.24}$$

$$\sum_{k \in K} \gamma_{k,e,q} = b_q \cdot \sum_{r, 0 \le r < R} \chi_{r,q} \cdot H^e_{r,q} \qquad \forall q \in Q_c \tag{5.25}$$

$$\gamma^i_{k,e,q} \le \gamma_{k,e,q} \qquad \forall q \in Q_c, \forall e \in E, \forall k \in K, \forall i, \alpha_q \le i \le \omega_q \tag{5.26}$$

Constraint (5.23) ensures that there is exactly one starting interval for a demand q, which is routed over the network (i.e., if y_q=1). Constraint (5.24) sets $a_{q,i}$ =1 for τ_q consecutive intervals starting with $st_{q,i}$ = 1; this ensures that the segments for demand q are sent continuously in τ_q consecutive intervals, starting from interval $st_{q,i}$.

Constraints (5.25) and (5.26) ensure that all segments for demand q are sent using the same b_q channels during each interval. As before, if wavelength continuity constraint needs to be enforced, then constraints (5.11) and (5.12) must also be added.

The above variables and constraints can be used directly for the fixed window model as well. As long as the start and end times (α_q and ω_q) satisfy the condition $\tau_q = \omega_q - \alpha_q + 1$, the entire demand will be sent as a single segment, using the same physical route and set of wavelengths. We note that an ILP formulation that directly models the continuous demands will be more efficient than the above method of restricting a more generalized segmented model. In terms of the number of binary variables, the route assignment variables ($\chi_{r,q}$) and demand accommodation variables (y_q) would be the same for both cases. But, ILP-C requires more wavelength assignment variables (since they must be defined for each interval); also the demand starting variables ($st_{q,i}$) must be defined as integer variables, which is not necessary in the continuous model. This results in increased complexity for ILP-C compared to traditional formulations. However, it is important to note that even for the traditional continuous model, the optimal formulations become computationally intractable for practical networks, hence heuristics are ultimately needed in both cases (continuous and segmented).

5.5.3 Allow Segments to Follow Multiple Physical Routes (ILP-MR)

In our formulation, we have assumed that all segments for a demand use the same physical route, in order to reduce the chances of out-of-order segments at the receiver. However, this restriction can be relaxed if appropriate. In order to do this, we define a new binary variable $\chi^i_{r,q}$ to replace $\chi_{r,q}$ in the original formulation.

- $\chi^i_{r,q}$ = 1, if and only if the r^{th} physical route is selected to route demand q during interval i, 0 otherwise.

Finally, constraint (5.4) is replaced by constraint (5.27), given below, and variable $\chi_{r,q}$ is replaced by $\chi^i_{r,q}$ in constraints (5.5) and (5.6). A similar technique can also be used for backup lightpaths, if path protection is implemented.

$$\sum_{r, 0 \le r < R} \chi^i_{r,q} = a_{q,i} \qquad \forall q \in Q, \forall i, \alpha_q \le i \le \omega_q \tag{5.27}$$

5.5.4 Search All Paths Based on Flow Conservation Constraints (ILP-FC)

In our original formulation (ILP-S), we do not search for all possible paths over the physical topology, but have restricted our search space by adopting the approach used in [34]. However, we note that the exact link level formulation, obtained using flow conservation constraints, is also of interest, and requires only minor changes to our original ILP. Therefore, we are including the link-level formulation below. For the new formulation, we first replace the variables $\chi_{r,q}$ in ILP-S by the binary variables $\chi_{e,q}$ defined below.

- $\chi_{e,q}$ = 1 if and only if demand q is routed over physical link e, 0 otherwise.

Then, we simply replace constraints (5.4) - (5.6) in ILP-S by the new constraints (5.28) - (5.30) as defined below.

$$\sum_{e:e(u\to v)\in E}\chi_{e,q} - \sum_{e:e(v\to u)\in E}\chi_{e,q} = \begin{cases} y_q & if\ u = s_q \\ -y_q & if\ u = d_q \\ 0 & otherwise \end{cases} \qquad \forall q \in Q, \forall u \in N \quad (5.28)$$

Eq. (5.28) is the standard flow constraint [85], and is used to find the best route over the physical topology, for each established lightpath.

$$\chi_{e,q} + a_{q,i} - \sigma_{e,q}^{i} \le 1 \qquad \forall q \in Q, \forall e \in E, \forall i, \alpha_q \le i \le \omega_q \qquad (5.29)$$

$$\chi_{e,q} \ge \sigma_{e,q}^{i} \qquad \forall q \in Q, \forall e \in E, \forall i, \alpha_q \le i \le \omega_q \qquad (5.30)$$

Constraint (5.29) ensures that if both $\chi_{e,q}$ = 1 and $a_{q,i}$ = 1, then $\sigma_{e,q}^{i}$ must be set to 1. Constraint (5.30) states that if $\chi_{e,q}$ = 0, then $\sigma_{e,q}^{i}$ must be 0.

5.5.5 Limiting the Number of Segments (ILP-NS)

For some applications, it may be desirable to ensure that the maximum number of individual segments for a demand does not exceed a pre-specified limit S_{max}. In order to do this, we introduce a new binary variable $\psi_{q,i}$, which indicates if interval i is the beginning of a new segment for demand q. We also add the new constraints given below.

- $\psi_{q,i}$ = 1, if and only if interval i is the start of a new segment for demand q, 0 otherwise.

The following constraints should be added to the original formulation:

$$a_{q,i} - a_{q,i-1} \le \psi_{q,i} \qquad \forall q \in Q, \forall i, \alpha_q \le i \le \omega_q \tag{5.31}$$

$$a_{q,i} \ge \psi_{q,i} \qquad \forall q \in Q, \forall i, \alpha_q \le i \le \omega_q \tag{5.32}$$

$$a_{q,i-1} + \psi_{q,i} \le 1 \qquad \forall q \in Q, \forall i, \alpha_q \le i \le \omega_q \tag{5.33}$$

$$\sum_{i,\alpha_q \le i \le \omega_q} \psi_{q,i} \le S_{\max} \qquad \forall q \in Q \tag{5.34}$$

Constraint (5.31) indicates interval i is the start of a new segment (sets $\psi_{q,i}$ = *1*) if interval i is active, but the previous interval i-1 is not. Constraint (5.32) states that only active intervals can be the start of a new segment, and (5.33) states that if interval i -1 is active, then interval i cannot be the start of a segment. Finally (5.34) ensures that the maximum number of segments does not exceed S_{max}.

5.6 RWA Heuristic for Segmented Scheduled Demands (H-SSD)

The ILP formulations discussed in the previous sections become computationally intractable for large networks. In this section we present our heuristic for RWA of Segmented Scheduled Demands (H-SSD) without any wavelength conversion. The heuristic can be used for practical networks with a large number of demands. For a given fiber network with a specified number of channels per fiber and a set of segmented scheduled demands, H-SSD tries to allocate resources in a way that accommodates as many demands as possible. We have used the following additional notation to describe the heuristic:

- $S_{e,i}$: The set of free channels (channels not allocated or temporarily locked to any lightpath) on the link e at the interval i.
- M_q: The set of potential active intervals i, for the demand q, where $\alpha_q \le i \le \omega_q$.
- A_q: The set of active intervals i, for the demand q, where $\alpha_q \le i \le \omega_q$.
- O_e: The set of intervals during which link $e \in E$ may become congested.
- Q_{alloc}: The set of allocated demands.
- $Q_{unalloc}$: The set of unallocated demands.
- $r_{q,i}$: The set of edges in the selected (least-congested) route for demand q during the interval i.

- $W_{r_{q,i}}$: The set of free channels that can be allocated to demand q along its selected route $r_{q,i}$, during the interval i. $W_{r_{q,i}} \leftarrow \bigcap_{e \in r_{q,i}} S_{e,i}$.
- $C^{r}_{q,i}$: The maximum congestion (in terms of the number of channels) on any edge for the r^{th} route of the demand q during the interval i.
- $K_{e,i}$: The set of channels (wavelengths) on each link $e \in E$ that have been allocated or temporarily locked for a lightpath at the interval i.
- η_q: The set of intervals during which the demand q can be routed using only uncongested links.
- ζ_q: The set of intervals during which the demand q uses at least one congested link.

