ABSTRACT

WANG, HUI. Efficient Decomposition Techniques for Traffic Grooming Problems in Optical Networks. (Under the direction of Dr. George N. Rouskas.)

Traffic grooming refers to a class of optimization problems developed to address the gap between the wavelength capacity and the traffic demands of individual connections in optical WDM networks. In this work, we focus on static traffic grooming problem and its subproblems, with an objective function of minimizing the total number of lightpaths. A lightpath is a path of physical links with a particular wavelength reserved on each link, and it is used as a metric of network cost.

We first review the existing work in this area and a typical integer linear program (ILP) formulation in the literature, and we identify two challenges related to this formulation in terms of scalability and wavelengths fragmentation. We then make three contributions that address the scalability challenges.

First, we propose a new solution approach that decomposes the traffic grooming problem into two subproblems that are solved sequentially: (1) the virtual topology and traffic routing ($VTTR$) subproblem, that does not take into account physical topology constraints, and (2) the routing and wavelength assignment (RWA) subproblem, that reconciles the virtual topology determined by $VTTR$ with the physical topology. The decomposition is exact when the network is not wavelength limited. We also propose three algorithms that use a partial LP relaxation technique driven by lightpath utilization information to solve the $VTTR$ subproblem efficiently. Our approach delivers a desirable tradeoff between running time and quality of the final solution.

Second, instead of viewing the network as a flat entity, we further consider networks with hierarchies, where the second level of hierarchy consists of hub nodes, and the first level is formed by the remaining nodes. Hierarchical traffic grooming facilitates the control and management of multigranular WDM networks. We first survey heuristic hierarchical grooming algorithms. We then apply the above decomposition approach onto hierarchical traffic grooming. We define the hierarchical virtual topology and traffic routing (H-VTTR) problem, and we present a suite of ILP formulations to solve it. The formulations represent various tradeoffs between solution quality and running time. We also explore the performance of the formulations under various direct lightpath thresholds, traffic loads, and number of hubs. The formulations are cross-compared with the baseline VTTR formulation.

Finally, we consider the traffic grooming problem with multi-class traffic, where traffic classes may be defined across various dimensions, including: (1) quality of Service (QoS) requirements, in which case the network may provision different lightpaths to support different levels of QoS; (2) the user or group of users, in which case the network operator may groom traffic of different organizations (e.g., competitors) onto different sets of lighgtpaths; and/or (3) privacy or security considerations. Given such a classification of user traffic, and regardless of how the various classes are defined, we impose the additional constraints that grooming onto a given lightpath is permitted if and only if the traffic to be groomed belongs to the same class. We refer to this new problem as multi-class virtual topology and traffic routing (MC-VTTR). We develop a solution to MC-VTTR and quantify the overhead of supporting multiple classes in terms of solution quality and running time.

Efficient Decomposition Techniques
for Traffic Grooming Problems
in Optical Networks

by
Hui Wang

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

Operations Research
Computer Science

Raleigh, North Carolina

2013

APPROVED BY:

Dr. Rudra Dutta Co-chair of Advisory Committee	Dr. Yahya Fathi
Dr. William J. Stewart	Dr. George N. Rouskas Co-chair of Advisory Committee

DEDICATION

To my parents,
Ping Zhang and Haoxu Wang,
for their endless love and support.

BIOGRAPHY

Hui Wang is a Ph.D. candidate in Opertions Research with co-major in Computer Science at North Carolina State University (NCSU). She was born in Shijiazhuang, Hebel, China in 1985, and spent the first 18 years of life in her hometown. She received her B.S. degree in Applied Mathematics from Peking University in July 2007. She directly joined the Ph.D. program in Operations Research at NCSU after graduating from college. She received her Master degrees in Operations Research and Statistics from NCSU in 2009 and 2011, respectively.

She has successfully completed the Graduate Leadership Development Series program in Spring, 2013. She received a 2013 SAS Summer Fellowship in Revenue Management and Pricing Optimization, and was recognized as a SAS Student Ambassador in 2011. She is a member of the Honor Societies Mu Sigma Rho, Omega Rho, and Phi Kappa Phi. She has interned at United Airlines Inc., for two summers in 2010 and 2011. During her graduate study, she has been actively involved in student organizations. She is the secretary of NCSU Women in Computer Science. She served as the President of the Operations Research Graduate Student Association and the NCSU INFORMS Student Chapter, and the Chair of the University Graduate Student Association Teaching Effectiveness Committee in the academic year of 2009. She is also a student member of the Institute of Electrical and Electronics Engineers (IEEE) and the Institute for Operations Research and the Management Sciences (INFORMS)

After graduation, she will join Quantcast as a Modeling Engineer. Quantcast is a late-stage, pre-IPO startup in San Francisco that is disrupting the multibillion dollar advertisement industry by perfecting targeted display advertising.

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. George N. Rouskas, for his dedicated guidance and mentorship, especially in multi-disciplinary study and research. Without your help, this thesis would not be possible. I am grateful for your encouragement in my exploring many opportunities. You have definitely set my direction and attitude in professional career. I am also very grateful to the rest of my committee: Dr. Rudra Dutta, for his valueable suggestions; Dr. Yahya Fathi and Dr. William Stewart, for their insightful comments and questions.

Much gratitude to my parents, Ping Zhang and Haoxu Wang, who raised me, provided me the best education opportunities, and gave me constant love and support. Thank you for believing in me.

To Shengfan Zhang, Andrea Villanes, Queqing Wang, Yuan Geng, Ziyu Xiao, Min Li, Yao Liu and all of my friends, thank you for bringing so much fun into my graduate study, and thank you for sharing my happy and sad moments. Special thanks to Baris Kacar, you made me want to be a better person.

At last, I'd like to thank NSF (Grant CNS-1113191) for the financial support of my research.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Traffic Grooming, ILP Formulation, and Challenges

1.1 Introduction and Related Work

In the modern world, communication services delivered via the Internet touch all of society and affect all aspects of human life. To accommodate the exponential growth of demand in communications, infrastructure that can support ever increasing amounts of traffic is highly needed. Optical networks have been commonly used as the backbone infrastructure of Internet services, since they deliver high performance in terms of both throughput and QoS. The physical structure of an optical network consists of a set of nodes and optical fibers that interconnect them. With the help of wavelength division multiplexing (WDM) technology, it is possible to transmit traffic on different wavelengths within the same optical fiber simultaneously. The data rate of a single wavelength can be up to 40 Gbps, while higher rates are becoming commercially available. Therefore, the capacity of each wavelength can be significantly higher than the magnitude of individual traffic demands.

The concept of traffic grooming was introduced in the mid-1990s to address the gap between the channel capacity and individual traffic demands in optical networks. The key idea is to aggregate individual traffic requests onto wavelengths so as to improve bandwidth utilization across the network and minimize the use of network resources. Many variants of traffic grooming have been studied in the literature. Online versions of the problem target network environments in which traffic demands arrive in real time. Since future demands are not known in advance, the main objective of online problems is to minimize blocking probability or maximize throughput. Heuristics for solving online traffic grooming problems have been proposed in [16, 21, 22, 44].

Offline traffic grooming is a fundamental network design problem that has been shown to be NP-hard [15]. Network design problems are typically formulated as integer linear programs

(ILPs) and assume the existence of a traffic matrix representing the demands between node pairs. Basic ILP formulations of the problem are available in [14] and [46]. Typically, the objective is to minimize the total network cost while satisfying all demands (e.g., as in [10,25]), or to maximize the total revenue by satisfying as many traffic demands as possible given certain capacity (wavelength) constraints (e.g., as in [46]). Since electronic equipments that terminate lightpaths represent a large fraction of the overall network cost, the number of lightpaths established to carry the traffic demands is usually taken as the metric to minimize [25]. Other cost functions have also been considered, including the electronic switching cost of grooming traffic between lightpaths at intermediate switches [13], and power consumption in optical networks [43].

One essential concern about the ILP formulations is that they are solvable only for small network topologies [43]. For larger topologies representative of deployed networks, the ILP formulation cannot be solved to optimality within reasonable amounts of time (e.g., a few hours). Therefore, the offline problem has been addressed either using heuristic algorithms [8,45] or by manipulating the ILP formulation using decomposition or column generation techniques. In [21], the original ILP is decomposed into two simpler ILPs: one that addresses only the traffic routing and lightpath routing subproblems and is solved first; and another that addresses the wavelength assignment problem only and takes as input the solution of the first ILP. In [41], a multi-level decomposition method is introduced to address the multi-layered routing and multi-rate connection characteristics of traffic grooming. In [29], the objective is to design a ring network that is able to satisfy any request graph with maximum degree at most δ. The cases of $\delta = 2$ and $\delta = 3$ were solved by graph decomposition.

Column generation techniques were developed in [12,37]. Given that the main difficulty in solving the problem has to do with selecting from among an exponential number of possible paths to route the traffic demands, a heuristic algorithm using column generation for a path-based formulation of the problem was developed in [12]. The key idea was to generate an optimal subset of paths efficiently. A hierarchical optimization method was proposed in [37]. The method first deals with the grooming and routing decisions using column generation to find the dual bounds and a rounding heuristic to find an integral solution; wavelength assignment was then carried out in a second step.

Traffic grooming problem has been studied over various communication modes. The point-to-point mode has been extensively studied on SONET ring network. In addition to point-to-point traffic demands, traffic grooming has been studied in the context of other communication modes, including: one-to-many, many-to-one, and many-to-many. For the one-to-many communication mode, [24] serves as a review for typical multicast traffic grooming problems; in [3], an exact approach that assumed multicast sessions were routed in advance and allowed wavelength conversion and optical splitting is presented. [32] investigates the dynamic multicast traffic grooming problem for mesh networks. In many-to-one mode, several source nodes communicate with one single destination node, like in resource discovery and data collection. [36] addresses the many-to-one traffic grooming problem such that all traffic requests are satisfied with the least number of required hight layer light termination equipments, and together minimizing the total number of used wavelengths, with a dynamic programming style heuristic algorithm. Many-to-many communication allows several users interact with each other. For example, distance learning, multimedia conference, and distributed computing all require many-to-many communication mode. In many-to-many communication mode, also called group communication mode, a session is formed by members (which refer to a group of users), where each member in this group transmits its traffic to every other member in the same group, [33]. Usually, the requirements for bandwidth for these user applications are much lower than the capacity of single wavelength. Thus, it becomes substantial to groom users' traffic demands at optical level. Many-to-many communication requires that each node in WDM network is capable of duplicating the incoming traffic into several copies, each copy going to a different output port.

Existing approaches to obtaining optimal solutions to the traffic grooming problem face serious scalability challenges as well as wavelength fragmentation issues; we discuss each challenge in Sections 1.5.1 and 1.5.2, respectively. The lack of scalability of optimal methods makes it difficult to characterize the performance of heuristic algorithms in realistic topologies, and severely limits the application of "what-if" analysis to explore the sensitivity of network design decisions to forecast traffic demands, capital and operational cost assumptions, and service price structures.

1.2 The Concept of Traffic Grooming

Traffic grooming [31] refers to a class of optimization problems developed to address the gap between the wavelength capacity and the traffic demands of individual connections in optical WDM networks. Fiber links currently support multiple wavelengths, each operating at data rates in the order of 10-40 Gbps, with rates of 100 Gbps or higher likely to be commercially available in the near future. Although certain applications, including some related to large e-Science experiments or peta-scale data mining and analytics, can make efficient use of the large capacity of each wavelength channel, the vast majority of connections has bandwidth requirements that are small compared to the available data rates. Hence, there is a need for packing the sub-wavelength traffic demands into lightpaths for efficient transport across the optical network.

Wavelength division multiplexing (WDM) is the technique to transmit traffic on different wavelengths on a single optical fiber simultaneously. The WDM technology makes it possible to have several traffic components multiplexed on the same fiber, each of them assigned to a different wavelength. The physical topology of optical WDM networks consists of several nodes that are connected with optical fibers, where the fibers are called links. Each link can have at most W wavelengths, and the traffic signals in the network propagate through the optical fiber at different wavelengths. Each wavelength has a capacity C, where C is measured in units of traffic. The detailed structure of each node is shown in Figure 1.1. Basically, each node is equipped with two parts, an optical cross connect (OXC), and a digital cross connect (DXC). Each OXC can transfer the optical signal from the incoming fiber onto the same wavelength to the outgoing fiber. If the OXC has a converter, then it can pass the optical signal onto a different wavelength on the outgoing fiber. The OXC device makes it possible to create an optical channel path along links.

A lightpath is a path of physical links in which a particular wavelength on each link is reserved for this lightpath, so that the signal carried on the lightpath remains in optical from the source node to the destination node. On each lightpath, no optical to electronic signal conversion takes place. When the lightpath is terminated at its destination node, the optical signals are converted into electronic signals by transceivers for further processing.

By establishing lightpaths, we build a virtual topology over the physical topology of fiber links, to allow direct communication between two not directly fiber connected nodes by a clear channel of reserved wavelengths on each fiber link, so that various pairs of nodes can be connected. Since the traffic signals are transferred at different wavelength on the optical fibers, thus, the network can be alternatively considered as a set of nodes interconnected by lightpath. Each traffic component may traverse through a series of lightpaths from source to destination nodes. For example, in Fig. 1.1, the traffic component from node 1 to node 4 is carried over the connection shown in a dashed line. The connection uses two lightpaths: first from node 1 to node 3, second from node 3 to node 4. At node 2, the signal is switched optically, but at node 3, the first lightpath is terminated and the signal is switched to the second lightpath electronically. Several lightpaths can share the same fiber link, but with different wavelengths assigned, it is easy to see that the number of lightpaths shared on one single link cannot exceed the number of wavelengths supported by the optical fiber.

Optical fiber links support multiple wavelengths. Currently, each link operates at data rates in order of 10 - 40 Gbps, and in the near future, 100 Gbps is likely to be deployed for commercial use. Compared with the large capacity of each wavelength channel on an optical fiber link, most of the communications between different nodes require much smaller bandwidth. Thus, it is a natural idea to combine low speed traffic components to fill in the large bandwidth of single wavelength, in order to utilize the wavelength capacity as much as possible. This gives rise to the concept of traffic grooming in optical networks.

Referring to Fig. 1.1 as an example, the lightpath from node 1 to node 3 currently carries

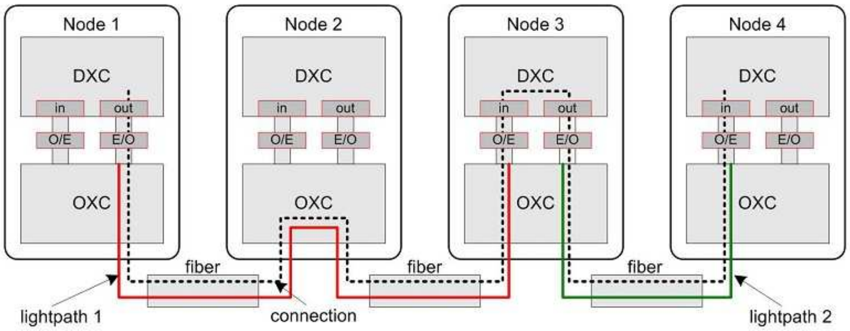

Figure 1.1: Illustration of an optical WDM network and traffic grooming

the traffic demand from node 1 to node 4, however, it is possible to groom traffic demand from node 1 to node 3 onto this lightpath, as long as there is still bandwidth left. Similarly, the second lightpath from node 3 to node 4 may carry other traffic demands as well, for example, traffic demand from node 2 to node 4 can be carried on a lightpath from node 2 to node 3 and the current lightpath from node 3 to node 4. We note that a lightpath from node i to node j does not necessarily carry traffic demands originating at node i or terminating at node j. As a result of such grooming, the overall network utilization may be improved significantly than the case that each traffic demand is carried on a direct lightpath from source node to destination node.

1.3 Basic ILP Formulation and Challenges

Consider a connected graph $G = (\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ denotes the set of nodes and $\mathcal{L}$ denotes the set of directed links (arcs) in the network. We define $N = |\mathcal{N}|$ and $L = |\mathcal{L}|$ as the number of nodes and links, respectively. Each directed link l consists of an optical fiber that may support W distinct wavelengths indexed as $1, 2, 3, \ldots, W$. Let $T = [t^{sd}]$ denote the traffic demand matrix, where t^{sd} is a non-negative integer representing the traffic demand units from source node s to destination node d. In general, traffic demands may be asymmetric, i.e., $t^{sd} \neq t^{ds}$. We also make the assumption that $t^{ss} = 0, \forall s$. Finally, we denote C as the capacity of a single wavelength channel in terms of traffic demand units.

We are interested in minimizing the total number of lightpaths used in the network; such an objective minimizes the use of critical resources and provides ample flexibility for future expansion of the network. Hence, we consider the following minimization problem that we refer to as *TG*.

Problem 1.3.1 (TG) *Given graph G, number of wavelengths W, wavelength capacity C, and traffic demand matrix T, establish the minimum number of lightpaths to carry all traffic demands.*

Let us define the following sets of decision variables:

- t_{ij}^{sd}: integer variable that indicates the amount of traffic, as a multiple of unit demand, from node s to node d carried on lightpaths from node i to node j;
- b_{ij}: integer variable that indicates the number of lightpaths from node i to node j;
- b_{ij}^{l}: integer variable that indicates the number of lightpaths from node i to node j which traverse link l; and
- $c_{ij}^{l,w}$: binary variable that indicates whether a lightpath from node i to node j uses wavelength w on link l.

Let us further denote the set of links going out of, and coming into, node n as L_n^+ and L_n^-, respectively. With these notations, the *TG* problem can be formulated as the following ILP, adapted from [14]:

$$\text{minimize:} \quad \sum_{i,j \in \mathcal{N}} b_{ij} \tag{1.1}$$

Subject to:

Virtual topology and traffic routing constraints:

$$\sum_{s,d} t_{ij}^{sd} \leq b_{ij} C, \quad i,j \in \mathcal{N} \tag{1.2}$$

$$\sum_{j} t_{ij}^{sd} - \sum_{j} t_{ji}^{sd} = 0, \quad i \in \mathcal{N} \setminus \{s,d\}, s,d \in \mathcal{N} \tag{1.3}$$

$$\sum_{j} t_{sj}^{sd} = t^{sd}, \quad s,d \in \mathcal{N} \tag{1.4}$$

$$\sum_{j} t_{js}^{sd} = 0, \quad s,d \in \mathcal{N} \tag{1.5}$$

$$\sum_{j} t_{dj}^{sd} = 0, \quad s,d \in \mathcal{N} \tag{1.6}$$

$$\sum_{j} t_{jd}^{sd} = t^{sd}, \quad s, d \in \mathcal{N} \tag{1.7}$$

Lightpath routing constraints:

$$\sum_{l \in L_n^+} b_{ij}^l - \sum_{l \in L_n^-} b_{ij}^l = 0, \quad n \in \mathcal{N} \setminus \{i, j\}, i, j \in \mathcal{N} \tag{1.8}$$

$$\sum_{l \in L_i^+} b_{ij}^l = b_{ij}, \quad i, j \in \mathcal{N} \tag{1.9}$$

$$\sum_{l \in \mathcal{L}_i^-} b_{ij}^l = 0, \quad i, j \in \mathcal{N} \tag{1.10}$$

$$\sum_{l \in L_j^+} b_{ij}^l = 0, \quad i, j \in \mathcal{N} \tag{1.11}$$

$$\sum_{l \in L_j^-} b_{ij}^l = b_{ij}, \quad i, j \in \mathcal{N} \tag{1.12}$$

Wavelength assignment constraints:

$$\sum_{w} c_{ij}^{w,l} = b_{ij}^l, \quad i, j \in \mathcal{N}, l \in \mathcal{L} \tag{1.13}$$

$$\sum_{i,j} c_{ij}^{w,l} \leq 1, \quad \forall w, \quad l \in \mathcal{L} \tag{1.14}$$

$$\sum_{l \in L_n^+} c_{ij}^{w,l} = \sum_{l \in L_n^-} c_{ij}^{w,l}, \quad n \in \mathcal{N} \setminus \{i, j\}, i, j \in \mathcal{N}, \forall w \tag{1.15}$$

$$\sum_{l \in L_i^+} c_{ij}^{w,l} \leq b_{ij}, \quad i, j \in \mathcal{N}, \forall w \tag{1.16}$$

$$\sum_{l \in L_i^-} c_{ij}^{w,l} = 0, \quad i, j \in \mathcal{N}, \forall w \tag{1.17}$$

$$\sum_{l \in L_j^+} c_{ij}^{w,l} = 0, \quad i, j \in \mathcal{N}, \forall w \tag{1.18}$$

$$\sum_{l \in L_j^-} c_{ij}^{w,l} \leq b_{ij}, \quad i,j \in \mathcal{N}, \forall w \tag{1.19}$$

Integrality constraints:

$$t_{ij}^{sd},\ b_{ij},\ b_{ij}^{l} : \text{integer}; \quad c_{ij}^{l,w} : 0, 1; \quad w = 1, 2, \ldots, W. \tag{1.20}$$

The virtual topology and traffic routing constraints (1.2)-(1.7) determine the lightpaths to be established (i.e., the virtual topology of the network) and the routing of traffic demands on the virtual topology. The capacity constraint (1.2) ensures that a sufficient number of lightpaths is established between each node pair. Constraints (1.3)-(1.7) are multi-commodity flow equations that find the route on the virtual topology of lightpaths for each traffic demand.

The routing constraints (1.8)-(1.12) are multi-commodity flow equations that determine the physical route for each lightpath, where each lightpath corresponds to a single commodity. Constraint (1.8) ensures that the number of incoming lightpaths is equal to the number of outgoing lightpaths at any intermediate node. Constraints (1.9)-(1.10) and (1.11)-(1.12) are the lightpath constraints at the origin node and sink node, respectively, of each lightpath.

The wavelength assignment constraints (1.13)-(1.19) enforce the two wavelength constraints: (a) expression (1.14) represents the distinct wavelength constraint which guarantees that each wavelength may only be used once on any link, i.e., no two lightpaths sharing a common link may use the same wavelength; and (b) multi-commodity flow equations (1.15)-(1.19) represent the wavelength continuity constraint by ensuring that each link on the same lightpath is assigned the same wavelength. Finally, constraint (1.13) ensures that each lightpath will only use one wavelength.

The above formulation, and most formulations studied in the literature that use link-related variables, suffer from two main challenges: scalability and wavelength fragmentation. We discuss each of these challenges in the next two subsections, respectively.

1.4 Traffic Grooming Subproblems by Logical Decomposition

Logically, the traffic grooming problem consists of four conceptual subproblems: virtual topology design subproblem, traffic routing subproblem, lightpath routing subproblem, and wavelength assignment subproblem. To solve for optimality, the four subproblems cannot be considered independently from each other, instead, they need to be considered jointly.

- **Virtual Topology Design:** establish the minimum number of lightpaths to carry all traffic, as shown in Fig. 1.2.

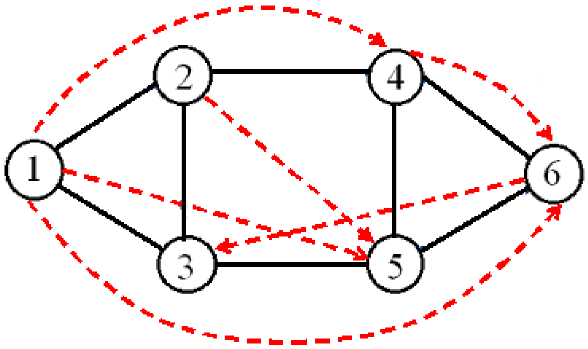

Figure 1.2: Illustration of virtual topology design and traffic routing subproblem

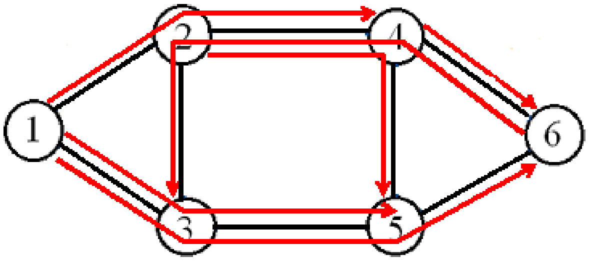

Figure 1.3: Illustration of lightpath routing subproblem

- **Traffic Routing:** determine the routing of individual traffic demands on the lightpaths determined by the virtual topology design subproblem.

- **Lightpath Routing:** determine the routing of lightpaths over the physical links, as shown in Fig. 1.3. For instance, the lightpaths from node 1 to node 6 in Fig. 1.3 is routed over the physical path $(1, 3, 5, 6)$.

- **Wavelength Assignment:** assign wavelengths to lightpaths without clash, as shown in Fig. 1.4. For instance, the lightpath from node 1 to node 6 is assigned the red wavelength in Fig. 1.4.

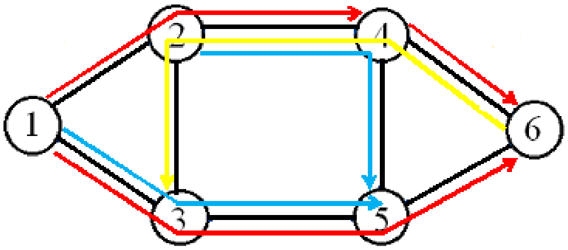

Figure 1.4: Illustration of wavelength assignment subproblem

1.5 Challenges

1.5.1 Scalability

The scalability of the formulation depends directly on its size, which, in turn, is determined by the number of variables and constraints. The above formulation consists of $N^2(N-1)^2$ integer variables $\{t_{ij}^{sd}\}$, $N(N-1)$ integer variables $\{b_{ij}\}$, $N(N-1)L$ integer variables $\{b_{ij}^l\}$, and $N(N-1)LW$ binary variables $\{c_{ij}^{l,w}\}$, for a total of $O(N^4 + N^2 LW)$ variables. Also, there are $O(N^3)$ routing constraints, $O(N^3 W)$ wavelength constraints, and $O(N^3)$ grooming constraints, for a total of $O(N^3 W)$ constraints in the formulation. Given that current technology may support up to $W = 100$ wavelengths per link, it becomes clear that the ILP formulation can be applied directly only to very small networks. In our experience from [43], this ILP formulation may take tens of hours to solve, using commodity hardware, on networks with as few as a dozen nodes.

To illustrate the scalability challenges of the formulation, we used CPLEX to solve problem instances on ring networks of size between $N = 4$ and $N = 12$ nodes. Since the ring topology is relatively sparse, the running time on ring networks is a (generally, quite loose) lower bound of the running time on mesh networks of the same size. All problem instances were solved by running the IBM Ilog CPLEX 12 optimization tool on a cluster of identical compute nodes, each with dual Woodcrest Xeon CPU at 2.33GHz with 1333MHz memory bus, 4GB of memory and 4MB L2 cache. Figure 1.5 plots, in logarithmic scale, the CPU running time (in seconds) it takes for CPLEX to find an optimal solution, as a function of the ring size N; each data point is the average of ten problem instances for the given network size. We imposed a limit of 100 CPU hours for CPLEX to find a solution; if it fails to do so within the limit, the process was terminated and this fact was indicated by plotting the corresponding data point in the light gray area in the figure labeled ”tLim”. As we can see, an increase in the size of the ring

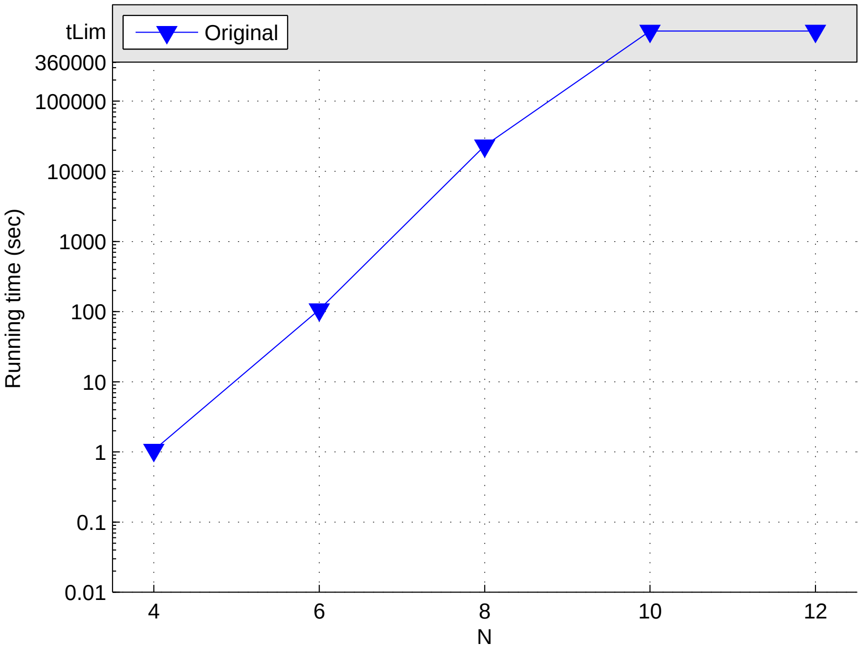

Figure 1.5: CPLEX running time to find an optimal solution, as a function of N the number of nodes of the ring network

network of two nodes results in a two order of magnitude increase in the running time. As a result, it is not possible to solve 10-node ring networks within the 100-CPU-hour limit (more than four days) that we imposed. Even for an 8-node ring, it takes about 10 CPU hours to find an optimal solution, a significant amount of time for such a small and sparse network.

