

2001

Regular graph-based logical topology design in multi-hop optical networks.

Anwar Ariful. Haque
University of Windsor

Follow this and additional works at: <http://scholar.uwindsor.ca/etd>

Recommended Citation

Haque, Anwar Ariful., "Regular graph-based logical topology design in multi-hop optical networks." (2001). *Electronic Theses and Dissertations*. Paper 3297.

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

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

UMI[®]

NOTE TO USERS

Page(s) not included in the original manuscript and are unavailable from the author or university. The manuscript was microfilmed as received.

15 and 31

This reproduction is the best copy available.

UMI

Regular Graph based Logical Topology Design in Multi-hop Optical Networks

by

Anwar Haque

A Thesis

Submitted to the Faculty of Graduate Studies and Research
through the School of Computer Science in Partial
Fulfillment of the Requirement for the Degree of
Master of Science at the University of Windsor

Windsor, Ontario, Canada
2001



National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-62220-7

Canada

Anwar Haque 2001

© All Rights Reserved

Abstract

Existing approaches for logical topology design and routing for multi-hop optical networks become intractable for large networks. One approach which has been used is to treat the logical topology design problem separately from the routing problem which can be solved as a LP problem. The straightforward formulation of the LP problem has been reported but this is also feasible only for relatively smaller networks since the basis size for the simplex method is $O(n^2)$ where n is the number of nodes in the network. In this paper, by exploiting the special structure of the routing problem, we present an efficient column generation scheme embedded into the revised simplex method. This approach makes it feasible to handle networks with relatively large number of nodes. To study the approach experimentally we have used a number of traffic based heuristics for generating the logical topologies. These include a variation of the well known HLDA heuristic and two simple traffic based heuristics to generate logical topologies based on regular graphs. Many researchers feel that regular graphs are not well suited for wide area optical networks. The interesting result is that logical topologies based on regular graphs perform quite well compared to others. This suggests that it is useful to consider regular graphs as possible topologies for wide area networks and should be included as potential candidates for large wide area networks.

to my Parents...

Acknowledgements

I take this opportunity to thank my advisors Dr. Arunita Jaekel and Dr. Yash Aneja for their continuous support and encouragement throughout my graduate studies. I am grateful to Dr. Subir Bandyopadhyay who guided me to the completion of this thesis. This work would not have been achieved without their support and advice. I am also grateful to Dr. Richard Caron and Dr. F. Kabanza for their appreciated and valuable suggestions. Finally, I want to express my gratitude to my family and friends for their support and love over these years.

Contents

Abstract	iii
Acknowledgement	v
List of Tables	ix
List of Diagrams	x
Chapter 1 Introduction	1
1.1 Motivation	1
1.2 Thesis Organization	2
Chapter 2 Review of Optical Network Concepts	3
2.1 Components of Optical Network	4
2.2 Wavelength Division Multiplexing	5
2.3 Lightpath and Wavelength Conversion	6
2.4 Physical and Logical Topology	8
2.4.1 Physical Topology	8
2.4.2 Logical topology	8
2.5 Singlehop Network	9
2.6 Multi-Hop Network	10
2.7 Wavelength Routed Optical Network (WRON)	11
2.8 Regular Multihop Topologies	12
2.8.1 GEMNET Architecture	13
2.9 Benefit of using Regular Topologies for congestion	14

Chapter 3 Problem Specification and Previous Approaches	15
3.1 The Problem	15
3.2 General Parameters and variables for Problem Formulation	15
3.3 Previous Approaches	17
3.3.1 MILP Formulation of the Problem	19
3.3.1.1 Limitations of MILP Formulation	22
3.3.2 Heuristics and LP Formulation	24
3.3.2.1 Heuristics	24
3.3.2.2 LP Formulation for Routing	28
3.3.2.3 Limitations of Heuristics and LP formulation	29
3.3.3 Meta Heuristic Based Approach	31
3.3.3.1 Limitations of Meta Heuristic based Approach	31
Chapter 4 Our Approach	32
4.1 Logical Topology Design subproblem	33
4.1.1 Heuristics for Topology Design	34
4.2 Routing Subproblem	46
4.2.1 LP formulation of the Problem	47
Chapter 5 Experiments	56
5.1 Methodology	56
5.2 Test Data	58
5.3 Results	60

Chapter 6 Conclusions and Future Work	62
6.1 Conclusion	62
6.2 Future Work	63
References	64
Appendix 1	69
Vita Auctoris	76

List of Diagrams

The General Structure of a Laser	4
Wavelength Division Multiplexing (WDM)	5
Wavelength Routed network with and without wavelength Converter	7
An example of a 4 node Physical Topology	9
Corresponding Logical Topology for physical topology	9
Wavelength Routed Optical Network (WRON)	12
Example of GEMNET architecture	14
Propagation delay on logical link	16
Flow Conservation	21
Minimizing Congestion	29
A 4-node network where 3 is the number of wavelengths per link	35
Logical Topology for heuristic 2	46
Bicriteria Algorithm	54
20 node European Optical Network (EONNET)	59
14 node NFSNET	60

List of Tables

Number of variables and constraints in MILP formulation	23
Performance Comparison between different heuristics	27
Number of Variables and Constraints in LP formulation	31
Small, Medium and Large network	56
t-test values for our experimental data	58

Chapter 1

Introduction

The 21st century has already been labeled by some as 'the century of light'. This label is very appropriate considering the current growth rate of optical networks. The rapid growth and advancement of optical fiber technology has opened the door for WDM (Wavelength Division Multiplexing) based optical networks. This will allow us to enjoy a very high speed data transmission with a huge bandwidth. The demands of a variety of multimedia services ranging from digitized voice, video, high-resolution graphics to distributed database requires secure and fast transmission of high volume of traffic over a large area. Optical fiber is the premiere medium to satisfy these needs, because of its huge bandwidth, low signal distortion, low signal attenuation and low power requirement. All optical wavelength division multiplexed networks using wavelength routing are considered to be the potential candidates for the next generation of wide area optical networks [24].

1.1 Motivation

The problem of determining a good logical topology for a given network and finding the corresponding routes between any s-d pair has been the focus of considerable research. This is an important problem because it determines the traffic load on the links and of the network and thus helps determining the maximum traffic a network can handle. Most previous approaches have used optimization techniques to solve this problem. This usually results in a logical topology which is an arbitrary graph, constructed in such a way as to

minimize the congestion of the network. The main drawback of this approach is that as the number of nodes in the network grows, the problem quickly becomes intractable. The result is that even for moderate sized networks the approach is infeasible. In this thesis we have proposed a new approach for logical topology design and flow assignment. We separate the problem into two parts. First we design the logical topology using a greedy heuristic. However, instead of an arbitrary graph, we ensure that our logical topology is a regular graph with attractive properties such as low diameter and rich interconnections. We expect to get a lower value of congestion by exploiting these properties. We have used the GEMNET shuffle exchange network as our target architecture in this thesis. Once the logical topology has been designed, we formulate the associated optimal routing for the given topology by developing an efficient implementation of the simplex method for the corresponding linear program. This is done by exploiting the special structure of this linear program, allowing us to solve the routing problem for much larger networks. We present an efficient column generation scheme embedded into the revised simplex method. This approach makes it feasible to handle networks with relatively large number of nodes.

1.2 Thesis Organization

In chapter 2, we provide a literature review of optical networks to give an overview of the concepts used in wavelength routed multi-hop optical network. In chapter 3, we outline the existing problem specification and briefly discuss the existing approaches to logical topology design and their limitations. In Chapter 4, we discuss our approach in detail. In

Chapter 2

Review of Optical Network Concepts

Much of the developments in the telecommunication and internet industry is made possible by huge bandwidth of fiber optics technology. Demand for bandwidth is increasing day by day. Lightwave technology has influenced our expectation for meeting the high bandwidth demands in telecommunication networks and modern internet world. Optical fiber can operate at a much higher rate than any other communication medium at present. Many attractive features like low cross talk, immunity to electromagnetic interference, short delay, high security have made it a potential candidate for the backbone of the next generation high-speed network

The logical topology design problem based on Wavelength Division Multiplexing (WDM) technology in optical networks has been studied extensively in last decade and there is ongoing research in this area. In this chapter, we give a brief introduction to Optical Networks and its basic components. We also discuss single hop and multi hop networks, WDM, Wavelength Routed Optical Network (WRON), logical and physical topologies. One of the main objectives of this thesis is to exploit the structural properties of regular graphs as logical topologies. We have chosen GEMNET as the target regular topology and we give a brief review of this architecture in section 2.8.1.

2.1 Components of Optical Network

Following is a brief description of optical devices used for optical transmissions [31] :

LASER : LASER is an acronym for *Light Amplification by Stimulated Emission of Radiation*. It produces intense high-powered beams of coherent light (light which contains one or more distinct frequencies). [31]

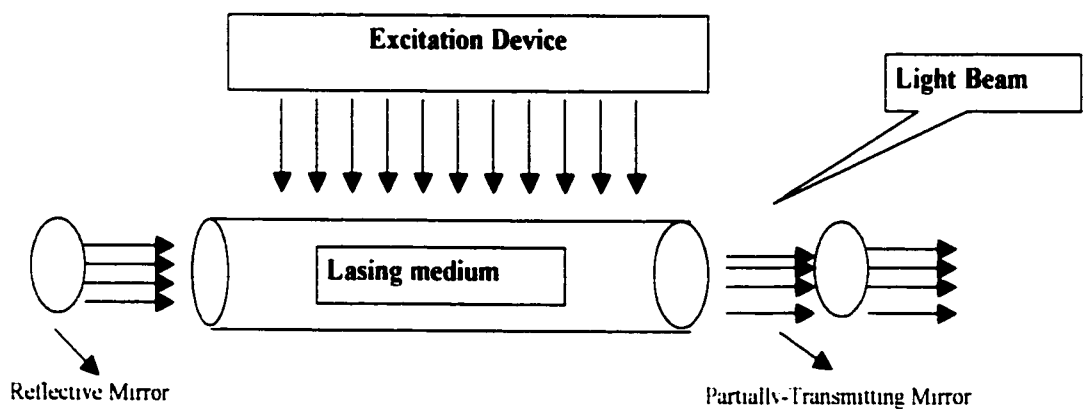


Figure 2.1: The General Structure of a Laser

Optical Amplifier : This device boosts the amplitude of the optical signal without converting optical signal to electrical domain.

Router : A router determines the next node to which a packet should be forwarded in order to reach its destination.

Multiplexer : A wavelength division multiplexer combines individual light signals coming from various ports into a single outgoing port.

Demultiplexer : wavelength division demultiplexer receives a composite light signal from an incoming port and routes it to several outgoing ports. Multiplexers and Demultiplexers are mainly used in routers [26, 27].

2.2 Wavelength Division Multiplexing

A WDM system uses a multiplexer at the source to multiplex several wavelengths on to a single fiber and demultiplexes the composite signal at the receiving end with the help of a demultiplexer [10]. When the demand on a link exceeds its capacity, WDM turns out to be a more cost effective solution compared to laying new fibers [11]. Due to reuse of each channel, the relative positions of each nodes (each user) can be changed dynamically based on the traffic fluctuations [12]. In figure 2.2 wavelengths λ_1 , λ_2 and λ_3 are combined at multiplexer and carried over on a single fiber where optical amplifier has been used to maintain the purity of the signal and at the demultiplexer end those distinct wavelengths are recovered.

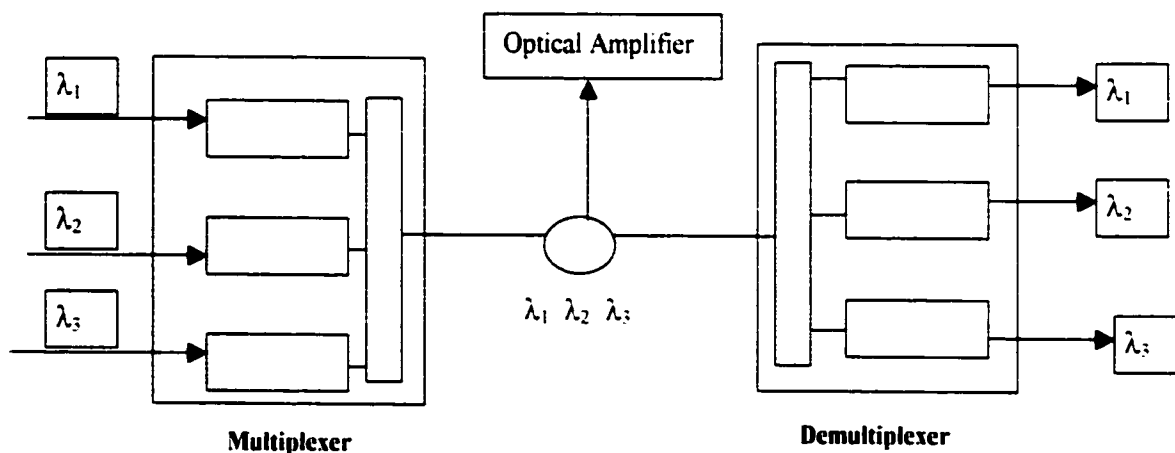


Figure 2.2: Wavelength Division Multiplexing (WDM)

Wavelength can be termed as the distance between points on corresponding phases of two consecutive cycles of a wave. The wavelength corresponding to a signal is related to its velocity, v , and frequency f , by $\lambda = v/f$.

2.3 Lightpath and Wavelength Conversion

A lightpath consists of a path through the network between end nodes and a wavelength on that path [24]. A light path can be viewed as a pipeline between two communicating end nodes with a particular wavelength associated with it. A lightpath provides a single hop communication between two nodes. Two light paths that share a link must necessarily use different wavelengths [24].

