NOTE TO USERS

This reproduction is the best copy available.

UMI
Heuristic Methods for Backup Lightpath Routing in WDM Networks

By

Pravesh Mahtani

A thesis submitted in conformity with the requirements for the degree of Masters of Applied Science and Engineering Graduate Department of Electrical and Computer Engineering University of Toronto

© Copyright by Pravesh Mahtani (2001)
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.
Heuristic Methods for Backup Lightpath Routing in WDM Networks
Pravesh Mahtani
Masters of Applied Science and Engineering, 2001
Department of Electrical and Computer Engineering
University of Toronto

Abstract

Reliability is of prime importance in future WDM networks. The work presented here focuses on path protection in mesh-based WDM networks. Since backup lightpaths consume resources, an effort is made to reduce the resource burden imposed on the network. The idea of resource sharing for backup lightpaths is investigated to meet this end. Online, distributed schemes for path protection are discussed and compared from the standpoint of resource consumption. Two new backup routing schemes are introduced in this work. It is shown that these new backup routing algorithms (Educated_DFS and Unrestricted_TH) perform dramatically better than other conventional routing methods. This is proven through a series of simulations using networks of different topologies and a variety of network loadings. Two additional problems are investigated:

1. The effect of introducing multiple reliability service classes into the network; and

2. The issue of pre-allocating port mappings to backup lightpaths.
Acknowledgements

I would like to thank Professor Leon-Garcia for his invaluable guidance, help and advice throughout the duration of my research.

I would also like to express my gratitude to my peers in the Network Architecture Labs at the University of Toronto, for their comments and insight into my work.

Special Thanks goes to Linda Espeut for administrative support.

Finally thanks to NSERC and the Communication Group at the University of Toronto for Financial Support.
Table of Contents

List of Figures .............................................................................................................................. vi
List of Graphs .................................................................................................................................. vii
List of Tables ..................................................................................................................................... xi
List of Terms and Acronyms .......................................................................................................... xii

Chapter 1
Introduction ......................................................................................................................................... 1
1.1 Motivation .................................................................................................................................... 1
1.2 Network Model .......................................................................................................................... 3
1.3 Working Towards a Dynamic WDM Network ............................................................................. 5
1.4 Protecting Against Network Failure .......................................................................................... 8
1.5 Shared Path Protection in WDM Networks ............................................................................... 9
1.6 Online Lightpath Routing – The Optimal Solution ................................................................ 10
Summary ........................................................................................................................................ 11

Chapter 2
Principles and Assumptions .......................................................................................................... 13
2.1 Working Assumptions ................................................................................................................ 13
2.2 Fundamental Principles ............................................................................................................. 16
  2.2.1 Routing Decision .................................................................................................................. 16
  2.2.2 Resource Reservation ........................................................................................................... 16
Summary ........................................................................................................................................ 19

Chapter 3
Heuristic Backup Lightpath Routing Solutions ........................................................................... 20
3.1 Conventional Schemes ............................................................................................................... 20
  3.1.1 Shortest Path First Routing (Scheme 1) .............................................................................. 20
  3.1.2 Joint Routing (Scheme 2) ................................................................................................... 22
3.2 Intelligent routing – Finding the Lowest Cost backup path ..................................................... 25
  3.2.1 Educated Depth First Search (Scheme 3) ........................................................................... 26
  3.2.2 Avoiding Troublesome Hops (Scheme 4) .......................................................................... 31
  3.2.3 A Restricted Troublesome Hop Version (Scheme 5) ................................................................. 33
Summary ........................................................................................................................................ 34
List of Figures

Figure 1.1: Envisioned WDM Backbone ......................................................................................... 3
Figure 1.2: Wavelengths in a Fiber ................................................................................................. 4
Figure 1.3: Optical Cross-Connect Functionality ........................................................................... 4
Figure 1.4: Link State Packets Distributed Amongst OXCs .......................................................... 7
Figure 1.5: Signaling Protocol to Establish a Connection .............................................................. 7
Figure 1.6: Link Protection ............................................................................................................. 9
Figure 1.7: Path Protection ............................................................................................................ 9
Figure 1.8: Protecting Lightpaths (No Backup Sharing) .............................................................. 10
Figure 1.9: Protecting Lightpaths (Exploiting Backup Sharing Possibilities) ......................... 10
Figure 2.1: Illustrating the Concept of Shared Risk Link Groups (two logically separate links which are at risk to the same physical failure) ....................................................................... 14
Figure 2.2: An OXC With Five Links ........................................................................................... 17
Figure 2.3: Port Mappings for the OXC of fig 2.2 ........................................................................ 18
Figure 2.4: Computing Required Protection Bandwidth on a Link ............................................. 19
Figure 3.1: Illustration of the Educated Depth First Search Scheme ........................................ 27
Figure 3.2: Utility of the Shared Protection Lambda Link-State Metric ....................................... 28
Figure 3.3: Sample Protection Table for one of an OXC's links .................................................. 31
Figure 4.1: Network A, Mesh-Torus ............................................................................................ 36
Figure 4.2: Network B .................................................................................................................. 36
Figure 4.3: Network C (UUNET - N. America) ............................................................................ 37
Figure 4.4: Sample Protection Table Including a Mixture of LF and LNF connections ........... 69
Figure 4.5: Difficulty Routing an LNF Connection - Shortest Path Between S and D Depicted by Dotted Line ......................................................................................................................... 71
Figure 4.6: Scenario 7; Percentage of Connections Refused, as a Function of Network Load – ALL SILVER vs MIXED Service Class Distribution ............................................................................. 77
Figure 5.1: (a) Five Backup Lightpaths Sharing 3 Protection Lambdas on a Link (b) Individual Lightpaths Multiplexed Onto Specific Lambdas ........................................................................ 84
Figure 5.2: Pathological Case – More Protection Lambdas Required Over Link A-B When Pre- assigning Lambdas to Lightpaths ........................................................................................... 84
Figure 5.3: Modeling Lambda-to-Lightpath Mapping as a Graph Coloring Problem ................... 85
List of Graphs

Graph 4.1: Links per Node Distribution (Network A) ................................................................. 36
Graph 4.2: Links per Node Distribution (Network B) ................................................................. 36
Graph 4.3: Links per Node Distribution (Network C) ................................................................. 37
Graph 4.4: Scenario 1; Percentage of Network Resources Occupied by Backup and Primary
Lightpaths on Network A - A Comparison Between Schemes ................................................. 42
Graph 4.5: Scenario 1; Average Number of Hops Traversed by a Backup Lightpath on Network A
- A Comparison Between Schemes .............................................................................................. 42
Graph 4.6: Scenario 2; Ratio of Total Number of Backup Resources Used to Total Number of
Primary Resources Used, as a Function of Offered Load - A Comparison Between
Schemes ........................................................................................................................................ 47
Graph 4.7: Scenario 2; Average Resource Usage (primary and backup) per Connection, as a
Function of Offered Load - A Comparison Between Schemes ...................................................... 48
Graph 4.8: Scenario 2; Average Resource Usage per Primary Lightpath, as a Function of Offered
Load - A Comparison Between Schemes ....................................................................................... 48
Graph 4.9: Scenario 2; Average Number of Additional Protection Lambdas Required per Backup
Lightpath, as a Function of Offered Load - A Comparison Between Schemes ......................... 49
Graph 4.10: Scenario 2; Average Number of Hops Traversed by A Backup Lightpath, as a
Function of Offered Load - A Between Schemes ........................................................................... 49
Graph 4.11: Scenario 2; Percentage of Connection Requests Refused, as a Function of Offered
Load - A Comparison Between Schemes ....................................................................................... 50
Graph 4.12: Scenario 3; Ratio of Total Number of Backup Resources Used to Total Number of
Primary Resources Used, as a Function of Offered Load - A Comparison Between
Schemes ........................................................................................................................................ 53
Graph 4.13: Scenario 3; Average Resource Usage (primary and backup) per Connection, as a
Function of Offered Load - A Comparison Between Schemes ...................................................... 54
Graph 4.14: Scenario 3; Average Resource Usage per Primary Lightpath, as a Function of
Offered Load - A Comparison Between Schemes ........................................................................... 54
Graph 4.15: Scenario 3; Average Number of Additional Protection Lambdas Required per Backup
Lightpath, as a Function of Offered Load - A Comparison Between Schemes ........................... 55
Graph 4.16: Scenario 3; Percentage of Connection Requests Refused, as a Function of Offered
Load - A Comparison Between Schemes ....................................................................................... 55
Graph 4.17: Scenario 2; Average Number of Admission Control Failures Before Connection
Success, as a Function of Offered Load - A Comparison Between Schemes ............................. 58
Graph 4.18: Scenario 3; Average Number of Admission Control Failures Before Connection
Success, as a Function of Offered Load - A Comparison Between Schemes ............................. 58
Graph 4.19: Scenario 2; Utility of the Shared Protection Bandwidth Link Metric – Comparing Educated_DFS With and Without the Metric

Graph 4.20: Scenario 4; Ratio of Total Backup Resources Used to Total Primary Resources Used, as a Function of Offered Load - No Change in Link Bandwidth

Graph 4.21: Scenario 4; Ratio of Total Backup Resources Used to Total Primary Resources Used, as a Function of Offered Load - Link Bandwidth Multiplied by 2

Graph 4.22: Scenario 4; Ratio of Total Backup Resources Used to Total Primary Resources Used, as a Function of Offered Load - Link Bandwidth Multiplied by 5

Graph 4.23: Scenario 5; Ratio of Total Backup Resources Used to Total Primary Resources Used, as a Function of Offered Load – A Comparison Between Schemes

Graph 4.24: Scenario 5; Average Resource (Primary + Backup) Usage Per Connection, As a Function of Offered Load – A Comparison Between Schemes

Graph 4.25: Scenario 5; Average Resource Usage per Primary Lightpath, as a Function of Offered Load – A Comparison Between Schemes

Graph 4.26: Scenario 5; Average Number of Additional Backup Lambdas Reserved per Backup Lightpath, as a Function of Offered Load – A Comparison Between Schemes

Graph 4.27: Scenario 5; Percentage of Connection Requests Refused, as a Function of Offered Load – A Comparison Between Schemes

Graph 4.28: Scenario 5; Inspecting Filled Links, As a Function of Offered Load – SPF Routing

Graph 4.29: Scenario 5; Inspecting Filled Links, As a Function of Offered Load – Joint Routing

Graph 4.30: Scenario 5; Inspecting Filled Links, As a Function of Offered Load – Educated_DFS

Graph 4.31: Scenario 5; Inspecting Filled Links, As a Function of Offered Load – Unrestricted_TH

Graph 4.32: Scenario 5; Inspecting Filled Links, As a Function of Offered Load – Restricted_TH

Graph 4.33: Scenario 6; Average Resource Usage per Connection, as a function of LNF Request Type – Moderate Offered Load

Graph 4.34: Scenario 6; Average Resource Usage per Connection, as a function of LNF Request Type – High Offered Load

Graph 4.35: Scenario 6; Average Resource Usage per Primary Lightpath, as a Function of LNF Request Type – Moderate Offered Load

Graph 4.36: Scenario 6; Average Resource Usage per Primary Lightpath, as a Function of LNF Request Type – High Offered Load

Graph 4.37: Scenario 6; Average Additional Resources Required per Backup Lightpath, as a Function of LNF Request Type – Moderate Offered Load
Graph 4.38: Scenario 6; Average Additional Resources Required per Backup Lightpath, as a Function of LNF Request Type – High Offered Load .......................................................................................... 73

Graph 4.39: Scenario 6; Ratio of Number of Links Whose Worst Case Failure is a Link to Number of Links Whose Worst Case Failure is a Node, as a Function of LNF Type Requests – High Offered Load .......................................................................................... 73

Graph 4.40: Scenario 6; Ratio of Number of Links Whose Worst Case Failure is a Link to Number of Links Whose Worst Case Failure is a Node, as a Function of LNF Type Requests – Moderate Offered Load .................................................................................. 73

Graph 4.41: Scenario 6; Percentage of Connection Requests Refused, as a Function of LNF Type Requests – High Offered Load .......................................................................................... 74

Graph 4.42: Scenario 6; Percentage of Connection Requests Refused, as a Function of LNF Type Requests – Moderate Offered Load .................................................................................. 74