1.5.2 Wavelength Fragmentation

Now let us turn our attention to a second issue with the basic formulation. Note that the objective of the ILP is a function that depends only on the number of lightpaths and is independent of the number of wavelengths used to color the lightpaths. Specifically, the number of available wavelengths W is part of the input to the problem, and the only constraint is to color the lightpaths using no more than W distinct colors. Consequently, the ILP solver will not make any attempt to minimize the number of wavelengths as long as the wavelength constraints are satisfied. In other words, the ILP solver will stop as soon as it has determined a solution with the minimum number of lightpaths, without checking whether there exists another solution with the same objective value that uses fewer wavelengths. Therefore, the number of wavelengths in the final solution may be higher than necessary, and as a result, a critical network resource

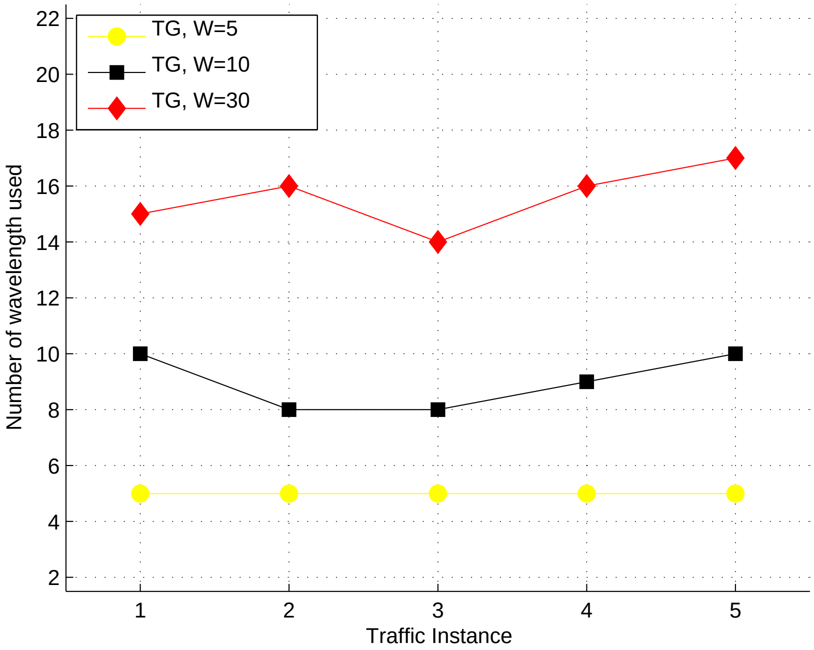

Figure 1.6: Wavelength usage comparison for five traffic instances, ring network with $N = 6$ nodes

may be severely fragmented.

To illustrate how serious this issue can be, we used the above ILP formulation to solve five problem instances on a 6-node ring network, each with a different traffic matrix. We solved each instance three times, each time providing as input a different value for the number W of available wavelengths, namely, $W = 5, 10, 30$. All instances can be solved with fewer than $W = 5$ wavelengths (as we will show later), hence all three solutions we obtained for a given instance used the same number of lightpaths.

Figure 1.6 plots the number of wavelengths used by each of the three optimal solutions to the five problem instances. It is clear that the number of wavelengths used in the solution is an increasing function of the number W of available wavelengths input to the formulation. In particular, the optimal solution obtained with a high input value of W may use more than three times as many wavelengths as the optimal solution obtained with a low input value of W. Given that the minimum number of wavelengths required for a feasible solution is not known *a priori*, the designer may be tempted to use a high input value for W (as long as that many wavelengths are indeed available) to ensure that the ILP solver will find a solution. Such an approach will result in severe fragmentation of wavelengths, one of the critical resources in optical networks, increasing the cost of deployment and limiting future expansion of the network. We emphasize that all ILP formulations in which the number W of wavelengths is taken as a constraint face similar fragmentation challenges.

1.6 Thesis Contributions and Organization

As we have shown in Section 1.5.1 and Section 1.5.2, the classic ILP formulation of traffic grooming problem suffers from two issues: the scalability, and wavelength fragmentation. However, the novel decomposition approach we propose in Chapter 2 provides solutions to both of the two issues. It decomposes the original traffic grooming problem into two subproblems: the VTTR subproblem and the RWA subproblem, and solves them sequentially. As we will prove in Chapter 2, given the number of wavelengths is not limited, the solution obtained from this sequential decomposition approach is optimal to the original traffic grooming problem. In order to tackle the scalability issue, three iterative algorithms based on partial LP relaxation are presented to deliver a desirable tradeoff between running time and quality of solution of VTTR subproblem. As for the wavelength fragmentation issue, since the objective function of the second subproblem - RWA problem, is minimizing the number of total wavelengths, the wavelength fragmentation problem is naturally resolved.

We further extend the decomposition approach to hierarchical network topologies, where nodes are divided into hub and non-hub groups for the purpose of facilitating the control and management of multigranular WDM networks. Correspondingly, we define the hierarchical virtual topology and traffic routing (H-VTTR) problem. We also formally formulated the H-VTTR problem into ILP for the first time. We also present several variants of the ILP formulation to demonstrate various tradeoffs between running time and solution quality. We then provide the performance study of all the H-VTTR ILP formulations and baseline VTTR formulation under different direct lightpath thresholds, traffic loads, and number of hubs.

The third contribution is on supporting multi-class traffic. The need of grouping traffic demands into different classes may arise naturally under several conditions, including QoS requirements, the user or group of users, and the privacy or security considerations. In order to accomodate different traffic classes, we only allow traffic components within the same class to be groomed onto a lightpath. We then define multi-class virtual topology and traffic routing (MC-VTTR) problem, and provide the ILP formualation of it. We also quantify the cost of supporting multiple traffic classes in terms of solution quality and running time.

The rest of this thesis is organized as follows. In Chapter 2, we propose a novel decomposition approach for traffic grooming problem, and efficient iterative algorithms based on partial LP relaxation are presented. Chapter 3 focuses on hierarchial traffic grooming, and a suite of ILP formulations representing different tradeoffs between solution quality and running time are presented. In Chapter 4, we explore the cost by support multi-class traffic grooming. At last, in Chapter 5, we discuss directions of future work and conclude this thesis.

Chapter 2

A Novel Decomposition Approach and Efficient Algorithms

As we have shown in Chapter 1, existing approaches of obtaining optimal solutions to the traffic grooming problem face severe scalability challenges as well as wavelength fragmentation issues. In this chapter, we develop a new decomposition algorithm and partial LP relaxation technique for the traffic grooming problem.

2.1 A New Decomposition of Traffic Grooming

We decompose the *TG* problem defined earlier in Section 1.3 into two subproblems, the *virtual topology and traffic routing (VTTR)* subproblem, and the *routing and wavelength assignment (RWA)* subproblem.

2.1.1 Virtual Topology and Traffic Routing (VTTR)

The *VTTR* subproblem is defined as follows:

Definition 2.1.1 (VTTR) *Given the number N of nodes in the graph G of TG, the wavelength capacity C, and traffic demand matrix T, establish the minimum number of lightpaths to carry all trafic demands.*

The *VTTR* subproblem can be expressed as minimizing the objective function (2.1) under the virtual topology and traffic routing constraints (2.2)-(2.7). The output of the problem is a set of lightpaths $\{b_{ij}\}$, as well as the routing $\{t^{sd}_{ij}\}$ of the traffic demands $\{t^{sd}\}$ over these lightpaths. Since the traffic routing subproblem of *VTTR*, i.e., routing of demands on a given set of lightpaths, is NP-hard [15], *VTTR* itself is NP-hard. The ILP formulation for *VTTR* is presented as follows.

$$\text{minimize:} \quad \sum_{i,j \in \mathcal{N}} b_{ij} \tag{2.1}$$

Subject to:

Virtual topology and traffic routing constraints:

$$\sum_{s,d} t_{ij}^{sd} \leq b_{ij} C, \quad i, j \in \mathcal{N} \tag{2.2}$$

$$\sum_j t_{ij}^{sd} - \sum_j t_{ji}^{sd} = 0, \quad i \in \mathcal{N} \setminus \{s, d\}, s, d \in \mathcal{N} \tag{2.3}$$

$$\sum_j t_{sj}^{sd} = t^{sd}, \quad s, d \in \mathcal{N} \tag{2.4}$$

$$\sum_j t_{js}^{sd} = 0, \quad s, d \in \mathcal{N} \tag{2.5}$$

$$\sum_j t_{dj}^{sd} = 0, \quad s, d \in \mathcal{N} \tag{2.6}$$

$$\sum_j t_{jd}^{sd} = t^{sd}, \quad s, d \in \mathcal{N} \tag{2.7}$$

Note that the *VTTR* problem does not take as input the network graph G, only the traffic demand matrix T (and, hence, the number of nodes, N). Consequently, the output of the problem is simply the set of lightpaths to be established but *not* the (physical) paths that these lightpaths take in the network. Therefore, the *VTTR* subproblem is very different than the *GR* subproblem in the traffic grooming decomposition studied in [21]: the *GR* subproblem takes as input the network graph G and determines not only the set of lightpaths but also their (physical) paths (but not wavelengths) in the network.

The *VTTR* subproblem is similar in concept to the virtual topology problem studied in the context of multihop broadcast-and-select (BAS) networks [26]. In multihop BAS networks it is not possible to establish direct connections between every pair of nodes, hence some traffic demands may need to be routed via intermediate nodes – just as traffic demands need to be routed over multiple lightpaths in our problem. Furthermore, just as BAS networks do not impose any physical topology constraints on the formation of direct connections due to their all-to-all broadcast nature, the *VTTR* subproblem does not impose any physical topology constraints on the formation of the virtual topology of lightpaths. The virtual topology determined by *VTTR* will be reconciled with the physical topology using the second subproblem, as we discuss shortly.

Ignoring the physical topology constraints in the definition of the *VTTR* subproblem has two major benefits. First, the running time for finding an optimal solution depends only on the size (i.e., number N of nodes) of the network, not its topology. Hence, the running time of a problem instance with a given demand matrix T would be identical for a sparse ring network and a dense mesh network of the same size. Second, the problem formulation includes the integer variables $\{b_{ij}\}$ and $\{t_{ij}^{sd}\}$, but it does not include any binary variables. Therefore, it is possible to employ partial LP relaxation techniques so as to reduce the time required to find solutions that are close to the optimal one; we describe an algorithm that uses such techniques in the following section.

2.1.2 Routing and Wavelength Assignment (RWA)

The routing and wavelength assignment (RWA) problem is one of selecting a path and wavelength for each lightpath, subject to capacity and wavelength constraints.

Definition 2.1.2 (RWA) *Given the graph G of TG and the set of lightpath demands $\{b_{ij}\}$ determined by the solution to VTTR, route the lightpaths on the physical topology of G and assign a wavelength to each lightpath so as to minimize the number of distinct wavelengths required.*

The *RWA* problem is a fundamental problem in optical network design, and has been studied extensively. In [42], an exact ILP formulation based on maximal independent sets (MIS) was developed that solves the *RWA* problem in rings of size up to $N = 16$ nodes (the maximum size supported by SONET technology and hence *de facto* maximum size of deployed ring networks) in just a few seconds, several orders of magnitude faster than earlier known solutions. New formulations that solve the *RWA* problem in mesh networks up to two orders of magnitude faster than existing techniques can be found in [27, 28]. Therefore, we solve the *RWA* subproblem using the techniques in [27, 28, 42] rather than using the corresponding part of the formulation of the *TG* problem in (1.8)-(1.19).

2.1.3 Sequential Solution to the VTTR and RWA Problems

We propose to solve the *TG* problem by sequentially solving its two subproblems:

1. Solve the *VTTR* subproblem to obtain the set $\{b_{ij}\}$ of lightpaths to be established, and the routing of traffic demands $\{t^{sd}\}$ over these lightpaths.

2. Solve the *RWA* subproblem to find a path and wavelength for each lightpath in the set $\{b_{ij}\}$ so as to minimize the number of distinct wavelengths used in the solution.

Recall that the first step of the solution produces a set $\{b_{ij}\}$ of lightpaths that are determined only by the traffic demands and are not tied to the physical topology of the network. However, the second step routes the lightpaths over the physical links of the network, hence ensuring that the final solution is consistent with the network topology.

The following two lemmas state the properties of this sequential solution.

Lemma 2.1.1 *Let $P^\star_{TG}$ and $P^\star_{VTTR}$ denote the number of lightpaths returned by the optimal solutions to the TG and VTTR problems, respectively. Then:*

$$P^\star_{VTTR} \quad \leq \quad P^\star_{TG}. \tag{2.8}$$

Proof. The *VTTR* subproblem is a relaxed version of the original *TG* problem with constraints (1.8)-(1.19) removed. Hence, the objective value of an optimal solution to *VTTR* cannot be greater than that of an optimal solution to *TG*.

Lemma 2.1.2 *Let $W^\star_{RWA}$ be the number of wavelengths returned as the optimal solution to the RWA subproblem that takes as input the optimal solution $S^\star_{VTTR}$ of the VTTR subproblem. If $W^\star_{RWA} \leq W$, where W is the number of available wavelengths given as input to the original TG problem, then $S^\star_{VTTR}$, together with the lightpath routing and wavelength assignment determined by the RWA subproblem, is an optimal solution to TG.*

Proof. According to Lemma 2.1.1, the number of lightpaths in the solution $S^\star_{VTTR}$ is such that $P^\star_{VTTR} \leq P^\star_{TG}$. After the *RWA* is solved, the routing and wavelength assignment of the lightpaths in $S^\star_{VTTR}$ satisfy all the physical topology and wavelength assignment constraints. Hence, the final result of sequentially solving the two subproblems is also a feasible solution to the original problem *TG*, i.e., $P^\star_{VTTR} \geq P^\star_{TG}$, from which the result of the lemma follows.

The practical implication of Lemma 2.1.2 is that whenever the network is not wavelength (bandwidth) limited, sequentially solving the *VTTR* and *RWA* subproblems will yield an optimal solution to the original *TG* problem that also minimizes the number of wavelengths used for the given set of lightpaths.

2.2 Three Efficient Algorithms for *VTTR*

In this section we develop solution techniques for the *VTTR* subproblem, an NP-hard problem for which we are not aware of any scalable solutions.

2.2.1 Partial LP Relaxation of *VTTR*

Linear programming (LP) relaxation of an ILP is the problem that arises by relaxing the integrality constraints on the relevant decision variable of the original problem. Since the original ILP formulation has stronger constraints than its LP relaxation, in the case of minimization problems such as the one we consider in this work, the optimal value of the LP relaxation provides a lower bound of the original ILP formulation. Although LP relaxation sacrifices optimality, the relaxed problem can be solved in polynomial time as a linear program in time that may be orders of magnitude lower than the time to solve the original ILP.

Definition 2.2.1 *(VTTR-rlx) Given the number of nodes N in the graph G of TG, the wavelength capacity C, and traffic demand matrix T, establish the minimum number of lightpaths to carry all traffic demands while allowing fractional lightpaths to exist between any pair of nodes.*

VTTR-rlx can be derived from *VTTR* by replacing the integer variables $\{b_{ij}\}$ with non-negative real variables $\{\bar{b}_{ij}\}$. Since the integrality constraints on variables $\{t_{ij}^{sd}\}$ are maintained, *VTTR-rlx* represents a *partial* LP relaxation of *VTTR*, and can be formulated as the following mixed integer linear program (MILP).

Lemma 2.2.1 shows how a feasible solution to *VTTR* can be obtained from any feasible solution to *VTTR-rlx.*

Lemma 2.2.1 *Let $\{\bar{b}_{ij}\}$ and $\{t_{ij}^{sd}\}$ represent a feasible solution to VTTR-rlx. Then, $\{\lceil\bar{b}_{ij}\rceil\}$ and $\{t_{ij}^{sd}\}$ is a feasible solution to VTTR.*

Proof. We first note that $\{t_{ij}^{sd}\}$ satisfies constraints (1.3)-(1.7) automatically, since these constraints are also part of *VTTR-rlx.* Constraint (1.2) is also satisfied, since:

$$\sum_{s,d} t_{ij}^{sd} \quad \leq \quad \bar{b}_{ij}C \quad \leq \quad \lceil\bar{b}_{ij}\rceil C. \tag{2.9}$$

Finally, $\{\lceil\bar{b}_{ij}\rceil\}$ are integers, satisfying the integrality constraints of *VTTR.*

We compared the *VTTR* and *VTTR-rlx* on problem instances defined on a 16-node network. Recall that the *VTTR* subproblem of *TG* only takes into account the traffic demands $\{t^{sd}\}$ between nodes, not the physical topology of the network. We generated traffic instances by setting each traffic demand t^{sd} as a random integer in the range $[0, t_{max}]$. We let parameter $t_{max} = 10, 20, 30, 40, 50, 60$, and for each value of t_{max} we generated ten traffic matrices (i.e., problem instances) that were used to solve both *VTTR* and *VTTR-rlx.*

Table 2.1 presents the CPU time (in sec), averaged over the ten random instances, that CPLEX needs to solve the *VTTR* and *VTTR-rlx* problems for each value of t_{max}. We note

Table 2.1: CPU time comparison of *VTTR* and *VTTR-rlx*, $N = 16$

	CPU Time (sec)	
t_{max}	*VTTR*	*VTTR-rlx*
10	21629.1	0.184
20	21626.7	0.199
30	21626.6	0.200
40	21732.8	0.242
50	21740.4	0.259
60	21625.9	0.188

Table 2.2: Objective value comparison of *VTTR* and *VTTR-rlx*, $N = 16$

	Objective Value (# of lightpaths)		
	VTTR	*VTTR-rlx*	
t_{max}	(best available)	(optimal)	(rounded-up)
10	101.7	74.1	217.9
20	173.2	150.0	274.2
30	250.6	226.6	340.1
40	327.1	302.6	423.6
50	389.3	366.4	480.1
60	468.2	443.5	558.5

that the CPU times do not vary much across the values of t_{max}, but solving the partial LP relaxation *VTTR-rlx* takes a fraction of a second whereas solving the *VTTR* ILP takes longer than the six-hour limit we imposed.

Table 2.2 compares the best available solutions to *VTTR* obtained within the six-hour limit, to the optimal solutions to *VTTR-rlx*, in terms of the objective value (i.e., number of lightpaths). For each row of the table (i.e., a specific value of t_{max}), the values shown are averages over the corresponding ten traffic instances. However, the optimal solution to partial LP relaxation *VTTR-rlx* is a lower bound, but not necessarily a feasible solution to *VTTR*. Therefore, in the table we also present the objective value of the feasible solution obtained as described in Lemma 2.2.1, i.e., by rounding up the real values $\bar{b}^{\star}_{ij}$ of the optimal solution to *VTTR-rlx*.

From the two tables we make two important observations. First, the integral constraints of the lightpath variables b_{ij} play an important role in increasing the complexity of the branch-and-bound process of the ILP solver. Second, rounding up the real lightpath values $\bar{b}^{\star}_{ij}$ results in a large optimality gap. Based on these observations, in the next section we develop three iterative algorithms that strike a good balance between running time and quality of solution.

2.2.2 Lightpath Utilization

Consider the optimal solution $\{\bar{b}^{\star}_{ij}\}$ to the *VTTR-rlx* problem and the corresponding feasible solution $\{\lceil \bar{b}^{\star}_{ij} \rceil\}$ to *VTTR*, obtained by rounding up all the lightpath variables. Let us define:

$$U_{ij} = \frac{\bar{b}^{\star}_{ij}}{\lceil \bar{b}^{\star}_{ij} \rceil}, \quad \bar{b}^{\star}_{ij} > 0. \tag{2.10}$$

The quantity U_{ij} represents the utilization of the lightpaths from node i to node j in the rounded-up feasible solution. When the utilization is high (i.e., U_{ij} is close to 1.0), the corresponding lightpath resources are used effectively in the solution; furthermore, rounding up the corresponding lightpath variable to obtain a feasible solution makes only a small contribution to the optimality gap. The opposite is true when the utilization of a set of lightpaths is low.

We now present three efficient algorithms based on partial LP relaxation and lightpath utilization in the following two subsections. The key idea of all the three algorithms is to treat the integer constraints on lightpath variables b_{ij} as *lazy* constraints, and activate only a subset of them (or variants) at each utilization threshold setting.

2.2.3 ***VTTR-rlx***(U_{thr})

This iterative algorithm treats the integer constraints on lightpath variables b_{ij} as *lazy* constraints, and activate only a subset of them at each iteration. Initially, we start by solving the partial LP relaxation *VTTR-rlx* in which none of the integrality constraints on $\{b_{ij}\}$ are activated. If all lightpath variables in the optimal solution are integer, then this is a feasible (and optimal) solution to *VTTR*. Otherwise, we examine the solution to identify all lightpath variables $\bar{b}_{ij}$ with a utilization $U_{ij} < u_{thr}$, where u_{thr} is some threshold that is initialized to a small value, e.g., $u_{thr} = 0.1$. We then activate the integrality constraints on the identified variables, i.e., we solve a modified version of *VTTR-rlx* in which the identified variables are replaced by integer variables b_{ij}. We repeat this process, increasing the threshold value u_{thr} at each iteration, until one of the following stopping criteria is satisfied:

1. all lightpath variables in the solution are integer, and hence represent an optimal solution to *VTTR*;

2. the threshold u_{thr} reaches a predetermined value[1]H ; or

[1] Note that, if this predetermined upper bound H on u_{thr} is 1.0, then the algorithm will continue until the integrality constraints on all lightpath variables are activated, i.e., the full *VTTR* problem will be solved in the

3. the improvement in the value of the objective function over the previous iteration is less than a predetermined minimum value δ.

A combination of the above criteria may be used, e.g., stop whenever the threshold has reached a predetermined value or the improvement over the previous iteration is less than δ, whichever is satisfied first.

This ascending utilization *VTTR(U_{thr})* iterative algorithm can be described by these steps:

1. Initialization: $i \leftarrow 0; u_{thr} \leftarrow 0.1$.

2. Solve *VTTR-rlx* to obtain the optimal solution and determine the objective value P_i of the corresponding feasible solution obtained by rounding up all non-integer lightpath variables.

3. Calculate U_{ij} for all non-integer lightpaths in the optimal solution and modify *VTTR-rlx* to activate the integrality constraints for the variables for which $U_{ij} \leq u_{thr}$.

4. If the stopping criterion is satisfied, return the current solution; otherwise set $i \leftarrow i + 1; u_{thr} \leftarrow u_{thr} + 0.1$ and repeat from Step 2.

We note that, at each iteration of the algorithm, a tighter partial LP relaxation of *VTTR* is considered, generally requiring longer time to solve. On the other hand, the objective value of the solution improves with each iteration. By selecting an appropriate stopping criterion, the iterative algorithm may be designed to deliver a desirable tradeoff between running time and quality of the final solution.

2.2.4 *VTTR-rlx*(U_l, U_h)

We define U_l and U_h, $0 \leq U_l \leq U_h \leq 1$, as low and high thresholds, respectively on lightpath utilization. We consider a modified version of *VTTR-rlx* in which $\bar{b}_{ij}$ is fixed to $\lceil \bar{b}^*_{ij} \rceil$, if $U_{ij} \geq U_h$, and to $\lfloor \bar{b}^*_{ij} \rfloor$, if $U_{ij} \leq U_l$. In practice, the following two sets of *equality constraints* are activated on the lightpath variables $\bar{b}_{ij}$ with a utilization $U_{ij} \leq U_l$ or $U_{ij} \geq U_h$:

$$\bar{b}_{ij} = \lceil \bar{b}^\star_{ij} \rceil \quad \forall\, i, j:\; U_{ij} \geq U_h \tag{2.11}$$

$$\bar{b}_{ij} = \lfloor \bar{b}^\star_{ij} \rfloor \quad \forall\, i, j:\; U_{ij} \leq U_l \tag{2.12}$$

We let *VTRL-rlx*(U_l, U_h) denote the modified LP relaxation of *VTTR* in which the variables $\bar{b}_{ij}$ are set to be equal to the floor (respectively, ceiling) of the corresponding optimal solution obtained from *VTTR-rlx* if $U_{ij} \leq U_l$ (respectively, $U_{ij} \geq U_h$).

last iteration. More practically, the upper bound on u_{thr} should be strictly less but close to 1.0, e.g., $H = 0.8$. Such an upper bound would guarantee that all lightpaths in the final solution are highly utilized while allowing room for expansion, i.e., accommodating new traffic demands without having to create additional lightpaths.

The key idea behind the equality constraints (2.11) and (2.12) introduced in the formulation of *VTRL-rlx*(U_l, U_h) can be explained by using an airline analogy in which lightpaths corresponds to scheduled flights. For lightpaths with low utilization, constraint (2.12) forces the fractional lightpaths to zero; when solving the modified problem, the traffic carried by these fractional lightpath will be redirected to other lightpaths. In the airline analogy, this is equivalent to canceling flights that are close to empty. Note that the deletion of lightpaths (respectively, the canceling of flights) may cause some traffic (respectively, passengers) to take a longer route to their destination; however, from the point of view of network design, this may be an acceptable tradeoff if it leads to a smaller network cost. On the other hand, when lightpaths have high utilization, constraint (2.11) forces the fractional lightpath to become a full lightpath. As a result, the extra capacity of this new full lightpath becomes available to carry traffic that is potentially redirected by fractional lightpaths that were deleted.

The question that arises is how to determine the pair of thresholds (U_l, U_h), i.e., the specific partial LP relaxation of *VTTR* that provides a desired tradeoff between running time and quality of solution. To this end, we propose an iterative algorithm that uses a local search technique to select the pair (U_l, U_h). The iterative algorithm treats the integer constraints on lightpath variables $\{\bar{b}_{ij}\}$ as *lazy* constraints, activating only a subset of them at each iteration based on how they relate to the current pair of utilization thresholds.

The iterative algorithm starts by solving the partial LP relaxation *VTTR-rlx* in which none of the integrality constraints on $\{\bar{b}_{ij}\}$ are activated. If all lightpath variables in the optimal solution are integer, then this is a feasible (and optimal) solution to *VTTR*. Otherwise, the solution is examined to identify all lightpath variables with a utilization U_{ij} outside the interval $[U_l, U_h]$, and the corresponding *VTRL-rlx*(U_l, U_h) variant is solved. This process is repeated, increasing the threshold value U_l and decreasing U_h at each iteration, until one of the following stopping criteria is satisfied:

1. all lightpath variables in the solution are integer, and hence represent an optimal solution to *VTTR*;

2. the threshold pair (U_l, U_h) reaches a predetermined value, or

3. the improvement in the value of the objective function over the previous iteration is less than a predetermined minimum value δ.

A combination of the above criteria may be used, e.g., stop whenever the thresholds have reached a predetermined value or the improvement over the previous iteration is less than δ, whichever is satisfied first.