To establish a lightpath, it is required that the same wavelength be allocated to all the links in the path. This requirement is known as the *Wavelength-continuity constraint*. Converting data arriving on one wavelength along a link into another wavelength at an intermediate node and forwarding it along the next link is referred to as *wavelength conversion*. The device which is used for wavelength conversion is known as wavelength-converter.

Consider the example in Figure 2.3(a). Two lightpaths have been established in the network: (i) between Node 1 and Node 2 on wavelength λ_1 , and (ii) between Node 2 and Node 3 on wavelength λ_2 . Now suppose a lightpath between Node 1 and Node 3 needs to be set up. Establishing such a lightpath is impossible (note that from node 1 to node 2 λ_2 is available and from node 2 to node 3 λ_1 is available) even though there is a free wavelength on each of the links along the path from Node 1 to Node 3. This is because the available wavelengths on the two links are different. In the figure 2.3(b), a wavelength converter at Node 2 is employed to convert wavelength from λ_2 to λ_1 . The new lightpath between Node 1 and Node 3 can now be established by converting the wavelength λ_2 into

λ_1 using wavelength converter at node 2 and use the λ_1 wavelength from node 2 to node 3 to complete the lightpath. Notice that a single lightpath in such a *wavelength-convertible* network can use a different wavelength along each of the links in its path. Thus, a network without wavelength converter may suffer from higher blocking as compared to network which allows wavelength conversion. Most optical networks do not allow wavelength conversion (in the optical domain) due to the high cost of optical wavelength converters. We also assume that there is no (optical) wavelength conversion when we design our networks.

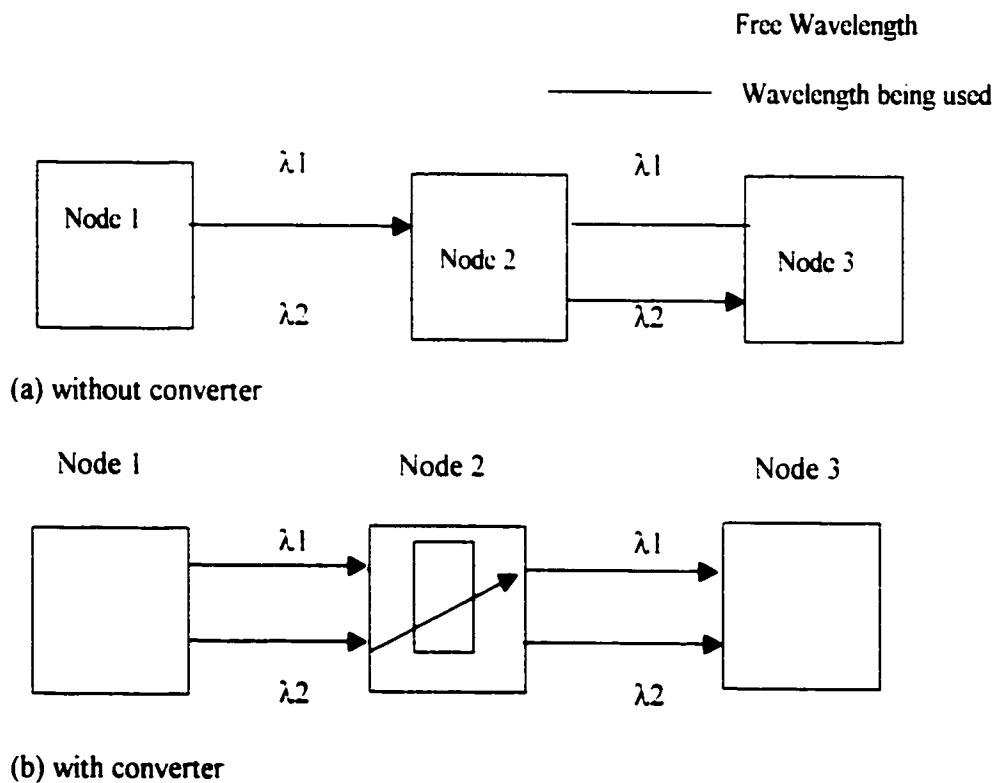


Figure 2.3: Wavelength routed network with and without wavelength converter

2.4 Physical and Logical Topology

2.4.1 Physical Topology

The physical topology of a network is the physical set of routing/end nodes and the fiber optic links connecting them upon which one sets up light paths between end nodes [27]. Light paths can be set up between end nodes on a physical topology. An example of physical topology is shown in figure 2.4 .

2.4.2 Logical topology

A logical topology is the set of all light paths that have been set up between end nodes. An example of logical topology (for the physical topology in Figure 2.4) is shown in Figure 2.5 . The logical topology is a directed graph corresponding to the end nodes for which light paths have been set. Logical topology can be easily reconfigured to adopt new changes in the network.

Physical topology and traffic matrix are two major input parameters for designing a logical topology. Traffic matrix of a given physical topology represents a long term traffic average of that particular network. Thus, once a logical topology is designed for a given physical topology and traffic matrix, it is not changed unless traffic pattern or physical topology (partially or as a whole) changes.

The logical out degree of an end node is the number of lightpaths that originate from that node and similarly the logical in-degree is the number of lightpaths that terminate

at that node [24]. For example, in figure 2.5, the logical out-degree and logical in-degree of all the nodes are 2.

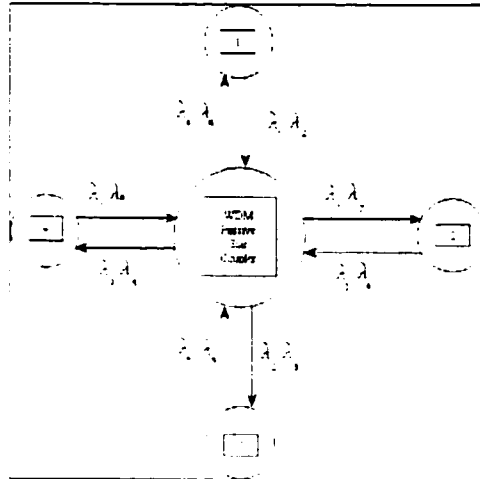


Figure 2.4 An example of a 4 node Physical Topology

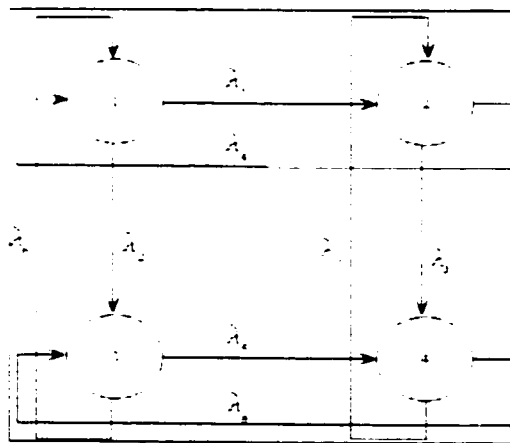


Figure 2.5 : Logical topology corresponding to physical topology in figure 2.4

2.5 Singlehop Network

In single-hop networks a signal transmitted from source reaches its destination directly without going through other end nodes (or routers) [28]. These networks are all-optical

by their nature as signals continue to remain in the optical domain throughout their transmission [20]. This results in very fast communication. In [20] the author has investigated this architecture in detail. Rainbow-I [31] from IBM and LAMDANET [12] from Bellcore are two popular research prototypes used to study these type of networks. Single-hop networks are generally not very scalable.

2.6 Multi-Hop Network

Multihop networks are used in situations where a direct link between a source and a destination node is, in general, not available [28]. In a multi-hop network, information from a source is routed through several intermediate nodes before reaching its destination. This architecture was originally introduced by Acampora in [2]. Other important contributions include [4], [6], [21] and [23].

In multi-hop architecture, packets traveling from one node to another may be routed via intermediate nodes in a sequence of 'hops', before reaching the destination. Let us consider again the network in figure 2.5. Node 1 can communicate with node 2 and node 3 directly via wavelength λ_1 and λ_2 , which means node 1 is logically connected to node 2 and node 3. Suppose now node 1 wants to communicate with node 4. Since there is no direct connection between them, the communication must be routed through one or more intermediate nodes. For instance, node 1 should reach node 4 in 2 hops through node 2 ($1 \rightarrow 2 \rightarrow 4$) or through node 3 ($1 \rightarrow 3 \rightarrow 4$). There will be a wavelength conversion at node 2 from λ_1 to λ_3 . The conversion is done in the electronic domain. Thus, the signal is

converted from optical domain to electrical domain and then reconverted to optical domain.

2.7 Wavelength Routed Optical Network (WRON)

Wavelength routing means that optical signals can be selectively routed, in the optical domain, based on their wavelengths. These types of networks are referred to as Wavelength Routed Optical Network (WRON). WRON is the combination of single-hop and multi-hop network and they show the characteristics of both. A Wavelength Routed Switch (WRS) is used at intermediate router nodes. Optical Networks using wavelength routing are considered to be the backbone for high speed future networks [24]. Figure 2.6 shows an optical network with 4 nodes. There are 3 lightpaths established over the physical network $A \rightarrow C$, $C \rightarrow A$ and $B \rightarrow A$. Communication from node A to node C is an example of single-hop communication since communication is done by one single lightpath $A \rightarrow C$. Communication from node B to node C is an example of multi-hop communication since this communication is performed along the path $B \rightarrow A \rightarrow C$ which uses two lightpaths $B \rightarrow A$ and $A \rightarrow C$.

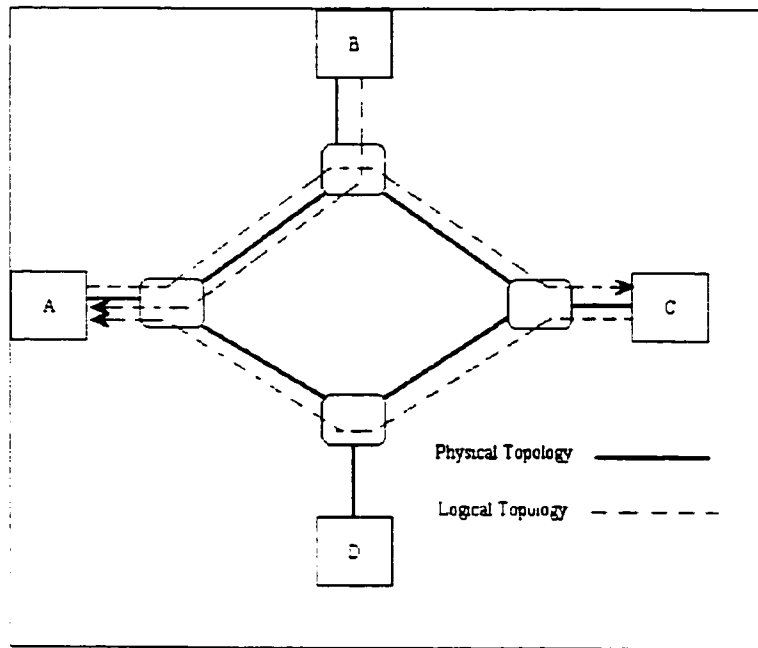


Figure 2.6 Wavelength Routed Optical Network (WRON)

Since routing mechanism in WRON remains in optical domain, there is no opto-electronic conversion needed. In WRS, routing is performed based on their input ports and wavelengths and thus opto-electronic conversion is not required to read the addresses of the packets.

2.8 Regular Multihop Topologies

The logical topology of a multihop network can be either regular (symmetrical), such as ShuffleNet, Toroid, de Bruijn graph, GEMNET, hypercube etc. or irregular (asymmetrical). Most previous approaches to logical topology design result in arbitrary graphs. In this thesis we want to explore the use of regular graphs as potential candidates for logical topologies for multi-hop networks.

We have selected GEMNET as the regular topology in our studies because of the

following reasons :

- (a) the interconnection pattern in a GEMNET is simple
- (b) the network can be defined for any number of nodes (most of the regular graphs have rigid restrictions about the number of nodes) and
- (c) this topology was specifically proposed for multi-hop networks [29]

2.8.1 GEMNET Architecture

In this section we give a brief review of the GEMNET architecture, which we have selected as the target regular architecture for our logical topologies. GEMNET has a simple routing scheme and low diameter, $O(\log N)$, where N is the number of nodes in the network. GEMNET can be defined for any arbitrary value of N .

The GEMNET architecture [29] is a Generalized shuffle exchange network of $n = k \times m$ nodes each of degree d . The nodes are arranged in k columns, with m nodes per column. In this study, we restrict ourselves to single column GEMNET architecture (i.e. $k = 1$), which can be defined for any arbitrary number of nodes and reduces to a de Bruijn graph for $N = d^m$. For a single column GEMNET $m = \log_d n$, and each node has a label $i \in \{0, 1, 2, \dots, d-1\}$. A logical node u having label i will have an edge v having a label j iff $j = d \times i - p \pmod{n}$ and $p \in \{0, 1, 2, \dots, n-1\}$.

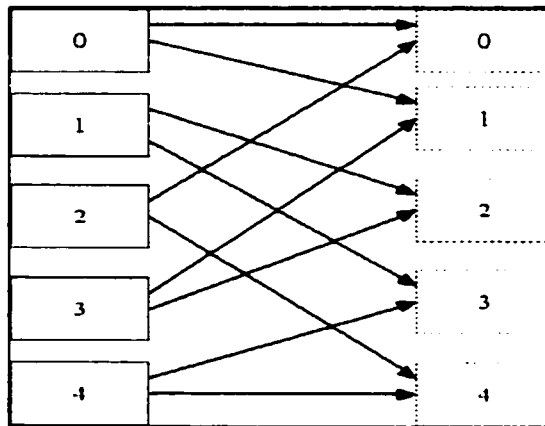


Figure 2.7: Example of GEMNET architecture

Figure 2.7 shows a single column GEMNET with 5 nodes, each of degree 2. The nodes are represented by squares, with the labels shown inside each node. In figure 2.7, we have shown a copy of the nodes in a second column in order to illustrate the interconnections more clearly.