H-SSD allows different segments of a demand to use different routes, but constraints similar to those in section 5.3 may be easily incorporated here as well. In H-SSD, when allocating resource to a demand q, the following constraints must be met:

- The number of WDM channels allocated on a link does not exceed $|K|$, in any interval i.
- The number of active intervals for a demand q equals its total demand holding time τ_q.

Overall, the entire algorithm is implemented in several phases as indicated in Fig. 5.1.

a) PHASE 0: Initialization

Initially, none of the demands have been allocated resources, so we set $Q_{alloc} \leftarrow \phi$ and $Q_{unalloc} \leftarrow \phi$. We also initialize the link capacities and set the values of $S_{e,i}$ to indicate that all channels on all links are free. During this phase, we also input R pre-computed routes, over the physical topology, for each node-pair.

b) PHASE 1: Contention-free Resource Reservation

The next phase of the heuristic (PHASE 1, lines 4 - 20 in Fig. 5.1) allocates resources only to those demands, which can be allocated without any contention with other demands. This phase is decomposed into 2 sub-phases - resource reservation and contention-free allocation.

Resource reservation: During the resource reservation phase of the heuristic (PHASE 1a, lines 6 - 12 in Fig. 5.1), each demand, $q \in Q_{unalloc}$, is processed one by one. For each demand q and each interval $i \in M_q$, we select the least-congested route r (from among the R pre-computed routes between s_q and d_q) for routing demand q during interval i. The congestion of a route, during interval i, is defined as the maximum number of channels allocated on any link e on that route, and is calculated based on the previously processed demands in

PHASE 1a. If the selected route has sufficient capacity to support demand q during interval i, we temporarily "lock" a set of b_q channels for demand q for that interval (lines 10 - 11). Otherwise, we identify each link e on the selected route r, which does not have sufficient resources and add interval i to O_e (line 12). In other words $i \in O_e$ indicates that link e may become congested during interval i. The order, in which the demands are processed, in PHASE 1a, is not important. This is because line 12 (for step 1a) simply states that edge e is congested during interval i, it does not matter which demands are causing the congestion. At the end of step 1a, the set of congested links for each interval has been determined. It is important to note that the main goal of the resource reservation phase is to identify the best candidate intervals for each demand q (i.e., the intervals least likely to interfere with resource allocation of other demands). However, no resources are actually allocated during this phase; they are just temporarily "locked" for future use, if needed.

Contention-free Allocation: During contention-free allocation (PHASE 1b, lines 13 - 19 in Fig. 5.1), the goal is to allocate resources to the demands which can be accommodated using only those intervals during which the congested links can be avoided. In order to do this, we first identify the intervals $i \in \zeta_q$, i.e., the intervals during which demand q may need to use channels on a "congested" link e. We set $i \in \zeta_q$, if $i \in O_e$ (as determined during PHASE 1a) and the link e is on the route selected (in PHASE 1a) for demand q during interval i. Then, $\eta_q = M_q - \zeta_q$ represents the set of intervals available for allocating demand q using "uncongested" links only. If $|\eta_q| \geq \tau_q$, then we simply select τ_q intervals from the set η_q and allocate b_q channels for demand q during each selected interval. Otherwise, demand q cannot be accommodated during the current iteration of PHASE 1. In case demand q is successfully accommodated, any locked resources for this demand during the remaining intervals in M_q are released. We note that the set of intervals η_q does not necessarily consist of consecutive intervals. It simply contains the intervals i during which q may be allocated without contention with other demands. Since a demand q is allocated only if it can be done without using any congested links in each active interval, such allocation will never prevent any other demands (processed later) from being allocated. Therefore, in PHASE 1b (as in PHASE 1a), the order of processing the demands is not important. Therefore, even though we process one demand at a time, it does not guarantee that all segments of an allocated demand will occupy adjacent intervals.

Each time a demand is successfully allocated in this phase, certain resources that were reserved during PHASE 1a, but are no longer needed, are released and become available for other demands. This means that a link e that was previously designated as "congested" during interval i may no longer be congested. This in turn may allow some additional demands to be accommodated using the newly uncongested links. So, this is an iterative process, and the whole process, i.e. PHASE 1a followed by PHASE 1b (lines 5 - 20), must be repeated until no more demands can be allocated using "uncongested" links only.

c) PHASE 2: Greedy Resource Allocation

In PHASE 2, any remaining demands, which could not be accommodated during PHASE 1, are processed using a "greedy" scheme (lines 22 - 33 in Fig. 5.1). If desired, the demands can be sorted, using *one* of the criteria given below. In case of ties, we arbitrarily pick one demand to be processed first.

- **criterion 1**: the unallocated demand set $Q_{unalloc}$ is sorted in an ascending order of the number of intervals needed.
- **criterion 2**: the unallocated demand set $Q_{unalloc}$ is sorted in a descending order of the demand priority ($q_{priority}$), where $q_{priority}$ = *(number of intervals still needed for demand q)/(number of intervals remaining until* ω_q*)*.

Then, each demand from the sorted list Q_{sort} is processed one by one. For each demand q, we compute the set $\eta_q = M_q - \zeta_q$ as before and allocate channels for q during these intervals. For the remaining segments of q, if the demand traverses a congested link during interval i, and a sufficient number of channels are available, we greedily allocate these to the demand, until we find τ_q intervals for demand q. If successful, then resources for demand q are allocated; otherwise demand q cannot be accommodated, and all resources for this demand (during all other intervals) are released. As in PHASE 1, there is no guarantee that the intervals for demand q will be consecutive.

The algorithm returns a set of demands Q_{alloc} that contains all the demands that can be handled under the given resource constraints, along with their routes and assigned channels.

As mentioned earlier, the H-SSD heuristic allows different segments of a demand to use different routes, and does not place any limit on the number of segments used. For comparison purposes, it would be interesting to see how such a limit on the number of segments will affect the quality of the solutions. Imposing a limit on the maximum number of segments would require a significant shift from the overall design philosophy used in H-SSD, since resources for different active intervals of a demand can no longer be allocated independently of each other. Therefore, to incorporate segment limits, we have used a modified heuristic, as outlined below.

1) Allocate as many demands as possible (to Q_{alloc}), using only a single segment for each demand. This is done using the algorithm given in [34], for the traditional *continuous* model.

2) Process each remaining demand $q \in Q_{unalloc}$ in a greedy manner.

a. Determine the minimum number of segments S_q required to allocate demand q, based on the resources already allocated to the demands in (Q_{alloc}).

b. **If** ($S_q \le S_{\max}$) **then**

Allocate resources to demand q

$Q_{alloc} = Q_{alloc} \bigcup \{q\}$

Else demand q is blocked.

5.6.1 Complexity Analysis of H-SSD Algorithm

In this section, we discuss the performance of the heuristic (H-SSD), i.e., how close the results are to the optimal solution when it can be found. We also comment on its time complexity. We shall treat the value of R as constant. Disregarding the initialization phase (PHASE 0), we present an informal discussion on the complexity of H-SSD as follows.

The time required for the *Resource Reservation* is bounded by $O(|Q| \cdot i_{\max} \cdot |E|)$, and the time for the *Contention-free Resource Allocation* is bounded by $O(|Q| \cdot i_{\max})$. For the practical design problems on wide-area topologies that are in use today [32, 93, 94], typically $i_{\max} \leq |Q|$ and $|E| \leq |Q|$. Therefore, we see that PHASE 1 is polynomially bounded over $|Q|$.

Finally, for PHASE 2, the resources are allocated in a greedy manner, for each remaining demand during each active interval. The sorting can be done in $O(|Q| \cdot \log |Q|)$ time using *heap-sort*. The remaining are bounded by $O(|Q| \cdot i_{\max})$. Hence, PHASE 2, and therefore, the overall time complexity of H-SSD is polynomial in $|Q|$. In contrast, the complexity of the corresponding ILPs is exponential. It is important to note that H-SSD does not require a specific bound on either the number of intervals i_{max}, or the number of edges $|E|$. However, in most practical cases $|E|$ and i_{max} are much less than $|Q|$, so the complexity of the heuristic depends primarily on $|Q|$ for practical topologies.

Heuristic: H-SSD

Input: $G = (N, E)$, $\mathcal{K}$ and $\mathcal{Q}$.

Value returned: $\mathcal{Q}_{alloc}$, and an appropriate RWA $\forall q \in \mathcal{Q}_{alloc}$.