Graph 4.43: Scenario 6; Percentage of Connection Requests Refused, as a Function of Moderate Offered Load – ALL SILVER vs MIXED Service Class Distribution .................................................. 77

Graph 4.44: Scenario 6; Percentage of Connection Requests Refused, as a Function of High Offered Load – ALL SILVER vs MIXED Service Class Distribution .................................................. 77

Graph 4.45: Scenario 6; Percentage of Connection Requests Refused, as a Function of the Proportion of Type GOLD Requests – SPF Routing .................................................................................. 80

Graph 4.46: Scenario 6; Percentage of Connection Requests Refused, as a Function of the Proportion of Type Gold Requests – SPF Routing .................................................................................. 80

Graph 4.47: Scenario 6; Percentage of Connection Requests Refused, as a Function of the Proportion of Type GOLD Requests – Joint Routing .................................................................................. 81

Graph 4.48: Scenario 6; Percentage of Connection Requests Refused, as a Function of the Proportion of Type Gold Requests – Joint Routing .................................................................................. 81

Graph 4.49: Scenario 6; Percentage of Connection Requests Refused, as a Function of the Proportion of Type GOLD Requests – Educated-DFS .................................................................................. 81

Graph 4.50: Scenario 6; Percentage of Connection Requests Refused, as a Function of the Proportion of Type Gold Requests – Educated-DFS .................................................................................. 81

Graph 4.51: Scenario 6; Percentage of Connection Requests Refused, as a Function of the Proportion of Type GOLD Requests – Unrestricted_Tc .................................................................................. 82

Graph 4.52: Scenario 6; Percentage of Connection Requests Refused, as a Function of the Proportion of Type Gold Requests – Unrestricted_Tc .................................................................................. 82

Graph 4.53: Scenario 6; Percentage of Connection Requests Refused, as a Function of the Proportion of Type GOLD Requests – Unrestricted_Tc .................................................................................. 82

Graph 4.54: Scenario 6; Percentage of Connection Requests Refused, as a Function of the Proportion of Type Gold Requests – Unrestricted_Tc .................................................................................. 82

Graph 4.55: Scenario 6; Percentage of Connection Requests Refused, as a Function of the Proportion of Type GOLD Requests – Restricted_Tc .................................................................................. 82

Graph 4.56: Scenario 6; Percentage of Connection Requests Refused, as a Function of the Proportion of Type Gold Requests – Restricted_Tc .................................................................................. 82
Graph D.1: Scenario 2; Ratio of Total Number of Backup Resources Used to Total Number of Primary Resources Used, as a Function of Offered Load – A Comparison Between Troublesome Hop Schemes
List of Tables

Table 3.1: Summary of Scheme Attributes ................................................................. 34
Table 4.1: Network A Characteristics ........................................................................ 36
Table 4.2: Network B Characteristics ......................................................................... 36
Table 4.3: Network C Characteristics ........................................................................ 37
Table 4.4: Scenario 2; Inspection of Filled Links in Network B – Unrestricted_TH ....... 45
Table 5.1: Determining The Resource Penalty of Pre-assigning Lambdas to Lightpaths. ... 88
# List of Terms and Acronyms

<table>
<thead>
<tr>
<th>Term/Abbreviation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>CR-LDP</td>
<td>Constraint-based Label Distribution Protocol</td>
</tr>
<tr>
<td>Educated DFS</td>
<td>Educated Depth First Search</td>
</tr>
<tr>
<td>IETF</td>
<td>Internet Engineering Task Force</td>
</tr>
<tr>
<td>IGP</td>
<td>Interior Gateway Protocol</td>
</tr>
<tr>
<td>IS-IS</td>
<td>Intermediate System to Intermediate System</td>
</tr>
<tr>
<td>LF</td>
<td>Connections demanding protection from link failures</td>
</tr>
<tr>
<td>LNF</td>
<td>Connections demanding protection from link and node failures</td>
</tr>
<tr>
<td>MP(Lambda)S</td>
<td>Multiprotocol Lambda Switching</td>
</tr>
<tr>
<td>MPLS</td>
<td>Multiprotocol Label Switching</td>
</tr>
<tr>
<td>OIF</td>
<td>Optical Internetworking Forum</td>
</tr>
<tr>
<td>OSPF</td>
<td>Open Shortest Path First</td>
</tr>
<tr>
<td>OXC</td>
<td>Optical Cross Connect</td>
</tr>
<tr>
<td>Restricted TC</td>
<td>Restricted Troublesome Component</td>
</tr>
<tr>
<td>Restricted TH</td>
<td>Restricted Troublesome Hop</td>
</tr>
<tr>
<td>RSVP</td>
<td>Resource Reservation Protocol</td>
</tr>
<tr>
<td>SRLG</td>
<td>Shared Risk Link Group</td>
</tr>
<tr>
<td>Unrestricted TC</td>
<td>Unrestricted Troublesome Component</td>
</tr>
<tr>
<td>Unrestricted TH</td>
<td>Unrestricted Troublesome Hop</td>
</tr>
<tr>
<td>Worst-Case Failure</td>
<td>A link’s worst-case failure is the hop or node whose fault results in the highest number of backup lightpath activations over that link.</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

This chapter will introduce the problem studied, and provide a roadmap for the rest of the thesis.

1.1 Motivation

The recent explosion in demand for bandwidth has telecommunication providers rushing to find ways to increase the size and capacity of their networks. Fortunately, the advent of wavelength division multiplexing (WDM) is helping satisfy the ever-increasing customer appetite. WDM partitions the enormous bandwidth of an optical fiber into separate and discrete wavelengths (or lambdas), each capable of carrying gigabits of data per second. Recently, there has also been technological improvement in the area of WDM networking, in which optical signals are transported across networks based solely on their wavelength. Now, any two nodes on a backbone can be connected by pipes of uniform bandwidth granularity, which span several links and are routed by optical cross-connects. These pipes, also called lightpaths, act as tunnels shuttling a myriad of digital payload signals from source to destination.

However, with more traffic flowing through an optical fiber, the impact of network failures becomes more severe. Cable cuts or component breakdowns can disrupt service for many customers over extended periods of time if no survivability mechanisms are in place. This underscores the need for fault tolerance in WDM networks. Furthermore, the emergence of mission-critical and real-time connections dictates the need for failure recovery strategies that can provide high availability guarantees and bounded restoration times for customers.

With this in mind, it is not surprising that many lightpath connections will have to come in pairs: one primary lightpath, which is normally operational and delivers the data between two endpoints, and a secondary lightpath, waiting on stand-by, ready to be activated if
the primary fails. Unfortunately, these secondary lightpaths consume resources that could otherwise be used to support new and additional connections. Therefore, it is essential that the resources required for protection purposes be kept as small as possible.

One way to help achieve the above stated goal is to permit multiple protection lightpaths to share resources (bandwidth, network components) amongst each other. The premise behind this solution is that network failures tend to occur independently and with low probability. Therefore, primary lightpaths that do not use the same network components are especially unlikely to fail at the same time. If a group of primary lightpaths never fail simultaneously, then their corresponding backup paths are never active at the same time. Thus, it makes sense to allow such backup lightpaths to share resources wherever possible.

Considerable research [1] [2] has been conducted for connection-oriented networks with the aim of developing algorithms that minimize the amount of network resources used by maximizing the degree of sharing amongst backup paths. However, the incumbent solutions involve complex optimization techniques that are only suitable in static networks. For dynamic networks, where new lightpath connections are being continuously setup and old ones torn down, these kinds of approaches are impractical. There is no alternative but to use online algorithms for establishing primary and secondary lightpath connections. Hence, the intent of this research was to analyze different online routing schemes for backup lightpaths with the objective of resource conservation.

The rest of this work is organized as follows: The remainder of this chapter is a discourse into the framework and context to which this research is relevant. Chapter 2 states assumptions and fundamental principles used in developing the ideas presented in the Chapter 3. Chapter 3 is a presentation of the backup lightpath routing schemes that were devised in this work. Chapter 4 provides the setting for the simulation environment that the schemes were compared upon. The results of the simulations are also provided, together with an in-depth analysis. Chapter 5 is focused on the issue of pre-allocating resources to individual lightpaths. Finally, Section 6 is a list of conclusions that can be derived from this work.
1.2 Network Model

This body of work is relevant to large-scale mesh-based WDM networks. Typically, these are nationwide backbones which act as ‘carriers’ for smaller network providers (see fig 1.1). Like any network, the WDM backbone, consists of nodes and links. In this case, the nodes are optical cross-connects (abbreviated as OXC hereafter) and the links are actually cables connecting two OXCs. Each cable houses a group of fibers, and each fiber in turn can support a given number of wavelengths (fig 1.2). The optical cross-connect is a complicated device (see [3] for a thorough description), but its function is straightforward. It is a black box that can switch a signal of wavelength \( \text{lambda}_\text{in} \) incoming on fiber port, \( \text{fiber}_\text{in} \), to output port \( \text{fiber}_\text{out} \) with outgoing wavelength \( \text{lambda}_\text{out} \) (fig 1.3).

Figure 1.1: Envisioned WDM Backbone
In practice, OXCs are not always so flexible. Current limitations in wavelength conversion and optical switch fabrics may preclude the mapping of any incoming <fiber,lambda> pair to any outgoing <fiber,lambda> pair. Such liabilities have been ignored in this work. A lightpath in the network can be uniquely identified by its route (i.e. the path it traverses), the particular fiber it resides on each link, and its wavelength on each fiber. Of course, no two operational lightpaths may share the same <fiber,lambda> pair on any link.

The view of the optical network presented above is a bit too simplistic. Other components such as optical amplifiers, regenerators, filters have not been alluded to. But they are not important to this discussion.

Under the envisioned framework, client devices (IP routers, ATM switches, etc.) outside of the network make requests to edge OXCs for connections to other client peers across the backbone. Based on these requests, an edge OXC may provision a lightpath
between itself and another edge node. Clients may also demand a certain degree of reliability for their connections. This will directly influence the kind of protection given to a lightpath. For instance, connections that constitute part of a virtual private network may require very high availability (up time). Lightpaths hosting mission critical services may additionally insist on extremely short and bounded recovery times. Most likely, a network provider will offer a range of protection options, and leave the choice to the clients [4].

Installing an optical connection in today's networks tends to be a long and arduous process [5]. Administrative issues aside, it is a difficult task because connections are normally placed with global optimality in mind. For example in a WDM network, a new lightpath may be called for in response to an updated traffic matrix. Often, this will engender a reevaluation of existing routes for the entire set of connections already within the network. Complex and time-consuming optimization algorithms are run offline to determine the best route and wavelength assignment for all lightpaths as a whole. Likely, the prevailing placement of lightpaths will be revamped to support the new connection. This is not a serious issue in static networks, since traffic patterns remain relatively stable and therefore changes are a rarity. But, for the dynamic networks of the future, customer demands change rapidly. Thus, lightpaths have to be provisioned or removed on the fly. As a result, online, heuristic lightpath establishment procedures are needed, whether or not they produce optimal solutions [6].

1.3 Working Towards a Dynamic WDM Network

In hopes of bringing expeditious lightpath provisioning to WDM networks, there has been a recent effort to introduce intelligence into optical networks [7]. In this regard, OXCs are autonomous devices, making independent decisions based on information communication to them by their peers. This view diminishes the role of the central network manager, which is vitally important in traditional optical networks. The work is now distributed amongst the OXCs; and this, in principle, should help in avoiding many of the bottlenecks associated with centralized systems [8]. The Internet Protocol (IP) is a distributed protocol that has proven to work in enhancing the scalability and flexibility of packet-switched networks. Therefore, standards bodies (IETF, OIF), are working
towards incorporating the IP control plane into the switched optical network model, rather than reinventing a new layer 3 protocol [9] [10].

In this new framework, OXCs will distribute link state information around the network using an Interior Gateway Protocol (IGP) such as Open Shortest Path First (OSPF) or Intermediate System-Intermediate System (IS-IS) [11]. Then OXCs can route and establish lightpaths themselves, based on their perception of network state, learned via link state packets (fig 1.4). Whereas conventionally, the intent of IGPs has been to convey knowledge of link connectivity, OXCs will now need to be aware of a host of other things (such as which links have free bandwidth available), if they are to have the capability of instantiating connections. This will require enhancements to current IGPs. This and other issues are already being addressed in the work being done in Multiprotocol Label Switching (MPLS) in the context of packet switched networks [12]. In fact, recently, a subset of MPLS (sometimes called MPL(ambda)S) has been borne out of industry standards groups with the objective of extending MPLS in order to capture the particularities and peculiarities germane to WDM networks. Some of these extensions will be outlined later.