The iterative algorithm can be described by these steps:

1. Initialization: $U_l \leftarrow 0.1$; $U_h \leftarrow 0.9$.

2. Solve *VTTR-rlx* with no integer constraints on $\{\bar{b}_{ij}\}$ activated. Calculate and record U_{ij} for all lightpath pairs in the optimal solution.

3. Solve *VTTR-rlx*(U_l, U_h). Find the new optimal solution, and determine the objective value of the corresponding feasible solution obtained by rounding up all non-integer lightpath variables.

4. If the stopping criterion is satisfied, return the current solution; otherwise set $U_l \leftarrow U_l + 0.1$; $U_h \leftarrow U_h - 0.1$ and repeat from Step 3.

We note that, at each iteration of the algorithm, a tighter partial LP relaxation of $VTTR$ with a larger number of equality constraints is considered, generally requiring longer time to solve. On the other hand, the objective value of the solution improves with each iteration. By selecting an appropriate stopping criterion, especially in terms of the threshold values on lightpath utilization, this algorithm may be designed to deliver a desirable tradeoff between running time and quality of the final solution.

We note that, by imposing the two sets of equity constraints, we manually fix the value of b_{ij}, which may lead to the reduction of solution accuracy. However, it also strongly reduce the possible combinations involving these b_{ij}s, which may highly reduce the corresponding running time. Thus, by running experiments, we aim to find a proper pair of (U_l, U_h) with the best solution quality in most cases.

2.2.5 *VTTR-rlx*(U_l, U_h)-Int

This algorithm is a variant of *VTTR-rlx*(U_l, U_h), such that, in addition to the two sets of equality constraints, we add one more set of integrality constraints.

1. In each iteration, the lightpath variables b_{ij} with $U_l \leq U_{ij} \leq U_h$ are set to be integer, i.e., the type of $\bar{b}_{ij}$ is changed from continuous to integral.

In the aspect of running time, by involving different numbers of integral constraints according to (U_l, U_h) value, the running time may have various increases. Thus, it will be useful to determine how many integral constraints to be added by choosing a proper pair of (U_l, U_h).

2.3 Numerical Results

In this section we investigate the $VTTR$ problem in terms of scalability (i.e., running time) and quality of solution. The $VTTR$ problem and its partial LP relaxations were solved by running the IBM Ilog CPLEX 12 optimization tool on a cluster of identical compute nodes with dual

Woodcrest Xeon CPU at 2.33GHz with 1333MHz memory bus, 4GB of memory and 4MB L2 cache.

Our study involves a large set of problem instances defined on several network sizes[2] with various random traffic loads. In particular, we consider networks with $N = 8$, 16, 24, and 32 nodes. In all the simulations, we set the wavelength capacity $C = 16$. For each network topology, we consider several problem instances. For each problem instance, the traffic demand matrix $T = [t_{sd}]$ is generated by drawing the (integer) traffic demands uniformly at random in the interval $[0, t_{max}]$. The values of t_{max} we used in the simulations are 20, 30, 40, and 50. Each data point in the figures we present in this section represents the average of 10 random problem instances for the stated values of the input parameters.

Unless otherwise stated, we set the relative optimality gap to 2% for all CPLEX runs. Consequently, CPLEX terminates when it finds a solution that is within 2% of the optimal for the problem at hand, rather than continuing until the problem is solved to optimality. Later in this section we will investigate how the running time required to solve the *VTTR* problem is affected by this optimality gap.

2.3.1 *VTTR-rlx*(U_{thr})

Figure 2.1 plots the running time of *VTTR-rlx*(U_{thr}) as a function of the threshold H. The algorithm runs until the threshold U_{thr} becomes equal to H, and this is used as the only stopping criterion. Note that, when $H = 0$, only the partial LP relaxation *VTTR-rlx* is solved, whereas when $H = 1$, the full *VTTR* problem is solved; for values of H between zero and one, intermediate versions of *VTTR-rlx* are solved whereby integrality constraints are imposed only on lightpath variables with utilization less than H, as we discussed in the previous section. The figure plots results for networks with $N = 12$ and various values of t_{max}. As expected, the running time increases as H increases, since larger values of H imply that the integrality constraints are impsoed on a larger number of lightpath variables. We also see that the running time is affected by the value of parameter t_{max}. As t_{max} increases, the overall traffic to the network increases as well, a larger number of lightpaths need to be established to carry the traffic, hence the optimal objective value is higher. Consequently, an optimality gap of 2% translates to a larger absolute difference (in the number of lightpaths) from the optimal objective value as t_{max} increases; in turn, CPLEX reaches this lower value faster.

For the remainder of the experiments in this section, we present results for problem instances with $t_{max} = 30$. With this value of t_{max}, the average size of demands between source-destination pairs is close to the capacity $C = 16$ of a wavelength. Results for other values of t_{max} exhibit the same behavior and are omitted.

[2] We remind the reader that the *VTTR* problem does not account for the physical topology of the network.

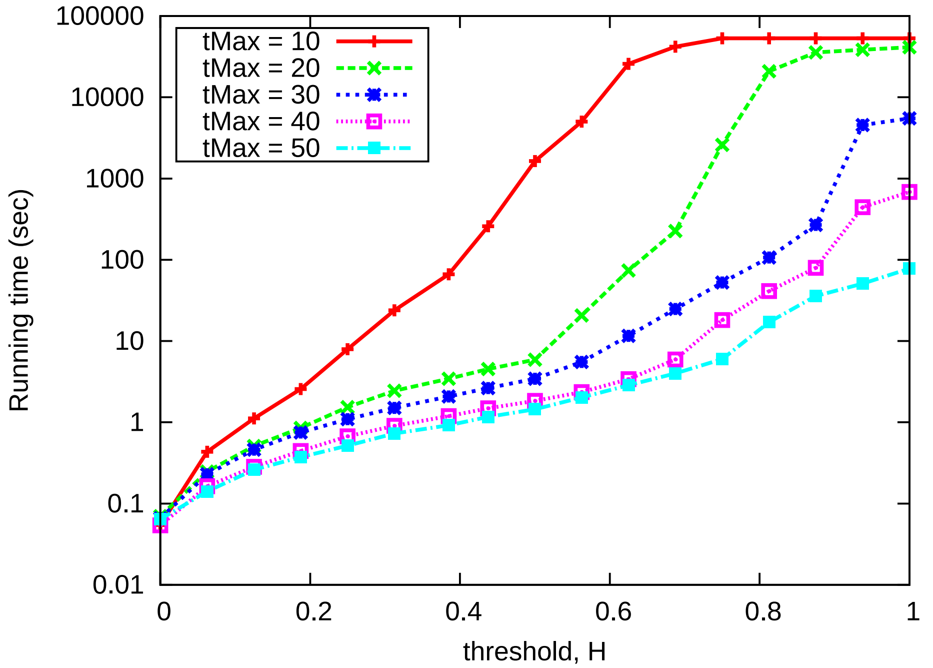

Figure 2.1: CPU time, as a function of H, $N = 12$, iterative algorithm $VTTR\text{-}rlx(U_{thr})$

Figure 2.2 plots the quality of the solution of the iterative algorithm as a function of the threshold H and for various network sizes. The quality of the solution for a given value of H is expressed as:

$$\frac{\sum \lceil \bar{b}^{\star}_{ij} \rceil}{\sum b^{\star}_{ij}}, \tag{2.13}$$

where the numerator is the feasible solution to $VTTR$ obtained by rounding up the lightpath variables in the optimal solution to the $VTTR\text{-}rlx$ problem for the given value of H, while $b^{\star}_{ij}$ is the optimal solution to the $VTTR$ problem (i.e., when $H = 1$). As expected, the quality of the solution starts high when $H = 0$ (i.e., all integer lightpaths are relaxed) and then decreases until it reaches optimality for $H = 1$. Importantly, for all network sizes shown in the figure, the solution is within 5% of the optimal as soon as $H = 0.8$. As we remarked earlier, such a value for H ensures that all wavelengths are highly utilized while also making it possible to accommodate future demands without necessarily setting up additional lightpaths.

Figure 2.3 plots the running time of the iterative algorithm as a function of the network size N. For these experiments, the iterative algorithm terminates when the lightpath utilization reaches the threshold $H = 0.8$. As we can see, the iterative algorithm makes it possible to solve the $VTTR$ problem for network sizes up to $N = 32$ within just a few hours without sacrificing

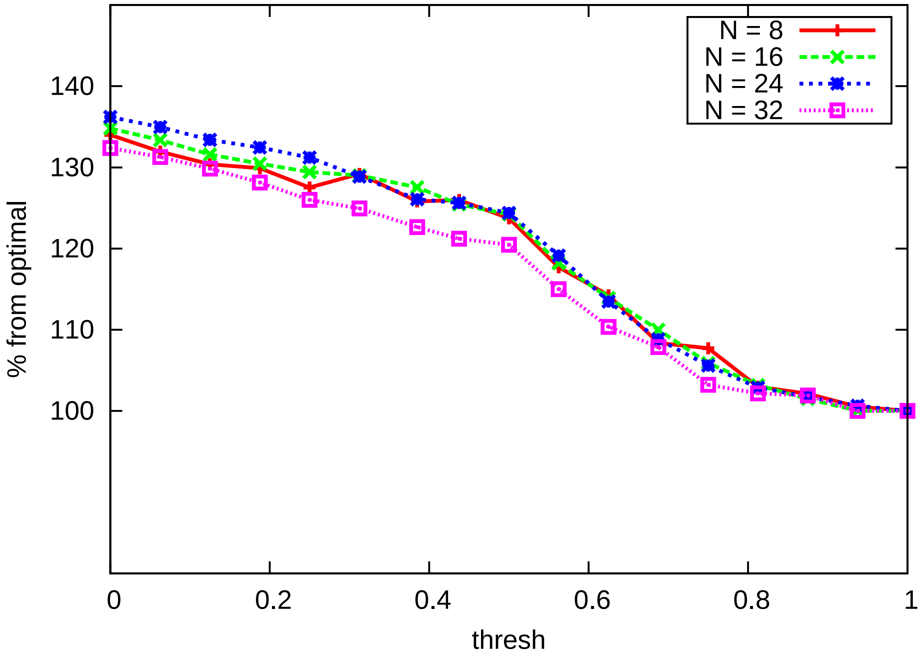

Figure 2.2: Quality of solution as a function of H, algorithm $VTTR\text{-}rlx(U_{thr})$, $t_{max} = 30$

much in terms of optimality (as Figure 2.2 indicates). Such problem instances are impossible to solve using the original formulation for the *TG* problem.

2.3.2 $\boldsymbol{VTTR\text{-}rlx}(U_l, U_h)$

Solution Quality

Figure 2.4 plots the quality of the solution of $VTTR\text{-}rlx(U_l, U_h)$ as a function of the pair of thresholds (U_l, U_h) and for various network sizes. The quality of the solution is defined similarly as:

$$\frac{\sum_{ij} \lceil \bar{b}^{\star}_{ij} \rceil}{\sum_{ij} b^{\star}_{ij}}. \tag{2.14}$$

The numerator in this expression is the value of the feasible solution to *VTTR* obtained by rounding up the lightpath variables in the optimal solution to the modified version of $VTTR\text{-}rlx(U_l, U_h)$ problem for the given value of a pair of thresholds. The denominator is the value of the objective function for the optimal solution to the *VTTR* problem (obtained within a 2% relative optimality gap, as we explained earlier). A low value of the above expression denotes a higher solution quality.

As expected, the quality of the solution starts away from optimal for *VTTR-rlx* (i.e., the

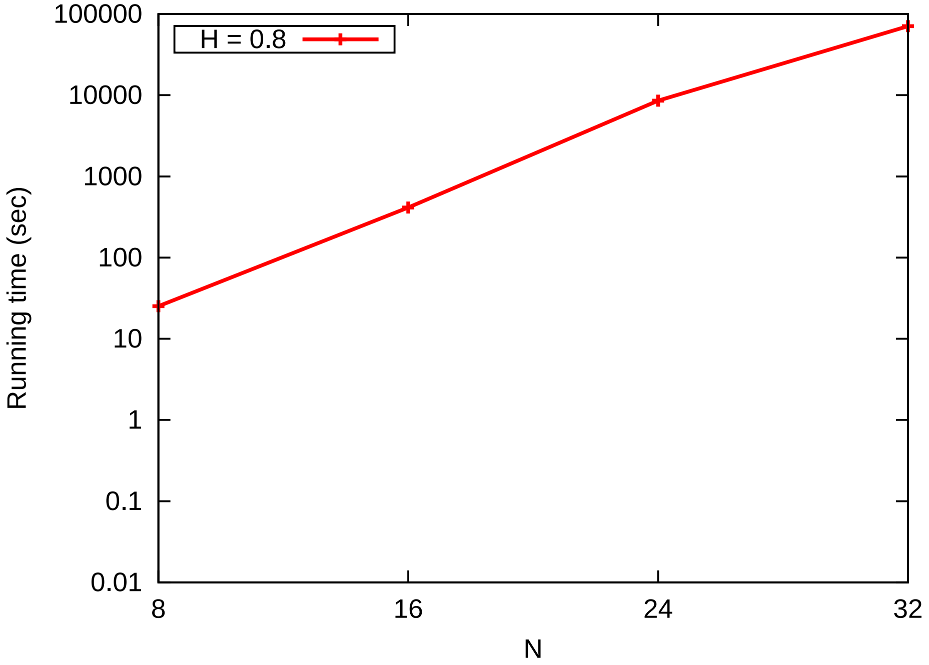

Figure 2.3: CPU time as a function of N, iterative algorithm $VTTR\text{-}rlx(U_{thr})$, $H = 0.8, t_{max} = 30$

point (0,0) in the figure) since all integer lightpath variables are relaxed; the solution quality then improves as U_l increases or U_h decreases. The best result is achieved for the threshold pair $(U_l, U_h) = (0.5, 0.6)$. Importantly, for all network sizes shown in the figure, the solution is about 11% from the optimal one as soon as $(U_l, U_h) = (0.5, 0.6)$; this worst case occurs for $N = 8$, and the gap decreases as the network size N increases. For the 32-node network, the gap is as small as 3%. Note that this pair of values for (U_l, U_h) ensures that no wavelength is under-utilized (i.e., it is filled to 50% at minimum) while also leaving some room to accommodate future demands without necessarily setting up additional lightpaths.

Scalability

Figure 2.5 plots the CPU time it takes to solve the $VTTR\text{-}rlx(U_l, U_h)$ problem as a function of the thresholds U_l and U_h; note that the pair (0,0) in the figure corresponds to the solution of the relaxed problem $VTTR\text{-}rlx$. The figure plots results for networks with $N = 16$ nodes and various values of t_{max}. As expected, the running time generally increases as U_l increases or U_h decreases, since in both cases equality constraints are imposed on a larger number of lightpath variables. We also see that the running time is not significantly affected by the value of parameter t_{max}. When t_{max} increases from 20 to 50, the running time only increases by a

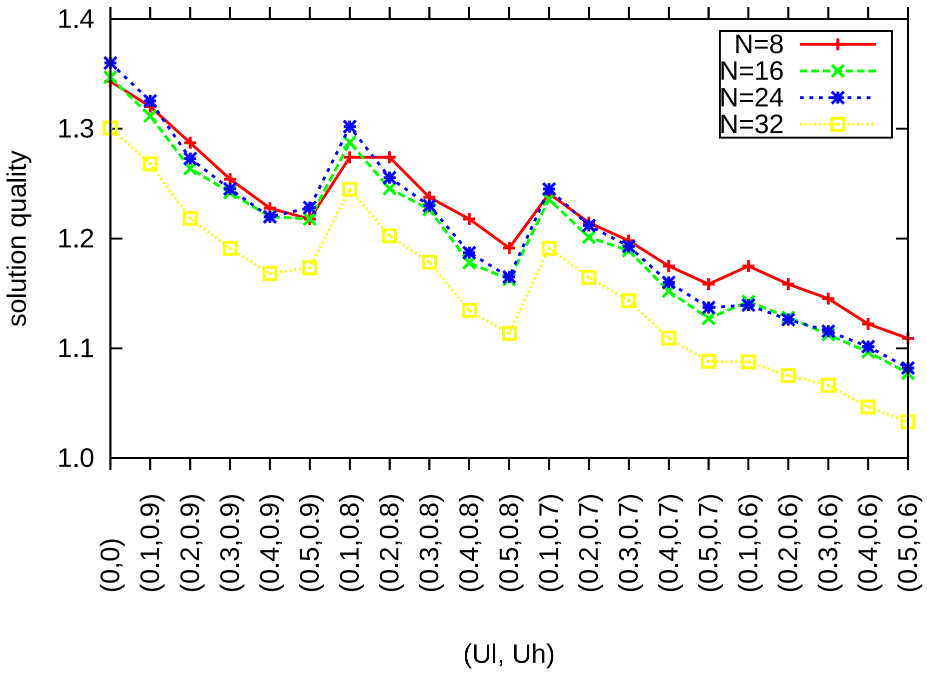

Figure 2.4: Solution quality as a function of (U_l, U_h), $t_{max} = 30$

few seconds. This shows that the algorithm is effective across a range of traffic loads.

Based on the last observation, for the simulations in the remainder of this section we have fixed the value of $t_{max} = 30$; with this value of t_{max}, the average size of demands between any source-destination pair is close to the capacity $C = 16$ of a wavelength. Results for other values of t_{max} exhibit the same behavior and are omitted.

Figure 2.6 plots the CPU time to solve *VTTR-rlx*(U_l, U_h) as a function of U_l. We plot results for a 24-node network, as they are representative of the running time trends for other network sizes. As shown in the figure, when U_h is kept fixed, the running time does not change significantly as a function of the lower threshold value U_l. However, as U_h decreases, the running time increases significantly. In other words, the value of the high threshold U_h has a stronger influence on the performance of the algorithm with respect to running time.

Figure 2.7 plots the running time as a function of the network size N. For this simulation, we set $(U_l, U_h) = (0.5, 0.6)$, the value that achieves the solution of highest quality (i.e., the smallest number of lightpaths) as we observed earlier. As we can see, it is possible to obtain a feasible solution to the $VTTR$ problem for network sizes up to $N = 32$ within one hour without sacrificing much in terms of optimality (as Figure 2.4 indicates). Such problem instances are impossible to solve within 24 hours by directly using the original ILP formulation for the $VTTR$ problem, even applying a 2% optimality gap.

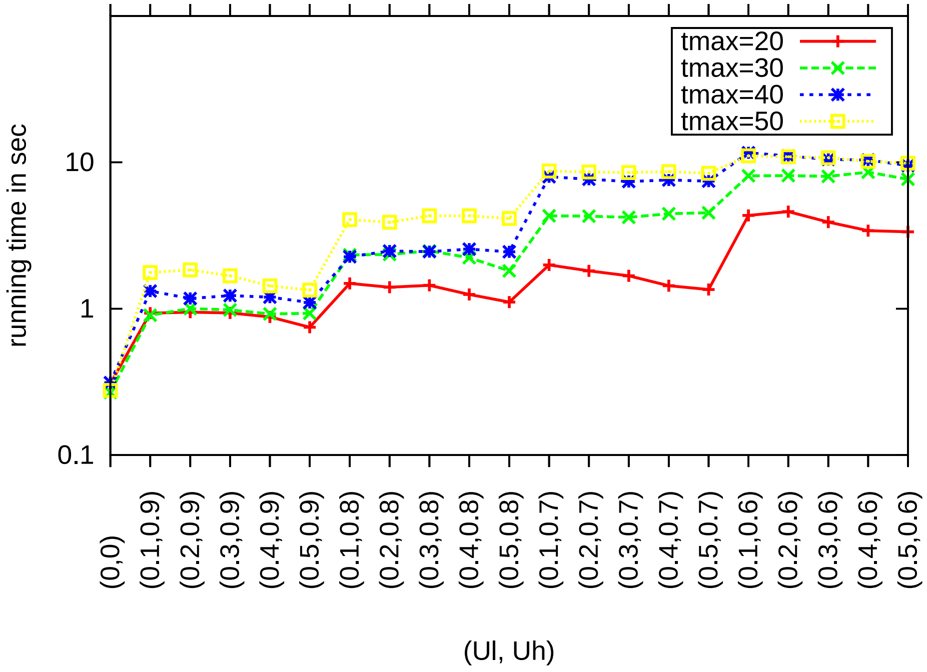

Figure 2.5: CPU time as a function of (U_l, U_h), $N = 16$

Finally, Figure 2.8 compares the running time as a function of network size of three methods for solving the $VTTR$ problem:

1. solving $VTTR$ to optimality;
2. solving $VTTR$ with a 2% relative optimality gap; and
3. solving $VTTR$-*rlx*(0.5,0.6) with a 2% relative optimality gap.

All simulations are allowed to run for as long as 24 hours. As we can see, the running time of the first method (i.e., directly solving the ILP formulation for the $VTTR$ problem to optimality), is extremely long: within the 24-hour time limit, this method can only solve network sizes up to 16 nodes. Setting the optimality gap to 2% (i.e., using the second method listed above) reduces the running time of finding a solution significantly, but even in this case it is not possible to solve 32-node networks within the 24-hour time limit. Notice that with the second method, the running time for the 8-node network is longer than the time it takes to solve the 16-node network. This drop in running time when the network size increases from 8 to 16 can be explained by making the observation that, with a fixed value of t_{max} (= 30 for these results), as the network size grows so does the offered traffic and the optimal objective value (i.e., number of lightpaths). Hence, CPLEX may take a shorter time to reach a solution that

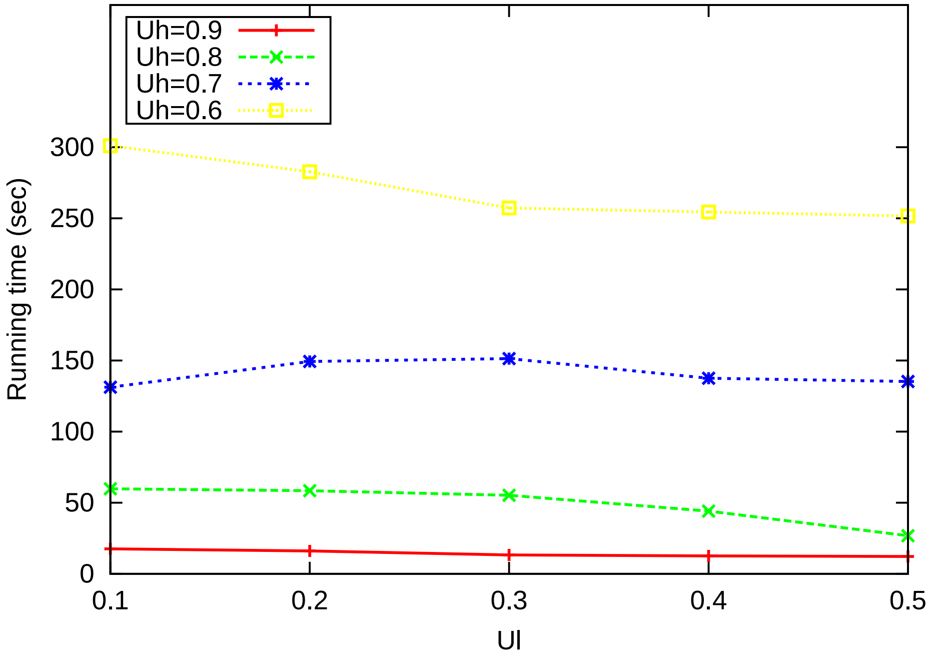

Figure 2.6: CPU time as a function of U_l, $N = 24, t_{max=30}$

is 2% from the optimal solution as the network size increases, as the absolute difference from the optimal solution is larger. Of course, as the network size increases further, the increase in the number of variables and constraints becomes once again the factor determining the running time; hence the increase as network size grows to 24 and beyond.

Finally, solving the modified *VTTR-rlx*(U_l, U_h) relaxed problem is significantly faster over all network sizes, and reduces the running time by more than one order of magnitude compared to solving *VTTR* directly within a 2% optimality gap. In particular, the *VTTR-rlx*(U_l, U_h) problem can be solved on the 32-node network in about 3000 seconds (i.e., less than one hour), while also obtaining a solution that is within 3% of the best solution (refer also to Figure 2.4) that we were able to obtain after running the second method for 24 hours. Note that in Figure 2.8, we used $(U_l, U_h) = (0.5, 0.6)$ as the pair of thresholds for the algorithm; however, depending on the relative importance of solution quality to running time, other pairs of thresholds may be applied to further reduce the running time (as shown in Figure 2.6).

Based on these results, we conclude that, for small networks, e.g., of size between 8-10 nodes, the *VTTR* problem can be directly solved by using the MILP formulation with or without imposing a 2% optimality gap. However, for larger networks, solving the *VTTR-rlx*(U_l, U_h) problem is more efficient and effective. The particular pair of thresholds (U_l, U_h) to use may be fine tuned using the iterative algorithm we described in the previous section.

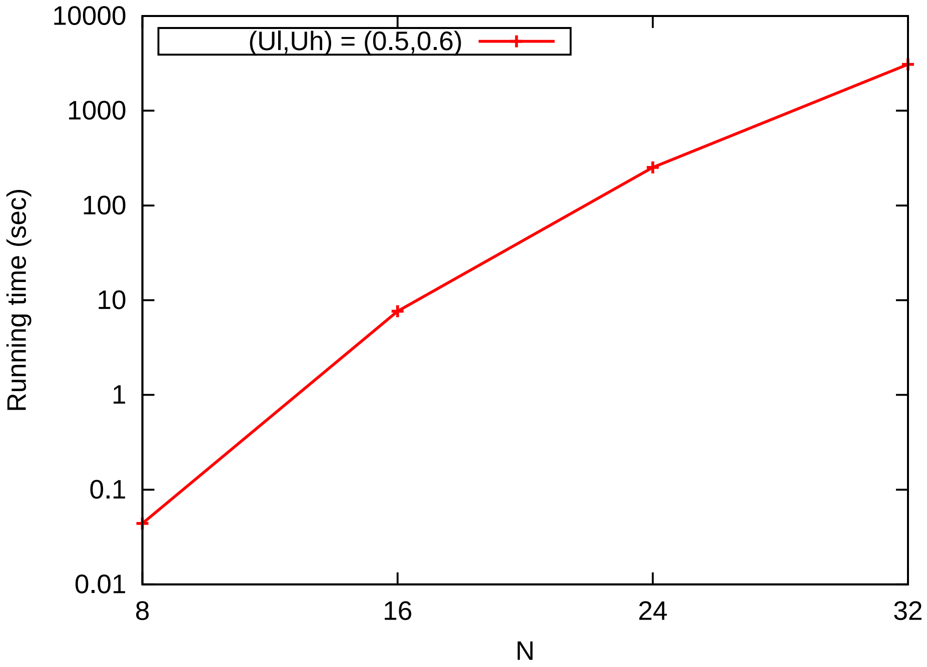

Figure 2.7: CPU time as a function of N, $(U_l, U_h) = (0.5, 0.6), t_{max} = 30$

2.3.3 *VTTR-rlx*(U_l, U_h)-Int

In Fig. 2.9, it plots the quality of solution against pairs of various threshold values for 8-, 16- and 24-node networks with $t_{max} = 30$, where $U_l = 0.3, 0.4, 0.5$ and $U_h = 0.6, 0.7, 0.8$. The quality of solution is defined similarly as in the previous subsection. A close to 1.0 value indicates a high solution quality. First, generally, when U_h is fixed, the solution quality is higher when U_l decreases; when U_l is fixed, the solution quality is higher when U_h increases. In other words, the more integrality constraints are, the better the quality of solution is. Unlike the *VTTR-rlx*(U_l, U_h) algorithm, for all the three network sizes, the most extreme threshold pair $(U_l, U_h) = (0.3, 0.8)$ provides the best solution quality (very close to 1.0). This can be explained as when $(U_l, U_h) = (0.3, 0.8)$, the most integrality constraints are present, hence it is the closest to the original VTTR problem. Second, the solution quality is more sensitive to U_l, i.e., as long as U_l remains low (=0.3), we have good solution quality within 3% of the best objective value obtained from solving VTTR with 2% optimality gap.