\\\\ PHASE 0: Initialization
Set $\mathcal{Q}_{alloc} \leftarrow \emptyset$ and $\mathcal{Q}_{unalloc} \leftarrow \mathcal{Q}$
Read in $\mathbb{R}$ routes between each pair of nodes. Initialize $\mathcal{M}_q$ and $S_{e,i}$.
\\\\ PHASE 1: Contention-free Resource Allocation
while (*demand can be added to* $\mathcal{Q}_{alloc}$) **do**
  \\\\ 1a. Resource Reservation
    for each demand $q \in \mathcal{Q}_{unalloc}$
      for each interval $i, \alpha_q \leq i \leq \omega_q$
        Select the least-congested route, $r_{q,i}$.
        if congestion of $r_{q,i}$ does not exceed $|\mathcal{K}| - b_q$ **then**
          Lock b_q wavelengths from $\mathcal{W}_{r_{q,i}}$.
        else $\mathcal{O}_e \leftarrow \mathcal{O}_e \cup \{i\}, \ \forall e \in r_{q,i}$.
  \\\\ 1b. Contention-free Allocation
    for each $q \in \mathcal{Q}_{unalloc}$
      for each interval $i \in \mathcal{M}_q$
        if $e \in r_{q,i}$ and $i \in \mathcal{O}_e$ **then** Update ζ_q.
        $\eta_q \leftarrow \mathcal{M}_q \backslash \zeta_q$.
        if $|\eta_q| \geq \tau_q$ **then** Select $\mathcal{A}_q$ such that $\mathcal{A}_q \subseteq \eta_q$ and $|\mathcal{A}_q| = \tau_q$
        Release unused resources for demand q.
end while
\\\\ PHASE 2: Greedy Resource Allocation
$\mathcal{Q}_{sort} \leftarrow$ sorted demands from $\mathcal{Q}_{unalloc}$.
for each $q \in \mathcal{Q}_{sort}$ **do**
    for each interval $i \in \eta_q$
      Allocate b_q channels from $\mathcal{W}_{r_{q,i}}$ for demand q in interval i.
    for each interval $i \in \zeta_q$
      if sufficient channels are available on each link $e \in r_{q,i}$
        Allocate b_q channels from $\mathcal{W}_{r_{q,i}}$ for demand q in interval i.
      if τ_q active intervals have been found for demand q **then** *Break*
    if (total active intervals for q is less than τ_q) **then**
      Release all resources for demand q in all intervals.
    else $\mathcal{Q}_{alloc} = \mathcal{Q}_{alloc} \cup \{q\}$.
end for
return ($\mathcal{Q}_{alloc}$).

Figure 5.1 : Pseudocode for H-SSD.

5.7 Experimental Results

In order to study the performance of our proposed segmented scheduled traffic model, we have tested our approach on a number of networks, varying in size from 6 to 20 nodes. For each network, we considered different number of demands and different amounts of available resources. In order to obtain a balanced perspective, the demands to be allocated were randomly generated in terms of all three criteria, i.e. the source-destination of each demand, the duration of each demand (between 3 and 6 hours, inclusive), and the bandwidth requirement for each demand. However, for a fair comparison, the same demand set was used to generate solutions using the different traffic models, e.g. fixed window, continuous sliding, and segmented models. For each size of demand set, 5 sets of demands were randomly generated and tested using different schemes. Each individual result presented here is the average of these 5 sets.

5.7.1 Results for the ILP

We were able to generate optimal solutions based on our ILP formulations, for small size networks (up to 6 nodes), using ILOG CPLEX solver [87]. Table 5.2 compares the total number of demands accommodated using the traditional scheduled traffic model and the ILP formulation (with wavelength converters) on the 6-node topology [56].

For these experiments, the number of demands were varied from 32 to 150 (indicated in the column $|Q|$), and the number of channels were varied from 4 to 16 (indicated in the column $|K|$). The column "Fixed" indicates the total number of demands that were accommodated by using the traditional *fixed* window approach, under the demand set and the number of channels given in the corresponding row. The columns "Continuous + 2hrs" and "Continuous + 4hrs" ("Sliding + 2hrs" and "Sliding + 4hrs"), indicate the same while using the traditional *continuous* (proposed *segmented*) sliding window approach, with the window size extended by 2hrs and 4hrs, respectively, over the fixed window demand holding time for each demand. The computation time ranged from several seconds (for 32 demands) to several hours for larger demand sets. As shown in the table, the ILP clearly outperforms the fixed window and the traditional sliding window approaches in all cases, with the amount of improvement increasing with the window size. We note that the fixed and continuous sliding window models are *holding-time-aware* models, which have already been shown to significantly outperform *holding-time-unaware* models. The performance improvements that can be achieved by our proposed model with respect to the holding-time-unaware approaches will be obviously much better.

$\lvert Q \rvert$	$\lvert K \rvert$	# of demand accommodated under the scheme				
		Fixed	Continuous +2 h	Continuous +4 h	Sliding +2 h	Sliding +4 h
32	4	22.8	24.2	24.8	29.2	31.2
50	4	28.3	30.3	31.8	39.4	42.2
80	8	55.7	61	61.7	74.4	77.2
100	8	65.3	68.5	69.5	88.2	92
150	16	116.7	123.7	124	141.8	146.8

Table 5.2: Comparison of total number of demands accommodated using the traditional scheduled traffic model and the ILP formulation (with wavelength converters) under different window size.

5.7.2 Results for H-SSD, Fault-free Case

To evaluate the quality of the solutions generated by the heuristic, we first compare the solutions to the optimal solutions (for cases where the optimal solution can be generated). For these cases, the heuristic solution is typically within 10% - 20% of the optimal solutions reported in Table 5.2. For larger networks, where the ILP is not able to find the optimal solution, we calculated an upper bound for the maximum number of accommodated demands based on an LP-relaxation of the proposed ILP-S, where all binary variables in ILP-S were declared as continuous variables. This relaxed formulation was solved using the CPLEX solver [87]. We note that the solution obtained by such LP-relaxation may lead to partially accommodated demands, or a single lightpath using multiple routes. In other words, it does *not* necessarily yield a feasible solution for our problem. However, it provides an upper bound in the sense that the actual optimal solution can never be better than that obtained using LP-relaxation. The actual optimal solution that can be achieved is typically lower than the upper bound, and the difference increases as the number of demands increases. Table 5.3 shows the percentage difference between the calculated upper bound and the heuristic solution for 14-node NSFNET topology with $|K|=16$ and window size extended by 2 hours, 4 hours, and 6 hours, respectively. The computation time required for calculating the upper bound ranged from a few seconds to several hours. The heuristic typically required only a fraction of a second to generate a solution. We see from the Table that the heuristic solutions are consistently within 20% - 25% of the upper bound (and likely even closer to the actual optimal solution).

$\|Q\|$	The percentage difference between the upper bound and the heuristic solution		
	Sliding +2 h (%)	Sliding +4 h (%)	Sliding +6 h (%)
100	0.00	0.00	0.00
150	6.00	4.67	2.67
200	15.58	13.50	9.00
300	23.27	23.18	17.41

Table 5.3: Comparison of H-SSD solutions to upper bound.

We studied the performance of the heuristic H-SSD using the well-known NSFNET (14 nodes, 21 bidirectional links and average degree of 3) and the ARPANET (20 nodes, 31 bidirectional links and average degree of 3.1) topologies

[86]. For these experiments, the number of demands was varied from 100 to 300. For each demand set, we compared the performance of our approach (using H-SSD) with both the *fixed* window and the traditional *continuous sliding window* approaches with the window size extended by 2hrs, 4hrs, and 6hrs over the fixed window demand holding time for each demand (the notation follows the similar convention as in Table 5.2 above). Even for the continuous model, ILP formulations become computationally intractable for practical networks. Therefore, we have used a modified version of an existing heuristic algorithm [34] to allocate resources for the continuous case.

Figure 5.2 (Fig. 5.3) shows the total number of demands that can be accommodated using H-SSD on the 14-node NSFNET (20-node ARPANET) network with no protection, when $|K|=16$ and criterion 2 (criterion 1) is used for sorting. We see that, in both cases, when the network is lightly loaded (i.e., the number of demands is lower), there is not much difference between the fixed window model and the proposed model. However, as the number of demands increases, the advantages of our *segmented* scheduled traffic model become more significant and the number of additional demands that can be accommodated (compared to the fixed window case) increases steadily. The results also confirm that increasing the length of the allowable time-window for each demand leads to improved performance, which is expected due to the increased flexibility in scheduling demands.