2.9 Benefit of using Regular Topologies for Congestion

Regular topologies such as GEMNET has symmetrical structure and thus the average hop distance is relatively small in such topologies. For a GEMNET architecture of n node the average hop distance is $O(\log(n))$ which results in shorter paths for routing packets. The maximum offered load on a link is called *congestion* [24]. The smaller the value of the *congestion*, the better the performance of the network [14].

In [14] authors showed that the de Bruijn Graph's attractive low diameter property minimizes the congestion on logical topology.

NOTE TO USERS

Page(s) not included in the original manuscript and are unavailable from the author or university. The manuscript was microfilmed as received.

This reproduction is the best copy available.

UMI

$$T = (t_{ij}) : \begin{bmatrix} 0 & .23 & .12 & .98 \\ .12 & 0 & .34 & .08 \\ .23 & .11 & 0 & .78 \\ .29 & .28 & .11 & 0 \end{bmatrix}$$

- A , represents the set of lightpaths (i.e. logical edges) established over $G_p(N, E_p)$, and $G_l(N, A)$ defines the logical topology.
- d_{ij} represents the delay from node i to node j over logical topology which is the sum of the propagation delay from i to j in physical topology. In [24] each logical link (i, j) is assumed to be established on the shortest propagation delay path from i to j . In Figure 3.1 (a) $d_{AB} = 5$ represents the propagation delay (on physical link) from node A to node B is 5 units. Similarly propagation delay from node B to node C is 3 units. In Figure 3.1(b) a logical link has been established on physical links (Figure 3.1(a)) from node A to node C and the delay on that logical link AC is $d_{AC} = d_{AB} + d_{BC} = 5 + 3 = 8$.

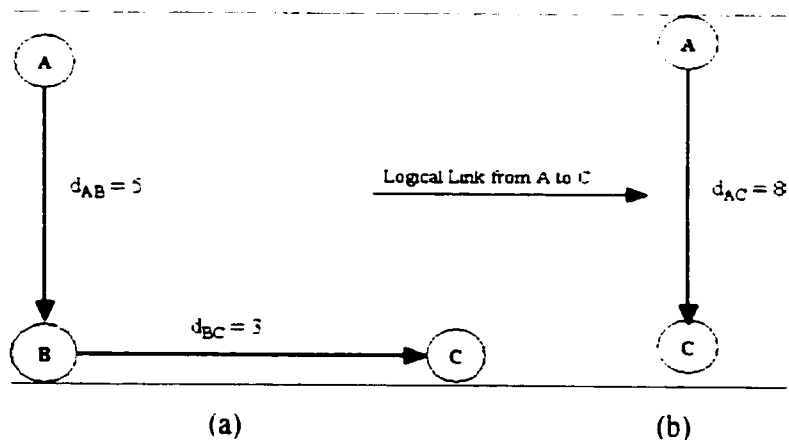


Figure 3.1 : Propagation delay on logical link

- d_{max} : maximum propagation delay on the physical topology between any s - d pair
- Δ , denotes the logical degree (in degree and out degree) of the network

Variables

- b_{ij} is the binary variable which represents the presence of logical edge between node i and j . If $b_{ij} = 1$, there is an edge between node i and j , otherwise edge does not exist between them.
- λ_{ij}^{sd} is the arrival rate of packets for (s,d) pair which is routed over arc (i,j)
- $\lambda_{ij} = \sum_{s,d} \lambda_{ij}^{sd}$ is the arrival rate of packets over arc (i,j) from all (s,d) pairs
- $\lambda^{max} : \max \{ \lambda_{ij} : (i,j) \in A, \}$ is the *congestion* which is defined as the maximum load offered on any link (i,j) . Our objective is to minimize this congestion.
- The delay bound is formulated as a uniform bound on the delay between any s - d pair which means the average delay between any s - d pair is restricted to at most α times the worst case propagation delay between any s - d pair, and is represented by αd_{max} where $\alpha \geq 1$.

3.3 Previous Approaches

If we set up $n(n-1)$ number of lightpaths where n is the number of nodes in the network, we will achieve the minimum congestion value for that particular network. But physical limitations such as number of wavelengths supported by a fiber as well as the number of transmitters and receivers per node makes this infeasible.

In last few decades considerable research effort has been focused on developing [1,9,15,17,28] and analyzing optimization models for designing logical topologies over a given physical fiber network. The approaches can be categorized into two major parts :

- 1) Formulating the problem as a combined design problem of logical topology and routing
- 2) Divide the whole problem into two sub-problem as :
 - a. Logical Topology Design problem
 - b. Routing problemand solve them individually

The first problem formulation involves Mixed Integer Linear Programming (MILP) formulation of the problem which generates a large number of constraints even for small sized network. The second approach generally uses heuristics to solve the first part and then formulate a Linear Programming (LP) for the routing sub problem.

Section 3.3.1 describes the details of MILP formulation of the problem and its limitations. Section 3.3.2 describes the second approach which uses heuristics to obtain logical topology and then formulate the routing problem as a LP problem. Details of these approaches are available in [24]. Section 3.3.3 describes the fundamentals of the Meta Heuristic based approach [1] using Tabu Search.

3.3.1 MILP Formulation of the Problem

The objective of all the MILP formulations that deal with wide area networks is to minimize a form of congestion. In [5], the authors have proposed a MILP formulation for the logical topology design problem. Given the traffic matrix and the physical topology of the underlying network, the MILP produces the logical topology and routing strategy that is to be used on it by minimizing the maximum traffic flowing on any logical edge.

The objective of this problem is to Minimize λ^{\max} subject to

Flow conservation at each node:

$$\sum \lambda_{ij}^{sd} - \sum \lambda_{ji}^{sd} = \begin{cases} t_{sd} & \text{if } i = s \\ -t_{sd} & \text{if } i = d \\ 0, & \text{otherwise.} \end{cases} \quad \forall i, s, d \in N, s \neq d \quad \dots \dots \dots (1)$$

Equation set (1) is multi commodity balance equations that account for routing of packet traffic.

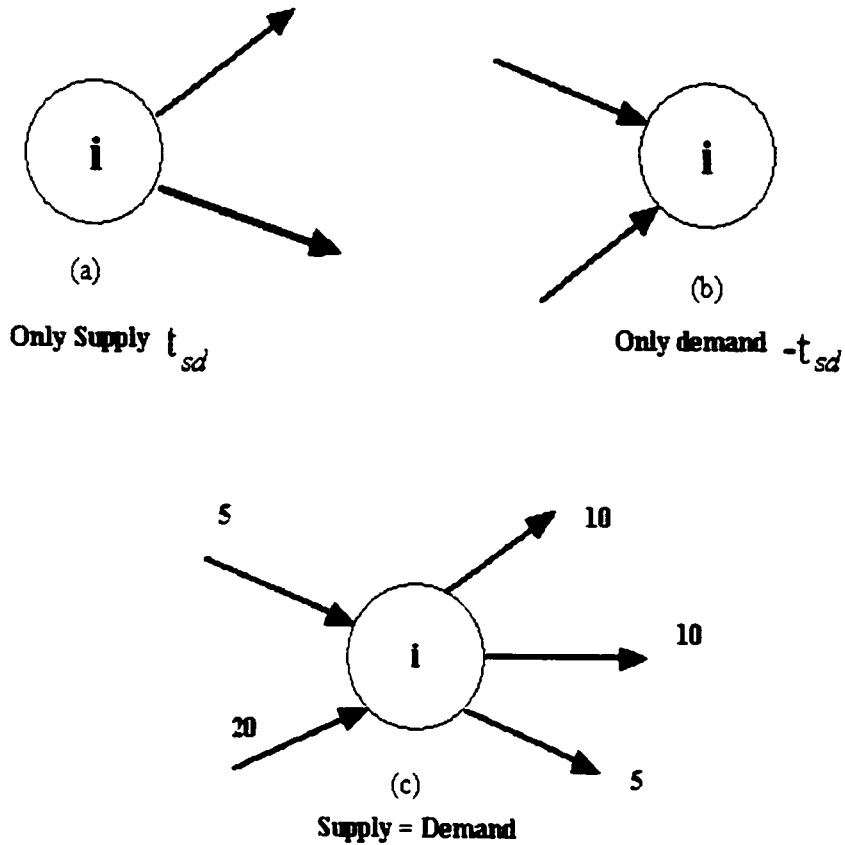


Figure 3.2 : Flow conservation

Figure 3.2 (a) represents the scenario when $i=s$ in equation 1. When $i=s$ in eq.1 , $\sum_j \lambda_{ij}^{sd}$ becomes t_{sd} and $\sum_j \lambda_{ji}^{sd}$ becomes 0 and thus eq. 1 gives us t_{sd} which represents supply. Figure 3.2 (b) represents the scenario when $i=d$ in equation 1. When $i=d$ in eq.1 , $\sum_j \lambda_{ij}^{sd}$ becomes 0 and $\sum_j \lambda_{ji}^{sd}$ becomes t_{sd} and thus eq. 1 gives us $-t_{sd}$ which represents demand. Figure 3.2 (c) represents the scenario when node i is neither a source nor destination node.

Total Flow on logical link :

$$\sum_{s,d} \lambda_{ij}^{sd} \leq \lambda^{\max}, \quad \forall i, j \in N, i \neq j \dots\dots\dots (2)$$

Constraint (2) guarantees that total flow on any link will not be greater than the congestion λ^{\max} which is the maximum load on any logical link.

$$\lambda_{ij}^{sd} \leq b_{ij} t_{sd}, \quad \forall i, j, s, d \in N, i \neq j, s \neq d \dots\dots\dots (3)$$

Constraint (3) satisfies that flow on a logical link (i,j) which is destined for $s-d$ pair will never be greater than the total supply of that commodity $s-d$.

Average Delay constraint

$$\sum_{i,j} \lambda_{ij}^{sd} d_{ij} \leq \alpha_{sd} d^{\max} \quad \forall s, d \in N, s \neq d \dots\dots\dots (4)$$

Constraint set (4) allows us to limit the average transmission delay $\alpha_{sd} d^{\max}$ is the maximum delay allowed for a commodity $s-d$. The higher the value of α , the lower the congestion would be since higher value of α allows more multi-hop communications. It is interesting to observe that tight value of α (small α) does increase the congestion, but this increase is not much significant [24].

We note that average delay constraint mentioned above does not take into account any delay involved with opto-electronic conversion at nodes along multihop paths.

Degree Constraints

$$\sum_j b_{ij} = \Delta, \quad \forall j \in N \quad \dots\dots\dots (5)$$

$$\sum_j b_{ji} = \Delta, \quad \forall i \in N \quad \dots\dots\dots (6)$$

Constraints set (5) and (6) imply a regular topology with in-degree and out-degree of every node to be Δ . Number of transmitters and receivers employed at routing nodes determine the degree of a node.

$$\lambda_{ij}^{sd} \geq 0, \quad \forall i, j, s, d \in N, i \neq j, s \neq d \quad \dots\dots (7)$$

$$h_{ij} = \{ 0, 1 \} \quad \forall i, j \in N, i \neq j \quad \dots\dots\dots (8)$$

The above MILP formulation allows traffic between any commodity $s-d$ to be split across multiple possible routes.

3.3.1.1 Limitations of MILP Formulation

The above MILP formulation which formulates a combined logical topology design and routing problem suffers from some major drawbacks.

The first and most important limitation of this approach is that the MILP gets computationally intractable for even moderately sized networks due to the presence of a large number of integer (0- 1) variables. Classical methods like branch and bound and other cutting plane approaches failed to solve the problem in a reasonable amount of time

and require enormous amount of computing power. The number of constraints and variables increases exponentially as the number of nodes grow in the network. There are $n^2(n-1)^2 - 1$ continuous variables, $n(n-1)$ binary variables and $n^3(n-1) - 2n^3$ constraints. For example, for a 14 node NFSNET [17] network , there are more than 33,000 continuous variables, 182 binary variables and more than 30,000 constraints. Table 3.1 shows how rapidly the number of variables and constraints increase as the number of nodes increased in the network.

Number of Nodes	Continuous Variables	Binary Variables	Constraints
9	5185	72	7290
16	57,601	240	69632
27	492,804	702	551,124
64	16,247,025	4032	17,039,360

Table 3.1 : Number of variables and constraints in MILP formulation

The second limitation is that this approach ignores wavelength constraints, which need to be considered for a complete logical topology design problem [8]. It simply assumes that enough wavelengths are available.

For these reasons, this MILP approach fails to give us an acceptable solution to the logical topology design and routing problem. Thus, rather than looking for an optimal topology one needs to develop alternate strategies to obtain “good” solutions to the above MILP.

3.3.2 Heuristics and LP Formulation

Since the aggregate MILP formulation was found to be infeasible for practical networks, attempts were made to simplify the problem by dividing it into two parts: first determine a logical topology using some heuristics and then find an optimal routing on that topology by formulating the routing as an LP problem.

In [24], for a given logical topology, the routing sub-problem is the linear program obtained by fixing the values of the b_{ij} in the MILP formulation of section 3.3.1 and is computationally quite tractable, at least for moderate sized networks.

3.3.2.1 Heuristics

In this approach, for the topology design sub-problem, heuristics are used to obtain the logical topology. In this section we briefly outline some of the heuristics discussed in [24]. Table 3.2 compares the performance of these heuristics for a 14-node NFS network.