In Fig. 2.10, it plots the CPU time for the same parameter settings as in Fig. 2.9. The first observation is that CPU time increases with the the size of network. Second, the CPU time increases when U_l decreases or U_h increases for 8-node network, while the CPU time is not very sensitive to the values of (U_l, U_h) when $N = 16$ and 24. Based on the observations from Fig. 2.9

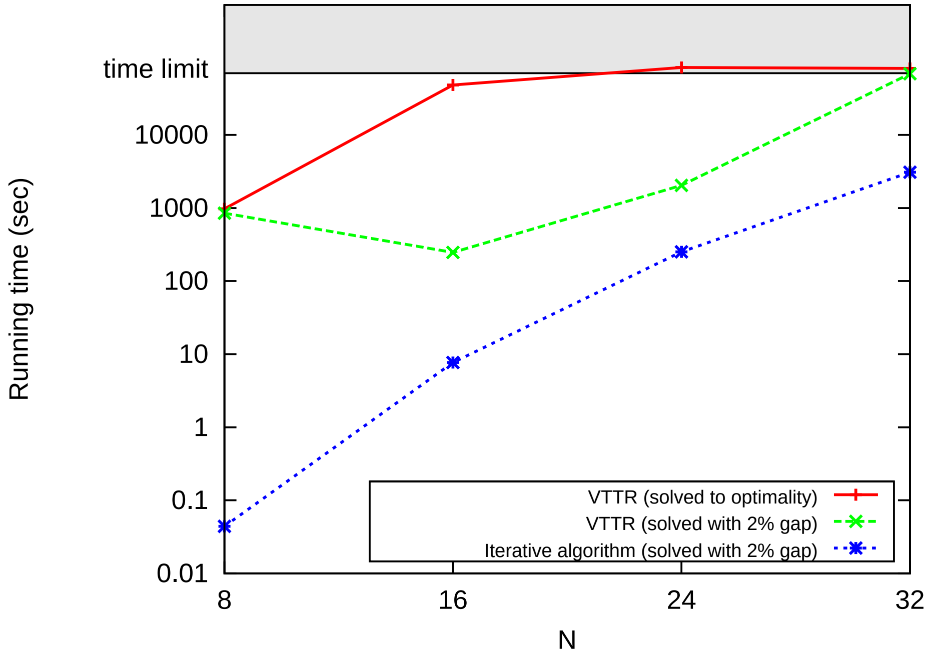

Figure 2.8: CPU time comparison as a function of network size N, $t_{max} = 30$

and 2.10, we can use $(U_l, U_h) = (0.3, 0.8)$ for all the three network sizes in order to obtain a good tradeoff between solution quality and running time. Depending on the practical needs, we may also choose different pairs of (U_l, U_h) values. For example, $(0.3, 0.6)$ can be used to obtain a one order of magnitude decrease in running time for 8-node network.

2.3.4 Wavelength Fragmentation

Let us now turn our attention to how sequentially solving the $VTTR$ and RWA subproblems addresses the wavelength fragmentation challenge we discussed in Section 1.5.2. Figure 2.11 is nearly identical to Figure 1.6 in that it plots the number of wavelengths used by solutions to the TG problem for five problem instances on a six-node ring network; each solution is obtained by providing the stated number W of wavelengths as input to the TG formulation. In addition, Figure 2.11 also includes the number of wavelengths obtained by sequentially solving the $VTTR$ and RWA subproblems on each of the five instances. As we can see, the sequential solution uses fewer wavelengths than any of the solutions to the original TG problem. This result confirms our earlier observations that if the network is not wavelength limited, then not only is the sequential solution optimal in terms of the number of lightpaths (the objective of the TG problem), but it also minimizes the number of wavelengths required to establish these lightpaths.

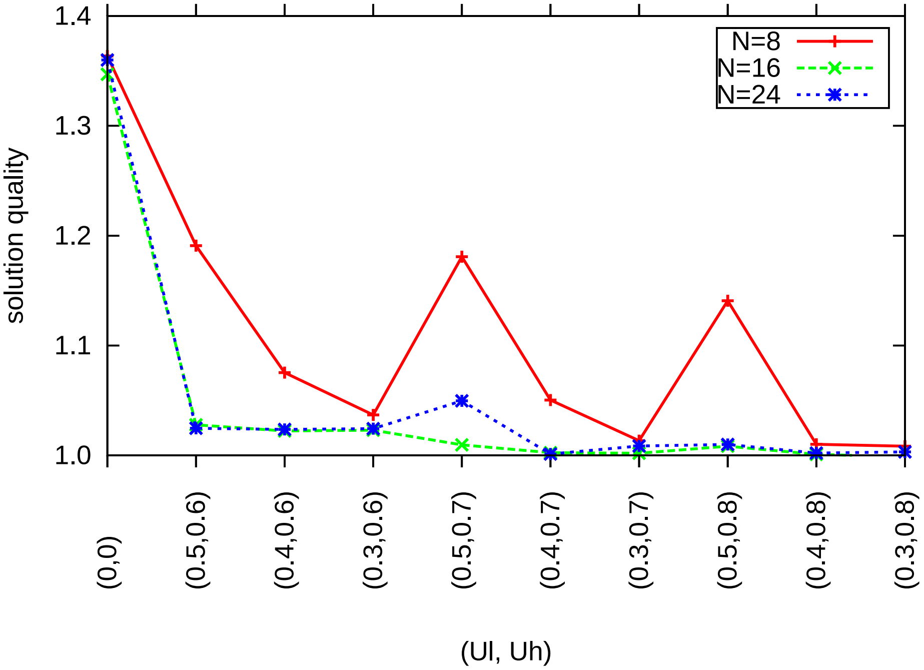

Figure 2.9: Solution Quality for *VTTR-rlx*(U_l, U_h)-Int, N=8, 16, 24, tmax=30

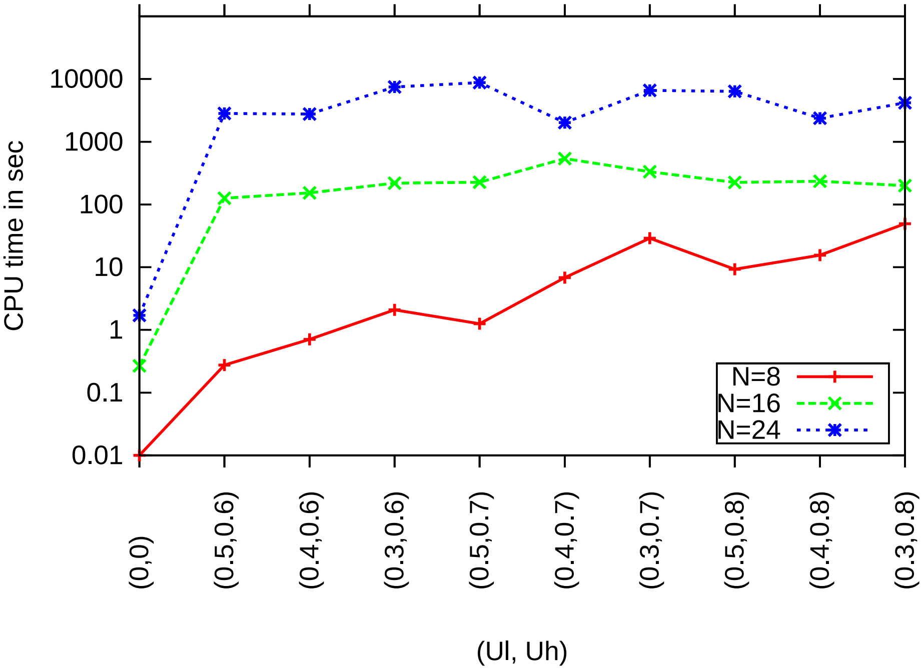

Figure 2.10: Running time (sec) for *VTTR-rlx*(U_l, U_h)-Int, N=8, 16, 24, tmax=30

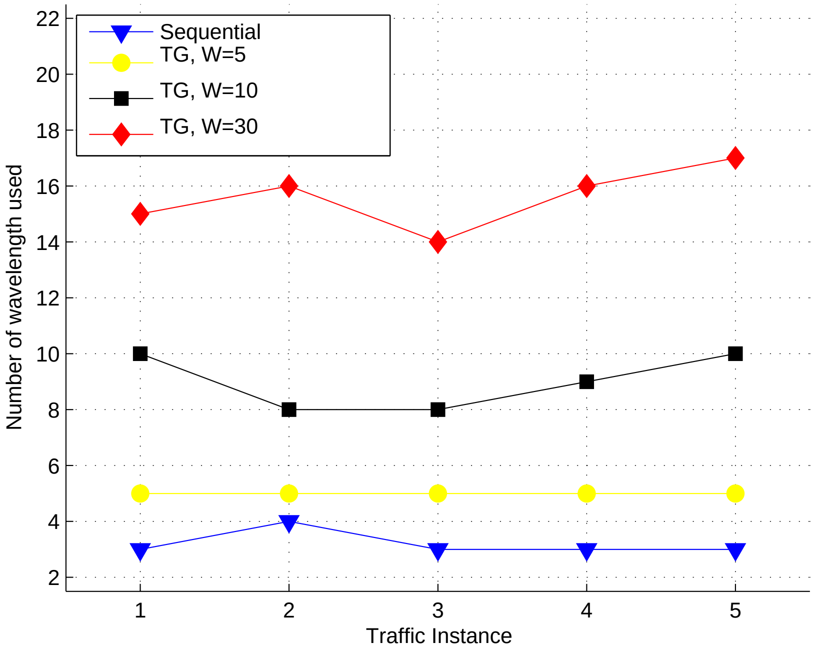

Figure 2.11: Wavelength usage comparison, 6-node ring network, $t_{max} = 12$

Chapter 3

Hierarchical Virtual Topology and Traffic Routing

As we mentioned earlier, the ILP formulation presented in Chapter 1 serves as the basis for reasoning about and tackling the traffic grooming problem. Unfortunately, as we have seen from Chapter 1, solving the ILP formulation directly does not scale to instances with more than a handful of nodes, and cannot be applied to networks of practical size covering a national or international geographical area. Based on this observation, in Chapter 2, we have presented a novel decomposition of traffic grooming problem on general topologies, which consists of two sequential subproblems - the VTTR and RWA subproblems. We have also developed three efficient heuristic algorithms for the VTTR subproblem to obtain a good tradeoff between solution quality and running time.

These studies regard the network as a flat entity for the purpose of traffic grooming. However, it is well-known that existing networks resources are typically managed and controlled in a hierarchical manner. The levels of the hierarchy either reflect the underlying organizational structure of the network or are designed in order to ensure scalability of the control and management functions. Accordingly, several studies have adopted a variety of hierarchical approaches to traffic grooming that, by virtue of decomposing the network, scale well and are more compatible with the manner in which networks operate in practice.

The rest of this chapter is organized as follows. In Section 3.1 we survey earlier work in hierarchical traffic grooming. In Section 3.2 we define the hierarchical virtual topology and traffic routing (H-VTTR) problem, and present several variants that arise naturally. We present a performance study of the problem variants in Section 3.3.

3.1 Related Work in Hierarchical Grooming

3.1.1 Ring Networks

Early research in traffic grooming focused on ring topologies [17], [38], [13], mainly due to the practical importance of upgrading the existing SONET/SDH infrastructure to support multiple wavelengths. A point-to-point WDM ring is a straightforward extension of a SONET/SDH network, but requires that each node be equipped with one add-drop multiplexer (ADM) per wavelength. Clearly, such a solution has a high ADM cost and cannot scale to more than a few wavelengths. Therefore, much of the research in this context has been on reducing the number of ADMs by grooming sub-wavelength traffic onto lightpaths that optically bypass intermediate nodes, and several near-optimal algorithms have been proposed in [13,38]. However, approaches that do not impose a hierarchical structure on the ring network may produce traffic grooming solutions, in terms of the number of ADMs and their placement, that can be sensitive to the input traffic demands.

The study in [17] was the first to present several hierarchical ring architectures and to characterize their cost in terms of the number of ADMs (equivalently, electronic transceivers or ports) and wavelengths for non-blocking operation under a model of dynamic traffic. In a single-hub ring architecture, each node is directly connected to the hub by a number of lightpaths, and all traffic between non-hub nodes goes through the hub. In a double-hub architecture, there are two hub nodes diametrically opposite to each other in the ring. Each node is connected to both hubs by direct lightpaths, and non-hub nodes send their traffic to the hubs for grooming and forwarding to the actual destination.

A more general hierarchical ring architecture was also proposed in [17]. In this architecture, shown in Figure 3.1, ring nodes are partitioned into two types: *access* and *backbone*. The set of wavelengths is also partitioned into access and backbone wavelengths. The access wavelengths are used to connect all nodes, including access and backbone nodes, in a point-to-point WDM ring that forms the first level of the hierarchy. At the second level of the hierarchy, the backbone wavelengths are used to form a point-to-point WDM ring among the backbone nodes only. This hierarchy determines the routing of traffic between two access nodes as follows. If the two access nodes are such that there is no backbone node along the shortest path between them, their traffic is routed using single-hop lightpaths over the access ring along the shortest path. Otherwise, suppose that b_1 and b_2 are the first and last backbone nodes, respectively, along the shortest path between two access nodes a_1 and a_2 (note that b_1 and b_2 may coincide). Then, traffic from a_1 to a_2 is routed to b_1 over the access ring, from there to b_2 over the backbone ring, and finally over the access ring to a_2.

A similar hierarchical ring structure was considered in [9], and it was shown that using local (access) and bypass (backbone) wavebands, P-port dynamic traffic (in which each node

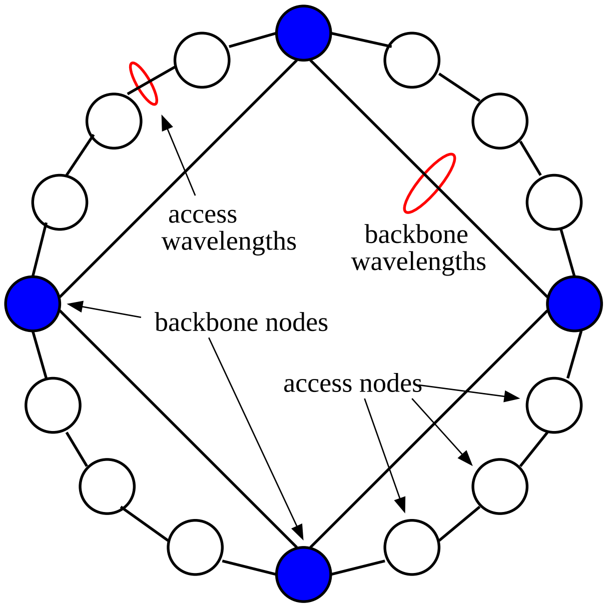

Figure 3.1: Hierarchical ring architecture with 12 access and 4 backbone nodes

is allowed to send and receive at most P wavelengths worth of traffic) can be supported with a minimum number of wavelengths.

A different hierarchical approach for grooming sub-wavelength traffic in ring networks was introduced in [34]. Specifically, the N ring nodes are grouped into K *super-nodes*, where each super-node consists of several consecutive ring nodes, as shown in Figure 3.2. The idea behind this partitioning is to pack (groom) all traffic from some super-node x to another super-node y onto lightpaths that are routed directly between the two super-nodes, bypassing intermediate nodes and hence, reducing the number of ADMs required. The study considered both uniform and distance-dependent traffic patterns, and, for each pattern, derived the number K of super-nodes, as a function of the number N of ring nodes and the granularity $C \geq 1$ of each wavelength, so as to minimize the number of ADMs; the granularity C is the number of unit traffic components that can be carried on a single wavelength.

Finally, [17] also proposes the decomposition of a ring into contiguous segments; these are similar to the super-nodes of [34] but are referred to as *subnets*. The ring network is organized in a hierarchical manner as a tree of subnets, where the root of the tree corresponds to a segment that consists of the entire ring. A tree node corresponding to a non-empty subnet s may be subdivided recursively into contiguous subsegments (subnets), and these become the children of subnet s in the tree. The set of wavelengths is also recursively partitioned into *transit* and *internal* sets at each node. Internal wavelengths are used within each subnet child of a node to carry traffic local to this subnet, whereas transit wavelengths are used to carry traffic between

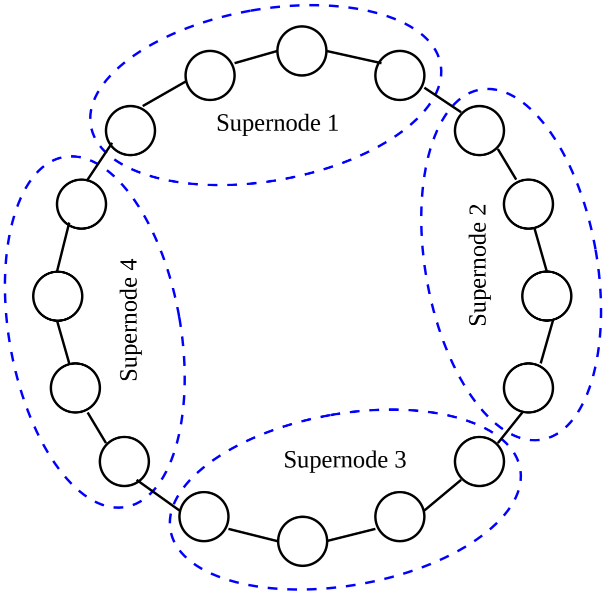

Figure 3.2: Ring architecture with 4 super-nodes, each of size 4

the subnet children of a node.

3.1.2 Torus, Tree, and Star Networks

A hierarchical approach for networks with a torus or tree topology was presented in [9], and is based on embedding rings on the underlying topology and then selecting hub nodes along each ring and using bypass wavelengths to interconnect the hubs. Consider first a $N \times M$ torus network, whose nodes are logically arranged on a grid of N rows and M columns. The network is viewed as a collection of N row-rings and M column-rings, and several nodes on each ring are designated as hubs; the hub selection is performed using an algorithm described in [9]. Traffic demands from some source s to a destination d are routed in three steps: from s to a hub h_1 in the same row as s along the appropriate row-ring; from h_1 to a hub h_2 in the same column as h_1 along the column-ring; and finally from h_2 to the destination d in the same row along a row-ring. This approach imposes a two-level hierarchy with non-hub nodes at the first level and hub nodes at the second level.

For tree networks, [9] proposes to embed a *virtual ring* in two steps: (1) using depth-first search to visit every node in the tree, and (2) locally arranging the tree nodes in a ring such that the nodes are connected in the ring in the order in which the corresponding tree nodes were first visited by the depth-first search. By defining hubs along the virtual ring, traffic components can be routed using the same algorithm we described for the hierarchical ring in Figure 3.1.

A traffic grooming algorithm for networks with a star topology was developed in [4]. The algorithm starts by creating lightpaths between the hub and each non-hub node s to carry all traffic originating and terminating at s. Such a solution provides maximum flexibility in terms of grooming, since traffic can be packed efficiently for transmission to the hub, and it can be groomed effectively there for forwarding to the destination. However, it usually requires a large number of lightpaths (equivalently, electronic ports). The algorithm then considers all traffic components in decreasing order of magnitude. Let t be a traffic component from some node s to another node d. The algorithm creates a direct lightpath from s to d to carry t, if there is an available wavelength for doing so; otherwise, no such lightpath is created. A direct lightpath is optically switched at the hub, bypassing electronic switching and grooming, and creating one has the potential to decrease the number of lightpaths by eliminating two lightpaths to/from the hub. The algorithm proceeds until all traffic components have been considered, and returns the solution with the minimum number of lightpaths. It was shown in [4] that this solution is close to optimal for a wide range of problem instances.

3.1.3 General Topology Networks

All the approaches we have discussed so far were developed for networks with topologies that are either symmetric (i.e., ring or torus) or contain no cycles (i.e., tree or star). A framework for hierarchical traffic grooming that is applicable to networks with a general topology was presented in [8]. The framework can be used for static or dynamic traffic, and for either sub-wavelength demands (to be groomed into lightpaths) or full-wavelength demands (to be groomed into wavebands). Although our discussion will consider only two levels of hierarchy, this approach can be extended in a straightforward manner to three or more levels of hierarchy to deal with networks of large size.

The traffic grooming problem involves the following conceptual subproblems (SPs) for sub-wavelength demands [14]:

1. *virtual topology SP:* find a set of lightpaths to carry the offered traffic;

2. *traffic routing SP:* route the traffic components over the lightpaths; and

3. *lightpath routing and wavelength assignment (RWA) SP:* assign a wavelength and path over the physical topology to each lightpath.

This is only a conceptual decomposition that helps in understanding and reasoning about the problem; in an optimal approach, the subproblems would be considered jointly in the solution. The first and second subproblems together constitute the grooming aspect of the problem. We will refer to these two subproblems as the *virtual topology and traffic routing (VTTR)* SP.

The hierarchical grooming approach, first described in [6], emulates the hub-and-spoke model used by the airline industry to “groom” passenger traffic onto connecting flights. Specifically,

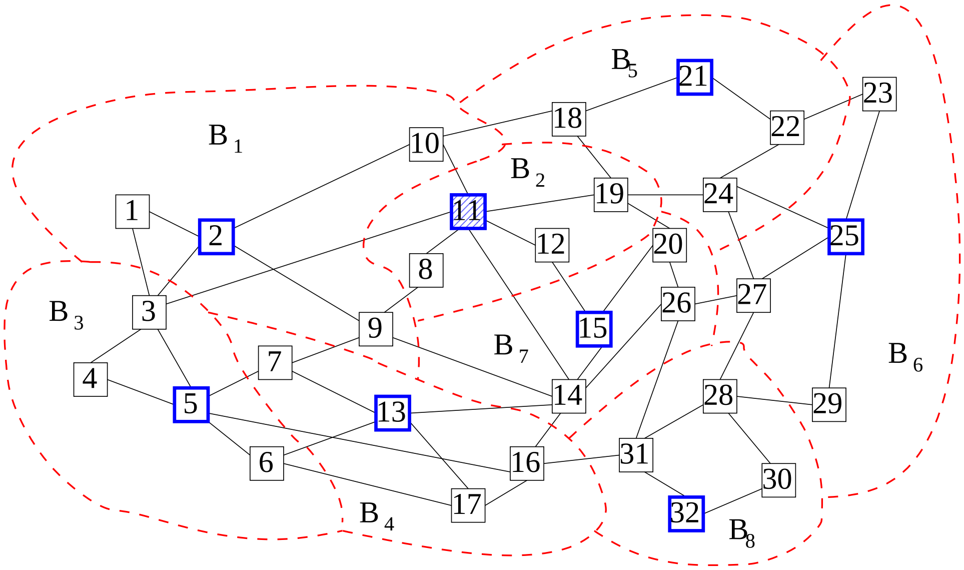

Figure 3.3: A 32-node network, partitioned into eight first-level clusters $B_1, \cdots, B_8$, with the corresponding hubs at the second level of the hierarchy

the network is first partitioned into clusters (or islands) of nodes, where each cluster consists of nodes in a contiguous region of the network. The clusters form the first level of the hierarchy, and may either correspond to independent administrative entities (e.g., autonomous systems), or may be created solely for the purpose of simplifying resource management and control functions (e.g., as in partitioning a single OSPF administrative domain into multiple areas). Within each cluster, one node is designated as the *hub*, and is responsible for grooming intra-cluster traffic as well as inter-cluster traffic originating or terminating locally. Hub nodes collectively form the second level of the hierarchy, and are expected to be provisioned with more resources (e.g., larger number of switching ports and higher capacity for grooming traffic) than non-hub nodes. Returning to the airline analogy, a hub node is similar in function to airports that serve as major hubs; these airports are typically larger than non-hub airports, in terms of both the number of gates ("ports") and physical space (for "switching" passengers between gates).

To illustrate this approach, let us consider the 32-node network in Figure 3.3. The figure shows a partition of the network into eight clusters, $B_1, \cdots, B_8$, each cluster consisting of four nodes. These clusters represent the first level of the hierarchy. Within each cluster, one node is the hub; for instance, node 2 is the hub for cluster B_1. The hub nodes of the eight first-level clusters form the second level of the hierarchy, and are responsible for grooming and routing inter-cluster traffic.

The main idea behind the hierarchical grooming strategy in [8] is to solve the first and second subproblems of the traffic grooming problem (i.e., construct the virtual topology and determine the routing of traffic components on it) *separately for each level of the hierarchy.* In the first step, each cluster is considered independently of the others, and a set of lightpaths is created to route local (intra-cluster) traffic, as well as inter-cluster traffic to and from the local hub. In the second step, lightpaths are created between the hub nodes to carry the inter-cluster traffic. Consequently, the problem of routing inter-cluster demands is divided into three simpler subproblems: routing the component to the local hub, from there to the remote hub, and then to the ultimate destination. Finally, given the set of inter- and intra-cluster lightpaths, the third subproblem can be solved on the underlying physical topology of the network using a standard RWA algorithm.

The hierarchical grooming algorithm for sub-wavelength demands consists of three phases [8], each discussed in the following three subsections.

Clustering and Hub Selection

The objective of this phase is twofold: (1) to partition the network nodes into some number k of clusters, denoted $B_1, \cdots, B_k$, and (2) to select one node in each cluster B_i as the hub, denoted h_i. Clearly, the number of clusters, their composition, and the corresponding hubs must be selected in a way that helps achieve the goal of minimizing the number of lightpaths and wavelengths required to carry the traffic demands. Therefore, the selection of clusters and hubs is a complex and difficult task, as it depends on both the physical topology of the network and the traffic matrix T. To illustrate this point, consider the tradeoffs involved in determining the number k of clusters[1]. If k is very small (but greater than one), the amount of inter-cluster traffic generated by each cluster will likely be large. Hence, the k hubs may become bottlenecks, resulting in a large number of ports at each hub and possibly a large number of wavelengths (since many lightpaths may have to be carried over the fixed number of links to/from each hub). On the other hand, a large value for k implies a small number of nodes within each cluster. In this case, the amount of intra-cluster traffic will be small, resulting in inefficient grooming (i.e., a large number of lightpaths); similarly, at the second-level cluster, $O(k^2)$ lightpaths will have to be set up to carry small amounts of inter-cluster traffic.

It was observed in [5,8] that the clustering and hub selection subproblem bears similarities to the classical k-center problem [18,20]. The objective of the k-center problem is to find a set S of k nodes (centers) in the network, so as to minimize the maximum distance from any network node to the nearest center. Thus, the set S implicitly defines k clusters with corresponding hub nodes in S. The k-center problem is NP-Complete, and the 2-approximation algorithm of [18]

[1] Note that in the special case of $k = 1$, there is a single cluster with one hub and $N - 1$ non-hub nodes, whereas in the special case $k = N$, there are N clusters, each with a single hub and no non-hub nodes.

was adapted in [5, 8] for the traffic grooming context.

A clustering algorithm designed specifically for hierarchical traffic grooming was presented in [7]. This work identified several grooming-specific factors affecting the selection of clusters and hubs, and developed a parameterized algorithm that can achieve a desired tradeoff among various goals. The algorithm partitions the network into clusters by considering: (1) the intra- and inter-cluster traffic, attempting to cluster together nodes with "dense" traffic in order to reduce the number of long inter-cluster lightpaths; (2) the capacity of the cut links connecting each cluster to the rest of the network, selecting clusters with a relatively large cut size so as to keep the number of wavelengths low; and (3) the physical shape of each cluster, attempting to avoid clusters with a large diameter. The algorithm also selects hubs on the basis of their physical degree, to prevent hub links from becoming bottlenecks. It was shown in [7] that this algorithm outperforms the k-center algorithm in terms of both the port and wavelength costs.

Hierarchical Virtual Topology and Traffic Routing

During this phase, the VTTR subproblem of the traffic grooming problem is solved. The outcome of this phase is a set of lightpaths for carrying the traffic demands, and a routing of individual traffic components over these lightpaths. The formation of the hierarchical virtual topology for traffic grooming follows three steps: formation of direct lightpaths, intra-cluster lightpaths, and inter-cluster lightpaths.