In MPLS networks (or MPL(ambda)S networks for that matter), connections are created by edge nodes called ingress nodes. Assuming the path route is known, the connection can be established using a signaling protocol such as RSVP-TE or CR-LDP [13]. In particular, the node downstream of the ingress is notified of the explicit route for the connection. Assuming the connection is accepted, the intermediate node reserves resources and initiates the same procedure for the next hop downstream along the route (see fig 1.5).
Figure 1.4: Link State Packets Distributed Amongst OXCs

Figure 1.5: Signaling Protocol to Establish a Connection
The connection is established once all nodes along the path have been notified. For WDM networks, the switching matrix of intermediate OXCs have to be altered to satisfy any <fiber,lambda> port mappings. In a packet switched network, signaling and control information is transmitted in band with data packets. For WDM networks, there may be a designated lambda channel on every link reserved for communicating this information between adjacent OXCs [14].

1.4 Protecting Against Network Failure

As with any communications network, faults can occur. Cables can break, OXC hardware can malfunction, software bugs may exist and so on. Whatever the case, data flowing through a failed component must be re-routed over working components if a connection is to be sustained. There are a number of ways to do this. Any such method can be classified as either reactive or proactive [15]. In reactive mechanisms, attempts to recover from the failure are made only after the failure occurs. For example, if a link cable breaks, OXCs attached to it could announce loss of connectivity via a link state update. Eventually, this may trigger some ingress OXCs to reroute any lightpaths of theirs that must have been brought down. Proactive schemes are those that take a priori measures to guard against network failures. Hence, the solution to a network failure is known beforehand. Proactive schemes can be further categorized as link protection or path protection. In link protection, bypass tunnels are established around links, only to be used if the link fails (fig 1.6). In path protection, a backup lightpath (physically disjoint from the primary lightpath) is installed between the same source and destination as the primary (fig 1.7). If any network failure occurs such that the primary is brought down, traffic is switched onto the backup lightpath, and customer connectivity is restored. Generally, path protection is known to be more capacity efficient than link protection [16]. Also, it is believed that path protection can still provide small enough recovery times to satisfy virtually all transport layer services [17]. As a result, the focus of this work has been on pre-allocated path protection.
1.5 Shared Path Protection in WDM Networks

If one is going to pre-allocate protection paths, it must be done with resource efficiency in mind. If each backup path had its own dedicated resources, then about 100% extra network resources would be used just for protection purposes! Fortunately, backup resources can be shared amongst different connections. Carrier grade optical networking equipment is usually designed to be robust [18], so barring any catastrophic
disasters (earthquakes or major fires), multiple simultaneous failures are not likely to occur simultaneously. Customers that are comfortable with this level of assurance are said to be content with the single component failure assumption [19]. For customers that accept this assumption, there is nothing wrong with permitting their backup lightpaths to mutually share resources with other backups, on the condition that the primary routes of their respective connections are physically disjoint from each other. When this condition is satisfied, no single failure can bring down more than one of these primary lightpaths at the same time – which means all these connections are protected from single component failures. The idea is illustrated in figs 1.8 and 1.9.

1.6 Online Lightpath Routing – The Optimal Solution

Consider a large backbone servicing a number of reliable connections, where backup lightpaths can share resources with other backups as long as all connections are protected against single component failures. Intuitively, there should exist a way to place
all primary-backup pairs so that the total number of consumed resources (lambdas) is minimized. As a matter of fact, this problem has already been solved in [19]. But the solution relies on complex optimization algorithms that will have to be run upon deletion of an old connection or acceptance of a new connection. Furthermore, already-established lightpaths may need to be reconfigured depending on the results of the algorithm. Disrupting already routed lightpaths is obviously not tolerable for reliable connections. For a dynamic network, where existing lightpaths must remain intact and future demands are unknown, using a greedy [20] routing approach is the obvious alternative.

The problem should be restated in a new way. Given the current state of the network, including the existing (and fixed) placement of lightpaths, what is the best primary-backup route for a new connection request between a given source and destination. Here, the “best” primary-backup route is the one that requires the least additional bandwidth. This question was also posed in [21] in the context of MPLS packet switched networks. In fact it was formulated as an integer linear programming problem that could be solved using iterative computer techniques (see appendix A). But this solution also has very high computational complexity (integer linear programs are NP hard).

Notwithstanding this issue, there are other problems with method. First, the solution is contingent on the knowledge of complete network state, which means it is really only amenable to centralized approaches. Centralized schemes are not favorable for many reasons. For example, when networks become larger, centralized network managers become flooded with network state updates, and requests for lightpath provisioning [8]. This can further impede an already slow connection establishment process. In addition, centralized agents themselves are at risk to malfunction. And any survivable network should not suffer from single points of failure - single failures that can disrupt the entire operation of the network. Hence, fast, distributed based schemes are better in this respect.

**Summary**

It should be stressed that the premise of this research was to develop online backup lightpath routing schemes for a distributed network environment. Consequently, this work leverages upon, and is applicable to the work done on MP(Lambda)S. The stated
objective of any routing methodology is to reduce backup resource consumption. Some related research has been conducted for real-time packet switch networks [20] [23], but the unique nature of this problem (specifically, dealing with discrete bandwidth lightpaths) gives way to new and better performing schemes.

**Contribution**

The contribution of this work is to provide two algorithms (Educated Depth First Search and Unrestricted Troublesome Hop scheme) that outperform conventional routing approaches from the standpoint of backup resource consumption. Through analysis and simulation, the superiority of these two algorithms is quantified under a variety of network scenarios. Additionally, modifications to these backup routing approaches are given in order to support connections with stricter fault tolerance constraints. Specifically, a new method for protecting primary lightpaths with constraints akin to traditional 1:1 protection is shown. This method generally requires less resource consumption than traditional 1:1 protection.
Chapter 2

Principles and Assumptions

Before describing specific backup routing schemes, certain guidelines and principles need to be established. This will be done in this chapter.

2.1 Working Assumptions

It should already be known that this work is relevant to WDM networks, and so every connection is of fixed and identical bandwidth. Other assumptions are stipulated below:

- The term network resource has specific meaning in the context of this work. A network resource is defined as a wavelength on a link. Two lightpaths sharing the same resource on a link effectively share the same \(<\text{fiber,lambda}\>\) pair on that link.
- Two types of failures are possible:
  1) Link failure, which is the failure of all fibers on a link cable.
  2) Node failure, which is the failure that renders an entire OXC, and all attached links inoperable. Node failures are less likely to occur in real networks \([18]\), so only link failures will be considered at first. Later, the possibility of node failures will be considered.

In real world networks, failures can be more specific. For instance, perhaps only one fiber within an entire link bundle gets damaged. Also, within an OXC, it is possible that only a small part of the switching fabric is faulty, as opposed to a full-fledged OXC breakdown. In such cases, only some of the lightpaths using the failed link (node) will need to be re-routed. But these examples are just subsets of the two failures listed above. The specific type of failure that does occur is not known beforehand, so protection needs to be provisioned assuming the worst single failure scenario. Entire links do fail in practice. During periods of construction, unsuspecting backhoes are notorious for accidentally cutting
network cables. In addition, power outages or fires can halt the operation of entire network nodes.

- A backup lightpath must be physically disjoint from its primary. By physical disjointness, it is meant that the backup should not reside over the same nodes (source and destination excluded) and links as the primary. This view may be too simplistic in practice. For instance, in fig 2.1, two separate links housed in the same conduit are shown. It is not unreasonable to believe these two links could fail at the same time. Situations like this have spurred the notion of a Shared Risk Link Groups (SRLG) [22]. Any lightpaths that are susceptible to simultaneous failure are considered to be part of the same SRLG. Therefore, it is more accurate to say that a backup path should be SRLG-disjoint from its primary.

![Logical View](image1.png)

![Physical View](image2.png)

Figure 2.1: Illustrating the Concept of Shared Risk Link Groups (two logically separate links which are at risk to the same physical failure)

- The link-state information must at the very least include information about link connectivity and the residual bandwidth available on a link. The unit of bandwidth granularity in a WDM network is a lambda. The residual bandwidth on a link is equivalent to the total number of free lambdas (aggregated over all of its fibers) on it. Other link state information will be introduced when needed.
- All connections are assumed to demand the same level of fault tolerance; that is pre-allocated path protection, in which backup resources can be shared with other backup lightpaths. Later, this restriction will be relaxed.
Lightpaths are not required to maintain wavelength continuity. All OXCs are assumed to have full wavelength conversion capabilities, and switching fabrics are presumed to be flexible enough to permit any <fiber,lambda> port mapping (i.e. switching fabrics are non-blocking).
2.2 Fundamental Principles

Before discussing methods for backup lightpath routing, some key concepts need to be outlined. Any method for backup lightpath establishment can be described as a two-step process:

1) Routing Decision
2) Resource Reservation

These two steps are elaborated further.

2.2.1 Routing Decision

As already mentioned, ingress OXCs route lightpaths based on their perceived knowledge of network state. Connections are placed with the intention of imposing the least cost burden on network resources. Commonly, the primary lightpath route is determined first, and the backup second. For the primary lightpath, the obvious choice is to select the shortest hop path to the destination, since dedicated resources need to be reserved on every link it traverses. It is recognized that other criteria (e.g. load balancing) may influence the primary path routing decision. However, incorporating multiple objectives into a routing policy typically makes the problem computationally intractable [20], so only resource minimization will be considered here. As far as backup routing is concerned, there are a number of alternatives. Chapter 3 will describe some of the approaches.

2.2.2 Resource Reservation

Backup lightpaths are not established in the same manner as primary lightpaths. For one thing, it may not be mandatory to reserve new resources on a given link, because the backup could possibly share existing protection lambdas with other backups already using that link. To ascertain whether an extra lambda is necessary to support a new backup lightpath, an OXC needs to know the primary routes for the new backup, and all other backups currently residing over the link in question. This means that the primary path must be successfully established before the resource reservation phase is initiated on the secondary path.
Furthermore, the signaling protocol for backup lightpath establishment (e.g. RSVP-TE or CR-LDP) must be enhanced so intermediate OXCs are made aware of the nodes and links that the primary path traverses. If an additional lambda is indeed required, one is picked from the pool of remaining free lambdas and reassigned to the set of shared protection lambdas. Fig 2.2 shows a sample OXC with five attached links servicing an arbitrary number of primary and secondary lightpaths. Primary lightpaths have fixed input-output port mappings. For example, primary ID 25 is switched from input \(<fiber,\lambda> \text{port} <1,2>\) on link 2 to output port \(<1,18>\) on link 1. Port maps for backup lightpaths are not fixed. Rather, each link has a group of protection lambdas set aside to support all the backup lightpaths over the link. For example port \(<1,1>\) on link 3 could support either backup ID 31 or 37. Reserved shared protection ports are shown along with the backup IDs they support. Finally any free ports are also specified.

![Figure 2.2: An OXC With Five Links](image)

\(^1\) Using RSVP as an example, the PATH message can be augmented by defining a new SESSION TYPE called 'Backup Path Establishment'. This SESSION TYPE will in turn be comprised of a SESSION ATTRIBUTE containing SUBOBJECTS that specify the corresponding primary links of the connection.
It is imperative to know just how many protection lambdas on a link are needed to accommodate a given number of backup lightpaths. Fig 2.4 depicts an arbitrary lightpath placement scenario for a sample network. All connections consist of a primary-backup lightpath pair, where the backup can share resources with other backups. Three backup lightpaths (dashed lines) have been provisioned over link f-g. How much spare bandwidth on link f-g needs to be set aside to support these three lightpaths? To answer this question, define a set $A_p_i$ as the list of primary links for connection $i$'s primary path. For example, $A_p_2$=(d-c,c-b,b-a,a-e). Do this for all three connections whose backups reside over f-g, and take the union of all resulting sets to form a superset, $\bigcup_{i=f-g} A_p_i$. Now, imagine the failure of each link in the superset (one at a time), and determine how many protection paths will be simultaneously activated over link f-g. The failure that results in the maximum number of backup activations over link f-g is deemed to be its worst-case failure, and the number of protection lambdas that f-g must reserve is equal to this maximum number of activations. For this particular scenario, there are two worst-case failures: the fault of link a-e, and the fault of link n-m, both of which invoke two backup activations over link f-g. Thus, link f-g should have two protection lambdas. The power of backup sharing is evident because only two protection lambdas are necessary to support the three backup lightpaths on link f-g.
Summary

Different approaches to establishing backup lightpaths were developed in this work. Any scheme will use the same resource reservation procedure, so what sets the various schemes apart is their ability to obtain routes that encourage high levels of backup sharing. The best ones are those that perform well and efficiently, without imposing unrealistic demands on the network infrastructure and protocols.
Chapter 3

Heuristic Backup Lightpath Routing Solutions

In this chapter a variety of methods for backup lightpath routing will be presented. Each method has its own advantages and drawbacks, so some key criteria should be defined from the outset. In particular, any approach should be evaluated based on:

1) Performance
2) Algorithm complexity
3) Link-state information requirements
4) Backup path setup latency time

Five schemes will be presented. In all except scheme 2 (Joint Routing), it is assumed that the primary path has already been established.