Heuristic Logical Topology Design Algorithm (HLDA)

HLDA stands for Heuristic logical topology design algorithm. It attempts to place logical links between nodes with very high traffic and proceeds in the order of decreasing traffic. The idea behind the heuristic is that routing most of the traffic in one hop may lower congestion [24]. However this algorithm does not account for delay constraints as it is more focused on the traffic.

Algorithm:

Step 1: Given a traffic distribution matrix $P = (p_{ij})$, make a copy $Q = (q_{ij})$.

Step 2: Select the source destination pair (i_{max}, j_{max}) with largest traffic

$$^4 i_{max}, j_{max} = \max_{ij} (q_{ij})$$

If all source destination pairs with non-zero traffic have already been tried go to step 4.

Step 3:

If node i_{max} has fewer than Δ_1 outgoing edges and j_{max} has fewer than Δ_1 incoming edges

find lowest available wavelength on the shortest propagation delay path between i_{max} and j_{max} in the physical topology.

If wavelength available then

Create logical edge (i_{max}, j_{max})

Find Source destination pair $i_1 j_1$ with next highest traffic

Set $^4 i_{max}, j_{max} = ^4 i_{max}, j_{max} - ^4 i_1, j_1$: go to step 2

else

$^4 i_{max}, j_{max} = 0$ go to step 2

else $^4 i_{max}, j_{max} = 0$ go to step 2

Step 4: If there are less than N edges then add as many remaining edges at random without violating the degree constraints and a wavelength can be found on the shortest path for a logical edge

In heuristic algorithms like HLDA only a subset of the traffic demands are considered while constructing the logical topology. This might result in a network with higher congestion in the edges that were ignored in the topology design process.

Traffic Independent Logical Topology Design Algorithm (TILDA)

As the name suggests this approach designs logical topologies regardless of the traffic on the network. The algorithm starts by placing logical edges over the physical topology between all one hop neighbors [24]. Then it places logical links between all two hop neighbors provided one doesn't exist before and continues. The algorithm checks for degree constraint violations at each stage. This approach attempts to minimize the number of wavelength used and is a very good choice while designing logical topologies if little or no information is available on the traffic over the network [27].

Minimum Delay Logical Design Algorithm (MLDA)

MLDA is only defined when Δ , is greater than the degree of physical topology. If that is the case then MLDA creates a pair of directed logical edges for each physical edge and after placing all of those logical edges, then MLDA starts using HLDA to place remaining logical edges. Thus, MLDA is capable of satisfying the tightest delay constraint which is physically realizable.

Following Table 3.2 shows the performance comparison between HLDA, TILDA and MLDA [24]. These values are obtained by solving the routing subproblem on the degree-4 logical topologies designed by HLDA, MLDA and TILDA for the 14-node NSFNET for various values of the delay parameter α for a particular traffic pattern [24]. In table 3.2 a 'X' indicates that the routing subproblem is infeasible.

α	1.0	1.5	2	α
HLDA	X	X	142.33	142.33
TILDA	217.38	210.85	210.85	210.85
MLDA	156.17	155.97	155.97	155.97

Table 3.2 : Performance comparison between different heuristics in terms of congestion values

Studies showed that HLDA requires a large number of wavelengths than MLDA or TILDA [24]. From the experimental results in [24], it is observed that HLDA achieves lower values of the congestion than the other algorithms. MLDA is also very close to HLDA in performance. TLDA is slightly poorer showing that traffic should be considered while designing the logical topology.

Another heuristic for topology design called RBDA was proposed in [14]. Below is the brief description of RBDA

RBDA stands for Regular graph Based logical topology design algorithm. RBDA is introduced and details of this algorithm has been studied in [14].

The objective of this algorithm is :

- 1) Select those node pairs (A,B) in the network such that there is a relatively high amount of traffic from A to B and
- 2) Ensure that those node pairs are assigned addresses such that , in the regular graph (De Bruijn) corresponding to the network, the path lengths from A to B for all such node pairs are as small as possible.

3.3.2.2 LP Formulation for Routing

For a given logical topology $G_l(N, A_l)$ which is obtained by the above mentioned heuristic algorithms, the MILP formulation (in section 3.3.1) reduces to a linear program. This LP formulation also generates a fairly large number of constraints and variables. It generates $n^2(n-1) - 1$ variables and $n(n^2-1) - |A_l|$ constraints. The LP problem for routing sub-problem is given below :

Min λ^{\max}

subject to :

$$\sum_j \lambda_{ij}^{sd} - \sum_j \lambda_{ji}^{sd} = \begin{cases} t_{sd} & \text{if } i = s \\ -t_{sd} & \text{if } i = d \\ 0, & \text{otherwise.} \end{cases} \quad \forall i, s, d \in N, s \neq d \quad \dots (1)$$

$$\sum_{s,d} \lambda_{ij}^{sd} \leq \lambda^{\max}, \quad \forall (i, j) \in A_l \quad \dots (2)$$

$$\sum_{i,j} \lambda_{ij}^{sd} d_{ij} \leq \alpha t_{sd} d^{\max} \quad \forall s, d \in N, s \neq d \quad \dots (3)$$

$$\lambda_{ij}^{sd} \geq 0, \quad \forall (i, j) \in A_l, s, d \in N, s \neq d \quad \dots (4)$$

The above LP generates more than 2500 variables and 2739+ $|A_l|$ constraints for a 14 node NFSNET which can be solved by using only large commercial LP solver. For larger networks even such LPs would be computationally intractable.

3.3.2.3 Limitations of Heuristics and LP formulation

Limitations of Heuristic Algorithms

The heuristic algorithms mentioned earlier generate an arbitrary graph for the logical topology. They do not enforce any regular topology and therefore cannot exploit the attractive properties of such topologies. For example, in the arbitrary graph generated by these algorithms there is no upper bound on the diameter of the graph. Consequently, certain paths between a given s-d pair may be very long (i.e. require large number of hops). In general, it is expected that longer paths will lead to higher congestion.

Figure 3.3 below is a simple illustration of this situation.

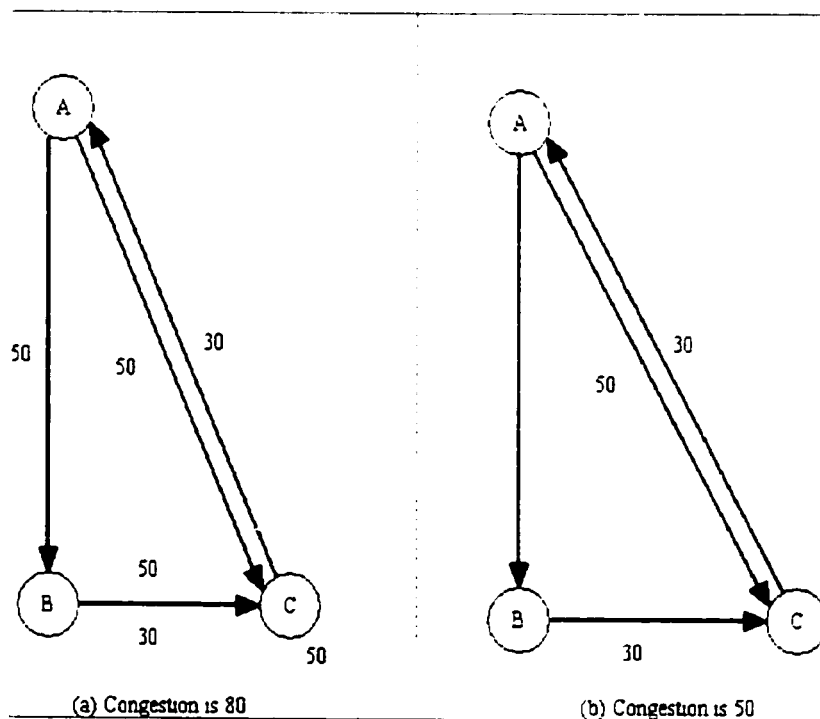


Figure 3.3 : Minimizing Congestion

In Figure 3.3 (a) 50 units of traffic is sent from A to C through the path $A \rightarrow B \rightarrow C$ requiring 2 hops and 30 units of traffic from B to A through the path $B \rightarrow C \rightarrow A$ requiring

2 hops which gives the congestion value of 80 (load on edge BC). On the other hand in figure 3.3 (b) A and B are sending same amount of traffic but over a different path. A is sending 50 units in a single hop from $A \rightarrow C$ and B is sending 30 units in a same way as in fig 3.3 (a) along path $B \rightarrow C \rightarrow A$. The resulting value of congestion is 50 (along $A \rightarrow C$) which is lower than congestion in fig 3.4 (a). Therefore, by reducing the number of hops required to transmit data from A to C we are able to reduce the congestion of the network. We expect that logical topologies with low diameter should perform better than irregular topologies, since there is an upper bound on the maximum path length.

Logical topologies generated by HLDA, TILDA or MLDA do not generate regular graphs. So, there is no upper bound on number of hops required to reach a destination.

TILDA is not an appropriate choice for logical topology design because it does not take into account the traffic information. HLDA uses shortest path to setup the corresponding light path. But if a wavelength is not available on the shortest path, HLDA does not look for any alternate paths. This may cause some high traffic commodities to use very long paths, leading to higher congestion.

Limitations of LP Formulation

Due to the fairly high number of variables and constraints, even the LP formulation can become computational intractable. This LP formulation generates $n^2(n-1) - 1$ variables and $n(n^2-1) - A$, constraints. Again, for 14 node NFSNET the LP has 2549 variables and $2739 + A$, constraints. Table 3.3 shows the growth of number of variables and

NOTE TO USERS

Page(s) not included in the original manuscript and are unavailable from the author or university. The manuscript was microfilmed as received.

This reproduction is the best copy available.

UMI

Chapter 4

Our approach

With the rapid improvements in technology, wide area WDM networks with a relatively large number of nodes will be quite realistic. Existing approaches using LP having a basis size $O(n^2)$ can become computationally intractable for determining the optimal routing scheme even when the logical topology is known. In this thesis, we have introduced a novel LP formulation [32] for optimally solving the routing problem for a given logical topology. Rather than using standard simplex method, we have exploited the underlying structure of the linear program and reformulated the LP allowing us to develop an efficient column generation scheme embedded into the revised simplex method. This new method uses a basis size l only. Since $l \leq n \times d$, where d is the degree of the logical topology and $d \leq n$, this method allows us to solve much larger problems. Our experiments indicates that our formulation dramatically improves the computation time to minimize the congestion as done in [24].

Our rationale for using a regular logical topology is that a regular topology has a low diameter and usually a large connectivity. This means that, in a regular topology, there is a large number of alternative paths from any source to any destination. In the de Bruijn graph, for example, it is known that there are a large number of node disjoint paths from a source to a destination, each having almost the same number of nodes as in the shortest path route. Since we use our new LP formulation to determine the routing, once we fix the logical topology, we are not restricted to the use of fixed, short routes typically

employed in communication using regular graphs. This means that optimization program can make full use of a) the rich interconnections available in a regular graph and b) the fact that numerous, relatively short, routes exist between any source-destination. We have used the GEMNET as the regular topology in our studies. The reasons are stated in Chapter 2, section 2.9.

We have looked at three classes of topologies :

- i) topologies having no graph theoretical properties, generated by traffic driven heuristics very similar to HLDA [24]
- ii) topologies based on the GEMNET architecture, derived using our heuristic driven by traffic considerations
- iii) hybrid topologies, obtained by inserting additional edges, determined by the techniques used in (i), on top of the GEMNET architecture generated in (ii)

In section 4.1 we have given our heuristics for logical topology design. In section 4.2 we have described the LP optimization techniques.

4.1 Logical Topology Design subproblem

4.1.1 Heuristics for Topology Design

In this section we will describe three simple traffic based heuristics for logical topology design. The first heuristic generates logical topologies whose structural properties are unknown. This approach is practically identical to the well known HLDA [24] and is used as a benchmark for the other topologies. The second generates logical topologies based on

the GEMNET where the edges are determined by traffic considerations. The third is a hybrid combining the first two heuristics. In our heuristics, we need to determine whether a particular lightpath from a specified source node S to a specified destination node D is feasible in the sense that there exists a path P in the physical topology from S to D such that it is possible to dedicate the same channel to all fibers in the path P , i.e. we don't allow any wavelength conversion. Once we have decided that a particular edge will be defined in the logical topology, we will actually dedicate the requisite channel in all fibers in path P to support this lightpath. When implementing our heuristics below, we have explored, using a breadth first search, all paths from S to D whose delay is less than $\alpha \text{ delay}_{S,D}$ where $\text{delay}_{S,D}$ is the delay along the shortest path from S to D and α is a predetermined constant.

Heuristic 1

The steps of heuristic 1 are very similar to that of HLDA [24]. The only difference is that we find a path P from source to destination using a breadth-first search rather than only the shortest path.

Algorithm:

Step 1: Pick the highest entry $(t_{s,d})$ from T and try to setup a lightpath between the corresponding source-destination pair by :

- (a) Check that in-degree and out-degree of the corresponding nodes will not be violated if a lightpath is setup
- (b) First try to establish a lightpath through the shortest path from source to destination providing that wavelength is available on that particular path

- (c) If it is not possible to setup the lightpath on the shortest path, try an alternative path to setup the lightpath with the restriction that the delay of the all path can be at most α times the delay of the shortest path.

Step 2: Modify traffic matrix by setting (t_{sd}) by 0.

Step 3: Repeat step 1 and 2 until all entries in T becomes 0.

We have provided an illustrative example of this Heuristic below :

Example of Heuristic 1:

Consider the following 4-node network. T represents the traffic matrix for this network.

Assume that number of available wavelengths per link is 2 and logical degree is 2.