Direct lightpaths for large traffic demands. During this step, "direct-to-destination" lightpaths are created between two nodes that exchange large amounts of traffic, even if these nodes belong to different clusters. Similarly, "direct to/from remote hub" lightpaths are created between some node s and a remote hub h if there is a sufficiently large amount of traffic between s and the nodes in h's cluster. Setting up such lightpaths to bypass the local and/or remote hub node has several benefits: the number of lightpaths in the virtual topology is reduced, the number of ports and switching capacity required at hub nodes is reduced (leading to higher scalability), and the RWA algorithm may require fewer wavelengths (since hubs will be less of a bottleneck).

Intra-cluster lightpaths. At this step, each cluster is considered independently of the others, and intra-cluster lightpaths are formed. Consider some cluster B with hub h. B is viewed as a *virtual star* such that the intra-cluster lightpaths within the cluster are formed by (1) having all traffic to (respectively, from) any node s of B from (to) nodes outside the cluster originate (terminate) at the hub h, and (2) applying the algorithm for star networks discussed in Section 3.1.2 to cluster B in isolation. Having all inter-cluster traffic originate or terminate at the hub imposes a hierarchical structure to the virtual topology of lightpaths: inter-cluster traffic, other than that carried by direct lightpaths set up earlier, is first carried to the local hub, groomed there with inter-cluster traffic from other local nodes, carried on lightpaths to the destination hub (as we discuss shortly), groomed there with other local and non-local traffic, and finally carried to the destination node. Also, recall that the lightpaths created by the star algorithm are either "single-hop" (i.e., from a non-hub node to the hub, or vice versa), or "two-hop" (i.e., from one non-hub node to another). Hence, the routing of the individual traffic components is implicit in the hierarchical virtual topology of each cluster.

Inter-cluster lightpaths. At the end of the intra-cluster grooming step, all traffic (other than that carried by the initial direct lightpaths) from the nodes of a cluster B with destination outside the cluster, is carried to its hub h for grooming and transport to the destination hub. In order to carry this traffic, a second-level cluster is considered, consisting of the k hub nodes of the first-level clusters. This cluster is also viewed as a *virtual star* with an associated traffic matrix representing the inter-cluster demands only. The inter-cluster lightpaths to carry these demands are then obtained by applying the star algorithm of Section 3.1.2 to this cluster in isolation. As with intra-cluster lightpaths, the routing of the individual traffic components is implicit in the topology.

Routing and Wavelength Assignment

The outcome of the virtual topology phase is a set of lightpaths and an implicit routing of the original traffic components over these lightpaths. The objective of this phase is to route the lightpaths over the underlying physical topology, and color them using the minimum number of wavelengths. The static RWA problem on arbitrary network topologies has been studied extensively in the literature [2, 11, 21, 23, 35], and any existing algorithm may be used in this case. Hence, by decoupling the grooming and routing of sub-wavelength traffic components onto lightpaths from the routing and wavelength assignment for these lightpaths, hierarchical grooming may capitalize on the the vast body of research on RWA algorithms.

3.1.4 Discussion

The hierarchical grooming framework for general topology networks that was presented in [8] and we summarized above has the following desirable characteristics:

- it is hierarchical, facilitating control, management, and security functions;
- it decouples the grooming of traffic components into lightpaths from the routing and wavelength assignment for these lightpaths: grooming is performed on a virtual hierarchy of clusters while RWA is performed directly on the underlying physical topology;
- it provisions only a few nodes (the hubs) for grooming traffic they do not originate or terminate;

- it handles efficiently small traffic demands: at the first level of hierarchy, nodes pack their traffic on lightpaths to the local hub; at the second level, demands among remote clusters are packed onto lightpaths between the corresponding hubs; and

- it allows for large traffic components to be routed on direct lightpaths, eliminating the cost of terminating and switching them at intermediate nodes.

Note, however, that the algorithm proposed in [8] to solve the VTTR subproblem imposes the following constraints on the virtual topology:

- It defines a cluster of nodes around each hub, and requires non-hub nodes to groom inter-cluster traffic towards their local hub only.

- In constructing the virtual topology, it considers each cluster at the first level of the hierarchy, as well as the cluster of hubs at the second level, in isolation.

- It constructs the virtual topology for each cluster by viewing it as a *virtual* star and applying the algorithm in Section 3.1.2, despite the fact that the physical topology of the cluster may be very different than that of a *physical* star.

With these constraints, the algorithm only takes a few seconds to construct the virtual topology for networks with fifty or more nodes [8]. On the other hand, the constraints exclude a large number of potential solutions hence the algorithm explores only a small fraction of the hierarchical virtual topology solution space. For instance, solutions that groom inter-cluster traffic from a node through a hub other than the node's local hub are disallowed. Similarly, at the second level of the hierarchy, grooming of traffic between hubs is only allowed via a *virtual star* topology.

In the following, we define several variants of the hierarchical VTTR problem so as to explore the spectrum of solutions between (1) the flat grooming approach that is the subject of most studies, and (2) the hierarchical grooming algorithm of [8].

3.2 Hierarchical VTTR Problem and Variants

Consider a connected graph $G = (\mathcal{V}, \mathcal{L})$, where $\mathcal{V}$ denotes the set of nodes and $\mathcal{L}$ denotes the set of directed links (arcs) in the network. We define $L = |\mathcal{L}|$ as the number of links. Each directed link $l \in \mathcal{L}$ consists of an optical fiber that may support W distinct wavelengths indexed as $1, 2, \ldots, W$. Let $T = [t^{sd}]$ denote the traffic demand matrix, where t^{sd} is a non-negative integer representing the traffic demand units to be established from source node s to destination node d. In general, traffic demands may be asymmetric, i.e., $t^{sd} \neq t^{ds}$. We also make the assumption that $t^{ss} = 0, \forall s$. Finally, we denote C as the capacity of a single wavelength channel in terms of traffic demand units.

We are interested in minimizing the total number of lightpaths used in the network; such an objective minimizes the use of critical resources and provides ample flexibility for future expansion of the network. Furthermore, we consider a hierarchical approach to traffic grooming that consists of the three phases we described in Section 3.1.3. More specifically, we assume that the first phase (hub selection) and the third phase (routing and wavelength assignment) are solved using the algorithms discussed in Sections 3.1.3. Therefore, for the remainder of this work we focus on hierarchical solutions to the virtual topology and traffic routing (VTTR) problem of the second phase.

To define the *hierarchical VTTR (H-VTTR)* problem, we assume that a set $\mathcal{H} \subset \mathcal{V}$ of hub nodes in the network is given. Hub nodes are nodes with traffic grooming capabilities. However, in contrast with the work in [8] (and the problem variant we discuss in the following subsection), no clusters are defined in the network; in other words, non-hub nodes are *not* assumed to be assigned to clusters and associated with a "local" hub. We also let $\mathcal{N} = \mathcal{V} \backslash \mathcal{H}$ be the set of non-hub nodes, and $K = |\mathcal{H}|$ be the number of hubs.

Definition 3.2.1 (H-VTTR) *Given the set $\mathcal{V}$ of nodes in the graph G, the set of hubs $\mathcal{H}$, the wavelength capacity C, and the traffic demand matrix T, establish the minimum number of lightpaths to carry all traffic demands, under two constraints: (1) only hub nodes may groom traffic that they do not themselves originate or terminate, and (2) no direct lightpaths between two non-hub nodes (i.e., nodes in $\mathcal{N}$) are allowed.*

We now present a formulation of the H-VTTR problem.

We use the following notation:

- $\mathcal{H}$ denotes the set of hub nodes, and $K = |\mathcal{H}|$ denotes the number of hub nodes.
- $\mathcal{N}$ is the set of non-hub nodes, and $N = |\mathcal{N}|$ represents the number of non-hub nodes.
- $\mathcal{L}$ denotes the set of (directed) physical links, and $L = |\mathcal{L}|$ is the number of links.
- $T = \{t^{sd}\}$ is the traffic demand matrix representing demands from any source node s to any destination node d.
- $\mathcal{Z} = \{(i,j) | i \in \mathcal{H} \text{ or } j \in \mathcal{H}, i \neq j\}$ is the set of pairs of nodes such that at least one node in the pair is a hub node; in other words, $\mathcal{Z}$ is the set of pairs of nodes between which direct lightpaths are allowed.

We also define these decision variables:

- b_{ij}, $(i,j) \in \mathcal{Z}$: the number of lightpaths originating at node i and terminating at node j.
- $t^{s,d}_{h_i,h_j}$: the amount of traffic from any source node s to destination node d carried on lightpaths from hub node h_i to hub node h_j.
- $t^{n,d}_{n,h_i}$: the amount of traffic from non-hub node n to any destination node d carried on lightpaths from n to hub node h_i.
- $t^{s,n}_{h_i,n}$: the amount of traffic from any source node s to a non-hub node n carried on lightpaths from hub node h_i to n.

With these definitions, we have the following multi-commodity flow formulation for the H-VTTR problem:

Objective function: minimize the number of lightpaths

$$\min \sum_{(i,j)\in\mathcal{Z}} b_{ij} \tag{3.1}$$

Constraints:

Capacity Constraint

$$\sum_{s,d\in\mathcal{N}\cup\mathcal{H},s\neq d} t^{sd}_{ij} \leq b_{ij}C, \quad (i,j)\in\mathcal{Z} \tag{3.2}$$

Flow Conservation Constraints at Intermediate Nodes

$$\sum_{h_j\in\mathcal{H},h_j\neq h_i} t^{sd}_{h_ih_j} + t^{sd}_{h_id} - \sum_{h_j\in\mathcal{H},h_j\neq h_i} t^{sd}_{h_jh_i} - t^{sd}_{sh_i} = 0, \\ s,d\in\mathcal{N}, h_i\in\mathcal{H} \tag{3.3}$$

$$\sum_{h_j\in\mathcal{H},h_j\neq h_i} t^{sd}_{h_ih_j} - \sum_{h_j\in\mathcal{H},h_j\neq h_i} t^{sd}_{h_jh_i} = 0, \\ s,d\in\mathcal{H}, h_i\in\mathcal{H}, h_i\neq s, h_i\neq d \tag{3.4}$$

$$\sum_{h_j\in\mathcal{H},h_j\neq h_i} t^{sd}_{h_ih_j} + t^{sd}_{h_id} - \sum_{h_j\in\mathcal{H},h_j\neq h_i} t^{sd}_{h_jh_i} = 0, \\ s\in\mathcal{H}, d\in\mathcal{N}, h_i\in\mathcal{H}, h_i\neq s \tag{3.5}$$

$$\sum_{h_j \in \mathcal{H}, h_j \neq h_i} t^{sd}_{h_i h_j} - \sum_{h_j \in \mathcal{H}, h_j \neq h_i} t^{sd}_{h_j h_i} - t^{sd}_{sh_i} = 0,$$
$$s \in \mathcal{N}, d \in \mathcal{H}, h_i \in \mathcal{H}, h_i \neq d \qquad (3.6)$$

Flow Conservation Constraints at Source Nodes

$$\sum_{h_i \in \mathcal{H}, h_i \neq s} t^{sd}_{sh_i} = t^{sd},$$
$$s \neq d, s, d \in \mathcal{H}, \text{ or } s, d \in \mathcal{N}, \text{ or } s \in \mathcal{N}, d \in \mathcal{H} \qquad (3.7)$$

$$\sum_{h_i \in \mathcal{H}, h_i \neq s} t^{sd}_{sh_i} + t^{sd}_{sd} = t^{sd}, \quad s \neq d, s \in \mathcal{H}, d \in \mathcal{N} \qquad (3.8)$$

$$\sum_{h_i \in \mathcal{H}, h_i \neq s} t^{sd}_{h_i s} = 0, \quad s \in \mathcal{H}, d \in \mathcal{H} \cup \mathcal{N} \qquad (3.9)$$

Flow Conservation Constraints at Destination Nodes

$$\sum_{h_i \in \mathcal{H}, h_i \neq d} t^{sd}_{dh_i} = 0, \quad s \in \mathcal{H} \cup \mathcal{N}, d \in \mathcal{H} \qquad (3.10)$$

$$\sum_{h_i \in \mathcal{H}, h_i \neq d} t^{sd}_{h_i d} = t^{sd},$$
$$s \neq d, s, d \in \mathcal{H}, \text{ or } s, d \in \mathcal{N}, \text{ or } s \in \mathcal{H}, d \in \mathcal{N} \qquad (3.11)$$

$$\sum_{h_i \in \mathcal{H}, h_i \neq d} t^{sd}_{h_i d} + t^{sd}_{sd} = t^{sd}, s \neq d, s \in \mathcal{N}, d \in \mathcal{H} \qquad (3.12)$$

Constraint (3.2) ensures that enough lightpaths are established to satisfy the traffic demand between each pair of nodes. Constraints (3.3) to (3.12) are the flow conservation constraints at intermediate nodes ((3.3-(3.6)), source nodes ((3.7)-(3.9)), and destination nodes ((3.10)-(3.12)). There are different flow conservation constraints depending on whether the source and destination nodes are hub or non-hub nodes, as this determines whether a drect lightpaths can be established between the two.

H-VTTR is a generalization of the VTTR problem we defined and studied in Chapter 2. Specifically, VTTR allows grooming of traffic to take place at any node in the network, as well as lightpaths to exist between any pair of nodes in the network. Therefore, if we let $\mathcal{H} = \mathcal{V}$ and $\mathcal{N} = \emptyset$, i.e., each node to be a hub node, H-VTTR reduces to VTTR. Note also that, because of the constraint on direct lightapths, traffic between two non-hub nodes has to be carried on at least two lightpaths via at least one hub node.

In the following subsections we introduce two variants of the H-VTTR problem, both inspired by the airline industry's hub-and-spoke model.

3.2.1 H-VTTR with Clustering (HC-VTTR)

The hierarchical VTTR with clustering (HC-VTTR) problem is a variant of H-VTTR that adopts the concept of clustering considered in [8]. Specifically, we assume that the set $\mathcal{V}$ of network nodes is partitioned into $K = |\mathcal{H}|$ clusters, $v_1, \ldots, v_K$, and that node $h_i \in \mathcal{H}$ is the hub node of cluster v_i. In HC-VTTR, traffic originating from, or terminating at, a non-hub node in cluster v_i may only be groomed with other traffic at the local hub h_i. More formally, we have the following definition.

Definition 3.2.2 (HC-VTTR) *Given the set $\mathcal{V}$ of nodes in the graph G, the set of hubs $\mathcal{H}$, a set of $K = |\mathcal{H}|$ clusters $\{v_1, \ldots, v_H\}$ such that each node $h_i \in \mathcal{H}$ is the hub of cluster v_i, the wavelength capacity C, and the traffic demand matrix T, establish the minimum number of lightpaths to carry all traffic demands, under three constraints: (1) only hub nodes may groom traffic that they do not originate or terminate, (2) traffic originating from, or terminating at, a non-hub node in cluster v_i may only be groomed with other traffic at the local hub h_i, and (3) no direct lightpaths between two non-hub nodes (i.e., nodes in $\mathcal{N}$) are allowed.*

The key idea in HC-VTTR is to ensure that grooming of traffic takes place "near" non-hub nodes (i.e., at their local hub). Local grooming handles small traffic demands efficiently, and it prevents solutions with long underutilized lightpaths. On the other hand, traffic between two non-hub nodes in different clusters must be carried on at least three lightpaths: from the source node to its local hub, then to the remote hub, and finally to the destination node. Although we omit the ILP formulation of the HC-VTTR problem, it is similar to that of the H-VTTR problem with additional constraints to prevent the establishment of lightpaths between a non-hub node and hubs other than the one in its cluster.

3.2.2 Hierarchical Grooming with Direct Lightpaths

The H-VTTR and HC-VTTR problems explicitly prevent direct lightpaths between non-hub nodes. Note, however, that if there is sufficient traffic between two non-hub nodes to fill a lightpath, forcing this traffic to travel via a hub node results in more lightpaths: sending the traffic directly to its destination requires only one lightpath, whereas sending it through one or more hubs requires at least two lightpaths without improving the grooming of other traffic (since this traffic takes up the whole capacity of these lightpaths). Our experience [8] also indicates that it is often cost-effective to establish partially filled direct lightpaths as long as these lightpaths have high utilization (i.e., the traffic between the two non-hub nodes is close to the capacity of a lightpath). Such high direct traffic demands may not present effective opportunities to groom other traffic on the same lightpaths; furthermore, including partially filled lightpaths in the solution makes it possible to accommodate future increases in traffic demands without the need to establish new lightpaths, an important consideration for long-term network planning.

We now formally define the H-VTTR problem with direct lightpaths (H-VTTR/DL):

Definition 3.2.3 (H-VTTR/DL) *Given the set $\mathcal{V}$ of nodes in the graph G, the set of hubs $\mathcal{H}$, the wavelength capacity C, the traffic demand matrix T, and a threshold $\theta, 0 < \theta \leq 1$, establish the minimum number of lightpaths to carry all traffic demands, under two constraints: (1) only hub nodes may groom traffic that they do not originate or terminate, and (2) direct lightpaths between two non-hub nodes (i.e., nodes in $\mathcal{N}$) are allowed only if the traffic between these nodes is at least equal to θC.*

The only difference between the H-VTTR/DL and H-VTTR problems is that in the former we allow direct lightpaths between non-hub nodes, whereas such lightpaths are not allowed in the latter. Therefore, the formulation of H-VTTR/DL is very similar to (3.1)-(3.12), with the following differences:

- The set $\mathcal{Z}$ is redefined as: $\mathcal{Z} = \{(i,j)|i,j \in \mathcal{H} \cup \mathcal{N}, i \neq j$ to allow direct lightpaths between any pair of nodes;
- Flow conservation constraints (3.7) and (3.11) are removed;
- Flow conservation constraints (3.8) and (3.12) are modified to apply to all $s, d \in \mathcal{H} \cup \mathcal{N}, s \neq d$; and
- The following constraints are added to ensure that direct lightpaths between a pair of non-hub nodes may be established only if the traffic between these nodes exceeds the given threshold θ:

$$b_{sd} = 0, \quad t^{sd} \leq \theta C, s, d \in \mathcal{N}. \tag{3.13}$$

Note that the fact that direct lightpaths may be established between non-hub nodes (something that is not allowed under H-VTTR) makes it possible to simplify the formulation by removing and modifying, respectively the above pairs of flow conservation constraints.

A similar HC-VTTR/DL problem with clustering can be defined, in which direct lightpaths between non-hub nodes, or a non-hub node and a remote hub, are allowed as long as the traffic between these nodes is at least equal to θC. The ILP formulation of the problem is omitted, but it is similar to the formulation of H-VTTR/DL with additional constraints.

The HC-VTTR/DL problem is identical to the one studied in [8]. But whereas the virtual topology algorithm developed in [8] treated each cluster in isolation as a virtual star and used a heuristic to determine the lightpaths, the ILP formulation we developed in this work considers the clusters in an integrated manner and solves the HC-VTTR problem optimally.

3.3 Numerical Results

In this section we evaluate the performance of hierarchical solutions to the VTTR problem in terms of two metrics: quality of solution (i.e., the number of lightpaths produced by the solution) and running time. Specifically, we compare the following five ILP formulations:

1. H-VTTR (the problem is defined in Section 3.2 and the ILP formulation is shown in Section 3.2);

2. H-VTTR/DL (the problem is defined in Section 3.2.2 and the formulation is shown in Section 3.2.2);

3. HC-VTTR (the problem is defined in Section 3.2.1);

4. HC-VTTR/DL (the problem is defined in Section 3.2.2); and

5. VTTR (the problem is defined in Section 3.1.3).

Note that the VTTR ILP formulation is shown in Section 2.1, which takes a flat view of the network such that grooming may take place at any node, not just hubs, and lightpaths are allowed between any pairs of nodes without any threshold constraints on the traffic demands. Since the four hierarchical formulations we presented in this paper are derived from the VTTR formulation by adding appropriate constraints, the solution to the VTTR formulation provides a lower bound for the solutions to the hierarchical formulations. (In fact, as we showed in [39], the solution to the VTTR formulation is a lower bound to the solution of the original traffic grooming problem, and it is optimal whenever the network is not wavelength limited.) Hence, we are interested in characterizing the performance of the hierarchical solutions relative to the baseline VTTR formulation.

In our study we consider four network topologies (link counts refer to directed links): the 14-node, 42-link NSFNet [30], shown in Fig. 3.4; the 17-node, 52-link German network [19],

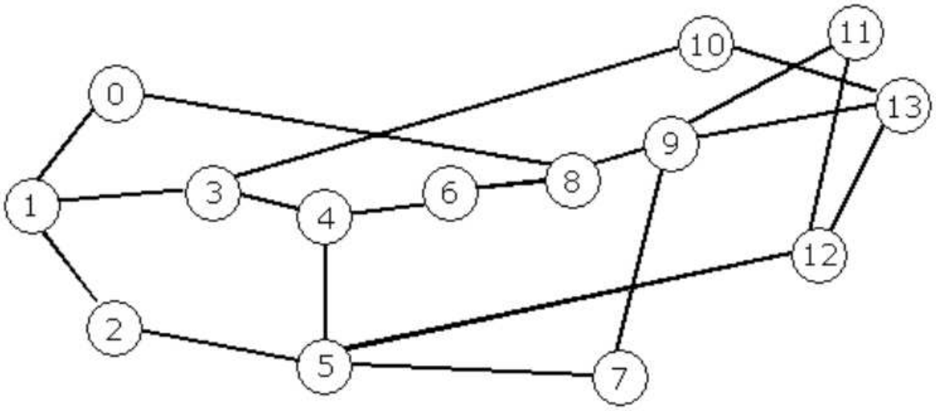

Figure 3.4: 14-node NSFnet

shown in Fig. 3.5; the 32-node, 106-link network shown in Figure 3.3; and the 47-node, 192-link network from [1], shown in Fig. 3.6.

For each problem instance, we generate the traffic matrix $T = [t^{sd}]$ by drawing each traffic demand t^{sd} uniformly and randomly in the interval $[0, t_{max}]$. Each data point in the following figures is the average of ten problem instances. For the experiments, we fix the wavelength capacity $C = 16$, and we vary the value of parameter $t_{max} = 10, 20, 30, 40, 50, 60$, to investigate various traffic loads. The results we present were obtained by running the IBM CPLEX 12 optimization tool on a cluster of identical compute nodes with dual Woodcrest Xeon CPU at 2.33GHz with 1333MHz memory bus, 4GB of memory and 4MB L2 cache. We imposed a 3% relative optimality gap in solving the optimization problems with CPLEX.

3.3.1 Direct Lightpath Threshold

Let us first investigate the impact of the threshold θ on the performance of the H-VTTR/DL and HC-VTTR/DL formulations. Figure 3.7 shows the objective value (i.e., number of lightpaths) in the solution of the H-VTTR/DL problem on the 32-node network under various traffic loads (i.e., values of parameter t_{max}). Figure 3.8 is similar but presents results for the HC-VTTR/DL formulation. We observe that the objective value is initially flat, and then slowly increases as the threshold value θ increases, across all traffic load values shown. This behavior can be explained by the fact that the constraints (3.13) on direct lightpaths become stricter as the threshold value θ increases, hence we expect that the solutions to also become worse. Nevertheless, lowering the threshold value below $\theta = 0.6$ has no discernible effect on the objective value, confirming our intuition that direct lightpaths are useful only if there is substantial traffic between two non-hub nodes. Therefore, for the experiments we present in the rest of this section, the threshold value

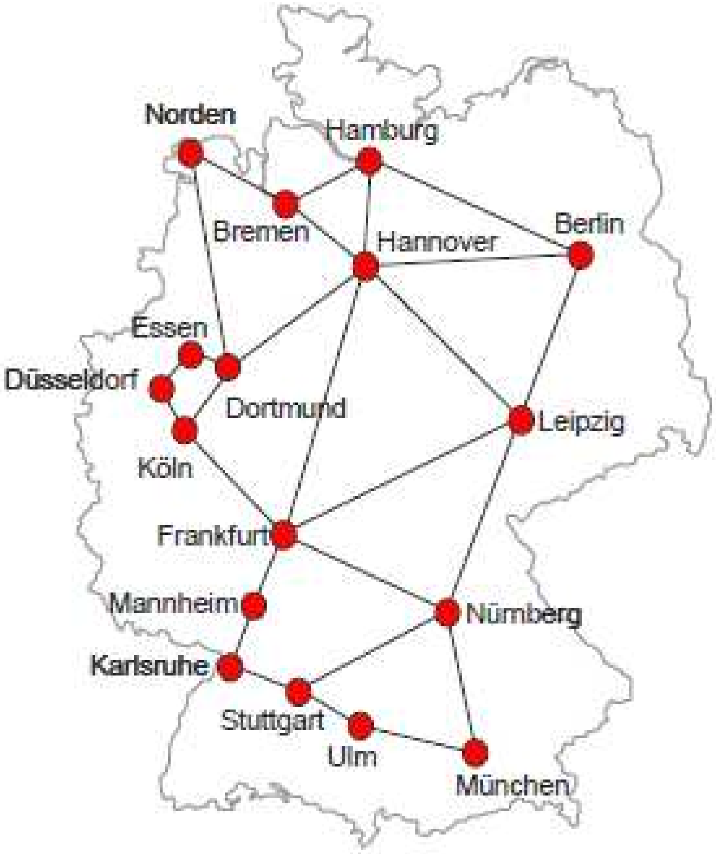

Figure 3.5: 17-node German Network

was fixed to $\theta = 0.6$.

3.3.2 Quality of Solution and Scalability

Figures 3.9 and 3.10 compare the five formulations above across the four network topologies, in terms of the objective value and the CPU time it takes CPLEX to solve them, respectively. For these experiments, we set $t_{max} = 40$, and we used the k-center algorithm [18] to determine the hubs for each topology. Specifically, we set the number of hubs to four for the 14- and 17-node topologies, and eight for the 32- and 47-node networks. We also set a time limit of two hours. As we can see, CPLEX was able to solve all the formulations within the time limit, except for the VTTR formulation on the 47-node network for which no integer solution was found within two hours; hence, the two figures do not present results for this formulation and topology.

Let us first refer to Figure 3.9 that compares the five formulations in terms of solution quality. We first note that the objective value increases with the size of the network topology,

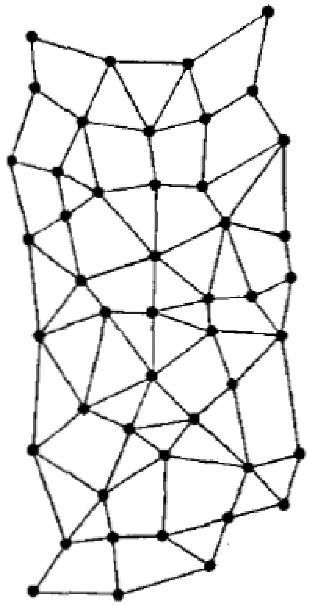

Figure 3.6: 47-node Network from [1]

as expected: for a given value of t_{max}, a larger network has more traffic to carry than a smaller one, requiring a larger number of lightpaths. We also note that the objective value obtained by solving the HC-VTTR formulation is always higher than that obtained by H-VTTR. Recalling the problem definitions, HC-VTTR includes more constraints than H-VTTR: in the former, traffic from a non-hub node must be groomed at the local hub, whereas in the latter it may be groomed at any hub node. Therefore, the solution to HC-VTTR cannot be better than that to H-VTTR. Also, the variants that allow for direct lightpaths (H-VTTR/DL and HC-VTTR/DL) lead to solutions that are better than variants that do not allow direct lightpaths (H-VTTR and HC-VTTR, respectively). Again, this result can be explained by the fact that allowing direct lightpaths increases the space of candidate solutions. Finally, the original VTTR formulation produces the best solution, as expected, for the three topologies for which a solution to this formulation was obtained within the time limit. However, in all three cases, the solution to H-VTTR/DL is very close to that of VTTR. Overall, the relative performance of the five formulations is consistent across the four topologies: VTTR leads to the best solution, followed by H-VTTR/DL, HC-VTTR/DL, H-VTTR, and HC-VTTR, in this order.