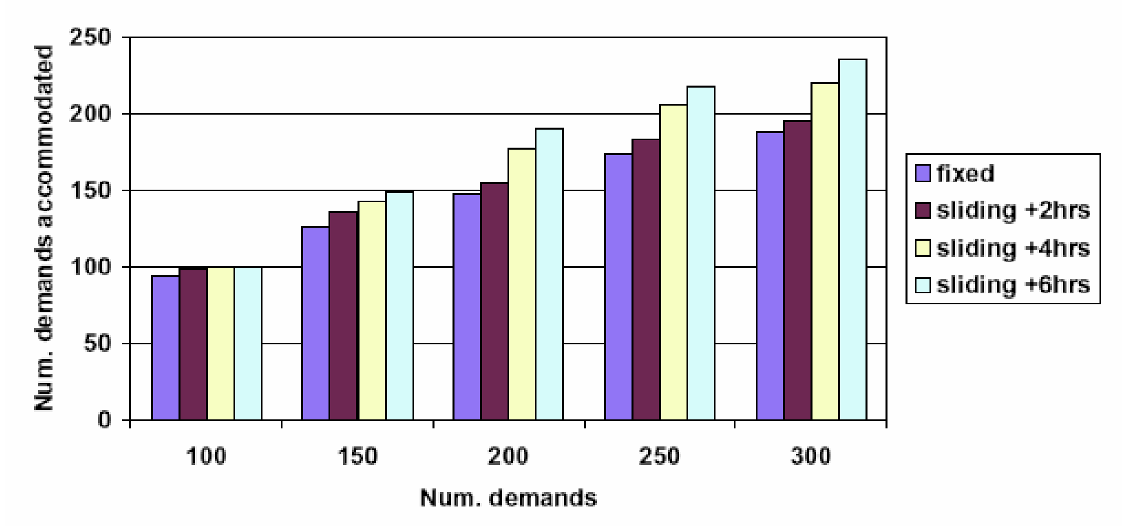

Figure 5.2: Number of demands accommodated using H-SSD for 14-node network.

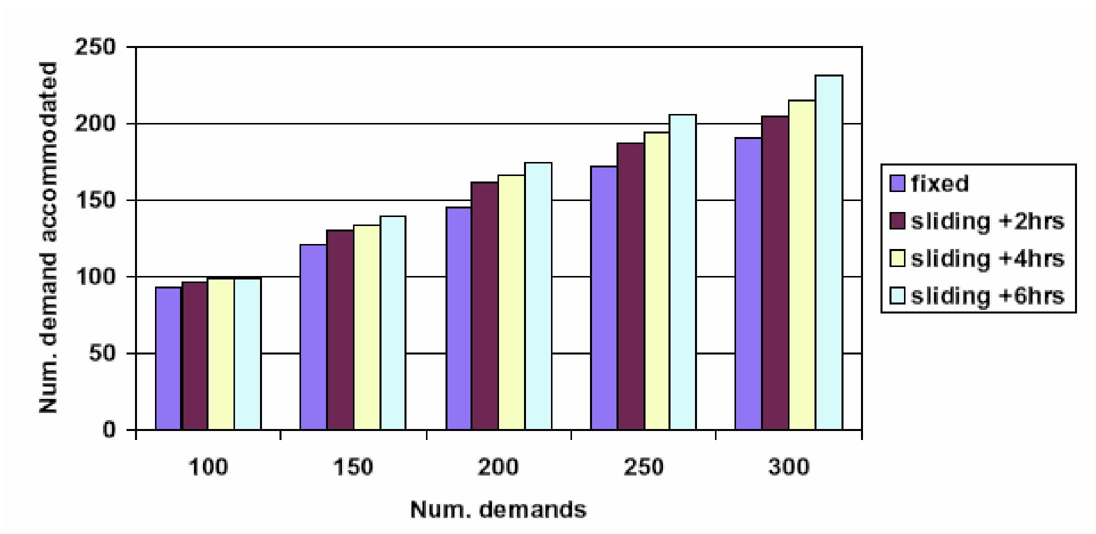

Figure 5.3: Number of demands accommodated using H-SSD for 20-node network.

In terms of the effectiveness of sorting the demands according to different attributes, the results for both criteria were close. Table 5.4 shows a comparison of the number of accommodated demands for the 14-node NSFNET topology, with $|K|=16$ and window size extended by 2 hours, when using different sorting criteria in PHASE 2 (including random selection of demands). For this configuration criterion 1 performed slightly better, and criterion 2 did not show any gains compared to random ordering. In general, we observed that sorting by criterion 1 performs slightly better in the case of smaller numbers of available channels per fiber and smaller window sizes, while criterion 2 provides better results when the number of channels per fiber and window size are increased. However, the gains are usually not substantial. So, the sorting step in PHASE 2 is optional and may be skipped if desired.

$\lvert Q\rvert$	$\lvert K\rvert$	The number of accommodated demands		
		Criteria 1	Criteria 2	Random
100	16	98	99	99
150	16	137	135	136
200	16	165	155	159
250	16	192	183	182
300	16	208	195	196

Table 5.4: The average number of accommodated demands for different sorting criteria in a 14-node network with window size extended by 2 hours.

The segmented scheduled traffic model not only outperforms the fixed window model, it also leads to significant improvements, in terms of the number of demands that can be accommodated by the network, compared to the traditional (i.e., continuous) *sliding window model*, which is currently the most flexible approach for scheduling lightpath demands. Figure 5.4 shows the percentage improvements, in the number of accommodated demands, obtained using H-SSD over the traditional fixed and sliding window models on a 14-node fault-free network with $|K|=16$ when criterion 2 is used for sorting. When calculating the improvement over the fixed window model, we used the results from the segmented "Sliding +6hrs" case for comparison. In all the other cases, we extended the window size by the same amount for both the continuous and segmented models, in order to get a meaningful comparison. In each group of bars, corresponding to a specific demand size and link capacity, the first (second) bar indicates the percent improvement obtained using the segmented model compared to the continuous sliding window model, when the window size is extended by 4 hours (6 hours). The third bar indicates the percent improvement of the segmented model over fixed window approach. As shown in Fig. 5.4, our approach consistently outperforms both the fixed and continuous sliding window models. The improvements range from 10% to 30%, and vary depending on the available resources and load on the network.

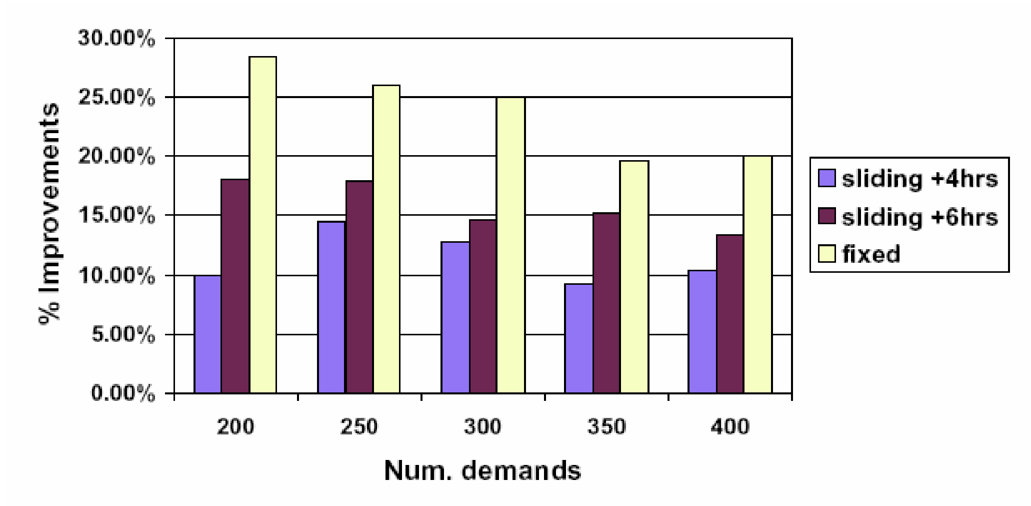

Figure 5.4: Percentage improvements in the number of demands accommodated by the network, obtained using H-SSD compared to continuous scheduled traffic model.