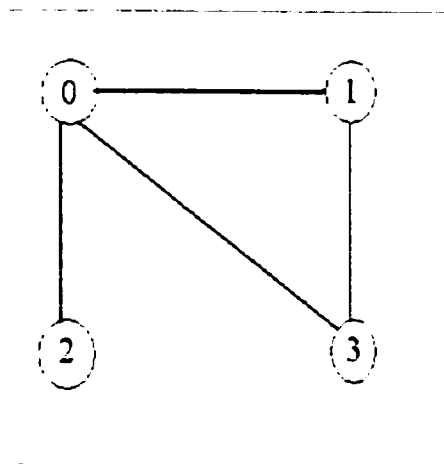


Figure 4.1 : A 4-node network where 2 is the number of wavelengths per link

Let us consider following traffic matrix T for the above 4-node network :

$$T = \begin{bmatrix} 0.0 & 0.0 & 0.4 & 0.2 \\ 0.5 & 0.0 & 0.8 & 0.6 \\ 0.0 & 0.9 & 0.0 & 0.1 \\ 0.3 & 0.2 & 0.7 & 0.0 \end{bmatrix}$$

Step 1: Pick up the 2→1 source destination pair since $t_{21}=0.9$ is the maximum traffic

Step 2: Establish a lightpath along shortest path 2→0→1 on wavelength λ_1 .

Step 3: Modify traffic matrix T by setting $t_{21}=0$. Traffic matrix T and in-degree and out-degree of nodes are shown below:

$$T = \begin{bmatrix} 0.0 & 0.0 & 0.4 & 0.2 \\ 0.5 & 0.0 & 0.8 & 0.6 \\ 0.0 & 0.0 & 0.0 & 0.1 \\ 0.3 & 0.2 & 0.7 & 0.0 \end{bmatrix}$$

Node	In-degree	Out-degree
0	0	0
1	1	0
2	0	1
3	0	0

Step 4: Pick up the 1→2 source destination pair since $t_{12}=0.8$ is the maximum traffic

Step 5: Establish a lightpath along shortest path 1→0→2 on wavelength λ_1 .

Step 6: Modify traffic matrix T by setting $t_{12}=0$. Traffic matrix T and in-degree and out-degree of nodes are shown below:

$$T = \begin{bmatrix} 0.0 & 0.0 & 0.4 & 0.2 \\ 0.5 & 0.0 & 0.0 & 0.6 \\ 0.0 & 0.0 & 0.0 & 0.1 \\ 0.3 & 0.2 & 0.7 & 0.0 \end{bmatrix}$$

Node	In-degree	Out-degree
0	0	0
1	1	1
2	1	1
3	0	0

Step 7: Pick up the 3→2 source destination pair since $t_{32}=0.7$ is the maximum traffic

Step 8: Establish a lightpath along shortest path 3→2 on wavelength λ_2 .

Step 9: Modify traffic matrix T by setting $t_{32}=0$. Traffic matrix T and in-degree and out-degree of nodes are shown below:

$$T = \begin{bmatrix} 0.0 & 0.0 & 0.4 & 0.2 \\ 0.5 & 0.0 & 0.0 & 0.6 \\ 0.0 & 0.0 & 0.0 & 0.1 \\ 0.3 & 0.2 & 0.0 & 0.0 \end{bmatrix}$$

Node	In-degree	Out-degree
0	0	0
1	1	1
2	2	1
3	0	1

Step 10: Pick up the 1→3 source destination pair since $t_{13}=0.6$ is the maximum traffic

Step 11: Establish a lightpath along shortest path 1→3 on wavelength λ_1 .

Step 12: Modify traffic matrix T by setting $t_{13}=0$. Traffic matrix T and in-degree and out-degree of nodes are shown below:

$$T = \begin{bmatrix} 0.0 & 0.0 & 0.4 & 0.2 \\ 0.5 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.1 \\ 0.3 & 0.2 & 0.0 & 0.0 \end{bmatrix}$$

Node	In-degree	Out-degree
0	0	0
1	1	2
2	2	1
3	1	1

Step 13: Pick up the 1→0 source destination pair since $t_{10}=0.5$ is the maximum traffic

Step 14: Since establishing the lightpath 1→0 will violate the out-degree of node 1, modify traffic matrix T by setting $t_{10}=0$. Traffic matrix T and in-degree and out-degree of nodes are shown below:

$$T = \begin{bmatrix} 0.0 & 0.0 & 0.4 & 0.2 \\ 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.1 \\ 0.3 & 0.2 & 0.0 & 0.0 \end{bmatrix}$$

Node	In-degree	Out-degree
0	0	0
1	1	2
2	2	1
3	1	1

Step 15: Pick up the 0→2 source destination pair since $t_{02}=0.4$ is the maximum traffic

Step 16: Since there is no wavelength available to establishing the lightpath 0→2 along the shortest path (note that there is no alternate path available), modify traffic matrix T by setting $t_{02}=0$. Traffic matrix T and in-degree and out-degree of nodes are shown below:

$$T = \begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.2 \\ 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.1 \\ 0.3 & 0.2 & 0.0 & 0.0 \end{bmatrix}$$

Node	In-degree	Out-degree
0	0	0
1	1	2
2	2	1
3	1	1

Step 17: Pick up the 3→0 source destination pair since $t_{30}=0.3$ is the maximum traffic

Step 18: Establish a lightpath along shortest path 3→0 on wavelength λ_1 .

Step 19: Modify traffic matrix T by setting $t_{30}=0$. Traffic matrix T and in-degree and out-degree of nodes are shown below:

$$T = \begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.2 \\ 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.1 \\ 0.0 & 0.2 & 0.0 & 0.0 \end{bmatrix}$$

Node	In-degree	Out-degree
0	1	0
1	1	2
2	2	1
3	1	2

Step 20: This process will continue until all the entries in T becomes 0.

Heuristic 2

When describing our heuristic for generating a logical topology based on the GEMNET architecture for a n node network where each node has a degree d , it is convenient to associate a distinct identifier Id_i and a distinct label $Label_i$ with the i^{th} node, in the network. Initially a label is not assigned to any node in the network and, after the heuristic is finished, every node has a distinct label associated with it.

In a network with n nodes, there are $n!$ ways of assigning labels to the nodes in the network. In this heuristic we are trying to connect, by a lightpath, as many pairs of nodes (u, v) as possible where there is a high traffic between u and v . We have to be careful that we do not violate the restrictions of the target logical topology. This is a greedy heuristic so that in each iteration, we assign a label to a selected node N on the basis that the current assignment of label l to N is the best assignment of a label to a node according to the current situation. Once a label has been assigned to a node, that label is never changed. An example of this heuristic is given later

Definition

A successor (predecessor) label of a node u having a label i in the logical topology is the label j of any node v such that an edge (respectively) exists in the logical topology.

Clearly, given the label of a node in a GEMNET architecture, any successor label or predecessor label can be determined immediately

In order to implement our heuristic we maintain three lists $L1$, $L2$ and $L3$ where

i) $L1$ is a list of all (identifier, label) pairs of nodes which have been assigned a label,

- ii) $L2$ is a list of all identifiers of nodes which have not been assigned a label and
- iii) $L3$ is a list of all labels which are not yet assigned to any node but will be the labels of nodes which are either the predecessor or the successor of a node whose identifier and label appears in $L1$.

In describing the heuristic, we have used a function *findBestMapping* with the three lists $L1$, $L2$ and $L3$ as arguments. This function attempts to match, if possible, a node in $L2$ with a label in $L3$ such that the objectives of our greedy heuristic are satisfied. If the match is possible, this function returns the (Identifier, Label) pair corresponding to the match it has found. If the match is not possible, the function returns a label of -1. This function is described later.

1. (Initializations)

- a) Pick the node X , such that there is a node Y where the traffic from X to Y is the highest entry in the traffic matrix T .
- b) Assign label 0 to node X .
- c) $L1$ = list consisting of the pair $(X, 0)$
- d) $L2$ = list consisting of the identifiers of all nodes except X
- e) $L3$ = list of all unassigned labels (i.e., excluding the label 0) which will be the labels which are either the predecessor or the successor of the node with label 0

2. while $L2 \neq \Phi$

3. {

```

(Id, L) = findBestMapping(L1, L2, L3).
4.  if (L == -1) exit. // topology design failed
    else
5.  {      Add the pair (Id, L) to the list L1. // Another node has an assigned label
        Remove the identifier Id from L2.
        Remove the label L from L3.
        Include in L3 all unassigned labels that will be the labels of nodes which
        are either the predecessor or the successor of the node with label L.
    }
}

```

We now describe the function *findBestMapping(L1, L2, L3)*.

```

1. Best_average_traffic_so_far = -9999
2.  repeat steps 3 - 7 for all label l in L3
3.  repeat steps 4 - 7 for all node identifier id in L2
4.  if there exists any predecessor label p of label l, which appears in list L1. such that
    a lightpath from the node assigned the label p to the node with identifier id is not
    feasible. do not consider the pair (l, id) any further.
    else
        { total_traffic_to_node = sum of all traffic from all nodes in list L1
          whose labels are predecessors of label l to the node with identifier id
          number_of_predecessor_nodes = number of all nodes in list L1 whose

```

labels are predecessors of label l

5. **if** there exists any successor label s of label l , which appears in list Ll , such that a lightpath from the node with identifier id to the node assigned the label s is not feasible, do not consider the pair (l, id) any further.

else

{

$total_traffic_from_node$ = sum of all traffic from the node with identifier id to all nodes in list Ll whose labels are successors of label l

$number_of_successor_nodes$ = number of all nodes in list Ll whose labels are successors of label l
6. $Average_traffic = (total_traffic_to_node - total_traffic_from_node) / (number_of_predecessor_nodes + number_of_successor_nodes)$
7. **if** ($Average_traffic > Best_average_traffic_so_far$)

; $Best_average_traffic_so_far = Average_traffic$

$Best_return_value_so_far = (l, id)$

;
8. **if** ($Best_average_traffic_so_far$ is -9999) return (null, -1)

else return $Best_return_value_so_far$

Example of Heuristic 2

Here we are explaining our heuristic 2 which is used to design a GEMNET based logical topology with degree two for a four node network having nodes A, B, C and D. Let the traffic matrix be T given below with rows 1, 2, 3 and 4 correspond to source nodes A, B, C and D and columns 1, 2, 3 and 4 correspond to destination nodes A, B, C and D.

$$T = \begin{bmatrix} 0.0 & 0.7 & 0.1 & 0.4 \\ 0.6 & 0.0 & 0.8 & 0.5 \\ 0.3 & 0.9 & 0.0 & 0.1 \\ 0.1 & 0.2 & 0.8 & 0.0 \end{bmatrix}$$

In this example, we assume that there are enough channels/fiber so that all lightpaths are feasible.

Step 1: The element in T having the highest traffic is $t_{3,2}$ which corresponds to the traffic from node C to node B. So node C is assigned label 0. the predecessors of label 0 are label 0 and label 2 and the successors of label 0 are labels 0 and 1. Thus $L1 = [(C, 0)]$, $L2 = [A, B, D]$ and $L3 = [1, 2]$.

Step 2: L2 is not empty.

Step 3: We can assign the node A, B or D the label 1 or 2. For example, if we allocate to A the label 1, we have a logical edge from node C (with a label 0) to node A (with a label 1). The traffic from C to A is 0.3. Since this is the only edge to consider, average traffic for this label allocation is 0.3. Considering all the 6 allocations, the best average traffic is

when we allocate label 1 to node B. (In view of our assumption above, all lightpaths are feasible). Thus *findBestMapping* returns the pair (B, 1).

Step 4: the label returned is 1 - it is not negative.

Step 5: We update the lists so that $L1 = [(C, 0), (B, 1)]$, $L2 = [A, D]$ and $L3 = [2, 3]$.

We include label 3 in L3 since label 3 is a successor of 1 which has not been allocated yet.

Step 2: L2 is not empty.

Step 3: We can assign the node A or D the label 2 or 3. Considering all the 4 allocations, the best average traffic is when we allocate label 3 to node A. Thus *findBestMapping* returns the pair (A, 3).

Step 4: The label returned is 3 - it is not negative.

Step 5: We update the lists so that $L1 = [(C, 0), (B, 1), (A, 3)]$, $L2 = [D]$ and $L3 = [2]$.

Similarly, in the next iteration, we get the pair (D, 2) and then the process terminates. The resulting logical topology is shown in Figure 4.1

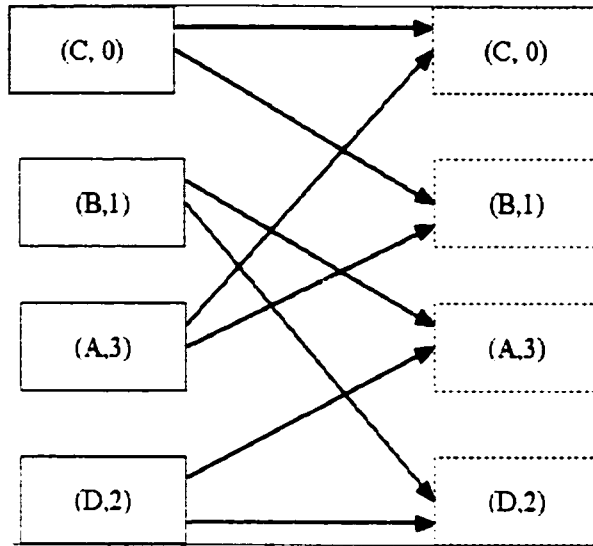


Figure 4.1: Logical Topology for heuristic 2

Heuristic 3

This is a combination of heuristics 1 and 2 in the sense that this gives us a topology where node has a degree d_l where we first implement a logical topology of degree d_l using heuristic 2. We keep in mind that fact that there are some nodes in this topology which have a degree $d_l - 1$ corresponding to the nodes which have a self loop. The remaining edges are then added using heuristic 1, our version of the HLDA algorithm.