3.1 Conventional Schemes

3.1.1 Shortest Path First Routing (Scheme 1)

This is perhaps the most obvious and straightforward tactic for backup lightpath routing. Here, the backup path is chosen to take the shortest path disjoint of the primary (recall the primary route is assumed to be already known). The main point of this method is to serve as a means for comparison against more sophisticated ones. If another scheme cannot perform sufficiently better than the Shortest Path First approach (also referred to as SPF Routing), it clearly has little value.

Algorithm

1. The ingress OXC models the network as a graph, with the cost of all links set equal to '1' by default. However, all links in common with the primary path are set to infinity\(^2\).

\(^2\) In the case of node disjointness, all links attached to the intermediate nodes of the primary path are also set to infinity.
2. Using Dijkstra's algorithm, the shortest path with respect to the link weights is found. If there are multiple equal cost paths, one is chosen at random. If no path exists, the procedure is aborted and the connection request is denied.

3. Resources are reserved along the resulting backup route, using the procedure from section 2.2.2. If during the admission control of any intermediate OXC, it is discovered that an extra protection lambda is needed, but no free lambdas are available, the process fails, and an error message (explaining the problem) is sent to the ingress. The ingress changes the cost of the offending link to infinity, and step 2 is repeated. Otherwise, the backup path has been successfully provisioned.

**Computational Complexity**
The cost of generating the graph is negligible (only the cost of links overlapping with the primary need to changed to infinity). Dijkstra's algorithm has a known complexity of \( O(n^2) \), where 'n' is the number of nodes in the network [25].

**Link-State Information Requirements**
No new link state information, beyond what was given in Section 2.1 is necessary.

**Setup Time Latency**
As with all methods, attempts to establish the backup lightpath cannot be made unless and until the primary path has been installed. Once the resource reservation phase has indeed started, it can only fail when an extra lambda is needed on a link, but none are available. In this event, the entire procedure must begin again. In this scheme, admission control failures are only expected to occur during periods of high network load.
3.1.2 Joint Routing (Scheme 2)

In the previous approach, the ingress could not initiate the backup path routing algorithm until it was aware of the primary lightpath route. This actually restricts the choice of backup path routes (because the primary links are immediately eliminated from consideration), and could work against finding low-cost backup paths. One solution is to obtain the shortest-hop disjoint pair of paths, which is defined as the two mutually disjoint paths that use the least total number of links. Here, the primary and backup are treated as one entity, and the objective is to locate the minimum-hop pair. Algorithms (such as one by Suurballe [26]) have been devised to accomplish this goal. One drawback to this method is that it necessarily integrates the primary path routing objective with that of backup routing. In the previous scheme (and for all forthcoming), the primary path routing policy can be separate and distinct from the stated backup path routing objective (which is backup resource minimization in this thesis). So the backup routing algorithms of the other schemes are applicable even when primary lightpaths are not placed with hop minimization in mind. For example, the primary lightpaths could be placed to balance the resource load across a network, while the backup lightpaths could be placed with the objective of conserving protection resources. This is not possible with the Joint Routing scheme because there is only one objective – total hop count minimization. Thus, in order to fairly compare the performance of one of the other backup routing schemes to Joint Routing, it is essential to always pick the shortest-hop primary lightpaths.

Algorithm
1. The network is modeled as a graph in which all links are assigned a cost of 1.

2. Using Suurballe's algorithm, the least-cost disjoint pair of paths is found. If multiple equal cost pairs exist, one is chosen at random. If two distinct paths do not exist, the procedure is aborted and the connection request is denied.

3. Resources must be reserved along the primary path and then the backup path. For the backup, if during the admission control of any intermediate OXC, it is discovered that an extra protection lambda is needed, but none exists, the process fails, and an error occurs.

---

3 When deriving the primary route, all links with no free available lambdas are reassigned a cost of infinity.
message (explaining the problem) is sent to the ingress. The ingress changes the cost of the offending link to infinity, and step 2 is repeated. Otherwise, backup path has been successfully provisioned.

Computational Complexity
The cost of generating the original graph is negligible. The cost of Suurballe’s algorithm is dominated by the following steps:
(i) An initial run of Dijkstra’s algorithm to obtain the primary route (and a shortest path tree). This step has an associated cost of $O(n^2)$, where ‘n’ is the number of nodes in the network.
(ii) A graph transformation, which costs $O(L)$, where ‘L’ is the number of links in the network.
(iii) Another run of Dijkstra’s algorithm (to obtain the backup route).

To be fair, the cost of obtaining the primary route should be excluded since it is not included in all other schemes. Then the computational cost is the cost of a single iteration of Dijkstra’s algorithm plus the cost of the graph transformation.

Link-State Information Requirements
No new link state information, beyond what was given in Section 2.1 is necessary.

Setup Time Latency
Once the primary path has been established, the latency involved in setting up the backup should be similar to the first scheme. However, if there is an admission control failure at an intermediate OXC, and the procedure needs to be restarted, the original primary path should also be torn down. But the ingress does not have to wait for primary resources to be released before it can recommence with the algorithm, so primary path teardown is assumed to contribute little to the overall backup path setup latency time. The new primary lightpath should be reserved before the backup lightpath can be

---

*The graph transformation is described as follows (assuming a directed graph G). Let $d_i$ represent the shortest path length from the source node to node i. Replace the cost of link (i,j), which is currently equal to ‘1’, with $1+d_i-d_j$. This needs to be done for all L links of graph G. Hence, the complexity is $O(L)$.*
reserved, so there will be an extra setup delay every time the process is repeated (again, this will normally only happen during high network loads).
3.2 Intelligent routing – Finding the Lowest Cost Backup Path

The two previous approaches to backup routing are not especially sophisticated. In both cases, the underlying philosophy is that minimizing hop count will in turn lead to lower resource usage. But this is a suspicious strategy, since a backup lightpath can be supported over some links without commanding any extra bandwidth. It is entirely possible that a backup lightpath traversing several hops could impose less resource demands than one placed over even a single hop route. Logically, it should be advantageous to be able to consistently find the actual lowest cost backup path. In fact, for connection-oriented packet-switched environments, [20] demonstrated that an algorithm that can obtain the backup route with the lowest incremental cost to the network is superior to other conventional routing schemes when it comes to reducing overall backup resource usage in a network.

For a distributed, connection-oriented packet-switched network, finding the backup path with the lowest incremental cost is close to a practical impossibility. Each node would need to know the routes and bandwidth for every connection in the network. Integrating this kind of information into IGPs is ludicrous. Link-state packets would have to advertise the connection IDs and bit-rates for the full set of primary paths and the full set of backup paths currently using the link. From this, ingress nodes could piece together the placement of all connections in the network. Placing the ID of just one connection in a link-state packet will occupy several bytes of space. So, in large, high-capacity networks, the link-state overhead becomes excessive.

For the WDM networks being discussed here, the task of obtaining the lowest cost backup path is not nearly so daunting. All lightpaths are of uniform bandwidth, and this leads to some useful properties. In particular, any attempts to share resources with other backups on a link becomes an all-or-nothing endeavor – either a backup can be placed over the link without requiring any additional protection bandwidth, or a whole new unit lambda needs to be reserved. So, as far as an ingress OXC trying to find a backup route in concerned, a link can be in one of two states (either an extra lambda will be needed or nothing will be needed to support the new backup lightpath). This simplification makes it easier to find the lowest cost-backup route, as will be shown in the next two schemes.
3.2.1 Educated Depth First Search (Scheme 3)

An ingress OXC can definitely find the minimum cost backup path if full routing information is at its disposal. But even if it isn't, it is still possible to search for the minimum cost backup path. In particular, an initial guess of the lowest cost path can be made. Then, intermediate OXCs along that path can be polled to see just how many extra lambdas over the entire route will be necessary if the path were used. If the cost is higher than anticipated, the perceived view of the network could be adjusted, and a new potentially lowest cost path derived. This process could be repeated until an actual lowest cost path is found. What is being described here is essentially a depth-first search for the minimum cost backup route. A more concrete explanation is now given.

Let the links that the ingress thinks will not require an extra lambda be termed "perceived-friendly". All other links are labeled "perceived-unfriendly". "Perceived-friendly" links are low-cost, while "perceived-unfriendly" links are high cost. Initially, the ingress is ignorant of the network state, but optimistic about its chances of finding a low cost path. Thus, all links are considered "perceived-friendly", at first. The ingress chooses a perceived lowest-cost path, and starts the resource reservation phase over its route. The signaling protocol of this phase is upgraded so intermediate OXCs along the links of the path are told whether or not the ingress node perceived their link as friendly. In the case where a perceived-friendly link actually needs an extra lambda, the intermediate OXC is programmed to fail the admission control process, and send an appropriate error message to the ingress. Upon learning of its mistake, the ingress can change the link's label to "perceived-unfriendly", and determine a new low cost backup path route (see fig 3.1). This procedure is repeated until an end-to-end path is found. Such a path will always be a minimum cost one.

After an ingress OXC is notified of a mistake, its next routing decision will be a more informed one. Path choices get wiser as the search progresses. Still, there is some concern that there may be too many mistakes made at the beginning. To help alleviate this problem, a new link-state metric called shared protection lambdas is defined. It indicates the total number of shared protection lambdas (aggregated over all fibers) currently residing over the link. This new metric is coined because it is reasonable to expect that a backup lightpath is less likely to demand additional bandwidth on a link already containing a large number of protection lambdas. This hypothesis can be qualified analytically.
Figure 3.1: Illustration of the Educated Depth First Search Scheme

(a) Primary lightpath established
(b) First attempt to establish a backup
(c) Link cost update and second attempt to establish a backup
(d) Link cost update and third (successful) attempt to establish a backup
Suppose the primary path is known, and the ingress is currently deciding amongst several candidate links to be the first hop for the backup path (see fig 3.2). Each of the links on the primary path hosts a number of primary lightpaths for other connections. Some of these other primary lightpaths may have protection paths that are already placed over b1, b2, or b3. For example, if link p1 fails, let the number of backup activations over link b3 be denoted as $M_{b3}^{p1}$. Also, denote the number of shared protection lambdas over link b3 as $E_{b3}$.

Based on prior analysis, if link b3 is going to be used as the 1st hop, no new protection bandwidth needs to be reserved over it if and only if:

$$E_{b3} \geq M_{b3}^{p_i} + 1, \text{ for all } i = 1, 2, 3$$

Unfortunately, without the complete routing information, the ingress has no idea what the value of $M_{b3}^{p_i}$ (or any $M_{b_i}^{p_i}$ for that matter) is. All that can be said is that it is better to choose routes that traverse links with large shared protection bandwidth so the above condition is satisfied more often. Following this reasoning, when the ingress is about to route the backup path, it makes sense to weight the links using the cost function $(E_{b_i})^{-1}$. Since the purpose of this cost function is to improve the search decisions, this scheme is called the "Educated Depth First Search" (abbreviated Educated_DFS).
**Algorithm**

1. Initially, the Ingress forms a weighted graph by assigning all links the cost,

\[ \text{Cost}(\text{link}) = \frac{1}{E}, \]

where \( E \) is equal to the aggregate number of shared protection lambdas on a link. Those links with no shared protection lambdas are assigned a cost=K (K is defined as a sufficiently large number), while those links traversed by the primary path are set to infinity. Those links with cost K are designated as "perceived unfriendly", and those with cost \( \leq 1 \) are defined as "perceived friendly".