Let us now turn our attention to Figure 3.10 that compares the running time for the five formulations. We observe that solving the HC-VTTR formulation takes the least amount of time, less than a second, on average, even for the 47-node network. Among the hierarchical formulations, the next fastest solution time is achieved by HC-VTTR/DL, followed by H-

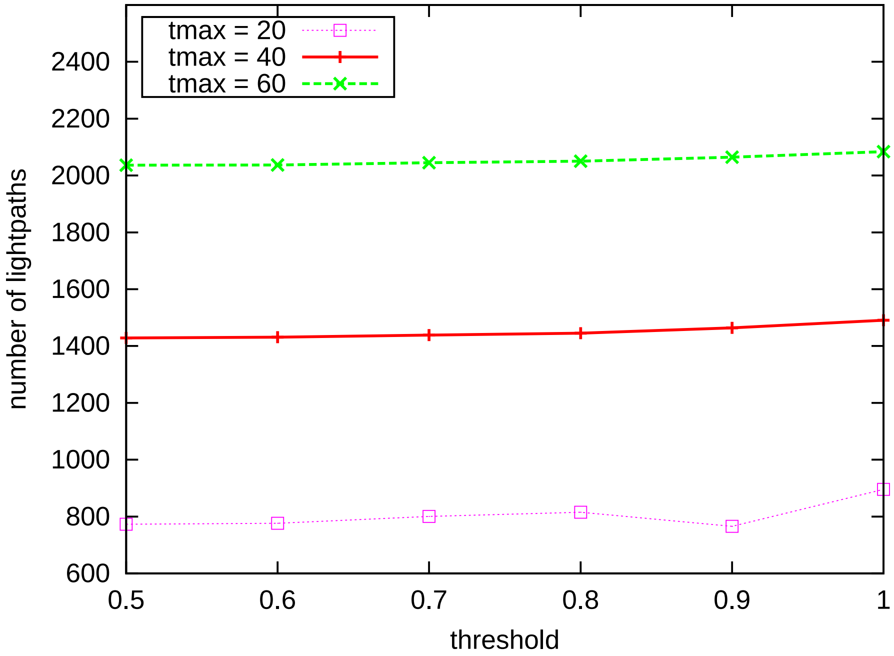

Figure 3.7: Objective value of H-VTTR/DL as a function of threshold value θ

VTTR and H-VTTR/DL. We also note that, for a given formulation, the running time is similar for the 14- and 17-node networks, and is also similar (but higher) for the 32- and 47-node networks. On the other hand, for the two small networks, the VTTR formulation that does not impose any hierarchical structure on the topology, takes about the same time as H-VTTR/DL, the hierarchical formulation with the worst running time. But whereas the running time of H-VTTR/DL increases by a small factor as we move from the 17- to the 32-node network, the running time of VTTR increases by almost three orders of magnitude; similarly, the running time of H-VTTR/DL increases slightly from the 32- to the 47-node network, but the running time of VTTR increases significantly and exceeds the two-hour limit we imposed. From these results, we conclude that imposing a hierarchical structure on the virtual topology is not beneficial in terms of running time when the size of the network is relatively small (in our study, up to 17 nodes). However, as the network size grows, flat solutions (i.e., VTTR) do not scale whereas hierarchical solutions scale quite well; indeed, it is at larger network sizes that one would expect the benefits of hierarchical structures to materialize. Overall, these results indicate that H-VTTR/DL represents the best tradeoff between running time and solution quality, as it takes, on average, about 100 seconds or less to obtain solutions close to the optimal (i.e., that obtained by VTTR).

In Figure 3.11 we compare the solution quality of the five formulations as a function of

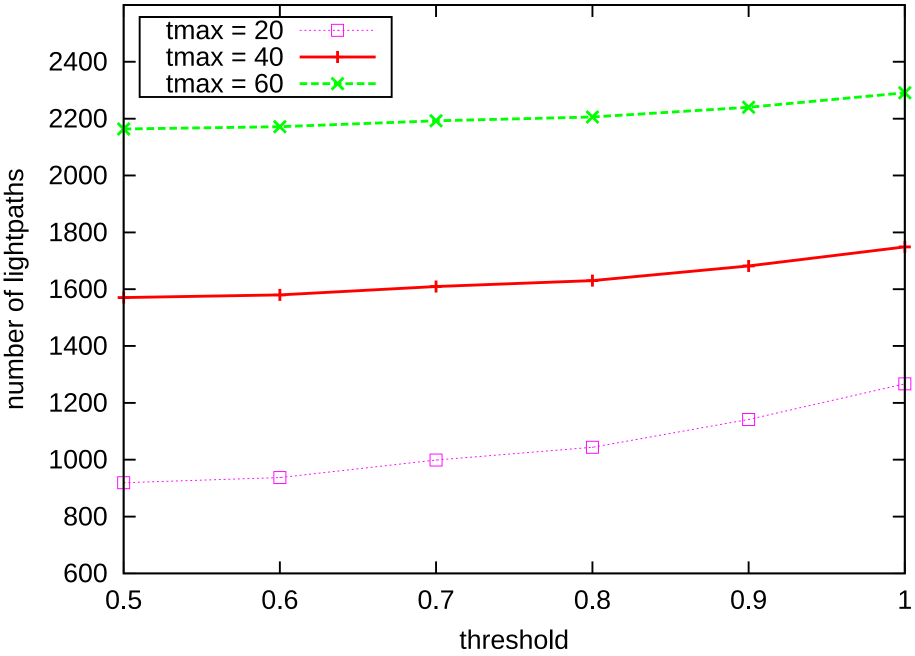

Figure 3.8: Objective value of HC-VTTR/DL as a function of threshold value θ

t_{max}, i.e., the traffic load. The results shown are for the 32-node network with eight hubs; the relative behavior of the various curves is representative of that for the other topologies. As the traffic load increases, the number of lightpaths increases almost linearly, but the rate of increase depends on the particular formulation. We also observe that the relative performance across the various values of t_{max} is similar to that in Figure 3.9, i.e., HC-VTTR requires the largest number of lightpaths, followed by H-VTTR and HC-VTTR/DL, while H-VTTR/DL and VTTR have very similar objective values. These reults further support our earlier conclusion that H-VTTR/DL provides the best tradeoff between running time and solution quality.

3.3.3 Effect of Number of Hubs

Finally, Figures 3.12 and 3.13 show the effect of the number K of hubs on the objective value and running time, respectively, for the four hierarchical grooming formulations. As we can see, the number of hubs has little effect on the number of lightpaths for formulations that allow direct lightpaths; this result is due to the fact that a good amount of traffic is sent over such direct lightpaths and hence the number of hubs is not very important. For H-VTTR, as the number of hubs increases, the solution improves; since H-VTTR allows a node to send traffic to any hub, increasing the number of hubs also increases the number of candidate solutions. HC-VTTR, on the other hand, requires each node to send its traffic to the local hub, hence increasing the

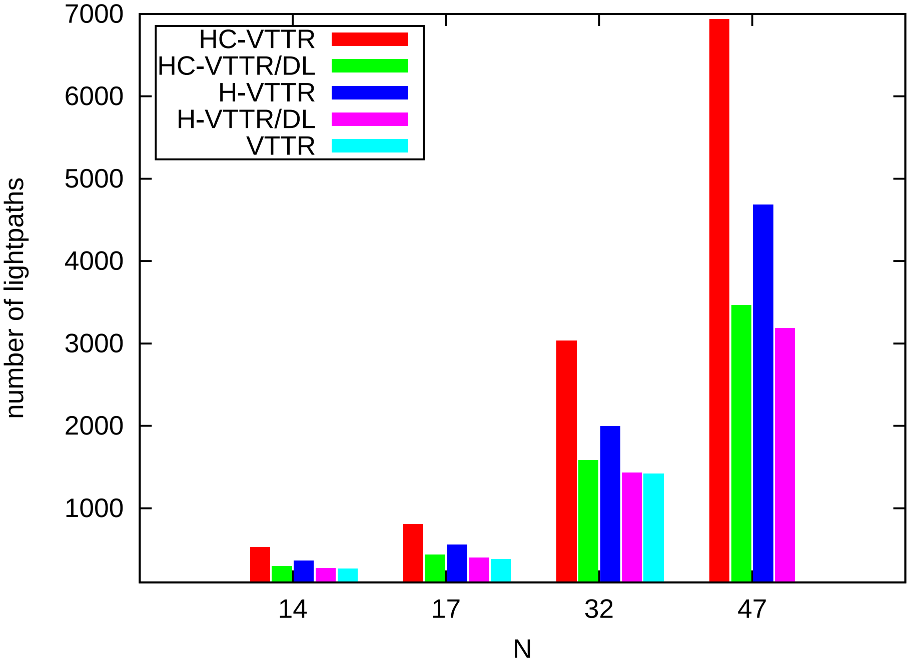

Figure 3.9: Objective value comparison, $t_{max} = 40$

number of hubs may also initially increase the overall number of lightpaths. As Figure 3.13 indicates, the running time increases with the number of hubs across all formulations, between one and two orders of magnitude $K = 2$ to $K = 8$. Therefore, if one of the formulations that allow direct lightpaths is adopted, these results indicate that a smaller number of hubs should be used.

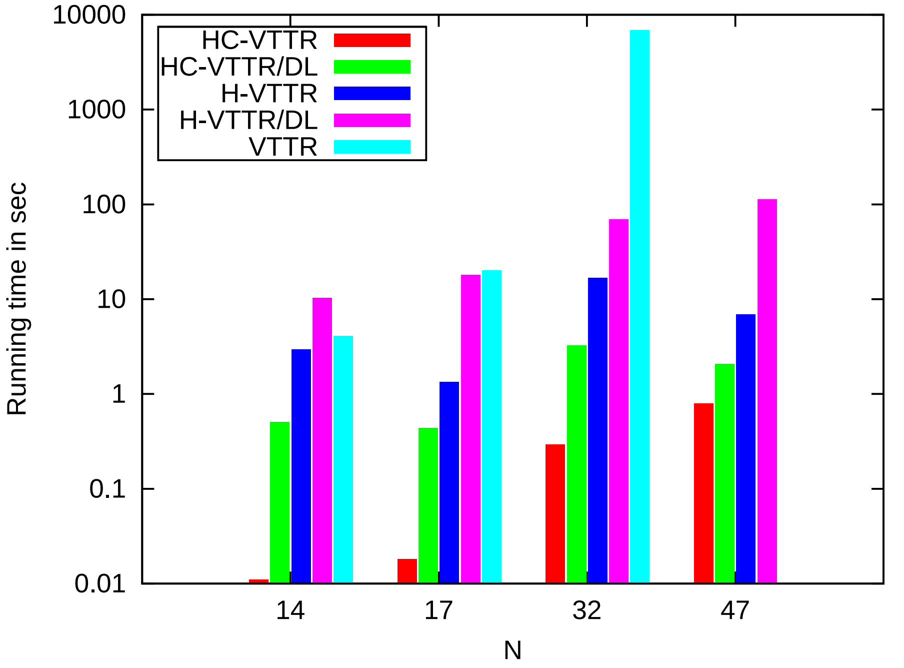

Figure 3.10: CPU time comparison, $t_{max} = 40$

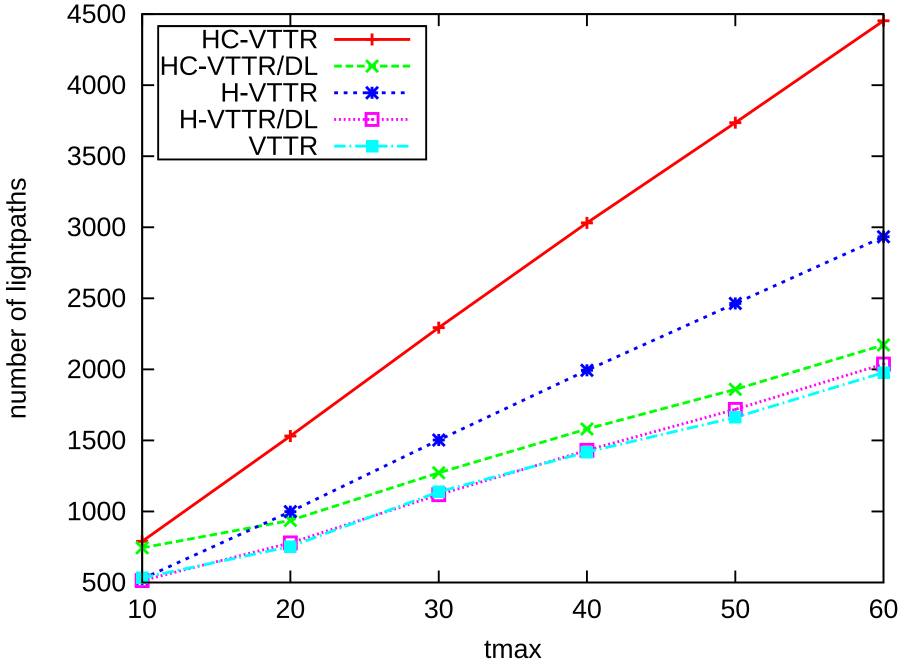

Figure 3.11: Objective value comparison, 32-node network, $K = 8$ hubs

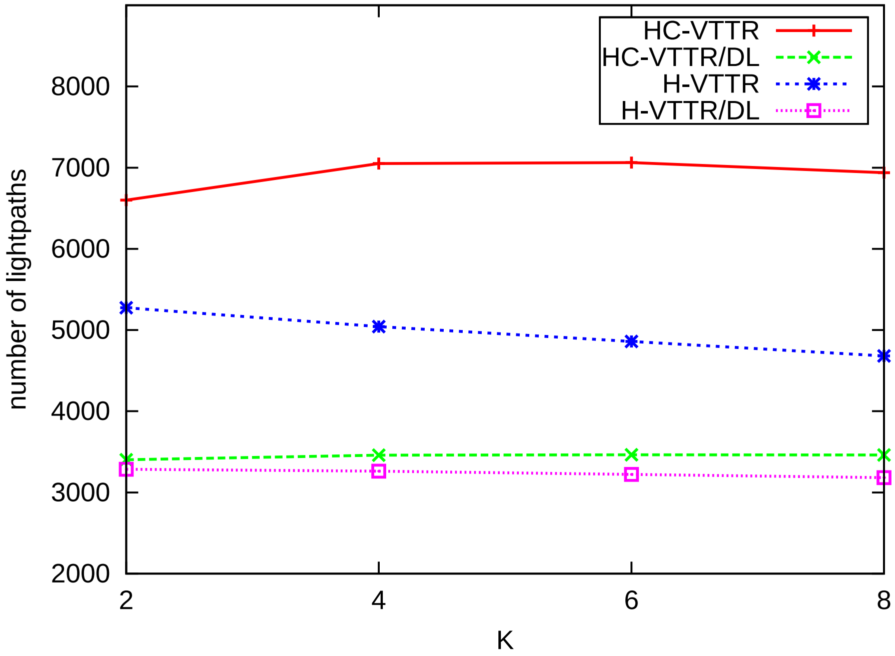

Figure 3.12: Objective value comparison, 47-node network, $t_{max} = 40$

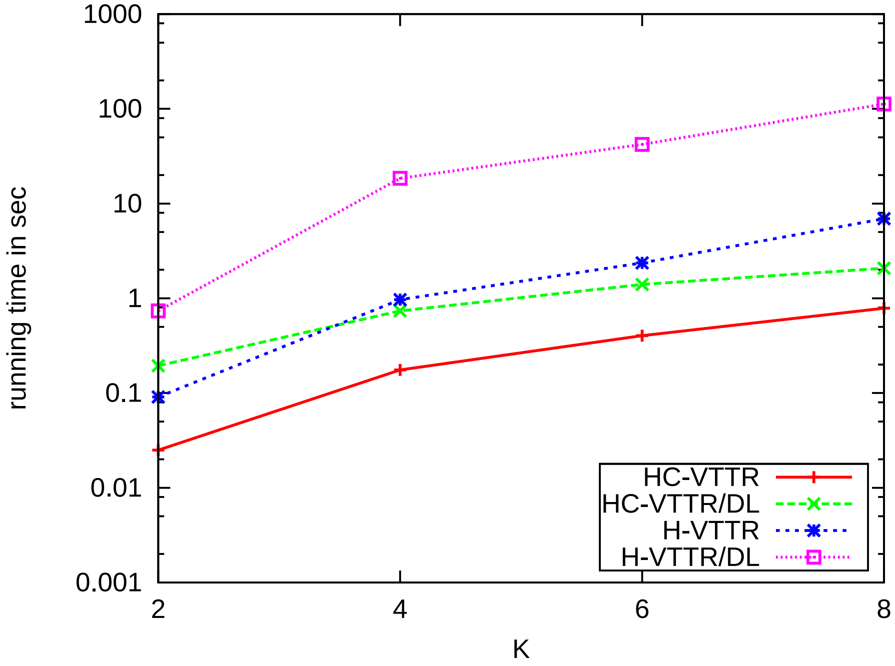

Figure 3.13: CPU time comparison, 47-node network, $t_{max} = 40$

Chapter 4

Multi-class Traffic Grooming

4.1 Introduction

In Chapter 3, we have studied the VTTR problem from the aspect of network hierarchy, and we have demonstrated that hierarchical traffic grooming is an efficient and scalable way to groom traffic in large-scale networks with realistic sizes. We have also developed a suite of ILP formulations representing various tradeoffs between solution quality and running time. In this chapter, we consider the VTTR problem from a different point of view - traffic demands.

A fundamental assumption underlying most traffic grooming studies has been that a lightpath may carry *any* traffic demands, without any constraints on which traffic components can be groomed on a particular lightpath. This is a reasonable assumption if the only objective is to minimize network cost or maximize revenue: more opportunities for grooming traffic lead to better utilization of lightpaths, which in turn results in either lower cost (for a fixed set of demands) or in more accepted connections (for a given level of network resources).

In this chapter, we introduce and study the concept of multi-class traffic grooming, whereby the network supports multiple classes of traffic such that a given lightpath may only be used to groom traffic from a certain class. In Section 4.2, we introduce the multi-class traffic grooming problem. In Section 4.3, we discuss the multi-class virtual topology and traffic routing (MC-VTTR) problem, and in Section 4.4, we develop algorithms to solve MC-VTTR. We evaluate these algorithms in Section 4.5.

4.2 Multi-Class Traffic Grooming

We consider an optical network with a physical topology represented as a graph $G = (\mathcal{N}, \mathcal{A})$, where $\mathcal{N}$ is the set of network nodes and $\mathcal{A}$ is the set of physical links (arcs) interconnecting the nodes. We assume that each physical link (arc) is directed and composed of a single fiber supporting W wavelength channels. The nodes are connected to each other by a pair of links in opposite directions. The capacity of each wavelength channel is measured in multiples of a unit traffic rate (e.g., OC-3) and is denoted by integer C.

We also assume that traffic carried by the network belongs to one of $K, K > 1$ distinct classes. Traffic classes may be defined across various dimensions, including:

- quality of Service (QoS) requirements, in which case the network may provision different lightpaths to support different levels of QoS;
- the user or group of users, in which case the network operator may groom traffic of different organizations (e.g., competitors) onto different sets of lighgtpaths; and/or
- privacy or security considerations, in which case traffic grooming encompasses flow isolation considerations [40].

Given such a classification of user traffic, and regardless of how the various classes are defined, we impose the following constraint on traffic grooming:

> *Multi-Class Grooming Constraint:* Grooming onto a given lightpath is permitted if and only if the traffic to be groomed belongs to the *same class*.

We let $t_k^{sd}, k = 1, \ldots, K$, denote the amount of class-k traffic originating at node s and terminating at node d, where quantity t_k^{sd} is an integer multiple of the unit traffic rate. We also let $T_k = [t_k^{sd}]$ denote the network traffic matrix for class k. The objective of the multi-class traffic grooming problem is to determine the virtual topology of lightpaths and the routing of traffic components $\{t_k^{sd}\}$ over these lightpaths, under the class-specific grooming constraints we mentioned above, so as to minimize the number of lightpaths.

This new variant of the traffic grooming problem balances the objective of minimizing network cost against the requirement to isolate classes of traffic. Clearly, the additional constraint of only grooming traffic within the same class is expected to limit the opportunities for grooming and drive the cost higher; we investigate this issue in Section 4.5. Nevertheless, such a cost increase may be acceptable given that the resulting solution provides higher security guarantees for which the users are likely to be willing to pay a premium.

4.3 Multi-Class Virtual Topology and Traffic Routing (MC-VTTR)

We first note that the special case $K = 1$ (i.e., a single traffic class encomassing all traffic demands) corresponds to the classical traffic grooming problem that has been studied in the past. In Chapter 2, we proposed a decomposition of the single-class traffic grooming problem into two subproblems that are solved sequenbtially:

1. *virtual topology and traffic routing (VTTR):* determine the minimum number of lightpaths to carry all traffic demands, and route these demands over the virtual topology; and
2. *lightpath routing and wavelength assignment (RWA):* find a wavelength and physical path for each lightpath in the virtual topology, so as to minimize the number of wavelengths.

As we have shown in Chapter 2, if the network is not bandwidth (wavelength) limited, this sequential solution has two important properties: (1) it is optimal for the original traffic grooming problem, in the sense that the number of lightpaths it produces is minimum for carrying the traffic demands, and (2) it also minimizes the number of wavelengths used. In Chapter 2, we presented LP relaxation techniques to solve the VTTR problem efficiently.

For the new traffic grooming variant we consider in this paper, we define the multi-class virtual topology and traffic routing (MC-VTTR) problem as follows:

Definition 4.3.1 (MC-VTTR) *Given the traffic demand matrices $T_k = [t_k^{sd}], k = 1, \ldots, K$, and the wavelength capacity C, establish the minimum number of lightpaths to carry all traffic demands under the multi-class grooming constraint.*

The ILP formulation of MC-VTTR may be derived from the formulation of VTTR by making the following two modifications:

- Expand each decision variable in the VTTR formulation to a set of K new decision variables, each corresponding to one of the traffic classes. Consider, for instance, variable b_{ij} in the original formulation, that is defined as the number of lightpaths from node i to node j. This variable is replaced by K variables $b_{ij,k}, k = 1, \ldots, K$, defined as the number of lightpaths from node i to node j that are used for grooming class-k traffic.
- Expand accordingly the set of constraints in the VTTR formulation to account for the class-specific decision variables.

We also note that, similar to the VTTR problem, MC-VTTR only takes as input the traffic demand matrices T_k, and hence, the number of nodes $|\mathcal{N}|$, but not the network graph G. Since the physical network topology is not taken into account, the running time to obtain the multi-class virtual topology can be orders of magnitude shorter than solving the original formulation that integrates the RWA subproblem. This virtual topology is then reconciled with the physical topology by running an appropriate RWA algorithm, e.g., the fast RWA techniques we have developed separately for ring [42] and mesh [27, 28] networks.

4.4 Multi-Class Grooming Algorithms

As a result of the multi-class grooming constrained, each lightpath created in the virtual topology carries traffic within a single class. Consequently, the virtual topology can be thought of as a collection of K disjoint sub-topologies, where the k-th sub-topology carries only traffic in class $k, k = 1, \ldots, K$. This arrangement naturally supports flow isolation: each traffic class is carried optically on its own sub-topology and hence, it does not interfere with traffic on other sub-topologies.

4.4.1 Separate Formulation

As the name implies, in the separate formulation we consider each traffic class in isolation, rather than considering all classes jointly as in the integrated formulation. Let us assume that the traffic classes are labeled $1, 2, \ldots, K$, in some arbitrary manner. This approach can be described by the following algorithmic steps:

1. Let $k \leftarrow 1$.

2. Solve the VTTR ILP formulation in Chapter 2 with the traffic matrix $T_k = [t_k^{sd}]$.

3. Let $k \leftarrow k + 1$; if $k \leq K$ then repeat from Step 2, otherwise terminate.

During the k-th iteration, this algorithm constructs the optimal virtual sub-topology for class-k traffic.

4.4.2 Baseline Formulation

In this approach, we simply solve the VTTR formulation in Chapter 2 with the traffic matrix $T = \sum_k T_k$. In other words, we assume that all traffic offered to the network belongs to the same class, hence there are no class-specific grooming constraints. This baseline formulation provides a lower bound on the number of lightpaths required to accommodate the traffic demands, as well as a reference value in terms of the computational effort (running time) necessary to obtain a solution. Hence, it can be used to characterize the additional resources involved in providing flow isolation guarantees.

4.5 Numerical Results

In this section, we investigate the relative performance in terms of solution quality and running time of the two formulations above by solving them on the IBM Ilog CPLEX 12 optimization tool on a cluster of identical compute nodes with dual Woodcrest Xeon CPU at 2.33GHz with 1333MHz memory bus, 4GB of memory and 4MB L2 cache.

Our study mainly concerns three factors that impact the performance of MC-VTTR: the traffic load t_{max}, the number of traffic classes K, and the skewness between classes, indicated by percentages P. The experiments are run on three network topologies. Specifically, the three networks are: the 14-node, 42-link NSFNet [30], shown in Fig. 3.4; the 17-node, 52-link German network [19], shown in Fig. 3.5; and the 32-node, 106-link network shown in Figure 3.3.

For each network topology, we consider three settings of the number of traffic classes K, where $K = 3, 4, 5$, two different sets of percentages associated with each K, and five levels of traffic loads, i.e., $t_{max} = 40, 60, 80, 100$. In running the baseline formulation, the traffic matrix $T = [t^{sd}]$ is generated by drawing the (integer) traffic demands uniformly at random in the interval $[0, t_{max}]$. For the separate formulation, the traffic matrices for different traffic classes are generated as follows:

1. Uniformly generate traffic components from $[0, t_{max}]$. The traffic matrix T obtained from this step is the same used in the corresponding baseline formulation.

2. Divide T into sub-traffic matrices according to percentages $P = (p_1, p_2, \ldots, p_K)$ by calculating $t^i_{sd} = t_{sd} \cdot p_i$

The percentages used for each number of traffic classes are listed below, where the latter setting indicates a more 'even' division.

1. $K = 3$: $P = (10\%, 30\%, 60\%), (30\%, 30\%, 40\%)$

2. $K = 4$: $P = (10\%, 10\%, 40\%, 40\%), (25\%, 25\%, 25\%, 25\%)$

3. $K = 5$: $P = (10\%, 10\%, 20\%, 30\%, 30\%), (20\%, 20\%, 20\%, 20\%, 20\%)$

Among all the experiments, each data point in the figures we present in this section represents the average of ten random problem instances for the stated values of the input parameters. We set the relative optimality gap to 2% for all CPLEX runs, wavelength capacity $C = 16$, and time limit to six hours (for the separate formulation, the time limit for each traffic class is set to $\frac{6hours}{K}$.) Consequently, CPLEX terminates either when it finds a solution that is within 2% of the optimal for the problem at hand, or when reaches the time limit; in the latter case, we simply record the best value obtained by CPLEX at that time.

4.5.1 Quality of Solution

The odd numbered Figures 4.1 - 4.17 plot the number of lightpaths against various traffic loads t_{max}. There are three lines in each of the plots, which refer to the two percentage settings of the separate formulation and the baseline formulation. As we expected, the baseline formulation always gives the lowest number of lightpaths. This is because the baseline formulation is a relaxed version of separate formulation without the additional constraints. For the two different percentage settings, there is no significant difference when t_{max} is relatively small, i.e., $t_{max} \leq 60$. When t_{max} is getting larger, it shows that the 'even' division has slightly smaller number of lightpaths. Comparing the three number of lightpaths plots for the same network topology, for example, Fig. 4.1, Fig. 4.3, and Fig. 4.5 for the 14-node network, we can see that more lightpaths are required to support more traffic classes, i.e., $K = 5$ requires more lightpaths than $K = 3$ for the same t_{max} value. All of the solution quality plots, i.e. the number of lightpaths plots, show consistent linearly increasing trend in the number of lightpaths along with traffic load increase. Furthermore, within the same figure, the slope for each line is similar, which means that for different t_{max} values, the absolute number of extra lightpaths required to support multiple traffic classes remains in the similar scale. In other words, the additional lightpaths represent a smaller portion of the number of lightpaths for the baseline formulation when t_{max} increases.