4.2 Routing Subproblem [32]

In this part, we focus on the optimal routing problem once a logical topology has been determined by our heuristic algorithms. Our resulting LP can be solved efficiently by using a working basis size of $|A_r|$ only using Generalized upper Bounded (GUB) simplex method [18]. Although the maximum value of $|A_r|$ is $n(n-1)$ but in practice this number is much more smaller than $n(n-1)$. An empirical studies [8] showed that the average size of

$|A_r|$ in many practical situations in $O(n)$. Hence, with the method proposed here, we can solve the optimal routing problem for much larger networks.

The scheme proposed here uses revised simplex method with column generation scheme [18] for determining a candidate column for entering into current basis.

Our proposed column generation scheme [18] provided an insight into path-delays of the paths used in sending t_{sd} units of traffic from s to d .

4.2.1 LP formulation of the Problem

Given a logical topology $G_l(N, A, J)$, our routing subproblem can be formulated as the following LP problem:

Min λ^{\max}
 subject to :

$$\sum_i \lambda_{ij}^{sd} - \sum_j \lambda_{ji}^{sd} = \begin{cases} t_{sd} & \text{if } i = s \\ -t_{sd} & \text{if } i = d \\ 0 & \text{otherwise.} \end{cases} \quad \forall i, s, d \in N, s \neq d \quad \dots (1)$$

$$\sum_{s,d} \lambda_{ij}^{sd} \leq \lambda^{\max}, \quad \forall (i, j) \in A, \quad \dots (2)$$

$$\sum_{i,j} \lambda_{ij}^{sd} d_{ij} \leq \alpha_{sd} d^{\max} \quad \forall s, d \in N, s \neq d \quad \dots (3)$$

$$\lambda_{ij}^{sd} \geq 0, \quad \forall i, j, s, d \in N, s \neq d \quad \dots (4)$$

Here λ_{ij}^{sd} is the amount of $(s \rightarrow d)$ pair traffic going over arc (i, j) , and λ^{\max} is the

congestion. We can consider a polytope defined by the set of constraints (1), (3) and (4) for traffic variables from s to d :

$$\sum_j \lambda_{ij}^{sd} - \sum_j \lambda_{ji}^{sd} = \begin{cases} t_{sd} & \text{if } i = s \\ -t_{sd} & \text{if } i = d \\ 0, & \text{otherwise.} \end{cases} \quad \forall i, s, d \in N, s \neq d \quad \dots (5)$$

$$\sum_{i,j} \lambda_{ij}^{sd} d_{ij} \leq \alpha_{sd} d^{\max} \quad \forall s, d \in N, s \neq d \quad \dots (6)$$

$$\lambda_{ij}^{sd} \geq 0, \quad \forall (i, j) \in A_i \quad \dots (7)$$

Let $W^{sd} = \{ w_1^{sd}, w_2^{sd}, w_3^{sd}, \dots, w_{k_{sd}}^{sd} \}$ be the set of extreme points for this polytope where each extreme point w_r^{sd} is an $|A_i|$ vector. Then any feasible solution λ^{sd} of the above problem can be expressed as a convex combination of the extreme points of the set W^{sd}

$$\text{Thus } \lambda^{sd} = \sum_{r=1}^{k_{sd}} x_r^{sd} w_r^{sd} \quad \dots (8)$$

$$\text{Where, } \sum_{r=1}^{k_{sd}} x_r^{sd} = 1 \quad \dots (9)$$

$$\text{And, } x_r^{sd} \geq 0, r = 1, 2, 3, \dots, k_{sd} \quad \dots (10)$$

Now, our original problem (1) to (4) becomes equivalent to the following problem :

$$\text{Minimize } \lambda^{\max} + M \sum_{s=d} a^{sd}, M \gg 0$$

$$\sum_{(s,d)} \sum_{r=1}^{k_{sd}} x_r^{sd} w_{r(i,j)}^{sd} - \lambda^{\max} + s_{(i,j)} = 0, \forall (i,j) \in A_t \dots\dots\dots (11)$$

$$\sum_{r=1}^{k_{sd}} x_r^{sd} + a^{sd} = 1, \forall s,d \in N, s \neq d \dots\dots\dots (12)$$

$$x_r^{sd} \geq 0, r = 1, 2, \dots, k_{sd} \in N, s \neq d \dots\dots\dots (13)$$

Here, $\{a^{sd}\}$ and $s_{(i,j)}$ are sets of artificial variables and slack variables respectively and M is a very large positive number.

Our reformulated LP problem (11) to (13) is a linear program with $n(n-1) + |A_t|$ constraints. For this LP, the revised simplex method would use a basis size of $n(n-1) + |A_t|$ at each iteration. However, this problem also satisfies the generalized upper bounded (GUB) structure [18]. Thus GUB simplex method would use only a working basis size of $|A_t|$ at each iteration.

Below we develop a column generation scheme using bicriterion approach [19] which enables the revised simplex method to obtain a desired column for entering into basis at each iteration.

The simplex multipliers set $\{\pi_{(i,j)} : (i,j) \in A_t\}$ is associated with the $|A_t|$ constraints in (11) and simplex multipliers set $\{\alpha_{sd} : s,d \in N, s \neq d\}$ is associated with $n(n-1)$ constraints in (12). If $\pi_{(i,j)} > 0$ for some (i,j) , then the corresponding slack variable $s_{(i,j)}$ would be a candidate to enter into basis and the leaving basic variable would be determined in usual

way. To determine if the current basis is optimal, we need to find out whether there exists any variable x_r^{sd} whose reduced cost is negative. So, we need to find out if

$$0 - \left(\sum_{(i,j) \in A_i} \pi_{(i,j)} w_{r(i,j)}^{sd} + \alpha_{sd} \right) < 0 \dots\dots\dots(14)$$

$$\text{or, } \sum_{(i,j) \in A_i} (-\pi_{(i,j)}) w_{r(i,j)}^{sd} < \alpha_{sd} \dots\dots\dots(15)$$

Note that, the set of values which is basically a column is given by an extreme point of the polytope defined by the constraints sets (8) to (10). Before determining if any such extreme point exists, we also need to determine whether the column corresponding to λ^{\max} enter the basis and once it enters it will not leave the basis. The column that corresponding to λ^{\max} is

$$\begin{bmatrix} -1 \\ -1 \\ \cdot \\ \cdot \\ -1 \\ \hline 0 \\ 0 \\ \cdot \\ \cdot \\ 0 \end{bmatrix}$$

Now, we need to find out whether such extreme points exists by solving the following LP problem for the (s→d) pair :

$$\text{Minimize } \sum_{(i,j) \in A_i} -\pi_{(i,j)} \lambda_{ij}^{sd}$$

$$\sum_j \lambda_{ij}^{sd} - \sum_i \lambda_{ji}^{sd} = \begin{cases} t_{sd} & \text{if } i = s \\ -t_{sd} & \text{if } i = d \\ 0, & \text{otherwise.} \end{cases} \quad \forall i, s, d \in N, s \neq d \quad \dots (16)$$

$$\sum_{i,j} \lambda_{ij}^{sd} d_{ij} \leq \alpha_{sd} d^{\max} \quad \forall s, d \in N, s \neq d \quad \dots (17)$$

$$\lambda_{ij}^{sd} \geq 0, \quad \forall (i, j) \in A_i \quad \dots (18)$$

If the optimal objective value of the above LP is less than α_{sd} , then the optimal solution provides a column to enter the basis and simplex method can proceed to the next iteration.

If, however, for every pair of (s,d) the optimal value to the above LP is α_{sd} or higher,

then there is no column to enter the basis and the current solution is optimal.

Ignoring the superscript sd , and defining $\lambda_{ij} = \lambda_{ij}^{sd} / t_{sd}$, the above problem becomes:

$$\text{Minimize } \sum_{(i,j) \in A_i} -\pi_{(i,j)} \lambda_{ij}$$

$$\sum_j \lambda_{ij} - \sum_i \lambda_{ji} = \begin{cases} 1 & \text{if } i = s \\ -1 & \text{if } i = d \\ 0, & \text{otherwise} \end{cases} \quad \forall i, s, d \in N, s \neq d \quad \dots (19)$$

$$\sum_{i,j} \lambda_{ij} d_{ij} \leq \alpha d^{\max} \quad \forall s, d \in N, s \neq d \quad \dots (20)$$

$$\lambda_{ij} \geq 0, \quad \forall (i, j) \in A_i \quad \dots (21)$$

In the absence of constraint (20), the above problem is a shortest path problem from s to d with $(-\pi_{(i,j)})$ as the length of the arc (i,j) . If the shortest path satisfies constraint (20) then clearly it is the optimal solution to the above problem. Otherwise any optimal solution to the above problem must satisfy constraint (20) as a strict equality. Instead of solving the above LP by the standard simplex method, we provide below an efficient scheme named as bicriteria algorithm.

Consider the following bicriteria linear program :

$$\text{Min } \left\{ \begin{array}{l} z_1 = \sum_{(i,j) \in A_s} (-\pi_{(i,j)}) \lambda_{ij} \\ \vdots \\ \{ \lambda_{ij} \in \Lambda \} \\ z_2 = \sum_{(i,j) \in A_s} \lambda_{ij} d_{ij} \end{array} \right\} \quad (22)$$

Here, $\Lambda \{ \lambda_{ij} \in \Lambda \}$ is the convex set of all feasible solutions to the constraints in (19) and (21). Every feasible solution in Λ is mapped by the two linear objective functions defined in constraint (22), into a point $z=(z_1, z_2)$ in the bicriteria space. Thus Z , the mapping of Λ is a convex polygon in the nonnegative quadrant of this bicriteria space. We define a solution $\lambda \in \Lambda$ to be efficient if for every $\lambda \in \Lambda$ such that $z_1(\lambda) \leq z_1(\lambda)$ and $z_2(\lambda) \leq z_2(\lambda)$ implies that $z_1(\lambda) = z_1(\lambda)$ and $z_2(\lambda) = z_2(\lambda)$. Similarly a point

$(z_1, z_2) \in Z$ is efficient if for every $z \in Z$ such that $z_1 \leq z_1$ and $z_2 \leq z_2$ implies that $z_1 = z_1$ and $z_2 = z_2$. From the mapping from Λ to Z it should be clear that $\lambda \in \Lambda$ is efficient if and only if its mapping is efficient. Now it is quite easy to show that any optimal solution λ^* to the problem defined by (19) to (21) maps into an efficient point of Z . Clearly the south-west segment of this polygon Z in the bicriteria object space contains all the efficient extreme points and its called the efficient frontier, thus can be viewed as a function $z_2 = f(z_1)$ and the breakpoints of this function correspond to the extreme point of Λ .

We provide below a scheme somewhat similar to binary search which searches this efficient frontier to solve the problem (19) to (21). We illustrate this scheme geometrically

Assume that an efficient frontier is given in figure 4.2. We first solve the shortest path problem using $\pi_{i,j}$ as the length of arc (i,j) . Let the shortest path value be $z_1^{(1)}$. Among all the shortest paths with value $z_1^{(1)}$, find the one with the least delay and let this point be $z_2^{(1)}$. This can be achieved by slightly modifying the Dijkstra's shortest path algorithm. This point $(z_1^{(1)}, z_2^{(1)})$ provides us with the first efficient extreme point $z^{(1)} = (z_1^{(1)}, z_2^{(1)})$. If $z_2^{(1)} < \alpha d^{\max}$, we would stop as the corresponding path provides the optimal solution. Otherwise, as in our diagram, we proceed to find the second efficient extreme point $z^{(2)} =$

$(z_1^{(2)}, z_2^{(2)})$. We obtain this by solving the shortest path problem by using d_{ij} as the length of the arc. If $z_2^{(2)} > \alpha d^{\max}$, we would stop as this indicates there is no feasible solution to this problem.