Figure 5.5 shows the relative effect of increasing the number of channels ($|K|=8$ to $|K|=16$), or increasing the window size (from 2 hours to 6 hours) on the number of demands that can be accommodated in the network. Obviously, when we increase the amount of available resources (in this case, the number of WDM channels per fiber), it is possible to handle many more demands. However, it is interesting to note that a portion of the additional demands could also have been accommodated by using a larger window size, which is a much simpler option. Of course, if the network is seriously under-resourced, it will eventually become necessary to increase the capacity of the network. Increasing both parameters, i.e., window size and fiber capacity together, always yields the best results, as expected. We note that increasing the window size or number of channels leads to a corresponding increase in demands allocated for the continuous model as well. We have not reported the results separately here, since they followed a pattern similar to the segmented model. Overall, we observed that the relative improvement of the segmented model compared to continuous sliding window remained fairly consistent for the different values of $|K|$ and window size.

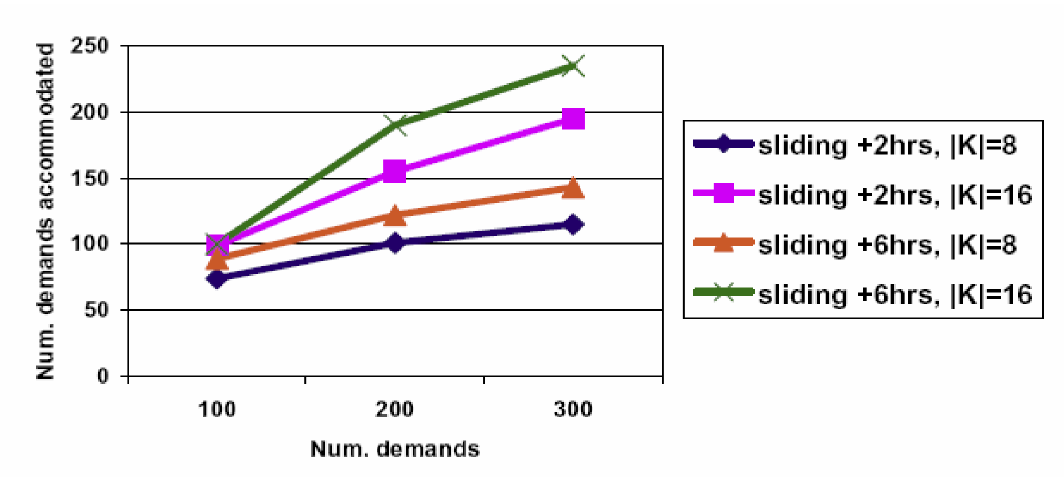

Figure 5.5: Effect of increasing fiber capacity or window size on a 14-node network.

5.7.2.1 Segment Allocation in Different Phases

In H-SSD, resource allocation for a demand can occur in Phase 1b (contention-free allocation) or Phase 3 (greedy allocation). Unless the demand size is very small (less than 50 demands), it is rare that the entire demand (i.e., all segments) can be allocated during Phase 1b. However, we have found that a non-negligible portion of the total segments of a demand can usually be allocated in a contention-free manner. Table 5.5

shows the average percentage of segments having contention-free allocation in a 14-node network with wavelength conversion for different demand sets and different window sizes. We see that as the size of the demand set or the window size increases, the proportion of segments that can be allocated in a contention-free manner decreases steadily. The first trend is expected since more demands lead to increased contention for resources. The decrease with window size occurs because a demand has more flexibility for when it is to be scheduled, and therefore can potentially overlap with many more demands.

$\|Q\|$	$\|\mathcal{K}\|$	The average percentage of segments having contention-free allocation		
		Sliding +2 h (%)	Sliding +4 h (%)	Sliding +6 h (%)
100	8	83.93	77.54	71.96
150	16	32.21	27.25	20.84
200	16	20.98	15.15	9.91
250	16	13.06	9.01	7.84
300	16	9.42	6.69	6.22

Table 5.5: The average percentage of segments having contention-free allocation in a 14-node network with wavelength conversion for different demand sets and different window sizes.

5.7.2.2 Results with Limited Number of Segments

In our original ILP formulation (ILP-S) and heuristic (H-SSD), no limits are imposed on the maximum number of segments allowed for decomposing a single demand. In order to investigate the effect of limiting the number of segments, we first calculated how many segments, on average, were actually being used for each demand. These results are shown in Table 5.6 for the 14-node topology, and similar values were observed for other topologies, including the 6-node network, using ILP. Although there were a few isolated cases where a demand was decomposed into 4 or more segments, the vast majority of the demands were actually accommodated using only 1 - 3 segments. As a result, putting a limit on S_{max}, the maximum allowable number of segments, had no significant effect on the performance of the proposed ILP and heuristic, even for S_{max} = 2 or 3. We note that S_{max} = 1 corresponds to the traditional continuous sliding window model, for which we have already reported the results.

$\|Q\|$	$\|K\|$	The average number of segments per demand		
		Sliding +2 h	Sliding +4 h	Sliding +6 h
100	8	2.1	2.1	2.3
150	16	1.7	1.9	2.1
200	16	1.7	2.0	2.1
250	16	1.7	2.1	2.3
300	16	1.8	2.1	2.3

Table 5.6: The average number of segments per demand in a 14-node network.

5.7.3 Results for H-SSD, Path Protection

Figure 5.6 shows the amount of improvements in terms of the total number of demands that the network can handle, obtained using H-SSD, compared to traditional (continuous) sliding window models, when dedicated path protection is implemented. The first (second) bar in each group indicates the relative improvement when the window size is extended by 2 hours (4 hours). The above results are for a 14-node network with $|K|=16$ when criterion 1 is used for sorting. Results for other networks show a similar pattern. The average improvement varies from about 12% (for a lower number of demands) to over 30% (for larger demand sets). The improvements over the fixed window model are much higher, varying from 35% to 45%.

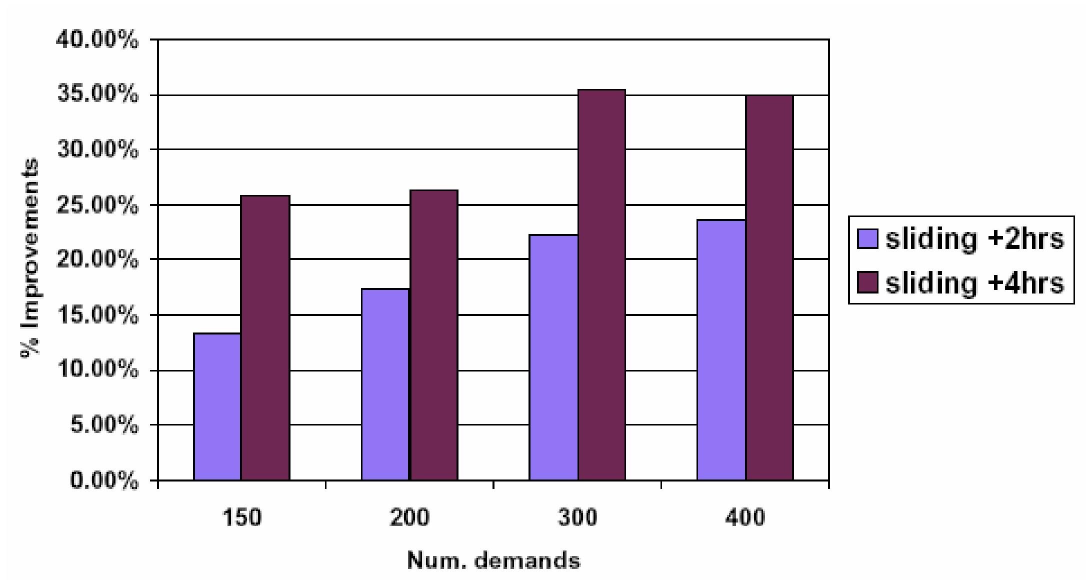

Figure 5.6: Percentage improvements in the number of demands accommodated using H-SSD compared to continuous sliding window model, for dedicated path protection.

Finally, Fig. 5.7 shows, as expected, that almost twice as many demands can be accommodated (using the same parameters as in Fig. 5.6) in a network with no protection versus one with dedicated path protection. More interestingly, we see that for both cases (i.e., fault-free and dedicated protection), increasing the window size from 2 hours to 6 hours leads to improved performance (by about 10% to 20% on average).