2. Then, it uses Dijkstra's algorithm to find the shortest path with respect to this graph. If multiple equal cost paths exist, one is chosen at random. If no path exists the procedure is aborted and the connection request is denied.

3. The ingress node attempts to establish the backup path over the chosen route using a resource reservation protocol. The reservation message should indicate to each intermediate OXC whether or not its upstream link is considered friendly or unfriendly. An intermediate OXC can only accept the request if:
   (i) It does not require an extra lambda to support the backup lightpath (i.e. sharing is possible).
   (ii) It requires an additional lambda, and the ingress node has labeled its upstream link as "unfriendly".

If all intermediate OXCs accept the connection, the backup lightpath has been established. Otherwise if an admission control process fails, an appropriate error message is returned to the ingress node. In this instance, the ingress checks the error message to see which OXC sent it. It will change the cost of the corresponding link in the following manner:
   (i) If the current cost is = K, it is changed to \( \infty \).
   (ii) If the current cost is \( \leq 1 \), it is changed to K (‘K’ is a sufficiently large number).

Return to step 2.
**Complexity**

Ingress OXCs are assumed to have a graph weighted as given by the algorithm. The graph can be dynamically updated as new link-state packets arrive. This graph is used to determine the first guess for the backup route. Modifications to the graph are made as the ingress learns of its mistakes. The processing cost of this is ignored. Dijkstra's algorithm is run upon each iteration of the algorithm. Dijkstra's algorithm has computational complexity $O(n^2)$, where 'n' is the number of nodes.

**Link-State Information Requirements**

As mentioned before, this scheme relies on introducing a new link state metric, which is the number of shared protection lambdas aggregated over all fibers on a link. This means that an extra field has to be incorporated into the original link state packet structure. The size of this field should be similar to that of the other fields already required. While using this new metric should have little or no impact on performance, it will help the ingress find the minimum cost path more quickly.

**Setup Latency**

The main drawback of this algorithm is the potential latency that may be involved in setting up a backup path. Every time an intermediate OXC rejects a request, a failure message needs to be propagated back to the ingress OXC, which must compute a new route, and initiate a new resource reservation phase. If there are too many rejections, the latency involved in setting up a backup path can be quite high. It is expected that there will be many rejections because the ingress makes a lot of guesses (especially at the beginning of the algorithm). Some may argue that long latency times are not of serious concern for the protection path, since it is normally not in use anyway. This issue aside, if the latency times become excessive, the state of the network may change so that the chosen path may no longer the best one.
3.2.2 Avoiding Troublesome Hops (Scheme 4)

In section 2.2, a mechanism for calculating the number of protection lambdas required over a link was given. In summary, the number of protection lambdas is equal to the number of simultaneous backup activations over the link in the worst-case single failure scenario. An intermediate OXC engaging in an admission control procedure for backup lightpath can use a similar approach to deduce whether an extra lambda is needed to support a new backup lightpath. Each OXC could construct a table like that in fig 3.2, to systematically store all the primary hops corresponding to the backup connections currently on an attached link. The table will be called a protection table, and an OXC should maintain a unique one for each of its attached links. In the sample shown, the table is divided into two columns. The first is a record of all the distinct primary hops associated with the entire set of backup connections over the link (this is identical to the superset \( \bigcup \mathcal{A}_p \) defined in section 2.2). Each row identifies a different primary link.

The second column of that row is a list of all the connection IDs for the backup lightpaths that need to be activated if the primary link in the first column fails.

<table>
<thead>
<tr>
<th>Primary Hop</th>
<th>Affected Connection IDs</th>
</tr>
</thead>
<tbody>
<tr>
<td>a-b</td>
<td>31,51,270,501,552</td>
</tr>
<tr>
<td>r-e</td>
<td>14,29,675</td>
</tr>
<tr>
<td>w-q</td>
<td>21,145,267,339,401</td>
</tr>
<tr>
<td>e-j</td>
<td>19,194,225</td>
</tr>
</tbody>
</table>

The prevailing worst-case failure is simply the hop on the row containing the most number of connection IDs. Refer to the worst-case fault as a "troublesome hop". For the example, there are two troublesome hops (a-b and w-q) in the table shown.

Protection tables are modified upon acceptance of a new backup lightpath or removal an existing one. During admission control of a candidate backup connection, the OXC learns the connection's primary route, so it can conceive how the protection table will be modified.
updated if the backup lightpath was accepted. A new protection lambda needs to be reserved if the updated table has a row with more connection IDs than the previous worst case. Of course, this can happen if and only if the primary lightpath associated with new backup connection traverses one or more of the current troublesome hops. For instance, in the sample protection table, a new backup connection whose primary path passed through troublesome hop a-b, could not be supported without an extra protection lambda. On the other hand, if the primary route did not include either troublesome hop, the backup would not impose any new resource demands over the link. This is a powerful concept, because it reduces the amount of information one needs to know about a link in order to determine if a backup lightpath can be placed over it without extra bandwidth. Given a known primary path, the cost of a backup route (measured in lambdas) is equal to the number of links along its backup route that have at least one troublesome hop that matches one of the primary hops of the primary lightpath.

A new proposition is spawned out of the above discussion. OXCs can advertise the corresponding troublesome hops of their attached links into link-state packets. Ingress OXCs armed with this information from every link, have enough knowledge to pick the lowest cost backup route, assuming the primary path is computed prior to and independently of the backup. Therefore, this approach offers the same performance as that outlined in [20] without the huge link state overhead.

Algorithm
The algorithm is contingent on the primary route being known

1. The ingress OXC generates graph according the following rules. The cost of a link is equal to 1 if the link's troublesome hops do not coincide with any of the primary hops. If there is a match, the cost of the link is assigned a value of K (where 'K' is a large number) provided free bandwidth is available. Otherwise the link cost is infinity.

---

5 This solution is specific to discrete bandwidth WDM networks. The notion of troublesome hops has no relevance to environments with connections of varying bandwidth.
2. The shortest path between the source and destination with respect to the link weights is obtained using Dijkstra’s algorithm. If multiple paths exist, one is chosen at random. If no path exists, the procedure fails and the connection is refused.

3. Resources are reserved over the chosen path. No admission control failures should occur in theory since the ingress knows which links are usable beforehand. At the end of this phase, the backup path has been established.

**Computational Complexity**
The graph cannot be generated until the primary route is known. Each link is assigned a cost on its own merits. The cost of this is \( O(L) \), where ‘L’ is the number of links in the network. Dijkstra’s algorithm (used to compute the backup route) costs \( O(n^2) \), where ‘n’ is the number of nodes in the network.

**New Link-State Information**
This scheme mandates that OXCs advertise the troublesome hops for each of their links. There may be more than one troublesome hop for a link. A new metric needs to be defined for link-state packets, and it should consist of subfields of wide enough size to label troublesome hop identifiers.

**Backup Path Setup Latency**
If a path has been computed, it should be established without admission control failure because the ingress OXC will not pick a link that doesn’t have enough bandwidth to accommodate the backup lightpath.

### 3.2.3 A Restricted Troublesome Hop Version (Scheme 5)
One grievance with the previous scheme is that link-state packet sizes can potentially become large if a link has many troublesome hops. It may be more prudent to fix the troublesome hop field size by restricting the number of troublesome hops that can be advertised. The unfortunate consequence of this constraint is that it will limit the knowledge of ingress OXCs. Poorer backup routes may be chosen when the ingress is not completely informed. In addition, it is no longer true that a backup path can always
be established without any admission control failures. It is useful to examine just how severely this restriction will hamper performance. So the previous scheme will be analyzed under two possibilities. The original approach will be referred to as the Unrestricted Troublesome Hop scheme (Unrestricted_TH for brevity), while the restricted version will be called the Restricted Troublesome Hop (or Restricted_TH) scheme.

**Summary**

The table below (table 3.1) is a summary of the attributes of each of the backup routing schemes.

<table>
<thead>
<tr>
<th>SCHEME</th>
<th>COMPLEXITY</th>
<th>LINK-STATE REQUIREMENTS</th>
<th>SETUP LATENCY</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPF routing</td>
<td>$N^2$</td>
<td>Nothing Beyond Basic Set</td>
<td>Low (except at high network loads)</td>
</tr>
<tr>
<td>Joint routing</td>
<td>$N^2+L$</td>
<td>Nothing Beyond Basic Set</td>
<td>Low (except at high network loads)</td>
</tr>
<tr>
<td>Educated_DFS</td>
<td>$kN^2$, $k=$average # of search attempts</td>
<td>Shared Protection Lambdas</td>
<td>Potentially High</td>
</tr>
<tr>
<td>Unrestricted_TH</td>
<td>$N^2+L$</td>
<td>Troublesome hop</td>
<td>Consistently Low</td>
</tr>
<tr>
<td>Restricted_TH</td>
<td>$N^2+L$</td>
<td>Troublesome hop</td>
<td>Low (except at high network loads)</td>
</tr>
</tbody>
</table>

Table 3.1: Summary of Scheme Attributes
Chapter 4

Performance Evaluation of Backup Lightpath Routing Schemes

The analysis of section 3 should have given the reader an understanding of the advantages and pitfalls of each scheme. While inferences as to which schemes perform better can be made, it is imperative to quantify the superiority of one scheme over another under a variety of scenarios. Therefore, the schemes were compared on different networks, and under different settings and constraints.

4.1-Simulation Environment

Three network topologies were considered. Network A (fig 4.1) is a 5X5 mesh-torus found in [20]. It consists of 25 nodes and 50 links. Network B (fig 4.2) is a 30 node, 59-link network originally used in [27] in their analysis of capacity placement strategies for mesh-survivable ATM networks. It is characteristic of typical long-haul networks. Network C (fig 4.3) is the topology of the North American UUNET backbone, and was derived from [28]. It consists of 61 nodes and 148 links and is representative of an actual real-world network. Tables 4.1 – 4.3 and graphs 4.1 - 4.3 summarize each network’s characteristics (including the links per node distribution, which is defined as the percentage of network nodes that are attached to ‘n’ links). For example, for network C, the links per node distribution is wide. As table 4.3 indicates, in network C, 30% of the nodes have only two attached links, whereas for network B, 7% have two attached links.
Table 4.1: Network A Characteristics

<table>
<thead>
<tr>
<th>Number of Attached Links</th>
<th>Percentage of Nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0%</td>
</tr>
<tr>
<td>2</td>
<td>0%</td>
</tr>
<tr>
<td>3</td>
<td>0%</td>
</tr>
<tr>
<td>4</td>
<td>0%</td>
</tr>
<tr>
<td>5</td>
<td>0%</td>
</tr>
<tr>
<td>6</td>
<td>0%</td>
</tr>
<tr>
<td>7</td>
<td>0%</td>
</tr>
<tr>
<td>8</td>
<td>0%</td>
</tr>
</tbody>
</table>

Figure 4.1: Network A, Mesh-Torus

Graph 4.1: Links per Node Distribution (Network A)

Table 4.2: Network B Characteristics

<table>
<thead>
<tr>
<th>Number of Attached Links</th>
<th>Percentage of Nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.00%</td>
</tr>
<tr>
<td>2</td>
<td>0.00%</td>
</tr>
<tr>
<td>3</td>
<td>0.00%</td>
</tr>
<tr>
<td>4</td>
<td>0.00%</td>
</tr>
<tr>
<td>5</td>
<td>0.00%</td>
</tr>
<tr>
<td>6</td>
<td>0.00%</td>
</tr>
<tr>
<td>7</td>
<td>0.00%</td>
</tr>
<tr>
<td>8</td>
<td>0.00%</td>
</tr>
</tbody>
</table>

Figure 4.2: Network B

Graph 4.2: Links per Node Distribution (Network B)
Figure 4.3: Network C (UUNET – N. America)

<table>
<thead>
<tr>
<th>NODES</th>
<th>LINKS</th>
<th>AVG. VERTEX DEGREE</th>
</tr>
</thead>
<tbody>
<tr>
<td>61</td>
<td>148</td>
<td>4.8</td>
</tr>
</tbody>
</table>

Table 4.3: Network C Characteristics

Links per Node Distribution (Network C)

Graph 4.3: Links per Node Distribution (Network C)
For each network, all links were composed of an equal (even) number of fibers, and each fiber supported the same number of wavelengths.