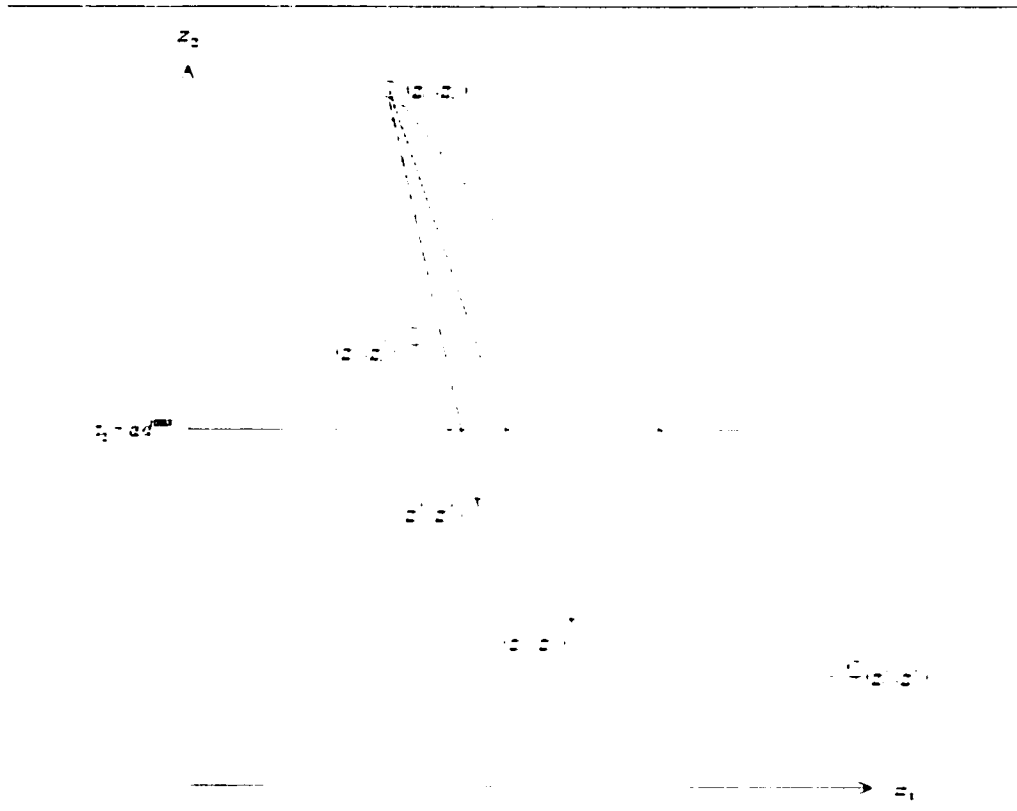


Figure 4.2: Bicriteria Algorithm

With these two starting points $z^{(1)}$ and $z^{(2)}$, we find the equation of the line passing through these two points. Suppose the equation is $az_1 + bz_2 = k$. We then solve our shortest path problem with $a(-\pi_{i,j}) + b(d_{ij})$ as the length of arc (i,j) . Geometrically, this is equivalent to moving this line in the south-west direction, parallel to itself. The next

efficient extreme point we get is $z^{(3)} = (z_1^{(3)}, z_2^{(3)})$. Since $z_2^{(3)} < \alpha d^{\max}$, $z^{(3)}$ will take the roll of $z^{(2)}$ and next we consider the line joining $z^{(1)}$ and $z^{(3)}$. This yield a new extreme point $z^{(4)} = (z_1^{(4)}, z_2^{(4)})$. Since $z_2^{(4)} < \alpha d^{\max}$, we now consider the line joining points $z^{(1)}$ and $z^{(4)}$ and solving the corresponding shortest path problem would give us a new extreme point $z^{(5)} = (z_1^{(5)}, z_2^{(5)})$. Since $z_2^{(5)} > \alpha d^{\max}$, $z^{(5)}$ will now replace the roll of $z^{(1)}$ and now we will consider the line joining extreme points $z^{(4)}$ and $z^{(5)}$. Moving this line does not produce any new extreme point. At this point, we find β such that $\beta z_2^{(5)} + (1 - \beta) z_2^{(4)} = \alpha d^{\max}$. Suppose $z^{(5)}$ and $z^{(4)}$ correspond, respectively, to paths given by λ^5 and λ^4 . Then $\beta z^{(5)} + (1 - \beta) z^{(4)}$ is an optimal solution to the problem.

The formal scheme for generating all efficient extreme points of a bicriteria transportation problem are detailed in [19]. The bicriteria scheme allows us to make an interesting observation. The optimal solution to problem is either at an extreme point (a path) with delay $< \alpha d^{\max}$ or a convex combination of two paths with average delay equal to αd^{\max} . Thus every possible non-slack variable in the optimal basis either correspondence to a path or a convex combination of two paths. Thus, in an optimal routing of traffic, every traffic carrying path with delay more than αd^{\max} can be paired with another traffic carrying path whose delay is less than αd^{\max} . In this sense the formulation comes close to achieving the desired objective of requiring that delay on every traffic carrying path should be limited to αd^{\max} .

Chapter 5

Experimental Results

The objective of our experiments are :

- (a) test efficiency of our LP (i.e. up to what size of networks our LP can handle in a reasonable amount of time)
- (b) compare different logical topologies generated by three heuristic algorithms

We have performed a number of experiments on different networks which are distinct from each other in terms of their physical connectivities, sizes and traffic matrices. We have categorized the physical network in terms of their size as small, medium and large networks shown in Table 5.1

TYPE	Number of Nodes
SMALL	9, 14
MEDIUM	16, 20, 27
LARGE	35, 50

Table 5.1: Small, Medium and Large network

5.1 Methodology

To study the performance of our LP formulation we generated a large number of logical topologies and determined the congestion for each one of them and made a comparative

study of the three heuristics for logical topology given in chapter 4, section 4.1.1. Our hypothesis was that the hybrid topology is better than the GEMNET and that the GEMNET is better than the HLDA topology when the number of nodes is large.

To test this hypothesis we performed paired t-tests and report here the corresponding p-values [30]. In this study we took a number of physical topologies ranging from 9 nodes to 50 nodes. Wherever possible, we have used existing networks as our physical topologies; in the remaining cases we have used synthetic physical topologies with a reasonable number of fibers coming into and out of every end-node. For each physical topology P with n nodes we followed the steps given below.

1. **repeat** steps 2 - 8 for all $i, 1 \leq i \leq k$
2. generate a $n \times n$ traffic matrix T_i , where all non-diagonal elements are determined using a random number generator generating a value between 0 and 1
3. **repeat** steps 4 - 5 for all $j, 1 \leq j \leq 3$
4. For a given physical topology P and traffic matrix T_i , generate logical topology L_{ij} using heuristic j .
5. Run the optimization program (Chapter 4) giving a value of congestions C_{ij}^*
6. $d_i^{1,2} = (C_i^1 - C_i^2)$
7. $d_i^{3,2} = (C_i^3 - C_i^2)$
8. $d_i^{1,3} = (C_i^1 - C_i^3)$
9. Compute the t-statistics and the p-value for the paired t-test comparing the three heuristics.

In this algorithm, k denotes the number of traffic matrices we generated for a given physical topology. For all the cases we used k (sample size) = 30. In step 4, we have used

heuristic j , for all $j, 1 \leq j \leq 3$. Here heuristic 1 (respectively 2 and 3) refer to the heuristic for HLDA (respectively GEMNET and hybrid topology) given in chapter 4. In steps 6, 7, 8 we calculate the proportional difference between the congestions for heuristic m and n , for all m and n . In step 8, we calculate the t -statistics and the p -value for all pairs of heuristics taken from chapter 4, so that $t^{1,2}(P^{1,2})$ is the t -statistics (p -value) for comparing the HLDA heuristic with the heuristic for GEMNET using $d^{1,2}_0, d^{1,2}_1, \dots, d^{1,2}_k$. Similarly $t^{1,3}(P^{1,3})$ and $t^{2,3}(P^{2,3})$ are used to compare the hybrid topology with the GEMNET and the HLDA with the hybrid topology. For all the topologies we used 3 as the degree of the logical topology. When designing hybrid topology we used 2 as the degree for the underlying GEMNET subgraph and 1 for the rest.

5.2 Test Data

The results of the experiments are given below. Detailed of our collected data is given in APPENDIX 1. Column 5 of this table also gives the average percentage improvement of the congestion value (%cv) when we use GEMNET as the topology rather than HLDA.

N	Network	$t^{1,2}$	$p^{1,2}$	% cv	$t^{1,3}$	$p^{1,3}$	% cv	$t^{2,3}$	$p^{2,3}$	% cv	cpu time (s)
9	(A)	-2.022	-0.026	-15	-9.07	-1.85	-8	-1.30	-1.01	-9	.9
14	(B)	1.764	0.044	3.98	-8.75	-0.19	-4.47	2.05	.019	5.69	7
16	(A)	2.403	0.011	8.67	0.111	0.456	-2.06	2.57	.007	8.77	13
20	(C)	4.384	0.00007	15.28	-2.37	-0.012	-9.26	6.34	≈ 0	21.4	37
27	(D)	3.747	0.0003	12.73	0.438	.332	.39	3.573	.0006	11.4	170
35	(D)	2.746	0.005	8.7	1.12	.135	2.2	1.77	.043	4.9	640
50	(A)	1.364	0.091	3.11	0.572	.285	0.32	.686	.249	.32	4071

Table 5.1 : t-test values for our experimental data

In Table 5.1 n stands for the number of nodes of the network and cpu time is the average time to compute the congestion for all the logical topologies generated for this physical topology. The networks we used are as follows :

(A) denotes that this is a synthetic network

(B) denotes that this is the NFS network (EONNET)

(C) denotes that this is the European optical network

(D) denotes that this is a subset of pacific /northwest gigapop network

Following are the two physical topologies (NFSNET and EONNET) that we have used for 14 node and 20 node network.

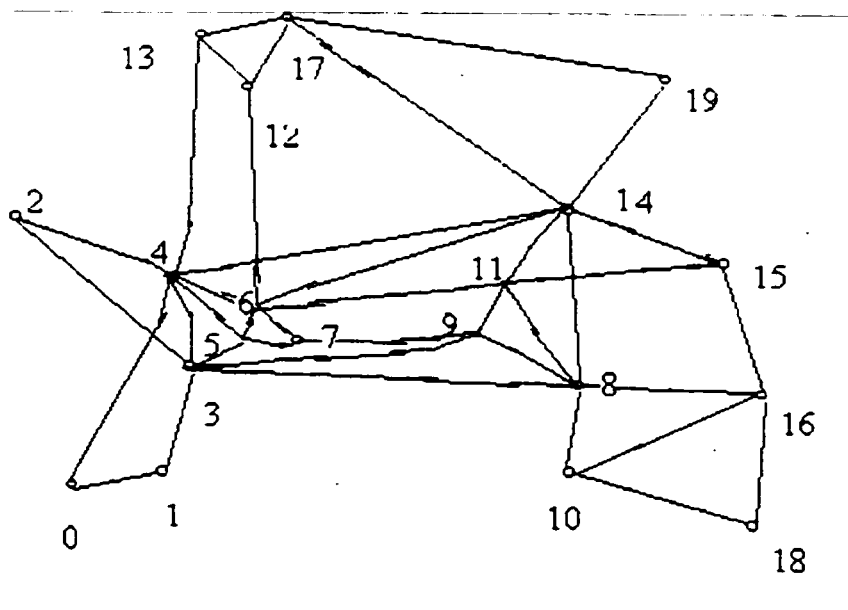


Figure 5.1: 20 node European Optical Network (EONNET)

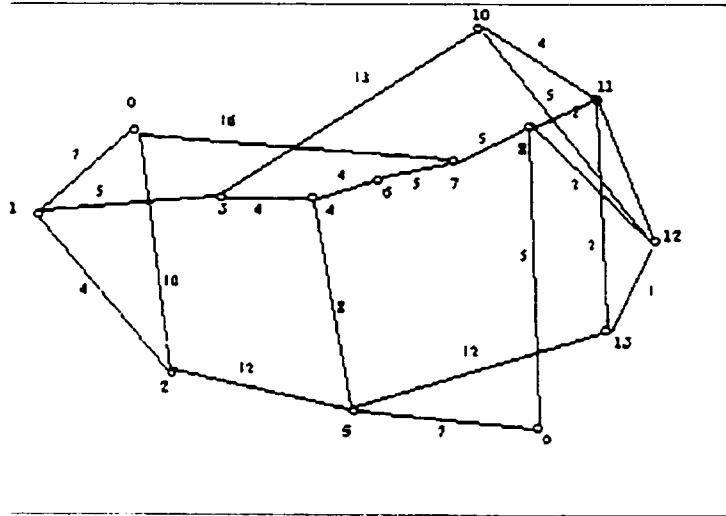


Figure 5.2 : 14 node NFSNET

5.3 Results

These experiments indicate that for larger networks, GEMNET works well compared to both HLDA and hybrid topology. To illustrate this further, column 5 gives the average percentage improvement of the value of congestion when we use the GEMNET rather than HLDA.

By looking at the $P^{1,2}$ values in column 4 we can say that GEMNET performs better than HLDA in moderate sized and large sized network. Average improvement of GEMNET over HLDA ranges from 3% to 15%. It is interesting to see that GEMNET does not perform better than HLDA in small network such as in 9 node network. $P^{1,3}$ values in column 7 does not support any particular behavior of hybrid topology on which we can make any comments. However, by looking at column 8 in table 5.1 we can say that the average improvement of hybrid over HLDA is very low which ranges from 0.32% to 2.2% . According to the Table 5.1 the hybrid topology performs slightly better than HLDA in

moderate and large sized network (in our experiment it performs slightly better in 27, 35 and 50 node networks). From the average improvement values in column 11 we can say that GEMNET performs much better than hybrid where improvement ranges from 0.32% to 21 % and p values in column 10 tells us about the strong statistical confidence of GEMNET over hybrid topology. It is also interesting to see that, hybrid outperforms GEMNET only in 9 node (small node network) network. Column 12 represents the average CPU time taken by our LP program to find the congestion for a particular size network.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

In this thesis we have introduced a novel LP formulation, which exploits the underlying structure of the linear program, for optimally solving the routing problem over a multihop optical network with a specified logical topology. This new method uses a basis size $O(L)$ where L is the number of lightpaths, while conventional techniques use a basis size of $O(n^2)$, where n is the number of nodes in the network. This leads to a dramatic reduction in the number of constraints and thus allows us to handle considerably larger networks than was previously possible. In the literature, optimal solutions to the routing problem for multihop networks have not been reported for networks with more than 14 nodes. As discussed in chapter 4, we are easily able to handle much larger networks

We have also investigated the use of regular logical topologies for implementing wide-area optical networks. It is interesting to note that in all our experiments, except for small networks, the regular topologies performed as well as or better than the irregular topologies, with improvements ranging from 3% to 15%. This suggests that regular logical topologies can be viable candidates for designing such wide-area networks. As shown in Table 5.1 (Chapter 5), there were several cases where the logical topologies produced a substantial improvement.

6.2 Future Work

We need to investigate the effect (if any) of the underlying physical topologies on the congestion of the network. In order to do this we must study the congestion for different physical topologies with same number of nodes and same traffic matrix.

We also need to study how regular and irregular topologies perform under different traffic conditions. From our experiments we see that, in certain cases, regular topologies provided a significant improvement. But in other cases the performance of regular and irregular topologies were comparable.

Further experiments are required to determine the conditions (e.g. uniformly high traffic) under which regular topologies are more suitable candidates.