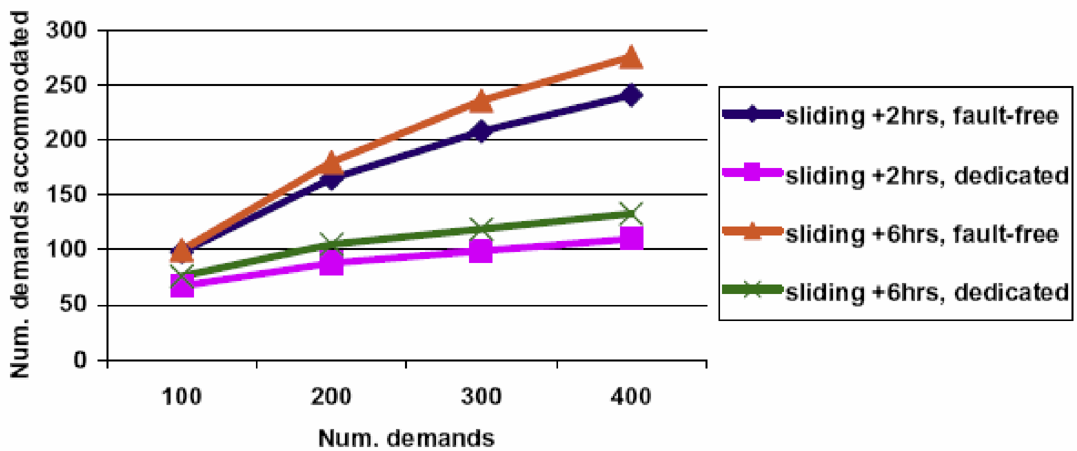

Figure 5.7: Comparison of the number of accommodated demands for fault-free networks versus dedicated protection.

5.8 Summary

The main contributions of this chapter are:

- We propose a new segmented scheduled traffic model, which is more flexible than the existing models for scheduled traffic and results in more efficient resource utilization.

- We present an ILP formulation that optimally solves the complete design problem and show that the traditional fixed and sliding window models for scheduled traffic can be treated as a special case of our proposed model.

- We present an efficient heuristic that can be used for larger networks with many scheduled demands.

- We demonstrate, through simulations that significant improvements can be achieved using our approach, even compared to existing *holding-time-aware* models, which already outperform *holding-time-unaware* models.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

By exploiting knowledge of the connection holding times of traffic demands, this thesis proposes new traffic grooming techniques to achieve more efficient resource utilization in WDM optical networks. Efficient integer linear program (ILP) formulations are presented to address the complete traffic grooming problem consisting of logical topology design, routing and wavelength assignment, and routing of traffic demands over the selected topology. Although individual demands may be short lived, it is desirable to have a logical topology that is relatively stable and not subject to frequent changes. Therefore, the objective is to design a *stable* logical topology that can accommodate a collection of low-speed traffic demands with specified setup and teardown times. The proposed approach results in lower equipment cost and significantly reduces overhead for connection setup/teardown. A new technique for energy efficient routing and scheduling of periodic demands is also presented to minimize the overall energy consumption of a network. Unlike most existing energy-aware traffic grooming approaches, our approach considers holding-time-aware demands, and adaptively selects the start times and routes for the demands in a way that allows the maximum number of lightpaths (from the specified base topology) to be switched off at any given time. Finally, this thesis introduces a new model, the *segmented sliding window model*, which is more flexible than the existing models for scheduled demands (including the sliding window model) and can be useful for a variety of applications requiring periodic use of bandwidth. We present ILP formulation*s* that optimally divide a demand into multiple segments (if necessary), schedule each segment in time and perform RWA for each of the individual segments. Simulation results demonstrate that the proposed model can lead to additional improvements, even over existing holding-time-aware models.

6.2 Future Work

Grid computing requires high performance computing with resource sharing and high speed communications to support large-scale, data-intensive applications involving geographically distributed resources. Owing to their high reliability, high throughput, and low latency, optical networks are ideal for meeting capacity and connectivity requirements of grid applications. The optical network architecture used for grid computing is generally known as *optical*, *lambda*, or *photonic* grid [95]. Different solutions have been proposed to provide protection and fault-tolerance in both grid computing and optical networks. But a new challenge in resilient optical grid network design is to consider the interrelation between computing and network resource usage and how they are affected by potential faults [96]. Therefore, it is necessary to develop integrated algorithms that jointly allocate computing (e.g. processor time) and network (e.g. link bandwidth) resources to improve resource utilization while guaranteeing service availability.

In spite of the lower power cost per bit of optical networks, it is expected that one of the most challenging issues in the next decade will be reducing the power requirement for such core networks. For both traditional and optical grid paradigms, the exact geographic location of the physical resource remains transparent to the user. Typically the user does not care where exactly a specific job is executed. So, it is possible to select one out of a number of possible destinations to execute a specific job. This is known as the *anycast* [97] principle. By exploiting the inherent flexibility of anycasting, we will propose new approaches to select the destination node and perform routing and wavelength assignment (RWA) to minimize overall energy consumption of a set of lightpath demands.

Bibliography

[1] R. Ramaswami and K. N. Sivarajan, Optical Networks: A Practical perspective, 2nd. Ed., Morgan Kaufmann Publishers, 1998.

[2] I. Chlamtac, A. Ganz and G. Karmi, Lightnets: topologies for high-speed optical networks, *Journal of Lightwave Technology*, 11 (5) (1993) 951-961.

[3] H. Zang, J.P. Jue and B. Mukherjee, A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks, *Optical Networks Magazine*, 1 (2000) 47-60.

[4] K. Zang and B. Mukherjee, A Review of traffic grooming in WDM optical networks: Architectures and challenges, *Optical Networks Magazine*, 4 (2) (2003) 55-64.

[5] A. Jaekel, A. Bari, Y. Chen and S. Bandyopadhyay, New techniques for efficient traffic grooming in WDM mesh networks, in: *IEEE ICCCN 2007*, pp. 303-308.

[6] A.E. Ozdaglar and D.P. Bertsekas, Routing and wavelength assignment in optical networks, *IEEE/ACM Transactions on Networking*, 11 (2) (2003) 259-272.

[7] C. Ou, K. Zhu, H. Zang, L. H. Sahasrabuddhe and B. Mukherjee, Traffic grooming for survivable WDM networks: shared protection, *IEEE Journal on Selected Areas in Communications*, 21 (9) (2003) 1367-1383.

[8] C. Xin, B. Wang, X. Cao, and J. Li, Logical topology design for dynamic traffic grooming in WDM optical networks, *Journal of Lightwave Technology*, 24 (6) (2006) 2267-2275.

[9] J. Kuri, N. Puech, M. Gagnaire, E. Dotaro and R. Douville, Routing and wavelength assignment of scheduled lightpath demands, *IEEE Journal on Selected Areas in Communications*, 21 (8) (2003) 1231-1240.

[10] B. Wang, T. Li, X. Luo, Y. Fan and C. Xin, On service provisioning under a scheduled traffic model in reconfigurable WDM optical networks, in: *BROADNETS 2005*, pp. 15-24.

[11] Y. Chen, A. Jaekel, and A. Bari, A new model for allocating resources to scheduled lightpath demands, *Computer Networks*, 55 (13) (2011) 2821-2837.

[12] N. Skorin-Kapov, P. Pavon-Marino, B. Garcia-Manrubia and R. Aparicio-Pardo, Scheduled virtual topology design under periodic traffic in transparent optical networks, in: *BROADNETS 2009*, pp. 1-8.

[13] Y. Chen, S. Phibbs, and A. Jaekel, Joint scheduling and virtual topology design for sliding periodic traffic, *Optical Switching and Networking*, 10 (3) (2013) 186 – 194.

[14] A. Jaekel, Y. Chen and A. Bari, Stable logical topologies for survivable traffic grooming of scheduled demands, *Journal of Optical Communications and Networking*, 2 (10) (2010) 793-802.

[15] G. Shen and R. Tucker, Energy minimized design for IP over WDM networks, *Journal of Optical Communication and Networking*, 1 (1) (2009) 176 - 186.

[16] J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang and S. Wright, Power awareness in network design and routing, in: *IEEE INFOCOM 2008*, pp. 457 - 465.

[17] F. Idzikowski, S. Orlowski, C. Raack, H. Woesner and A. Wolisz, Saving energy in IP-over-WDM networks by switching off line cards in low-demand scenarios, in: *14th Conf. Optical Network Design and Modeling 2010*, pp. 1 - 6.