The simulation environment was developed using a computer program written in C. The simulation was event driven, so real-time delays were not simulated. Full-Fledged protocol implementations (that is, the actual OSPF and RSVP-TE), were not simulated, but the salient features of each were incorporated into the simulation framework.

As mentioned before, for all schemes, OXCs were provided with the basic set of link-state information (link connectivity, number of free lambdas available). The Educated_DFS approach additionally included the number of shared lambdas as another metric. Both Troublesome Hop schemes required the troublesome hop(s) metric outlined in section 3.2.2. In the Restricted Troublesome Hop scheme, only one (arbitrarily chosen) troublesome hop was advertised, even when multiple were present.

Upon the addition or deletion of a lightpath, the link-state information was updated accordingly. All ingress OXCs were assumed to have correct and consistent link-state knowledge by the time any new connection request arrived.

Connections were deemed to be bi-directional, so a lightpath from OXC1 and OXC2, implied an identical lightpath following the same route, in the opposite direction. During the admission control for a lightpath (primary or secondary), if a new lambda needed to be reserved on a link, the lexicographically smallest free <fiber,lambda> pair was chosen. For primary lightpaths, the connection ID was explicitly mapped to that port. Backup lightpaths, on the other hand, were loosely associated with the set of designated protection lambdas on a link.

Initially, all connections were implicitly assumed to request the same degree of reliability. In particular, each connection was composed of a primary-backup pair, and the backup

---

6 As a result, an OXC reserving resources at lambda-fiber port <fiber,lambda> will also reserve resources on <fiber, lambda+1>.
lightpath could share resources with other backups. The only faults imagined were single link failures. Both these assumptions were modified in later simulations.
4.2 Scenario 1 – Performance on a Mesh-Torus

The first simulation was designed to test each algorithm's ability to find low cost backup routes. Network A was the topology of choice, due to its high connectivity (average vertex degree is 4), symmetry and regularity. There are often multiple equal-hop paths between any source and destination, so this offered wide range of potential backup routes for any scheme to select from. Each link was composed of a single bi-directional fiber pair with infinitely large bandwidth capacity. This was done to ensure routing decisions were not hampered by bandwidth constraints. Six hundred ordered connection requests (one between each of the 25x24=600 node pairs) were serviced. The number of lambdas reserved for primary lightpaths, and the number of lambdas reserved for secondary lightpaths were recorded. The experiment was repeated 5 times using different permutations of connection requests.

Also, for comparison purposes, the brute-force solution made reference to in [20] was also simulated. Here, any OXC is assumed to have full routing information, and the lowest cost backup route is always chosen. The cost of a link is equal to the amount of additional protection bandwidth it requires to support the backup lightpath. Hereafter, it will be referred to as the Han-Shin approach.

The Results
All results are the average of the values obtained from the individual experiments. Graph 1, shows the percentage of total network resources occupied by both primary and secondary lightpaths after all 600 connections were established. Results for each of the six schemes is given. In all cases, 30% of the network was used for primary reservations. This is an important fact to know when evaluating the performance of a backup routing scheme, because it means that any notable decrease in backup resource usage didn't come at the expense of an increase in primary resource usage.

The Educated_DFS and Unrestricted_TH schemes require the least amount of backup network resources, while the Han-Shin approach comes a close second. It may be

7 Results lie in 5% window of the reported value with 95% confidence.
surprising that the Han-Shin method produces slightly worse results than the other two, since all three attempt to obtain the lowest cost backup route. The reason is that they use different cost functions. In particular, under the Han-Shin definition, links that will not require extra protection bandwidth are assigned a cost of zero. But a shortest path algorithm does not discriminate between a one-hop path of cost zero and a N-hop path of cost zero. The unfortunate consequence is that the backup route may traverse more hops than necessary. This is seen from graph 4.4, which shows that a given backup path obtained from the Han-Shin scheme traverses more hops than any other scheme including the Educated_DFS and Unrestricted_TH schemes. Even though this has no bearing for the connection is question, unnecessarily occupying extra hops could take potential resource sharing opportunities away from subsequent backup lightpaths.

The performance of most schemes was superior to the standard SPF Routing approach. In fact, the best schemes used 50% less backup resources than SPF Routing. However, Joint Routing did not fare better than SPF routing. In this well-connected network, where multiple equal-hop paths between two nodes often exist, the solutions produced from these two schemes rarely differ. The results of the Restricted_TH approach lie somewhat in the middle. This indicates that considerable gains can be obtained by advertising all the troublesome hops.

Overall, the lesson learned from these simulations is that there is evident value to judiciously picking backup routes. Routing, with the intent of minimizing backup resource usage is clearly advantageous over a naïve approach like SPF-routing. Admittedly, the simulation setting was somewhat contrived. Hence, future simulations were run under more realistic conditions.
Graph 4.4: Scenario 1; Percentage of Network Resources Occupied by Backup and Primary Lightpaths on Network A - A Comparison Between Schemes

Graph 4.5: Scenario 1; Average Number of Hops Traversed by a Backup Lightpath on Network A - A Comparison Between Schemes
4.3 Scenario2 – Evaluating Performance Under Realistic Network Conditions (Part 1)

The schemes were next evaluated under a dynamic network environment. Network B was first used, and all links were composed of 10 bi-directional fibers, each capable of supporting 10 lambdas. Client requests were of two types:
1. Connection Establishment
2. Connection Removal

More specifically, connection establishment requests were structured as a Poisson process with a fixed mean arrival rate of 1 per unit time. Connection duration times were modeled as an exponential random process, with an average lifetime of $\alpha^{-1}$ units. At the conclusion of a connection’s lifetime, a request for connection deletion was made. The offered network load (given as $\text{arrival\_rate} \times \text{duration\_time} = 1 \times \alpha^{-1}$ Erlangs) was varied by changing alpha.

Client requests were handled sequentially. Any node could be named as the source or destination of a connection request, but a node was chosen in proportion to its access to bandwidth. For example, in fig 4.2 (shown earlier), there is a node attached to 2 of the networks 59 (undirected) total links. Therefore, the probability of it being a source or destination was $\frac{2}{2 \times 59} = \frac{1}{59}$.

The simulation was run until 1,000,000 connection requests were made, and the parameters recorded were as follows:

- The total number of primary resources used for all successfully established connections.
- The total number of protection resources used for all successfully established connections.
- The total number of connections requests refused.
- The number of admission control failures encountered during any attempts to establish a backup lightpath.
Results
The outcome of the simulation is illustrated by the series of graphs 4.6 - 4.11. Graph 4.6 depicts the ratio of the total protection resources used in establishing all backup lightpaths to the total primary resources used in establishing primary lightpaths, as a function of offered network load. The curve for each scheme is shown on the same graph. All curves have a similar shape, and this implies that the different schemes react similarly to conditions imposed by the network load. During very low network loads (< 200 Erlangs), there are few connections established in the backbone at any given time. With fewer backup lightpaths available, there are less sharing opportunities for any new backup lightpath, and this accounts for the high ratio. The problem quickly disappears when offered load increases, and the ratio becomes stable between 200-700 Erlangs. In this range, there is still enough free bandwidth on links so that primary lightpaths can continue to traverse the conventional shortest hop routes. In addition, the network is sufficiently loaded so that sharing opportunities are possible on many links. These facts can be verified by observing graphs 4.8, and 4.9, which show the average amount of resources required for primary and secondary lightpaths respectively. The curves of both graphs (and therefore also graphs 4.6 and 4.7) are relatively constant in the aforementioned loading range.

As the offered load increases beyond 700 Erlangs, links start to become fully occupied with lightpaths. Some connections have to be refused due to a lack of resources (see graph 4.11). Primary lightpaths that are established are often forced to take different (non shortest-hop) routes to reach destinations. This explains the rise of the curves in graph 4.8 at loads higher than 700 Erlangs. But finding backup paths isn’t so difficult because links with free lambdas are not as essential. In fact, by not taking shortest-hop routes, primary lightpaths inadvertently create new prospects for backups. There are a couple of reasons behind this claim. First, backup lightpaths might be able to take the shorter-hop routes that are not available to the primary (graph 4.10 illustrates this – the number of backup hops taken by a backup lightpath actually decreases when network load becomes high).

Besides this fact, a backup lightpath is also less likely to demand an extra lambda on a given link when network load becomes high. This can be reasoned as follows:
By default, primary lightpaths have been forced to always take the minimum-hop path available. Inevitably, this will mean some links become heavily used. Moreover, many of these heavily loaded links are likely to be the worst-case failure for a large number of network links. To prove this, a snapshot of the network state was taken during high network loads, after an arbitrary number of requests had been serviced. Any hop labeled as the worst-case failure for more than four network links (referred to as type-4 worst-case links henceforth) at that time was recorded. At moderately high network loads, it was observed that these hops were much more likely to be completely full than any other network hop (table 4.4 presents the observations based on the Unrestricted_TH scheme). For example, row 3 of table 4.4 shows that although type-4 worst-case links made up 68% of the total network, they comprised 100% of all the filled network links. Since a primary lightpath cannot be placed over these filled links, it will traverse other hops that are less likely (as it turns out) to be the worst-case failure of any given backup link. This can only help reduce backup resource usage, as is witnessed in graph 4.9. This also explains the dip in the backup:primary ratio at moderately high network loads (graph 4.6). When network loads become very high, even the once lightly loaded links are used frequently, so this phenomenon is not observed.

<table>
<thead>
<tr>
<th>Moderately High Network Loading</th>
<th>Percentage of Network Resources Used</th>
<th>Percentage of Network Links That are Type-4 Worst Case Failure Hops</th>
<th>Percentage of Filled Links That are Type-4 Worst Case Failure Hops</th>
</tr>
</thead>
<tbody>
<tr>
<td>46%</td>
<td>31%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>51%</td>
<td>34%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>74%</td>
<td>68%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>Very High Network Loading</td>
<td>79%</td>
<td>61%</td>
<td>78%</td>
</tr>
<tr>
<td>86%</td>
<td>54%</td>
<td>50%</td>
<td></td>
</tr>
<tr>
<td>87%</td>
<td>71%</td>
<td>73%</td>
<td></td>
</tr>
<tr>
<td>89%</td>
<td>81%</td>
<td>72%</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.4: Scenario 2; Inspection of Filled Links in Network B – Unrestricted_TH
The relative performance of different schemes validates what was observed earlier in scenario 1. Two of the schemes, (Educated_DFS, and Unrestricted_TH) produce near identical results, and outperform all other methods. Graph 4.8 demonstrates that the primary resource usage of all schemes is similar during any network load, which means any performance gains are directly related to the backup routing algorithms. In fact, these schemes reduce the required backup resources over regular SPF routing by more than 50% for most network loads.

Graph 4.11 indicates the connection refusal rate as a function of offered load. Again, the Educated_DFS and Unrestricted_TH scheme are superior to other schemes in this respect. For example, at an offered load of 1250 Erlangs, 12.2% of calls are blocked when SPF routing is used, but only 9.3% are blocked when the Educated_DFS scheme is implemented. As is expected, a scheme that works better to reduce backup resource usage will in turn allow more connections to be admitted into the network (hence the lower connection refusal rate). In all cases, it was found that over 80% of connection refusals happened at the primary routing stage (before the backup routing algorithms even commenced). This means that most connection refusals occur simply because sufficient free bandwidth wasn’t available to accommodate most primary lightpaths.
Graph 4.6: Scenario 2; Ratio of Total Number of Backup Resources Used to Total Number of Primary Resources Used, as a Function of Offered Load – A Comparison Between Schemes
Graph 4.7: Scenario 2; Average Resource Usage (primary and backup) per Connection, as a Function of Offered Load – A Comparison Between Schemes

Graph 4.8: Scenario 2; Average Resource Usage per Primary Lightpath, as a Function of Offered Load – A Comparison Between Schemes
Graph 4.9: Scenario 2; Average Number of Additional Protection Lambdas Required per Backup Lightpath, as a Function of Offered Load – A Comparison Between Schemes

Graph 4.10: Scenario 2; Average Number of Hops Traversed by A Backup Lightpath, as a Function of Offered Load – A Between Schemes
Graph 4.11: Scenario 2; Percentage of Connection Requests Refused, as a Function of Offered Load – A Comparison Between Schemes
4.4 Scenario 3 - Evaluating Performance Under Realistic Network Conditions (Part 2)

A similar experiment to scenario 2 was conducted, but this time using network C. All links were armed with 9 bi-directional fibers, and each fiber supported 12 lambdas. Connection requests were still modeled in the same manner as scenario 2, and the simulation was run until 1,000,000 requests for connection establishment were made.