4.5.2 CPU Time

The even numbered Figures 4.2- 4.18 plot the CPU time against various traffic loads. The running time for the baseline formulation remains stable across various t_{max} values, while the running times for the separate formulation shows a drop when t_{max} increases. The baseline formulation has the shortest running time, followed by the "even division" separate formulation, which is about one order longer, while the separate formulation with the more skewed percentage is slower by one or two more orders of magnitude. For the 32-node network, the running times for the two separate formulation with different percentages are very close to each other, which are both approximately at the time limit. These observations can be explained by the 2% relative optimality gap we set. When t_{max} is large, the total number of lightpaths required to carry all the traffic demands is large. Since the relative optimality gap remains the same, the absolute optimality gap is larger for larger t_{max} values than smaller ones. Thus, when t_{max} is large, it is easier to reach the absolute optimality gap, so that the running time is reduced.

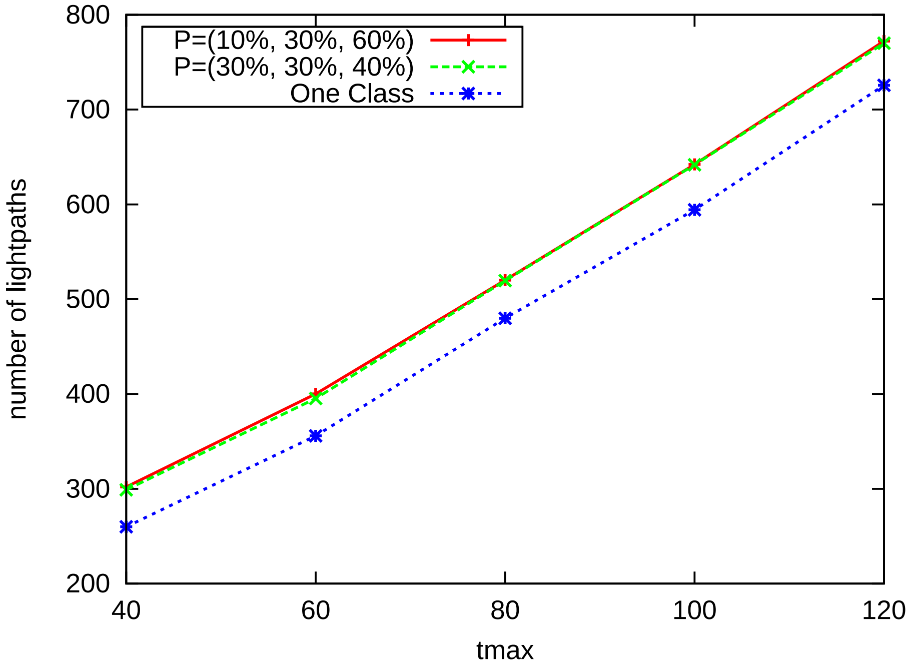

Figure 4.1: Number of lightpaths, 14-node network, $K = 3$ traffic classes

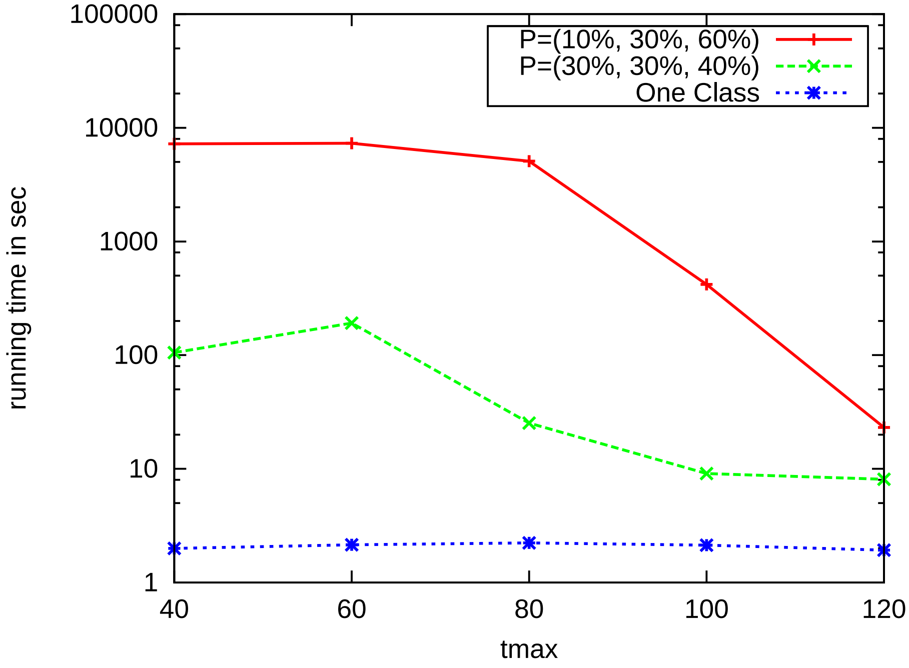

Figure 4.2: CPU time of 14-node network, $K = 3$ traffic classes

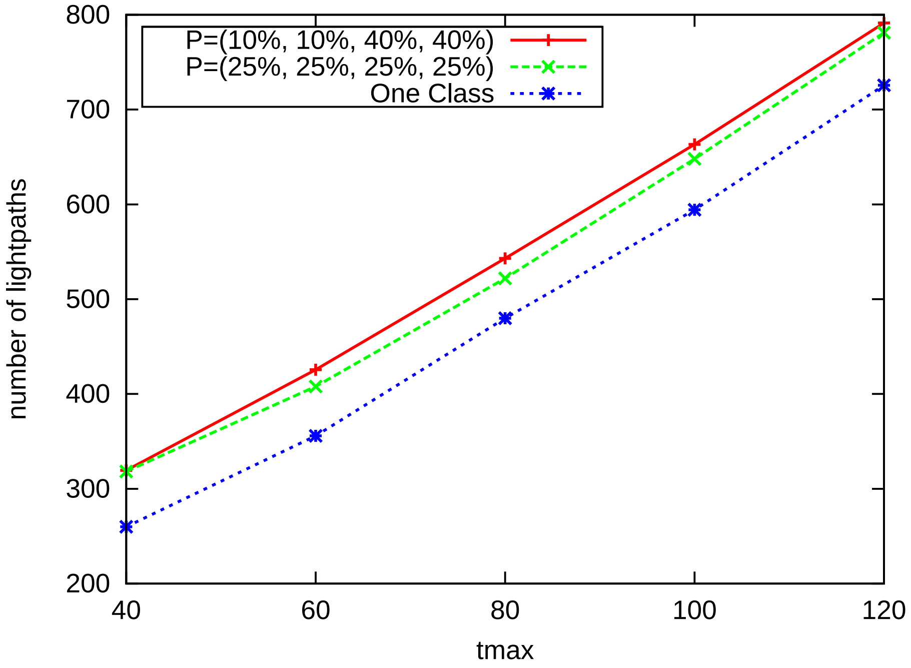

Figure 4.3: Number of lightpaths of 14-node network, $K = 4$ traffic classes

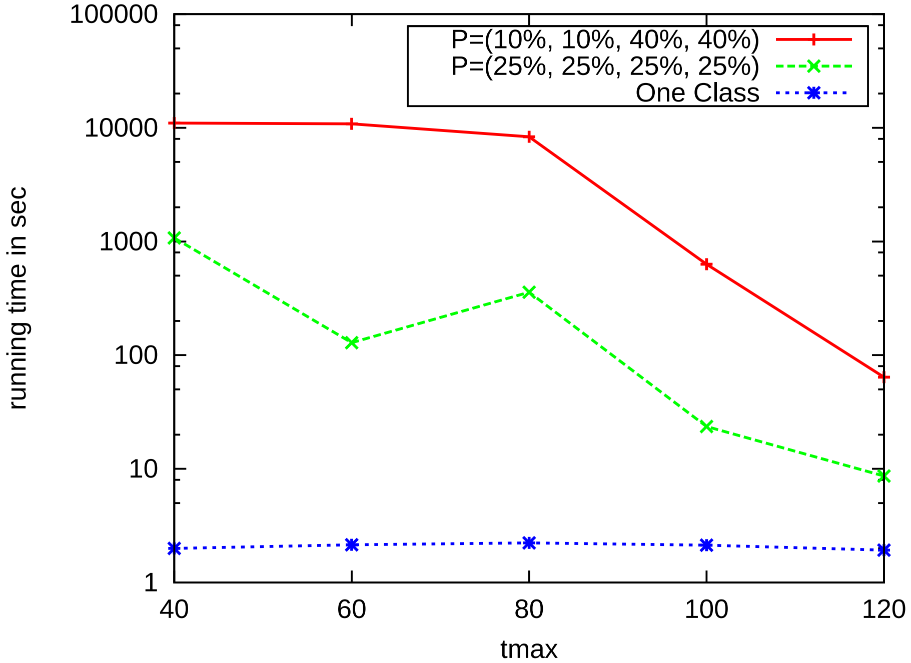

Figure 4.4: CPU time of 14-node network, $K = 4$ traffic classes

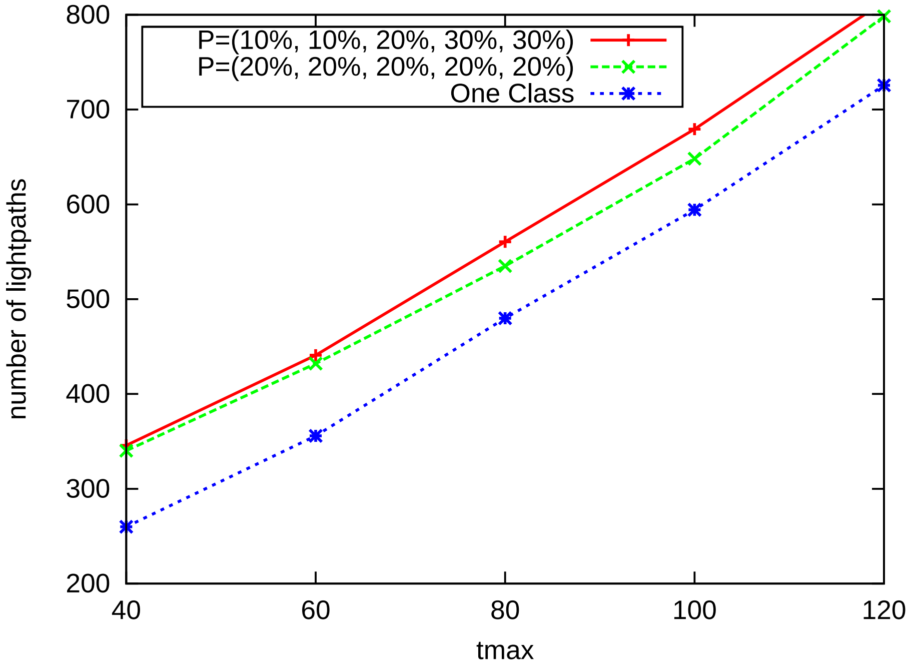

Figure 4.5: Number of lightpaths of 14-node network, $K = 5$ traffic classes

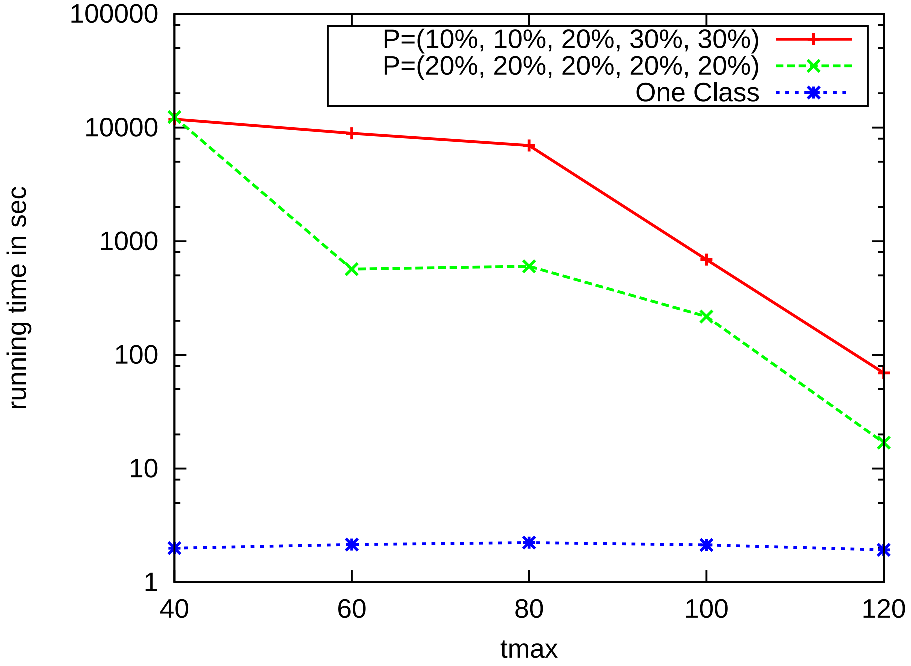

Figure 4.6: CPU time of 14-node network, $K = 5$ traffic classes

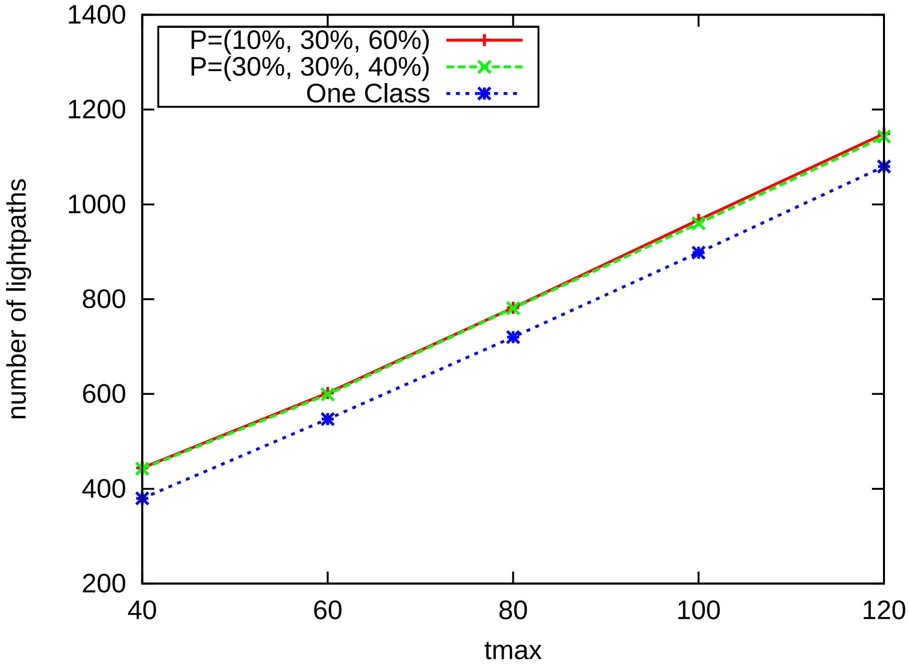

Figure 4.7: Number of lightpaths of 17-node network, $K = 3$ traffic classes

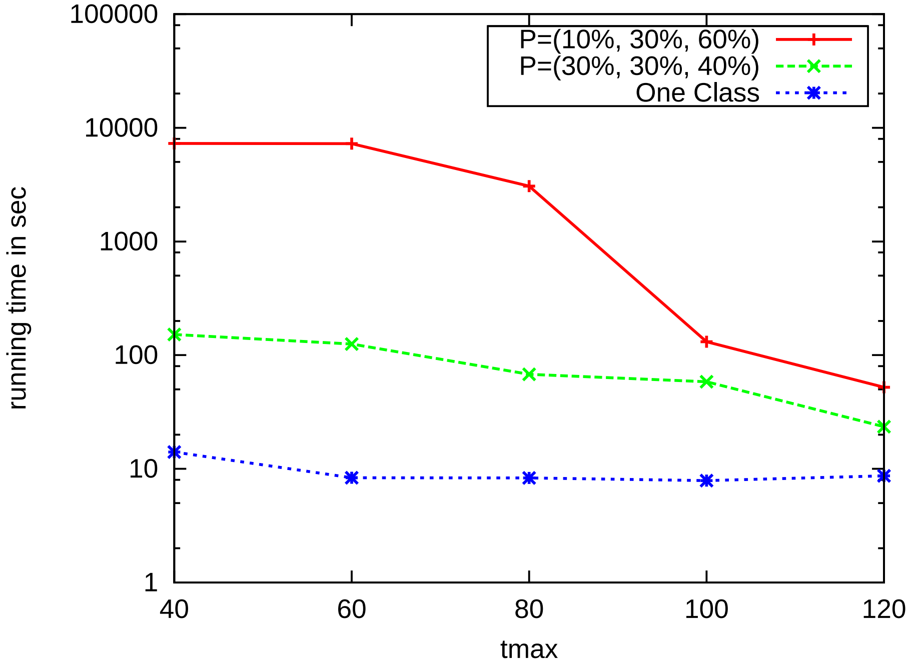

Figure 4.8: CPU time of 17-node network, $K = 3$ traffic classes

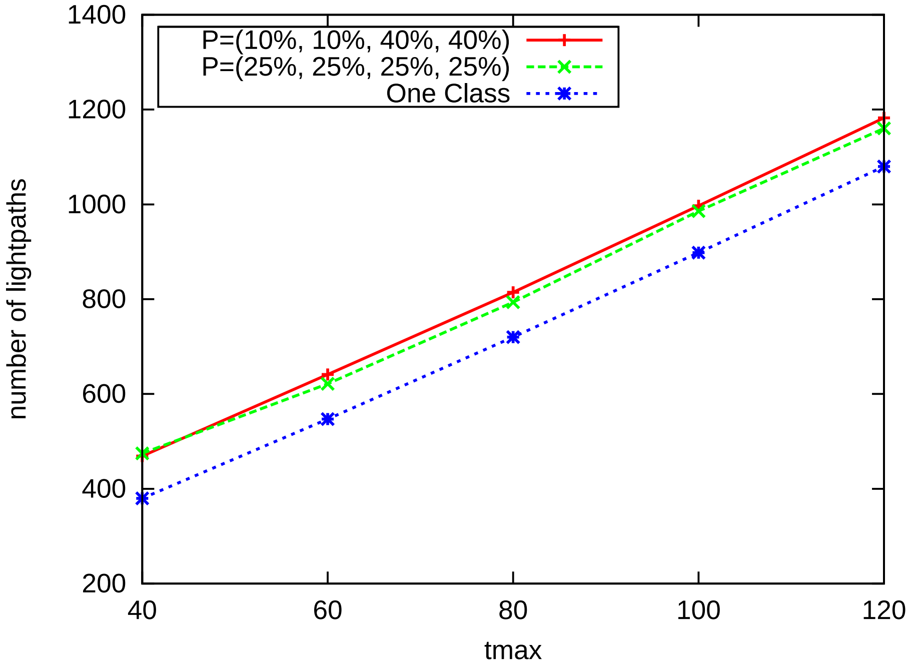

Figure 4.9: Number of lightpaths of 17-node network, $K = 4$ traffic classes

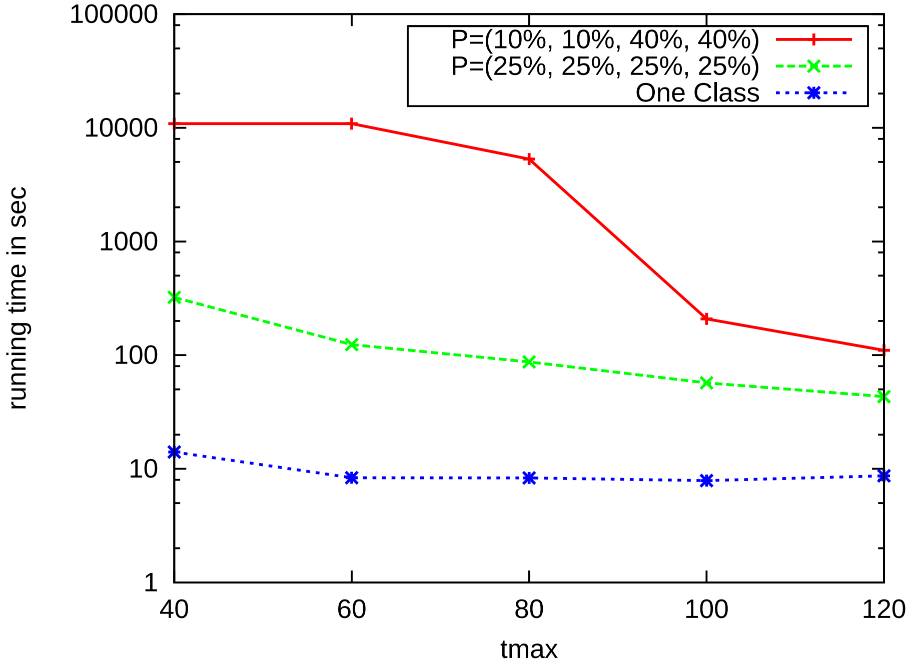

Figure 4.10: CPU time of 17-node network, $K = 4$ traffic classes

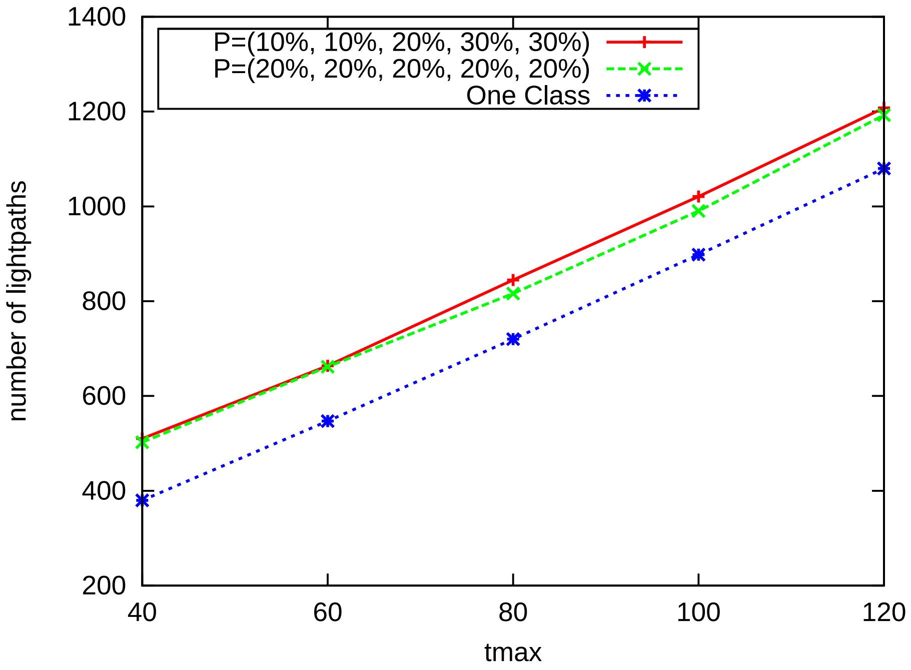

Figure 4.11: Number of lightpaths of 17-node network, $K = 5$ traffic classes

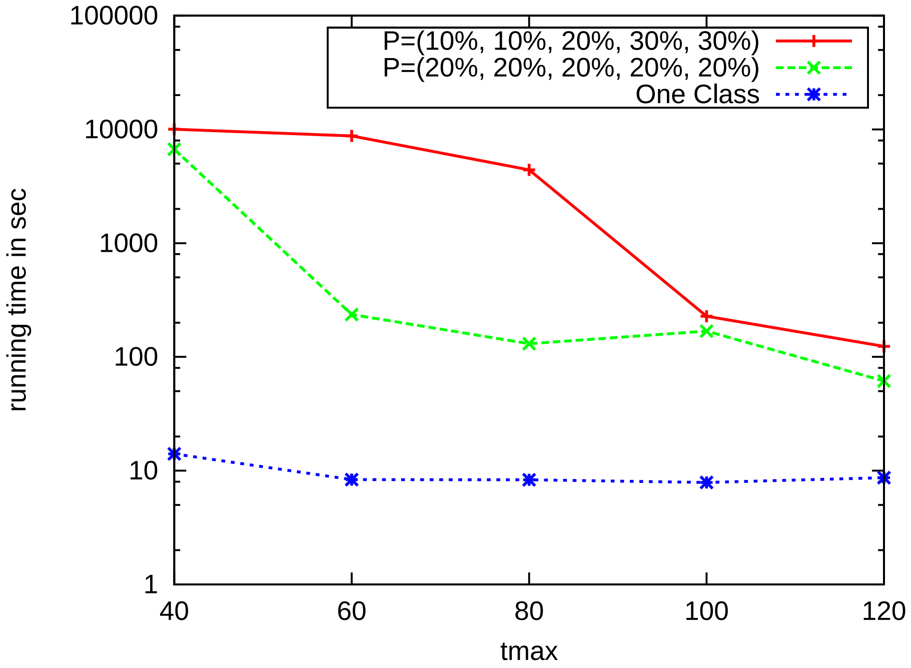

Figure 4.12: CPU time of 17-node network, $K = 5$ traffic classes

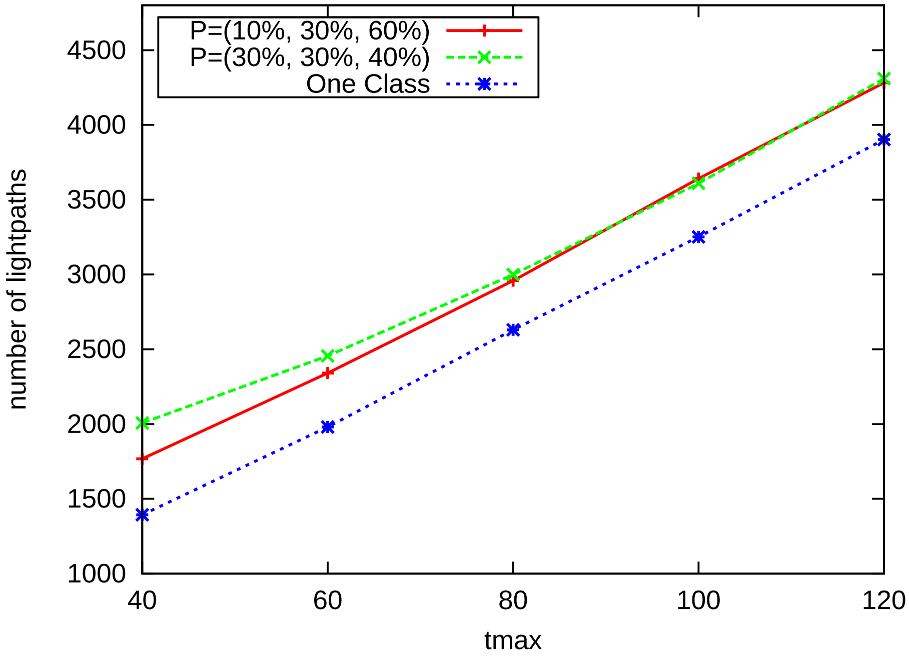

Figure 4.13: Number of lightpaths of 32-node network, $K = 3$ traffic classes

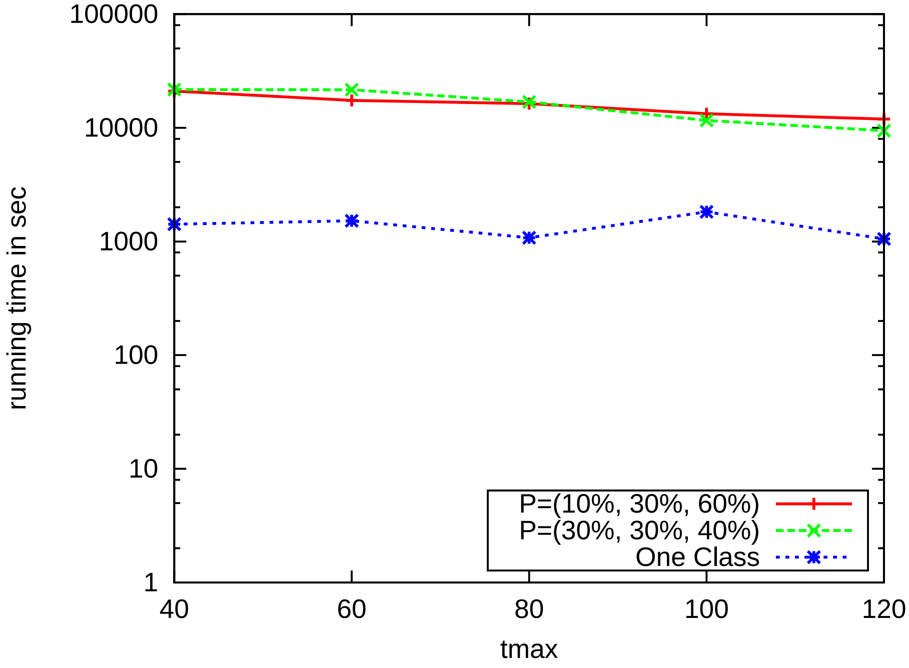

Figure 4.14: CPU time of 32-node network, $K = 3$ traffic classes

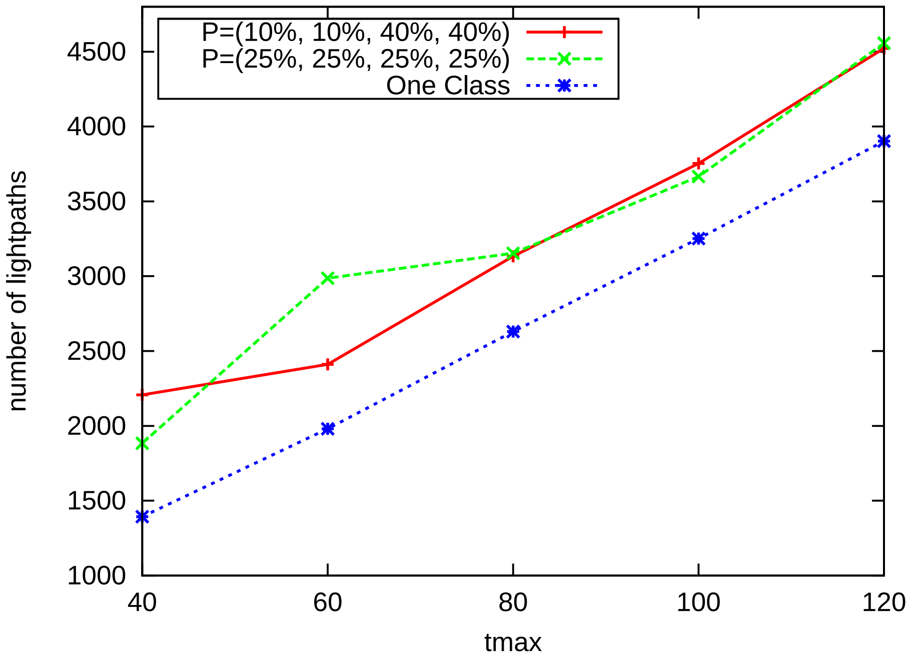

Figure 4.15: Number of lightpaths of 32-node network, $K = 4$ traffic classes

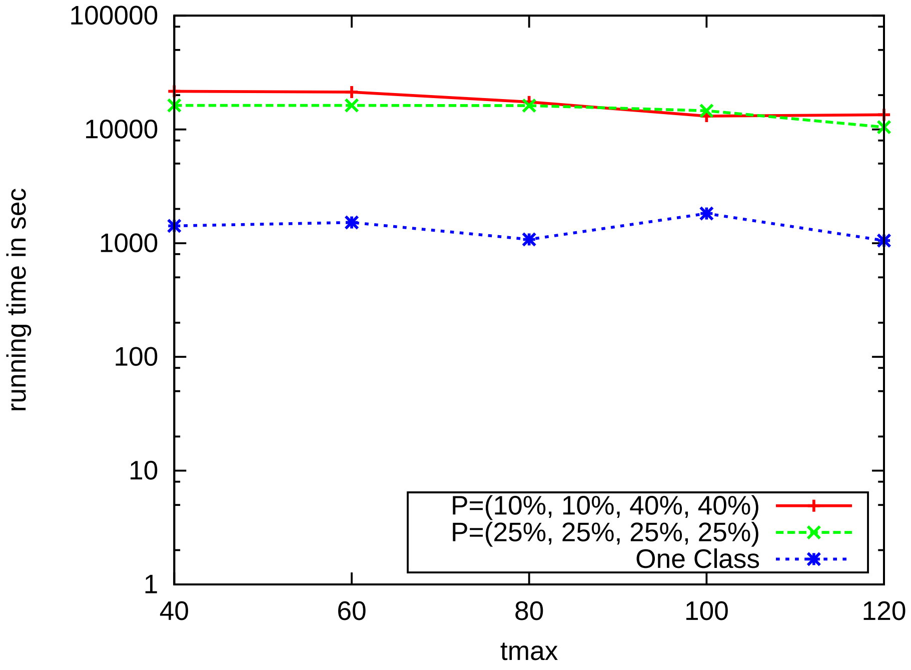

Figure 4.16: CPU time of 32-node network, $K = 4$ traffic classes

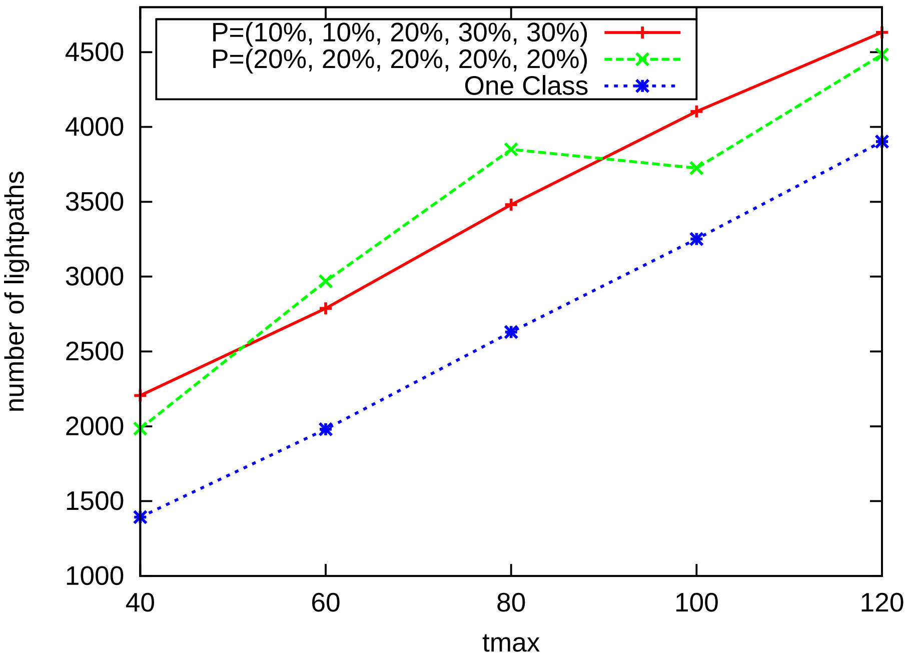

Figure 4.17: Number of lightpaths of 32-node network, $K = 5$ traffic classes

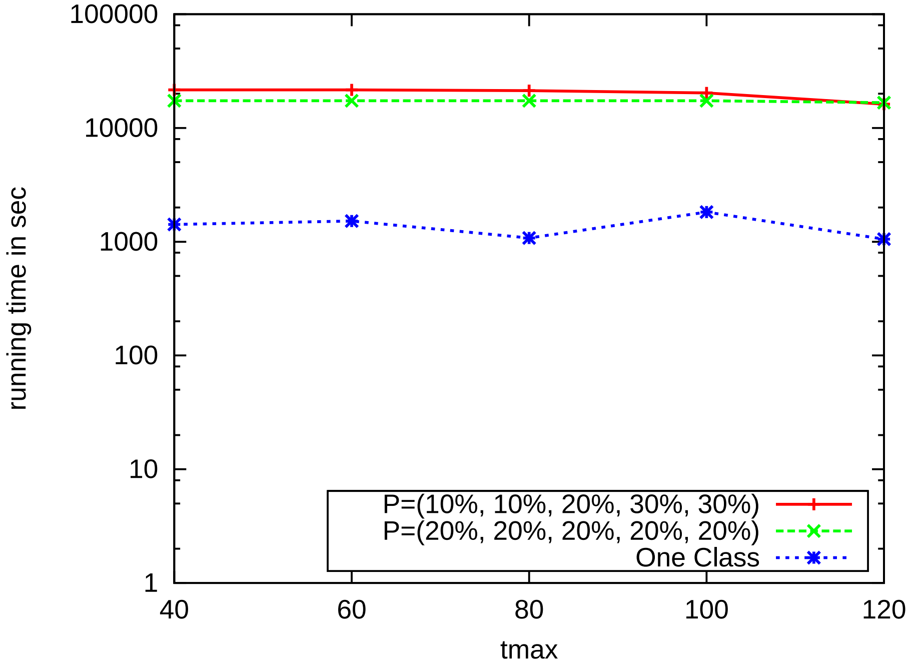

Figure 4.18: CPU time of 32-node network, $K = 5$ traffic classes

Chapter 5

Conclusions and Future Work

In this work, we reviewed the existing studies on traffic grooming in optical networks, and presented the basic integer linear program (ILP) formulation, which was the foundation and starting point of all the work in this thesis. We then identified two major issues with solving the ILP formulation directly: the scalability issue and wavelength fragmentation issue. We then make the contributions to address them.

We proposed a novel approach of decomposing the traffic grooming problem into two subproblems that are solved sequentially: the virtual topology and traffic routing (VTTR) subproblem, and the routing and wavelength assignment (RWA) problem. Unlike previous decomposition methods, our decomposition is exact, i.e., the solution from decomposition is optical to the original traffic grooming problem, as long as the network is not wavelength limited. The wavelength fragmentation issue is addressed by minimizing the number of wavelengths by solving the RWA subproblem. We also developed three iterative algorithm for VTTR problem based on partial LP relaxation, which address the issue of scalability by reaching desirable tradeoffs between solution quality and running time. We showed that this decomposition approach of the traffic grooming problem scales well to general network topologies, so that it enables network operators to carry out extensive ”what-if” analysis in network optimization.

The decomposition approach was further extended to hierarchical networks. The hierarchical traffic grooming framework showed that hierarchical traffic grooming is an efficient and scalable method to groom multigranular traffic in large-scale networks with a general topology. After defining the hierarchical virtual topology and traffic routing (H-VTTR) problem, we presented a suite of ILP formulation for H-VTTR problem with different grooming features. The formulations have been shown to perform well over a range of network topologies and traffic patterns, and scale to networks of realistic size.

We explored the overhead of supporting multiple traffic classes in VTTR problem, where traffic classes may be defined across various dimensions, including: (1) quality of Service (QoS) requirements, in which case the network may provision different lightpaths to support different levels of QoS; (2) the user or group of users, in which case the network operator may groom traffic of different organizations (e.g., competitors) onto different sets of lighgtpaths; and/or (3) privacy or security considerations. The way we used to maintain several traffic classes was by imposing the additional constraints that grooming onto a given lightpath is permitted if and only if the traffic to be groomed belongs to the same class. This new problem was defined as the multi-class virtual topology and traffic routing problem (MC-VTTR). The performance of MC-VTTR in terms of solution quality and running time was explored over a range of network topologies. The overhead was quantified to give an idea of the cost and benefit to support multiple traffic classes.

5.1 Future Work

The contribution of this thesis is mostly concentrated on a decomposition technique in traffic grooming problems under point-to-point communication mode. Some of future directions are listed below.

1. *Apply the decomposition approach to P2MP, MP2P, MP2MP Communication Modes.*

   The work of this thesis assumes that traffic demands originate at a single source node and terminate at a single destination node. However, the decomposition approach in Chapter 2, the hierarchical traffic grooming in Chapter 3, and the multi-class traffic grooming in Chapter 4 can all be adapted to other communication modes, including point-to-multipoint (P2MP), multipoint-to-point (MP2P), and multipoint-to-multipoint (MP2MP) modes. Since these communication modes are also commonly used in optical networks, it will be worth exploring the performance of the efficient decomposition technique in these contexts.

2. *Link Selection for Traffic Grooming (TG).*

   Instead of decomposing the original traffic grooming problem defined in Chapter 1 to tackle the scalability issue, we may consider to use link selection algorithms to reduce the size of the link-based ILP formulation by pruning redundant link decision variables, as proposed in [27]. Specifically, the link selection algorithm will only define link-related decision variables in the neighborhood of each traffic component source-destination pair (s, d). In this way, a great number of link-related decision variables would be ruled out, so that the size of the optimization problem will be highly reduced. Consequently, the running time may be reduced as well. One issue with link selection algorithm is that the solution will not be optimal to the original traffic grooming problem. But it will be interesting to explore the quality of new solution compared to the decomposition approach of Chapter 2.

3. *Waveband Grooming.*

   This thesis has been focused on sub-wavelength traffic grooming, i.e., aggregating low speed individual traffic components onto a lightpaths. However, the similar idea can be naturally extended to grooming wavelength into wavebands, i.e., waveband grooming. Waveband grooming considers to aggregate lightpaths onto a waveband, in order to minimize the number of optical cross-connects (OXCs) needed in designing an optical network.

REFERENCES

[1] P. Baran. On distributed communications networks. *IEEE Transactions on Communications*, 12(1):1–9, March 1964.

[2] S. Baroni and P. Bayvel. Wavelength requirements in arbitrary connected wavelength-routed optical networks. *IEEE/OSA J. Lightwave Technol*, 15(2):242–251, Feb. 1997.

[3] A.R.B. Billah, Bin Wang, and A. A S Awwal. Multicast traffic grooming in wdm optical mesh networks. In *Global Telecommunications Conference, 2003. GLOBECOM '03. IEEE*, volume 5, pages 2755–2760 vol.5, 2003.

[4] B. Chen, R. Dutta, and G. N. Rouskas. Traffic grooming in star networks. In *Proceedings of the First Traffic Grooming Workshop*, October 2004.

[5] B. Chen, R. Dutta, and G. N. Rouskas. On the application of k-center algorithms to hierarchical traffic grooming. In *Proceedings of the Second Traffic Grooming Workshop*, pages 295–301, October 2005.

[6] B. Chen, G. N. Rouskas, and R. Dutta. A framework for hierarchical grooming in WDM networks of general topology. In *Proceeding of BROADNETS 2005*, pages 167–176, October 2005.

[7] B. Chen, G. N. Rouskas, and R. Dutta. Clustering and hierarchical traffic grooming in large scale WDM networks. In *Proceedings of ONDM 2007*, May 2007.

[8] B. Chen, G. N. Rouskas, and R. Dutta. On hierarchical traffic grooming in WDM networks. *IEEE/ACM Transactions on Networking*, 16(5):1226–1238, October 2008.

[9] L.-W. Chen and E. Modiano. Dynamic routing and wavelength assignment with optical bypass using ring embeddings. *Optical Switching and Networking*, 1(1):35–49, 2005.

[10] A.L. Chiu and E.H. Modiano. Traffic grooming algorithms for reducing electronic multiplexing costs in WDM ring networks. *IEEE/OSA Journal of Lightwave Technology*, 18(1):2 –12, jan 2000.

[11] I. Chlamtac, A. Ganz, and G. Karmi. Lightpath communications: An approach to high bandwidth optical WANS. *IEEE Transactions on Communications*, 40(7):1171–1182, July 1992.