If the number of wavelengths available on a fiber is limited, it may affect the connectivity of the logical topology. We need to investigate the effect of the number of available wavelengths on topology design.

Finally, in our experiments the performance of the 'hybrid' algorithm was disappointing. We tried to incorporate the advantages of both regular and irregular topologies in the hybrid architecture and expected it to perform better. However, this was not the case. Different ways of generating the hybrid topology is currently under progress.

References

- [1] K. Achan, "Tabu search in multi-hop optical network design", Master's Thesis, University of Windsor, April 2000.

- [2] A. S. Acampora, "A multichannel multihop local lightwave network," Proc IEEE Globecom'87, Nov. 1987

- [3] F. Glover and M. Laguna, "Tabu search", Kluwer Academic Publishers, 1997

- [4] A. S. Acampora, M. Karol, and G. Hluchyj, "Multihop Lightwave Networks: A New Approach to Achieve Terabit Capabilities", IEEE Globecom '88, pp. 1478-84, July 1988.

- [5] F. Ayadi, J. Hayes, and M. Kavehrad, "Bilayered ShuffleNet: A New logical Configuration for Multihop Lightwave Networks." IEEE INFOCOM'93, pp. 1159-1163, Oct. 1993.

- [6] C. A. Bracket, A. S. Acampora, J. Sweitzer, G. Tangory, M. T. Smith, W. Lennon, K.-C. Wang, and R. H. Hobbs, "A scalable multiwavelength multihop optical network: A proposal for research on all-optical networks." IEEE/OSA Journal of Lightwave Technology, vol. 11, pp. 736-753, May/June 1993.

- [7] S. Banerjee and C. Chen, "Design of Wavelength-Routed Optical Networks for Circuit Switched Traffic," Proc. IEEE Globecom'96, London, Nov. 1996.

- [8] Rajiv Ramaswami, Kumar N. Sivarajan, " Routing and Wavelength assignment in all-optical networks", IEEE/ACM Trans. Networking, Vol. 3 (1995) 489-500.

- [9] I. Chlamtac, A. Ganz, and G. Karmi, "Lightnets: Topologies for high speed optical networks," IEEE/OSA Journal of Lightwave Technology, vol. 11, pp. 951-961, May/June 1993.

- [10] P. Cochrane, "Optical Network Technology", Chapman & Hall, 1995

- [11] Ori Gerstel, "On the future of wavelength routing networks", IBM T.J Watson research center, NY, USA, 1996

- [12] O. Gerstel, P. E. Green and R. Ramaswami, "Architecture for an optical network layer", Working Paper, IBM T.J Watson research center, NY, USA

- [13] P. E. Green, "Fiber Optic Networks", Prentice Hall, Englewood Cliffs, NJ 1992

- [14] Anwar Haque, A. Jackel, S Bandyopadhyay, and A. Sengupta, "Efficient Heuristic for Designing the Topology of Multi-hop Optical Networks," Photonics-2000, Dec. 18-20, in Calcutta, India.

- [15] J. F. P. Labourdette and A. S. Acampora, "Logically rearrangeable multihop lightwave networks," *IEEE Transactions on Communications*, vol. 39, pp. 1223-1230, Aug. 1991
- [16] N F Maxemchuk, "Routing in Manhattan street network", *IEEE transactions on communications*, vol. 35, pp. 503-512, May 1987.
- [17] B. Mukherjee, D. Banerjee, S. Ramamurthy and A. Mukherjee, "Some principles for designing a wide-area WDM optical network," *IEEE/ACM Transactions on Networking*, vol. 4, pp. 684-96, Oct. 1996
- [18] V. Chvatal, "Linear Programming". W.H. Freeman and Co. 1983
- [19] Y.P. Aneja and K.P.K. Nair, "Bicriteria transportation problem," *Management Science*, Vol.25 (1979) 61-71.
- [20] B. Mukherjee, "WDM-based local lightwave networks part I: Single hop systems," *IEEE Network*, vol. 6, pp. 12-27, May 1992
- [21] B. Mukherjee, "WDM-based local lightwave networks part II: Multihop systems," *IEEE Network*, vol. 6, pp. 20-32, July 1992

- [22] Rajesh M. K, "Algorithms for routing, wavelength assignment and topology design in optical networks", Ph.D. Thesis, Department of Electrical and Communication Engineering, Indian Institute of Science, Bangalore, India Nov. 1998
- [23] R. Ramaswami, "Multiwavelength optical network architecture", Tutorial, OFC 1996
- [24] Rajiv Ramaswami, Kumar N. Sivarajan, "Design of logical topologies for wavelength routed optical networks". IEEE Journal on Selected Areas in Communications, vol. 14, No.5, June 1996
- [25] K. Sivarajan and R. Ramaswami. "Lightwave Networks Based on de Bruijn Graphs." IEEE/ACM Transactions on Networking, vol. 2, No 1, pp. 70-79, Feb. 1994
- [26] K. Sivarajan and R. Ramaswami. "Multihop Networks Based on de Bruijn Graphs". IEEE INFOCOM '92, pp 1001-1011, Apr 1991
- [27] A. Venkateswaran and A. Sengupta. "On a scalable topology for Lightwave networks". Proc IEEE INFOCOM'96, 1996.
- [28] Z. Zhang and A. S. Acampora. "A heuristic wavelength assignment algorithm for multihop WDM networks with wavelength routing and wavelength reuse." IEEE / ACM Transactions on Networking, vol. 3, pp. 281-288, Jun 1995

- [29] J. Innes, S. Banerjee and B. Mukherjee, "GEMNET: A generalized shuffle exchange based regular, scalable and modular multihop network based on WDM lightwave technology", *IEEE/ACM Trans. Networking*, vol. 3, No. 4, Aug. 1995
- [30] Robert V. Hogg and Allan T. Craig, "Introduction to Mathematical Statistics", Macmillan 1978.
- [31] Biswanath Mukherjee, "Optical Communication Networks". McGraw Hill, 1997
- [32] Y. P. Aneja, "Routing in Wavelength Routed Optical Networks". 2001 IEEE Workshop on High Performance Switching and Routing, May 29-31, Dallas, Texas, USA

Appendix 1

Experimental Data

NODE	HLDA	GEMNET	HYBRID
9	3.29	3.01	3.6
	2.76	2.72	2.83
	3.09	2.65	2.08
	2.56	3.55	2.48
	2.32	3.92	1.93
	1.93	2.52	2.96
	2.06	2.29	2.18
	2.56	4.06	3.6
	2.97	3.39	3.03
	2.52	2.53	3.37
	4.09	2.49	3.04
	2.06	2.49	3.38
	2.97	4.36	3.26
	2.48	2.83	3.23
	2.35	2.76	2.99
	3.48	3.85	3.4
	3.42	3.76	3.42
	1.98	3.04	1.64
	3.72	3.29	3.18
	1.81	2.81	2.55
	3.01	2.56	3.92
	2.04	2.93	3.23
	2.69	2.86	2.74
	3.12	3.34	2.46
	4.09	2.75	3.01
	2.41	2.35	2.34
	2.22	2.54	2.32
	2.39	4.03	3.44
	3.77	2.94	2.81
	2.98	3.26	2.29

NODE	HLDA	GEMNET	HYBRID
14	5.86	5.34	5.95
	5.52	5.01	5.51
	6.55	5.76	6.96
	6.92	6.13	4.98
	5.24	5.78	5.42
	5.24	5.78	5.42
	5.28	6.09	9.24
	5.11	5.38	6.29
	4.47	5.93	6.29
	5.56	5.88	5.57
	5.21	5.1	4.82
	5.51	4.95	6.88
	5.32	5.59	5.58
	5.94	6.2	5.28
	5.67	5.23	5.65
	6.31	5.89	5.68
	5.43	4.37	4.87
	5.62	4.92	5.51
	5.56	4.13	6.61
	5.08	4.05	7.21
	5.82	5.81	6.05
	5.54	5.59	5.94
	5.33	5.37	5.66
	6.17	4.64	6.1
	4.95	5.79	5.54
	6.16	5.59	4.49
	5.76	5.57	5.36
	6.52	7.34	5.56
	6.05	4.28	4.67
	5.44	3.98	5.59

NODE	HLDA	GEMNET	HYBRID
16	7.41	7.13	7.31
	6.27	5.93	7.51
	7.93	6.16	6.13
	6.93	6.69	5.67
	5.57	4.63	5.52
	7.67	6.57	6.89
	3.9	3.01	4.12
	7.72	5.52	5.88
	7.57	5.54	6.52
	8.56	6.07	6.7
	6.16	10	8.49
	5.65	6.02	6.54
	5.94	5.53	6.66
	6.26	4.32	6.89
	6.34	5.18	7.37
	7.11	4.94	5.6
	5.6	5.91	7.92
	5.38	5.61	8.39
	7.21	4.7	6.54
	6.29	7.92	5.74
	4.96	4.42	7.62
	8.53	6.27	6.37
	7.34	4.87	6.01
	6.15	7.26	5.3
	7.61	5.52	6.79
	5.64	7.82	6.84
	6.16	6.23	5.16
	6.14	5.74	6.79
	6.44	6.49	6.32
	6.28	4.8	6.26

NODE	HLDA	GEMNET	HYBRID
20	7.32	6.44	7.54
	7.88	6.36	7.57
	6.9	6.6.8	8.96
	7.44	6.91	8.45
	9.36	7.57	10.51
	6.6	7.27	6.2
	8.6	7.08	9.14
	9.36	5.29	7.04
	7.73	5.66	8.86
	8.34	6.84	7.4
	7.89	6.73	7.25
	6.82	6.48	7.99
	7.06	6.81	8.87
	9.84	7.75	8.98
	6.9	5.26	7.13
	8.24	6.91	8.55
	9.1	6.72	9.65
	6.71	6.78	9.24
	7.14	7.26	7.75
	7.43	5.02	9.01
	7.91	6.16	8.48
	7.2	6.4	6.93
	7.39	6.2	7
	6.9	5.87	8.72
	7.72	6.98	8.27
	5.91	5.47	8.02
	7.57	6.54	6.84
	6.92	6.33	7.59
	5.94	6.88	7.86
	7.09	8.83	9.81

NODE	HLDA	GEMNET	HYBRID
27	11.54	9.42	13.44
	10.45	6.23	12.63
	12.32	12.64	13.48
	15.43	8.03	11.36
	12.21	10.83	13.09
	10.04	10.84	11.59
	12.11	9.77	10.62
	10.4	12.03	9.4
	11.27	10.21	10.45
	9.04	7.32	10.72
	10.14	6.22	9.71
	11.71	6.86	9.34
	11.38	7.24	9.21
	11.74	8.72	9.21
	8.89	7.26	8.8
	9.08	6.97	8.63
	8.94	6.41	9.93
	10.48	8.13	7.04
	10.51	11.14	8.54
	7.53	9.84	8.3
	9.25	7.56	8.58
	9.04	8.38	7.5
	8.5	9.51	9.08
	8.75	8.56	10.12
	10.26	8.63	10.32
	9.5	9.81	9.99
	8.01	9.75	9.32
	8.66	8.29	9.51
	12.69	9.19	13.12
	11.46	10.76	14.18

NODE	HLDA	GEMNET	HYBRID
35	15.56	12.08	14.45
	11.7	8.57	13.6
	9.64	7.69	9.6
	10.29	9.4	7.82
	11.78	6.95	7.96
	9.64	8.87	10.48
	8.44	7.23	7.79
	8.28	8.6	9.56
	8.22	7.57	10.3
	8.48	7.19	9.08
	11.44	9.44	8.23
	9.4	8.84	10.67
	9.22	10.52	8.75
	8.78	12.34	8.61
	10.82	9.32	8.35
	7.8	7.56	8.44
	10.33	10.15	10.77
	8.39	11.08	10.76
	11.64	8.56	12.16
	10.89	7.72	11.6
	9.05	7.98	12.02
	10.74	8.94	10.99
	10.58	9.07	8.63
	10.42	11.21	8.1
	9.84	7.74	9.18
	12.89	9.2	9.52
	13.23	7.37	10.13
	9.18	9.22	7.58
	9.01	12.32	9.25
	9.17	9.43	9.24

NODE	HLDA	GEMENT	HYBRID
50	9.53	11.42	8.66
	11.93	8.47	15.79
	11.09	10.39	10.05
	13.59	9.74	8.15
	13.44	11.68	9.99
	7.27	9.52	10.88
	10.59	8.98	8.94
	9.09	8.34	10.97
	10.84	6.79	8
	11.54	8.36	11.02
	11.19	8.47	8.83
	7.97	11.2	11.43
	8.2	13.77	10.26
	11.32	10.06	8.11
	8.06	10.66	8.86
	12.32	10.92	11.34
	9.76	11.56	9.24
	11.19	6.89	11.21
	12.05	12.91	8.48
	10.88	9.13	8.7
	10.62	10.97	10.07
	9.57	10.08	9.52
	10.13	8.82	9.15
	9.81	11.96	10.48
	10.48	10.5	11.1
	10.24	7.72	9.14
	13.55	13.11	11.74
	8.76	8.52	10.61
	12.82	9.57	16.85
	10.19	9.58	12.88

Vita Auctoris

Anwar Haque was born on Dec. 7, 1974 in Mymensingh – Bangladesh. He obtained his BS degree in Computer Science from North South University, Bangladesh in 1997. Besides, He received Diploma in Computer Studies from The National Computing Center, UK in 1993. Anwar is currently a candidate for the Master's degree in Computer Science at the University of Windsor, Canada. He is planning to pursue his Ph.D degree in Computer Science at University of Waterloo, Canada where he has been given admission. His research interests include topology design and routing issues in optical networks, topics in network management, performance evaluation and in mobile computing.