Results
Graphs 4.12 to 4.16 display the results of the simulation (the results at very low network loads have been omitted). The shape of the curves for every graph follows a similar trend to the results from scenario 2. Furthermore, the graphs convey the same message about the schemes. Two of the schemes (Educated_DFS and Unrestricted_TH) have virtually identical performance, and are better than the others. Once again, the Restricted Troublesome Hop scheme has performance somewhat in between the best and the worst.

Upon closer inspection, there are some discernable differences between these graphs and those of scenario 2. For example, graph 4.12 depicts the ratio of backup resource usage to primary resource usage at various network loads. It should be noticeable that as network loads become high, the curves corresponding to different schemes converge. This means that the performance gain of say the Educated_DFS method over SPF Routing is less compelling at high network loads. At low network loads the Educated_DFS and Unrestricted_TH schemes use 50% less backup resources than either Joint Routing or SPF Routing. But at high network loads (greater than 5000 Erlangs) the performance gain drops to about 20%. This is in contrast to the results of scenario 2 (see graph 4.12), where the relative performance between any two schemes is quite constant for all network loads. The reason can be traced to the intricacies of each topology.

As explained in section 4.1, Network C has a diverse links per node distribution. There are large number of nodes with only two attached links, and a substantial number with many attached links. The nodes with many links usually contend with a lot of traffic. For
example, some of these nodes (e.g. Salt Lake City, Denver, St Louis) act as gateways for coast-to-coast traffic. Not only must they source or sink local connection requests; they must also contend the high volume of throughway traffic. When network load increases, these links are the first to be filled, and many calls have to be rejected. The network picture is restricted to a set of nodes with fewer (in many cases only two) attached links. Working with a reduced set of links, options are limited, and so it is expected that the route decisions of the various schemes become less different. As a result (a fact verified by graph 4.15), the amount of backup resources used by each scheme converges when network load gets high.
Graph 4.12: Scenario 3; Ratio of Total Number of Backup Resources Used to Total Number of Primary Resources Used, as a Function of Offered Load – A Comparison Between Schemes
Graph 4.13: Scenario 3; Average Resource Usage (primary and backup) per Connection, as a Function of Offered Load – A Comparison Between Schemes

Graph 4.14: Scenario 3; Average Resource Usage per Primary Lightpath, as a Function of Offered Load – A Comparison Between Schemes
Graph 4.15: Scenario 3; Average Number of Additional Protection Lambdas Required per Backup Lightpath, as a Function of Offered Load – A Comparison Between Schemes

Graph 4.16: Scenario 3; Percentage of Connection Requests Refused, as a Function of Offered Load – A Comparison Between Schemes
4.5 Estimating Setup Latency

Earlier, it was pointed out that the Educated_DFS scheme could suffer from long setup latency times. The issue will be illuminated further here.

Establishing a backup path involves finding a backup route and reserving resources along all hops of the chosen route. This process has to be repeated if there is an admission control failure at an OXC along the route.

Mathematically the setup time can be given as:

\[ T_{\text{setup}} = t_{\text{comp}} + N(t_{\text{prop}} + t_{\text{ad,proc}}) \]

Where \( t_{\text{comp}} \) is the delay associated with computing a backup route,
\( t_{\text{prop}} \) is the average propagation delay per hop,
\( t_{\text{ad,proc}} \) is the average admission control delay per hop,
and \( N \) is the number of hops along the chosen route.

Which of the three components dominate the path setup time depends on many factors, such as network size, hop count, and perhaps most importantly, implementation issues. This makes it very difficult to estimate the setup latency time for any particular scheme. So, that will not be attempted here. However, for any scheme, the path setup phase has to be repeated upon every admission control failure. Therefore, it is possible to compute the average number of algorithm iterations for any of the methods.

In the simulations of scenarios 2 and 3, the number of admission control failures was recorded during each attempt to establish a backup lightpath. Graph 4.17 and Graph 4.18 show (for Network B and Network C respectively) the average number of admission control failures that occur before a backup path is successfully established. All schemes are shown. With the exception of the Educated_DFS scheme, the number of admission control failures per backup lightpath is negligible (in other words, the backup is usually successfully established on the first try for these schemes). With the Educated_DFS approach, there are a substantial number of admission control failures. The number of failures drops as network load increases, but even in the best case of extremely high network load, more than four (three failures, one success) backup establishment
procedures need to be run per connection. In other words, the ingress may have to invoke Dijkstra’s algorithm four times before connection success occurs. And, if the route computation time is dominated by Dijkstra’s algorithm (which it is for large network sizes [29]), then the time spent computing paths before successful backup establishment is much higher for the Educated_DFS than with any other scheme. Furthermore, each scheme uses the same resource reservation procedure, so if there are no admission control failures, then the time taken to complete the resource reservation phase for an identical path should be the same with all schemes. However, again, since the number of admission control failures encountered by the Educated_DFS scheme is significantly higher than any other, an ingress OXC using it will spend more total time reserving and releasing resources along paths compared to any other scheme.

The situation would be worse for the Educated_DFS, were it not for the cost function, $\frac{1}{\text{shared protection lambdas}}$. Graph 4.19 shows the average number of algorithm iterations in the case where links were not weighted with reference to this link-state metric (i.e. “perceived friendly” links are all assigned a cost equal to ‘1’). The graph indicates that on average, using the new link state metric helps eliminate a single admission control failure (and thus, an extra repetition of the algorithm).

The above analysis should convince the reader that the Educated_DFS scheme could potentially suffer from excessive backup path setup latency times.
Graph 4.17: Scenario 2; Average Number of Admission Control Failures Before Connection Success, as a Function of Offered Load – A Comparison Between Schemes

Graph 4.18: Scenario 3; Average Number of Admission Control Failures Before Connection Success, as a Function of Offered Load – A Comparison Between Schemes

Graph 4.19: Scenario 2; Utility of the Shared Protection Bandwidth Link Metric – Comparing Educated DFS With and Without the Metric
4.6 Scenario 4 – Changing Link Bandwidth

As demand for network services increase, link bandwidth will need to expand accordingly. It was worthy to know how the simulation results would change in response to changing link bandwidth. Therefore, the simulation of scenario 2 was repeated for higher link bandwidths. First, the number of fibers per bundle was increased from 10 to 20, and then to 50. Offered load was varied and the relevant parameters were recorded.

The results of these two experiments are shown in graph 4.21 and 4.22 (data at very low network loads was not plotted). The original graph from scenario 2 (4.20) is also shown for comparison purposes.

Each of the graphs appears virtually indistinguishable. The performance of a scheme at offered load ‘x’ is the same at offered load ‘2x’ when the link capacity is doubled, and offered load ‘5x’ when the capacity is increased by a factor of five. The overall conclusion here is that all the algorithms scale properly to changes in link bandwidth.
Graph 4.20: Scenario 4; Ratio of Total Backup Resources Used to Total Primary Resources Used, as a Function of Offered Load - No Change in Link Bandwidth.

Graph 4.21: Scenario 4; Ratio of Total Backup Resources Used to Total Primary Resources Used, as a Function of Offered Load - Link Bandwidth Multiplied by 2.

Graph 4.22: Scenario 4; Ratio of Total Backup Resources Used to Total Primary Resources Used, as a Function of Offered Load - Link Bandwidth Multiplied by 5.
4.7 Scenario 5 – Simulating on Networks with Non Uniform Link Bandwidth

It was also interesting to observe how schemes would perform in networks with variable size links. Thus, a new simulation was constructed using network C as the testbed. For the simulation, all fibers still supported an equal number of lambdas (36 to be exact). Instead, link bandwidth was adjusted by varying the number of fibers on a link. Network C (N. American UUNET network) was derived from [28], where links are actually described by the number of trunks between nodes and the bit rate over each trunk. This provided a basis to decide how the fiber count should be adjusted on each link of the simulation testbed. The smallest bandwidth trunks in [28] are single DS-3 (45 Mbps) lines. These were modeled as a single bi-directional fiber pair in the testbed. Any other link was scaled proportionally. For example, in [28] the link between Los Angeles and Phoenix is composed of 3 DS-3’s (4.5Mbps) and 8 OC-12’s (620Mbps). This translates to \[ \frac{3\times45 + 8\times620}{45} = 115 \] bi-directional fiber pairs in the testbed. The number of fibers for the rest of the links is given in Appendix B.

As with previous simulations, a node was chosen as a source or destination with a frequency proportional to its direct access to bandwidth (total number of fibers attached to it).

Results
Graphs 4.23 to 4.32 depict the results of this simulation. While the results still permit one to rank the schemes in the same order of performance, there are some key differences between these sets of graphs and the ones from scenario 3 (which was based on the same topology). For one thing, the data from these graphs do not change as dramatically with network load (compare graphs 4.23 - 4.26 to 4.12 - 4.15). This is especially the case with the best performing schemes, where the average resource usage (see graph 4.24) is consistently low over all network loads.

The source of these discrepancies is obviously related to the differences in link bandwidth allocation. In this simulation, the bandwidth of various links is distributed over a very wide range of values. Still, links can be categorized by bandwidth range.
The 82 links, composed of less than 20 bi-directional fibers per trunk were classified as low bandwidth links, while the rest (74) were classified as high bandwidth links. A snapshot of the network was taken for each of the schemes at a variety of network loads after an arbitrary number of connections had been serviced. All filled links were recorded. It was found that at loads below 3000 Erlangs, over 90% of filled links were low bandwidth links (see graphs 4.28 – 4.32). Even at loads higher than 3000 Erlangs, a disproportionately large number of filled links were of low bandwidth type. These graphs add enough evidence to explain what happens. Links between major hubs consist of many fibers (583 fiber pairs in some cases), while smaller cities are connected by only a few fibers (in some case only one fiber pair). As a result, most requests are for connections between major hubs, so primary and backup lightpaths between these major hubs reside over the fat pipes linking them together. The paths are also short which explains why primary and backup resource usage is low (see graphs 4.25 and 4.26). Even as network load becomes high, these high bandwidth links still have free available lambdas, so resource usage remains quite stable. Eventually, when network loads become very high, some of these high bandwidth links do become filled; so many primary lightpaths cannot be installed, and those that can may have to traverse longer paths. Backup lightpaths can still be placed on links with no free bandwidth, so backup resource usage remains relatively constant.

To exacerbate the situation, many low bandwidth links are filled at even low network loads. Many lightpaths that need to use such links cannot be established. This explains why a substantial number of connection requests are refused even at low and moderate network loads (graph 4.27). Furthermore, as offered load increases, and more of these smaller bandwidth links are removed from the routing picture, it is again not surprising that the performance of the difference schemes start to converge. To put it in another way, those schemes that work hard to exploit backup sharing possibilities are indirectly being penalized.
Graph 4.23: Scenario 5; Ratio of Total Backup Resources Used to Total Primary Resources Used, as a Function of Offered Load – A Comparison Between Schemes
Graph 4.25: Scenario 5: Average Resource Usage per Primary Linkpath as a Function of Offered Load – A Comparison between Schemes

Graph 4.26: Scenario 5: Average Resource (Primary + Backup) Usage Per Connection, as a Function of Offered Load – A Comparison between Schemes
Graph 4.26: Scenario 5; Average Number of Additional Backup Lambdas Reserved per Backup Lightpath, as a Function of Offered Load – A Comparison Between Schemes

Graph 4.27: Scenario 5; Percentage of Connection Requests Refused, as a Function of Offered Load – A Comparison Between Schemes
Graph 4.28: Scenario 5; Inspecting Filled Links, As a Function of Offered Load – SPF Routing

Graph 4.29: Scenario 5; Inspecting Filled Links, As a Function of Offered Load – Joint Routing

Graph 4.30: Scenario 5; Inspecting Filled Links, As a Function of Offered Load – Educated_DFS
Graph 4.31: Scenario 5; Inspecting Filled Links, As a Function of Offered Load – Unrestricted_TH

Graph 4.32: Scenario 5; Inspecting Filled Links, As a Function of Offered Load – Restricted_TH
4.8 Scenario 6 Node Disjoint Lightpaths

Not all customers are content with protection from link failures only. Optical cross-connects can malfunction, and therefore the possibility of node failures may need accounted for when establishing backup lightpaths. Borrowing terminology from [23], connections requiring protection from link failures will be called LF connections, while those that demand protection from both link and node failures will be called LNF connections.