[12] Milind Dawande, Rakesh Gupta, Sanjeewa Naranpanawe, and Chelliah Sriskandarajah. A traffic-grooming algorithm in wavelength-routed optical networks. *INFORMS Journal on Computing*, 19(4):565 – 574, 2007.

[13] R. Dutta and G. N. Rouskas. On optimal traffic grooming in WDM rings. *IEEE Journal on Selected Areas in Communications*, 20(1):110–121, January 2002.

[14] R. Dutta and G.N. Rouskas. Traffic grooming in WDM networks: past and future. *IEEE Network*, 16(6):46 – 56, nov/dec 2002.

[15] Rudra Dutta, Shu Huang, and George N. Rouskas. Traffic grooming in path, star, and tree networks: Complexity bounds and algorithms, April 2006.

[16] F. Farahmand, Xiaodong Huang, and J.P. Iue. Efficient online traffic grooming algorithms in WDM mesh networks with drop-and-continue node architecture. In *Broadband Networks, 2004. BroadNets 2004. Proceedings. First International Conference on*, pages 180 – 189, oct. 2004.

[17] O. Gerstel, R. Ramaswami, and G.H. Sasaki. Cost-effective traffic grooming in wdm rings. *Networking, IEEE/ACM Transactions on*, 8(5):618–630, 2000.

[18] T. Gonzalez. Clustering to minimize the maximum inter-cluster distance. *Theoret. Comput. Sci.*, 38:293–306, 1985.

[19] W.D. Grover and D.P. Onguetou. A new approach to node-failure protection with span-protecting p-cycles. In *Transparent Optical Networks, 2009. ICTON '09. 11th International Conference on*, pages 1–5, 2009.

[20] D. Hochbaum and D.B. Shmoys. A best possible heuristic for the k-center problem. *Math. Oper. Res.*, 10:180–184, 1985.

[21] J.Q. Hu and B. Leida. Traffic grooming, routing, and wavelength assignment in optical WDM mesh networks. In *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, volume 1, pages 4 vol. (xxxv+2866), march 2004.

[22] Shu Huang and Rudra Dutta. Research problems in dynamic traffic grooming in optical networks. In *Proc. 1st Int. Workshop on Traffic Grooming, San Jose, CA*, April 2004.

[23] Z.K.G. Patrocínio Jr. and G. R. Mateus. A lagrangian-based heuristic for traffic grooming in WDM optical networks. In *IEEE GLOBECOM 2003*, volume 5,1-5, pages 2767–2771, Dec 2003.

[24] A.E. Kamal. Algorithms for multicast traffic grooming in wdm mesh networks. *Communications Magazine, IEEE*, 44(11):96–105, 2006.

[25] V. R. Konda and T. Y. Chow. Algorithm for traffic grooming in optical networks to minimize the number of transceivers. *IEEE Workshop on High Performance Switching and Routing*, pages 218 – 221, 2001.

[26] J.-F.P. Labourdette and A.S. Acampora. Logically rearrangeable multihop lightwave networks. *IEEE Transactions on Communications*, 39(8):1223 –1230, aug 1991.

[27] Z. Liu and G. N. Rouskas. Link selection algorithms for link-based ilps and applications to rwa in mesh networks. In *Optical Network Design and Modeling (ONDM 2013), 17th Conference on*, 2013.

[28] Zeyu Liu and George N. Rouskas. A fast path-based ILP formulation for offline RWA in mesh optical networks. In *IEEE GLOBECOM 2013*, 2012.

[29] Xavier Muñoz and Ignasi Sau. Graph-theoretic concepts in computer science. chapter Traffic Grooming in Unidirectional WDM Rings with Bounded Degree Request Graph, pages 300–311. Springer-Verlag, Berlin, Heidelberg, 2008.

[30] B. Mukherjee, D. Banerjee, S. Ramamurthy, and B. Mukherjee. Some principles for designing a wide-area wdm optical network. *Networking, IEEE/ACM Transactions on*, 4(5):684–696, 1996.

[31] Ahmad E. Kamal Rudra Dutta and George N. (Eds.) Rouskas. *Traffic grooming for optical networks*. Springer, first edition, 2008.

[32] L.H. Sahasrabuddhe and B. Mukherjee. Light trees: optical multicasting for improved performance in wavelength routed networks. *Communications Magazine, IEEE*, 37(2):67–73, 1999.

[33] M.A. Saleh and A.E. Kamal. Many-to-many traffic grooming in wdm networks. *Optical Communications and Networking, IEEE/OSA Journal of*, 1(5):376–391, 2009.

[34] J. Simmons, E. Goldstein, and A. Saleh. Quantifying the benefit of wavelength add-drop in WDM rings with distance-independent and dependent traffic. *IEEE/OSA Journal of Lightwave Technology*, 17:48–57, Jan 1999.

[35] H. Siregar, H. Takagi, and Y. Zhang. Efficient routing and wavelength assignment in wavelength-routed optical networks. *Proc. 7th Asia-Pacific Network Oper. and Mgmt Symposium*, pages 116–127, Oct. 2003.

[36] R. Ul-Mustafa and A.E. Kamal. Many-to-one traffic grooming with aggregation in wdm networks. *Selected Areas in Communications, IEEE Journal on*, 24(8):68–81, 2006.

[37] B. Vignac, B. Jaumard, and F. Vanderbeck. A hierarchical optimization approach to optical network design where traffic grooming and routing is solved by column generation. In *INOC*, 2009.

[38] Peng-Jun Wan, G. Calinescu, and O. Frieder. Grooming of arbitrary traffic in sonet/wdm blsrs. *Selected Areas in Communications, IEEE Journal on*, 18(10):1995–2003, 2000.

[39] Hui Wang, Zeyu Liu, and George N. Rouskas. Scalable traffic grooming in optical networks. In *Asia Communications and Photonics Conference*, page AS3D.2. Optical Society of America, 2012.

[40] Hui Wang and G.N. Rouskas. Flow isolation in optical networks. In *Local Metropolitan Area Networks (LANMAN), 2011 18th IEEE Workshop on*, pages 1–6, 2011.

[41] W. Yao, G. Sahin, M. Li, and Byrav Ramamurthy. Analysis of multi-hop traffic grooming in WDM mesh networks. In *Broadband Networks, 2005. BroadNets 2005. 2nd International Conference on*, pages 165 – 174 Vol. 1, oct. 2005.

[42] E. Yetginer, Z. Liu, and G.N. Rouskas. Fast exact ILP decompositions for ring RWA. *IEEE/OSA Journal of Optical Communications and Networking*, 3(7):577 –586, july 2011.

[43] Emre Yetginer and George N Rouskas. Power efficient traffic grooming in optical WDM networks. *GLOBECOM 2009 2009 IEEE Global Telecommunications Conference*, 5:1–6, 2009.

[44] Hongyue Zhu, Hui Zang, Keyao Zhu, and B. Mukherjee. Dynamic traffic grooming in WDM mesh networks using a novel graph model. In *Global Telecommunications Conference, 2002. GLOBECOM '02. IEEE*, volume 3, pages 2681 – 2685 vol.3, nov. 2002.

[45] K. Zhu and B. Mukherjee. A review of traffic grooming in WDM optical networks: Architectures and challenges. *Optical Networks Magazine*, 4(2):55–64, March/April 2003.

[46] Keyao Zhu and B. Mukherjee. Traffic grooming in an optical wdm mesh network. *Selected Areas in Communications, IEEE Journal on*, 20(1):122–133, 2002.

APPENDIX

Appendix A

Full Formulation for Multi-class Traffic Grooming

Notation

L_k^+, L_k^- the set of links that go out node k and the set of links that come in node k.

Parameters

$\mathcal{Z}$ the set of node pairs. $\mathcal{Z} = \{(n_1, n_2) : n_1, n_2 \in \mathcal{N}, n_1 \neq n_2\}$.

i, j endpoints of a lightpath.

s, d source node and destination node of a demand.

l link l in the network.

R number of security levels in the problem.

r the rth security level.

W number of wavelength channels in each link.

w the wth wavelength channel in a link.

C the capacity of a wavelength channel.

t_r^{sd} the demand from node s to node d with security level r.

P_0 the energy consumption of constructing a lightpath.

$P_{\max}$ the energy consumption of a fully used lightpath including constructing consumption.

p the energy consumption rate. We have $P_{\max} = P_0 + C \cdot p$.

Decision Variables

$b_{ij,r}$ number of lightpaths starting at node i ending at node j with security level r. This variable is an nonnegative integer.

$b^{l}_{ij,r}$ number of lightpaths passing through link l, which start at node i and end at node j with security level r. This variable is an nonnegative integer.

$c^{w,l}_{ij,r}$ binary variable. The value is 1 if some lightpath starting at node i and ending at node j with security level r passes link l and wavelength channel w is assigned to it. Otherwise, the value is 0.

$t^{sd}_{ij,r}$ The amount of traffic from source s to destination d with security level r that is carried on lightpaths from node i to node j.

Objective Function

$$\min \; P_0 \sum_{(i,j)\in\mathcal{Z}} \sum_{r=1}^{R} b_{ij,r} + p \sum_{(s,d)\in\mathcal{Z}} \sum_{(i,j)\in\mathcal{Z}} \sum_{r=1}^{R} c^{sd}_{ij,r} \tag{A.1}$$

Constraints

— Lightpath Routing Constraints

$$\sum_{l\in L_k^+} b^{l}_{ij,r} - \sum_{l\in L_k^-} b^{l}_{ij,r} = 0$$

$$\forall (i,j)\in\mathcal{Z}, \forall k\in\mathcal{N}\backslash\{i,j\}, \forall r = 1,\ldots,R. \tag{A.2}$$

$$\sum_{l\in L_i^+} b^{l}_{ij,r} = b_{ij,r}$$

$$\forall (i,j)\in\mathcal{Z}, \forall r = 1,\ldots,R. \tag{A.3}$$

$$\sum_{l\in L_i^-} b^{l}_{ij,r} = 0$$

$$\forall (i,j)\in\mathcal{Z}, \forall r = 1,\ldots,R. \tag{A.4}$$

$$\sum_{l\in L_j^+} b^{l}_{ij,r} = 0$$

$$\forall (i,j)\in\mathcal{Z}, \forall r = 1,\ldots,R. \tag{A.5}$$

$$\sum_{l\in L_j^-} b^{l}_{ij,r} = b_{ij,r}$$

$$\forall (i,j)\in\mathcal{Z}, \forall r = 1,\ldots,R. \tag{A.6}$$

— Lightpath Wavelength Assignment Constraints

$$\sum_{w=1}^{W} c_{ij,r}^{w,l} = b_{ij,r}^{l} \qquad \forall (i,j) \in \mathcal{Z}, \forall r = 1, \ldots, R, \forall l \in \mathcal{L} \tag{A.7}$$

$$\sum_{r=1}^{R} \sum_{(i,j)\in\mathcal{Z}} c_{ij,r}^{w,l} \leq 1 \qquad \forall l \in \mathcal{L}, \forall w = 1, \ldots, W. \tag{A.8}$$

$$\sum_{l\in L_k^+} c_{ij,r}^{w,l} - \sum_{l\in L_k^-} c_{ij,r}^{w,l} = 0 \qquad \forall (i,j) \in \mathcal{Z}, \forall k \in \mathcal{N}\backslash\{i,j\}, \forall r = 1, \ldots, R, \forall w = 1, \ldots, W. \tag{A.9}$$

— Traffic Routing Constraints

$$\sum_{(s,d)\in\mathcal{Z}} t_{ij,r}^{sd} \leq b_{ij,r} \cdot C \qquad \forall (i,j) \in \mathcal{Z}, \forall r = 1, \ldots, R. \tag{A.10}$$

$$\sum_{(s,d)\in\mathcal{Z}} t_{ij,r}^{sd} \geq (b_{ij,r} - 1) \cdot C \qquad \forall (i,j) \in \mathcal{Z}, \forall r = 1, \ldots, R. \tag{A.11}$$

$$\sum_{j\in\mathcal{N}\backslash\{i\}} t_{ij,r}^{sd} - \sum_{j\in\mathcal{N}\backslash\{i\}} t_{ji,r}^{sd} = 0 \qquad \forall (s,d) \in \mathcal{Z}, \forall i \in \mathcal{N}\backslash\{s,d\}, \forall r = 1, \ldots, R. \tag{A.12}$$

$$\sum_{j\in\mathcal{N}\backslash\{s\}} t_{sj,r}^{sd} = t_r^{sd} \qquad \forall (s,d) \in \mathcal{Z}, \forall r = 1, \ldots, R. \tag{A.13}$$

$$\sum_{j\in\mathcal{N}\backslash\{s\}} t_{js,r}^{sd} = 0 \qquad \forall (s,d) \in \mathcal{Z}, \forall r = 1, \ldots, R. \tag{A.14}$$

$$\sum_{j\in\mathcal{N}\backslash\{d\}} t_{jd,r}^{sd} = t_r^{sd} \qquad \forall (s,d) \in \mathcal{Z}, \forall r = 1, \ldots, R. \tag{A.15}$$

$$\sum_{j\in\mathcal{N}\backslash\{d\}} t_{dj,r}^{sd} = 0 \qquad \forall (s,d) \in \mathcal{Z}, \forall r = 1, \ldots, R. \tag{A.16}$$