[18] A. Coiro, M. Listanti, A. Valenti and F. Matera, Reducing power consumption in wavelength routed networks by selective switch off optical links, *IEEE Journal of Selected Topics in Quantum Electronics*, 17 (2) (2011) 428 - 436.

[19] B. G. Bathula and J. M. H. Elmirghani, Green Networks: Energy efficient design for optical networks, in: *Wireless and Optical Communications Networks 2009,* pp. 1- 5.

[20] E. Yetginer and G. Rouskas, Power efficient traffic grooming in optical WDM networks, in: *IEEE GLOBECOM 2009*, pp. 1 - 6.

[21] S. Huang, D. Seshadri and R. Dutta, Traffic grooming: A changing role in green optical networks, in: *IEEE GLOBECOM 2009*, pp. 1 – 6.

[22] Y. Chen and A. Jaekel, Energy efficient grooming of scheduled sub-wavelength traffic demands, in: *IEEE Optical Fiber Communication Conf. 2011*.

[23] Y. Chen and A. Jaekel, Energy aware scheduling and resource allocation for periodic traffic demands, *Journal of Optical Communications and Networking*, 5 (4) (2013) 261 - 270.

[24] T. Li and B. Wang, On optimal survivability design in WDM optical networks under a scheduled traffic model, in: *DRCN 2005*, pp. 23-30.

[25] N. Skorin-Kapov, Heuristic algorithms for the routing and wavelength assignment of scheduled lightpath demands in optical networks, *IEEE Journal on selected areas in communications*, 24 (8) (2006) 2 – 15.

[26] B. Wang, T. Li, C. Xin, and X. Zhang, On survivable service provisioning in WDM networks under a scheduled traffic model, in: *GLOBECOM 2005*, pp. 1900-1904.

[27] T. Li and B. Wang, On optimal survivability design under a scheduled traffic model in wavelength routed optical mesh networks, in: *4th Annual Communication Networks and Services Research Conference 2006*.

[28] T. Li and B. Wang, Path protection based routing and wavelength assignment in wavelength division multiplexing optical networks under a scheduled traffic model, *Journal of Optical Networking*, 5 (2006) 575-588.

[29] M. Tornatore, C. Ou, J. Zhang, A. Pattavina and B. Mukherjee, Efficient shared-path protection exploiting the knowledge of connection-holding time, in: *OFC 2005*, pp. 65-72.

[30] A. Jaekel and Y. Chen, Quality of service based resource allocation for scheduled lightpath demands, *Computer Communications*, 30 (18) (2007) 3550-3558.

[31] W. Su, G. Sasaki, C. Su and A. Balasubramanian, Scheduling of periodic connections with flexibility, *Optical Switching and Networking*, 3 (3) (2006) 158-172.

[32] D. Andrei, H. Yen, M. Tornatore, C. U. Martel and B. Mukherjee, Integrated provisioning of sliding scheduled services over WDM optical networks, *IEEE/OSA Journal Optical Communications and Networking*, 1 (2) (2009) A94-A105.

[33] C. V. Saradhi, M. Gurusamy and R. Piesiewicz, Routing fault-tolerant sliding scheduled traffic in WDM optical mesh networks, in: *BROADNETS 2008*, pp. 197-202.

[34] A. Jaekel and Y. Chen, Resource provisioning for survivable WDM networks under a sliding scheduled traffic model, *Optical Switching and Networking*, 6 (1) (2009) 44-54.

[35] D. Andrei, M. Tornatore, C.U. Martel, D. Ghosal and B. Mukherjee, Provisioning subwavelength multicast sessions with flexible scheduling over WDM networks, *Journal of Optical Communications and Networking*, 2 (5) (2010) 241-255.

[36] N. Charbonneau and V. M. Vokkarane, A survey of advance reservation routing and wavelength assignment in wavelength routed WDM networks, *IEEE Communications Surveys and Tutorials*, 14 (4) (2012) 1037 - 1064.

[37] N. Charbonneau, A. G. Gadkar, B. H. Ramaprasad, and V. M. Vokkarane, Dynamic circuit provisioning in All-Optical WDM networks using lightpath switching, *Optical Switching and Networking*, 9 (2) (2012) 179 – 190.

[38] N. Charbonneau and V. M. Vokkarane, Dynamic circuits with lightpath switching over wavelength routed networks, in: *IEEE ANTS 2010*, pp. 22-24.

[39] B.H. Ramaprasad, A. Gadkar and V. M. Vokkarane, Dynamic anycasting over wavelength routed networks with lightpath switching, in: *IEEE HPSR 2011*, pp. 77-82.

[40] N. Charbonneau and V.M. Vokkarane**,** Dynamic Non-Continuous Advance Reservation over Wavelength-Routed Networks, in: *IEEE ICCCN 2011*.

[41] B. H. Ramaprasad, A. Somani, and V.M. Vokkarane**,** Dynamic non-continuous single slot advance reservation over wavelength routed network, in: *IEEE ICNC 2012*, pp. 861 – 865.

[42] M. Ahmed, S.M.H. Chowdhury, and M. Hasan, Fast preemptive task scheduling algorithm for homogeneous and heterogeneous distributed memory systems, in: *Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2008*, pp. 720-725.

[43] L. Sha, T. Abdelzaher, K-E Arzen, A. Cervin, T. Baker, A. Burns, G. Buttazzo, M. Caccamo, J. Lehoczky, and A. K. Mok, Real time scheduling theory: A historical perspective, *Real-Time Systems*, 28 (1-2) (2004) 101-155.

[44] M.G. Harbour, M. H. Klein, and J. P. Lehoczky, Fixed priority scheduling of periodic tasks with varying execution priority, in: *Proc. 12th IEEE Real-Time Systems Symposium 1991*.

[45] C. Han and H. Tyan, A better polynomial-time scheduling test for real-time fixed-priority scheduling algorithms, in: *Proc. 18th IEEE Real-Time Systems Symposium 1997*, pp. 36-45.

[46] M.L. Dertouzos, Control robotics: the procedural control of physical processes, *Information Processing*, 74 (1974).

[47] A. Radulescu and A. J. C. van Gemund, Preemptive task scheduling for distributed systems, *Euro-Par 2000 Parallel Processing*, LNCS Volume 1900/2000 (2000) 272-276.

[48] K. Kaur, A. Chhabra, and G. Singh, Heuristics based genetic algorithm for scheduling static tasks in homogeneous parallel system, *International Journal of Computer Science and Security*, 4 (2) (2010) 183-198.

[49] H. Izakian, A. Abraham, and V. Snasel, Comparison of heuristics for scheduling independent tasks on heterogeneous distributed environments, in: *Proceedings of the 2009 International Joint Conference on Computational Sciences and Optimization*, pp. 8-12.

[50] C.L. McCreary, A.A. Khan, J. Thompson, and M.E. McArdle, A comparison of heuristics for scheduling DAGs on multiprocessors, in: *Proceedings on Parallel Processing Symposium 2002*, pp. 446-451.

[51] J. Carretero, F. Xhafa, and A. Abraham, Genetic algorithm based schedulers for grid computing systems, *International Journal of Innovative Computing, Information and Control*, 3 (6) (2007) 1053-1071.

[52] A. M. Rahmani and M. A. Vahedi, A novel task scheduling in multiprocessor systems with genetic algorithm by using Elitism stepping method, *INFOCOMP Journal of Computer Science* (2008).

[53] A. Ravula and B. Ramamurthy, A tabu search approach for joint scheduling of resources in a Lambda grid network, in: *IEEE Globecom 2010*, pp. 1-6.

[54] X. Liu, C. Qiao, W. Wei, X. Yu, T. Wang, W. Hu, W. Guo, and M. Wu, Task scheduling and lightpath establishment in optical Grids, *Journal of Lightwave Technology*, 27 (12) (2009) 1796 - 1805.

[55] V. Lakshmiraman and B. Ramamurthy, Joint computing and network resource scheduling in a Lambda Grid network, in: *Proceedings of IEEE International Conference on Communications 2009*, pp. 1-5.

[56] K. Zhu and B. Mukherjee, Traffic grooming in an optical WDM mesh network, *IEEE Journal on Selected Areas in Communications*, 20 (1) (2002) 122-133.

[57] J. Q. Hu and B. Leida, Traffic grooming, routing, and wavelength assignment in optical WDM mesh networks, in: *INFOCOM 2004*, pp. 495-501.

[58] A. Jaekel, A. Bari, Y. Chen and S. Bandyopadhyay, Strategies for optimal logical topology design and traffic grooming, *Photonic Network Communications,* 19 (2) (2010) 223-232.

[59] S. Zhang and B. Ramamurthy, Dynamic traffic grooming algorithms for reconfigurable SONET over WDM networks, *IEEE Journal on Selected Areas in Communications*, 21 (7) (2003) 1165-1172.

[60] B. Wang, T. Li, X. Luo, and Y. Fan, Traffic grooming under a sliding scheduled traffic model in WDM optical networks, in: *IEEE Workshop on Traffic Grooming in WDM Networks 2004*.

[61] B. Garcia-Manrubia, R. Aparicio-Pardo, P. Pavon-Marino, N. Skorin-Kapov and J. Garcia-Haro, MILP formulations for scheduling lightpaths under periodic traffic, in: *ICTON 2009*, pp. 1-4.

[62] P. Pavon-Marino, R. Aparicio-Pardo, B. Garcia-Manrubia and N. Skorin-Kapov, Virtual topology design and flow routing in optical networks under multi-hour traffic demand, *Photonic Network Communications*, 19 (1) (2010) 42-54.

[63] G. Agrawal and D. Medhi, Lightpath Topology configuration for wavelength-routed IP/MPLS newtworks with Time-dependent traffic, in: *GLOBECOM 2006*, pp. 1 - 6.

[64] F. Ricciato and S. Salsano, A. Belmonte and M. Listanti, Offline configuration of a MPLS over WDM network under Time-varying offered traffic, in: *INFOCOM 2002*, pp. 57 - 65.

[65] R. Aparicio-Pardo, N. Skorin-Kapov, P. Pavon-Marino and B. Garcia-Manrubia, (Non-) Reconfigurable virtual topology design under multihour traffic in optical networks, *IEEE/ACM Transactions, Networking*, 20 (5) (2012) 1567 - 1580.

[66] W. Yao and B. Ramamurthy, Survivable traffic grooming with path protection at the connection level in WDM mesh networks, in: *BROADNETS 2004*, pp. 310 - 319.

[67] W. Yao and B. Ramamurthy, Survivable traffic grooming in WDM mesh networks under SRLG constraints, in: *IEEE Int. Conf. on Communications 2005*, vol. 3, pp. 1751-1755.

[68] A. Jaekel, A. Bari and S. Bandyopadhyay, Resilient traffic grooming for WDM networks, *Journal of Optical Networking*, 7 (5) (2008) 378-387.

[69] M. Gupta and S. Singh, Greening of the Internet, in: *Proc. Of the ACM SIGCOMM 2003*, pp. 19 - 26.

[70] J. Baliga, R. Ayre, K. Hinton, W. Sorin and R. Tucker, Energy consumption in optical IP networks, *Journal of Lightwave Technology,* 27 (13) (2009) 2391-2403.

[71] B. Bathula, M. Alresheedi and J. Elmirghani, Energy efficient architectures for optical networks, in: *Proc. London Communications Symposium 2009*.

[72] F. Musumeci, M. Tornatore and A. Pattavina, A power consumption analysis for IP-Over-WDM core network architectures, *Journal of Optical Communications and Networking*, 4 (2) (2012) 108-117.

[73] W. V. Heddeghem, F. Idzikowski, W. Vereecken, D. Colle, M. Pickavet and P. Demeester, Power consumption modeling in optical multilayer networks, *Photonic Network Communications*, 24 (2) (2012) 86 - 102.

[74] A. Muhammad, P. Monti, I. Cerutti, L. Wosinska, P. Castoldi, and A. Tzanakaki, Energy-efficient WDM network planning with dedicated protection resources in sleep mode, in: *IEEE Globecom 2010*.

[75] C. Cavdar, F. Buzluca, and L. Wosinska, Energy-efficient design of survivable WDM networks with shared backup, in: *GLOBECOM 2010*.

[76] A. Orgerie, L. Lefevre and I. Guerin-Lassous, Energy-efficient bandwidth reservation for bulk data transfers in dedicated wired networks, *Journal of Supercomputing*, 62 (3) (2012) 1139 - 1166.

[77] A. Gadkar, J. M. Plante, and V. M. Vokkarane, Manycasting: Energy-efficient multicasting in WDM optical unicast networks, in: *GLOBECOM 2011*, pp. 1 – 6.

[78] W. Xie, Y. Zhu, and J. P. Jue, Energy-efficient impairment-constrained 3R regenerator placement in optical networks, in: *IEEE ICC 2012*, pp. 3020 – 3024.

[79] A. Coiro, M. Listanti, A. Valenti and F. Matera, Power-aware routing and wavelength assignment in multi-fiber optical networks, *Journal of Optical Communication and Networking*, 3 (11) (2011) 816 - 829.

[80] A. Coiro, M. Listanti and A. Valenti, Dynamic power-aware routing and wavelength assignment for green WDM optical networks, in: *IEEE ICC 2011*, pp. 1 - 6.

[81] R. A. Scaraficci, N. L. S. da Fonseca, and M. R. Marcos, Algorithm for energy efficient lightpath establishment in WDM networks, in: *IEEE ICC 2012*, pp. 1454 – 1459.

[82] Q. Ho and M. Lee, A zone-based approach for scalable dynamic traffic grooming in large WDM mesh networks, *Journal of Lightwave Technology*, 25 (1) (2007) 261 – 270.

[83] S. Zhang, D. Shen and C-K Chan, Energy-efficient traffic grooming in WDM networks with scheduled time traffic, *Journal of Lightwave Technology*, 29 (17) (2011) 2577-2584.

[84] A. Jaekel**,** Y. Chen and A. Bari, Survivable traffic grooming for scheduled demands, in: *5th Int. Conf. on Broadband Communications, Networks and Systems 2008*, pp. 176 – 183.

[85] R. K. Ahuja, T. L. Magnanti and J. B. Orlin, Network flows, Prentice- Hall, 1993.

[86] M. Sridharan, M.V. Salapaka and A. Somani, A practical approach to operating survivable WDM networks, *IEEE Journal on Selected Areas in Communications*, 20 (1) (2002) 34-46.

[87] Ilog cplex 9.1. Documentation available at http://docs.hpc.maths.unsw.edu.au/ilog/cplex/9.1/pdf/usrcplex.pdf.

[88] Y. Chen and A. Jaekel, Energy aware resource allocation based on demand bandwidth and duration, in: *2nd International Workshop on Green Computing and Renewable Energy 2012*.

[89] I. Clamtac, A. Ganz and G. Karmi, Lightpath communications: An approach to high bandwidth optical WAN's, *IEEE Trans. on Communications,* 40 (7) (1992) 1171-1182.

[90] G. L. Nemhauser and L. A. Wolsey, Integer and combinatorial optimization, John Wiley and Sons, 1988.

[91] S. Ramaswami and B. Mukherjee, Survivable WDM mesh networks, Part I-Protection, in: *INFOCOM 1999*, pp. 744-751.

[92] A. K. Somani, Survivability and traffic grooming in WDM mesh networks, Cambridge University Press, 2006.

[93] B. Wang and T. Li, Survivable scheduled service provisioning in WDM optical networks with iterative routing, *Optical Switching and Networking*, 7 (1) (2010) 28-38.

[94] J.Y. Zhang, H. Mouftah, J. Wu and M. Savoie, Lightpath scheduling and routing for traffic adaptation in WDM networks, *Optical Communications and Networking*, 2 (10) (2010) 803-819.

[95] Y. Wang, Y. Jin, W. Guo, W. Sun and W. Hu, Joint scheduling for optical grid applications, *Journal of Optical Networking*, 6 (3) (2007) 304 - 318.

[96] Y. Chen, A. Bari, and A. Jaekel, Techniques for designing survivable optical grid networks, *Journal of Communications, Special Issue on Future Directions on Computing and Networking*, 7 (5) (2012) 391-399.

[97] C. Develder, B. Dhoedt, B. Mukherjee and P. Demeester, On dimensioning optical grids and the impact of scheduling, *Photonic Network Communications*, 17 (3) (2009) 255 - 265.

VITA AUCTORIS

Ying Chen obtained her B.C.S [H] Co-op and M.Sc. degrees in Computer Science from the University of Windsor in 2007 and 2009, respectively. She is currently a candidate for the Doctoral degree in Computer Science at the University of Windsor, Ontario and hopes to graduate in May, 2013.