The routing and resource reservation procedures for an LNF backup lightpath differ from the standard LF backup lightpath. In particular the following changes need to be made:

- The backup lightpath cannot share the same link or node (source and destination excluded) as the primary lightpath. If the primary route is known, the backup routing algorithm must exclude all links attached to any intermediate nodes along the primary path\(^8\). This is a much stronger constraint than the one enforced for routing LF backups.

- During the resource reservation phase of a backup lightpath for an LNF connection, intermediate OXCs should record the corresponding primary nodes in addition to the primary links. A sample protection table shown in fig 4.4, is for a link hosting some LF backup lightpaths and some LNF backups. Unlike the table of fig 3.3, there are rows designated for the primary nodes of LNF connections, along with the pertinent connection IDs. OXCs do not add the IDs of LF connections into rows related to nodes. In the sample table, the worst-case fault is the failure of node 'e', because it results in the highest number of backup activations (six) over the link. To support this enhancement, the signaling protocol should be augmented to indicate if the connection is of type LF or type LNF.

\(^8\) The modifications required for joint routing are given in [26].
Some changes specific to each scheme are also needed. The Troublesome Hop schemes are generalized to Troublesome Component schemes, as the worst-case fault of a backup link can be either a link or node failure. In the Unrestricted Troublesome Component approach (Unrestricted_TC), all troublesome components (nodes or hops) are advertised into link-state packets. But the notion of a troublesome node is only important to an ingress establishing an LNF connection. In the case where the ingress is attempting to install a regular LF connection, all advertised troublesome nodes are ignored. With the Restricted Troublesome Component approach (Restricted_TC), if multiple troublesome components exist, only one can be advertised into the IGP. In instances where a link had one or more troublesome nodes and also one or more troublesome hop, the troublesome hop was advertised. This was done because while a troublesome hop is relevant to both LF and LNF connections, a troublesome node is only relevant to LNF connections⁹.

To test the effects of the node-disjointness constraint on the schemes, a simulation was conducted using network B at two fixed offered loads (moderate load of 770 Erlangs, and a high load of 1000 Erlangs). Performance was evaluated as a function of the

---

⁹ A special case is where a troublesome node also happens to be one endpoint of a troublesome hop. Here, it was decided that link state packets would express the troublesome hop with an addendum that states which endpoint is also a troublesome node.
percentage of requests for LNF connections. Other parameters were the same as in scenario 2.

Results
Graphs 4.33 and 4.34 show the average resource usage per connection for moderate and high offered loads respectively. The resource usage is quite stable in both cases until the percentage of LNF requests increase beyond a threshold point (between 40% and 50% in both cases). Then, resource usage starts to increase. Each graph is separated into its constituent components (average primary resource usage and average backup resource usage to be precise) in graphs 4.35 to 4.38. The graphs indicate that primary resource usage remains relatively constant even when the percentage of LNF requests increase beyond the threshold. But, the backup resource usage increases dramatically.

To figure out why, a snapshot of the network state was taken after an arbitrary number of connections had been serviced. The protection table of every link was examined, and its worst-case failure recorded. The following observations were made. When the percentage of LNF requests is low (below the threshold), the worst-case fault of most backup links is primary hop failure. (See graphs 4.39 and 4.40, which show the ratio of the number of worst-case node faults to the number of worst-case link faults observed in the snapshot). This makes sense because for any link protection table, it is unlikely that rows associated with nodes will have many connection IDs when the percentage of LNF requests is low. Beyond the threshold point, worst-case faults tend to become node faults. In addition, because there are usually fewer nodes than links in any topology, it becomes more likely that a backup lightpath’s corresponding primary route traverses the worst-case fault of any potential backup link. As a consequence, backup lightpaths of LNF connections invariably require more resources. It may be difficult to find backup paths with available resources, a fact which is verified by observing the increase in blocking probability in accordance to an increase in percentage LNF requests (see graphs 4.41 and 4.42).

Inspecting the graphs 4.41 and 4.42 more closely, it may seem strange that even when the percentage LNF requests is low there are a sizeable number of connections refused for most schemes. The exception is with the Joint Routing scheme (see graph 4.41 in
particular). There are some situations where it is impossible to find a node disjoint backup path, if the primary path is routed beforehand. For instance, in fig 4.5, if the shortest hop route between nodes S and D is reserved for the primary lightpath, it is impossible to place a secondary node-disjoint path between these same two points. The same can be said if for certain node pairs in network B. In most schemes the primary is routed before the backup, which explains why some LNF connections have to be refused. However, with Joint Routing, the primary and backup are routed in tandem, and if a node-disjoint path pair exists, Suurballe's algorithm will find it. As can be seen from the graphs, at moderate network loads, where resource constraints are not an issue, the Joint Routing scheme has no problem finding node-disjoint lightpaths, while the other schemes have some difficulty. During high network loads, resource constraints become the dominant factor behind connection refusals, so this advantage of Joint Routing is less conspicuous.

![Figure 4.5: Difficulty Routing an LNF Connection – Shortest Path Between S and D Depicted by Dotted Line](image)

In summary, it was observed that permitting LNF connections into the network does not noticeably increase resource consumption, provided the percentage of LNF requests is kept below some limit. Also, in situations where a high performance scheme (Educated_DFS or Unrestricted_TC) cannot find a backup path for an LNF connection, it may be worthwhile to retry the entire primary-backup establishment procedure using Joint Routing instead.
Graph 4.33: Scenario 6; Average Resource Usage per Connection, as a function of LNF Request Type – Moderate Offered Load

Graph 4.34: Scenario 6; Average Resource Usage per Connection, as a function of LNF Request Type – High Offered Load

Graph 4.35: Scenario 6; Average Resource Usage per Primary Lightpath, as a Function of LNF Request Type – Moderate Offered Load

Graph 4.36: Scenario 6; Average Resource Usage per Primary Lightpath, as a Function of LNF Request Type – High Offered Load
Graph 4.37: Scenario 6; Average Additional Resources Required per Backup Lightpath, as a Function of LNF Request Type – Moderate Offered Load

Graph 4.38: Scenario 6; Average Additional Resources Required per Backup Lightpath, as a Function of LNF Request Type – High Offered Load

Graph 4.39: Scenario 6; Ratio of Number of Links Whose Worst Case Failure is a Link to Number of Links Whose Worst Case Failure is a Node, as a Function of LNF Type Requests – Moderate Offered Load

Graph 4.40: Scenario 6; Ratio of Number of Links Whose Worst Case Failure is a Link to Number of Links Whose Worst Case Failure is a Node, as a Function of LNF Type Requests – High Offered Load
Graph 4.41: Scenario 6; Percentage of Connection Requests Refused, as a Function of LNF Type Requests – Moderate Offered Load

Graph 4.42: Scenario 6; Percentage of Connection Requests Refused, as a Function of LNF Type Requests – High Offered Load
4.9 Scenario 7 – Introducing Multiple Service Classes

Implicit in all discussions thus far is that connections are assured protection from only single component failures. In reality, different connections demand different degrees of reliability, so a network needs to offer multiple levels of service. The schemes presented in this work were designed under the supposition that all connections are content with the single failure assumption. It is prudent to observe how the schemes are affected when multiple service class connections coexist within the same network.

To accomplish this, connection types were partitioned into three distinct service class types:

1) GOLD (commonly known as 1:1 protection) – Two disjoint lightpaths (primary + backup) are provided. Resources for the backup are dedicated and therefore not shared by any other backup lightpaths. Working traffic is switched to backup lightpath if a failure occurs on the primary.
2) SILVER\(^{10}\) (commonly known as 1:N protection) – Two disjoint lightpaths (primary + backup) are provided. Resources for the backup can be shared with other backups where appropriate.
3) BRONZE (no protection) – the connection is not offered a backup lightpath.

This list is by no means complete. Other reliability services (such as 1+1 protection [22]) are likely offered, but the above set captures the needs of many customers, and is sufficient for the purposes of this discussion.

As an initial experiment, a simulation was conducted using network B, under the same constraints as scenario 2. The only difference was that incoming calls could demand any of the three forms of service, under the following distribution:

- GOLD 25%
- SILVER 50%
- BRONZE 25%

\(^{10}\) Up to this point, all connections have been implicitly assigned type SILVER
This is a reasonable distribution at face value, because the resource burden imposed by GOLD level connections should be somewhat offset by the BRONZE connections.

The backup lightpaths of type GOLD were made to take the shortest disjoint free path available. The value of ‘N’ in type SILVER connections was not specified, so the degree of backup sharing was not directly restricted. Also, all connections were assumed to be of type LF (i.e. node disjointness was not considered).

The resource reservation phase for backup lightpaths had to be enhanced so intermediate OXCs knew the reliability type of the connection, and could reserve resources accordingly. Once again, the simulation was run until 1,000,000 connection requests had been made.

The results
Graph 4.43 shows the average resource usage per connection for the Restricted Troublesome Component scheme as a function of offered load (the curves of other schemes have been omitted for the sake of clarity). Included in this graph is the curve from scenario 2, where the service class distribution was (GOLD, SILVER, BRONZE)=(0%, 100%, 0%). The results show that more resources are needed to support the mixed service class connection set. This is true, despite the opposite protection requirements of the GOLD and BRONZE connections. Clearly, the dedicated protection demanded by GOLD connections can be very taxing on the network. Naturally, the high resource consumption of GOLD connections also leads to higher network blocking probability (see graph 4.44).
Graph 4.43: Scenario 7; Average Resource Usage (Primary + Backup) per Connection, as a Function of Offered Load – ALL SILVER vs MIXED Service Class Distribution

Figure 4.44: Scenario 7; Percentage of Connections Refused, as a Function of Offered Load – ALL SILVER vs MIXED Service Class Distribution
Solution to the Problem

One way to reduce the resource consumption of a connection is to redefine the manner in which is protected without compromising its reliability constraints. Gold connections are synonymous with 1:1 protection – a primary lightpath is protected by a dedicated backup lightpath, which remains on standby. Any connection receiving 1:1 protection is protected against any combination of faults with the exception of the simultaneous failure of a component along the primary route and a component along the backup. But the same protection can be provided without dedicating resources to GOLD backup lightpaths. To illustrate, consider a link with one protection lambda already serving a number of shared backup lightpaths (all type SILVER). Next, suppose a new GOLD backup lightpath is actually placed over the link without occupying an extra backup lambda (that is, it actually shares resources with the other SILVER backups). Based on earlier rules, this can only happen if the GOLD primary path does not overlap with the primary paths of the existing SILVER connections. Under this scenario, all connections (GOLD included) are protected against single component failures. Now, if multiple failures occur, there is a chance that two or more backup activations are attempted over the link. In the case where one of these backup activations is for a GOLD connection, the GOLD backup can be given priority, and thus temporary ownership of the protection lambda. This is allowed because SILVER type connections do not demand (nor do the associated customers pay for) protection from multiple failures. Therefore, a single backup GOLD lightpath can share resources with any number of SILVER backup lightpaths, provided appropriate precedence rules are set. Of course, two GOLD backup lightpaths cannot share the protection bandwidth or else there may be contention problems in the event of multiple failures. If these rules are enforced throughout the network, GOLD connections are protected from multiple failures (in a manner equivalent to standard 1:1 protection), while SILVER connections continue to be guaranteed protection from single failures.

To support these new regulations, modifications need to be made to the resource reservation phase of a backup lightpath. First, the corresponding primary hops of a GOLD backup must now be entered into the protection table of backup links (previously protection tables were only relevant to type SILVER connections). An intermediate OXC calculating the number of protection lambdas required over one of its links needs to compute two values:
1. The number of backup activations (GOLD or SILVER) that occur over the link in the worst-case failure scenario.

2. The total number of GOLD backup lightpaths presently residing over the link.

The number of protection lambdas required is equal to the higher of these two values.

Previously, under the old policy, GOLD backup lightpaths were placed on the shortest available path (disjoint of the primary), in order to reduce resource consumption. Now, since GOLD backups can be shared, they are routed under the same paradigm governing regular SILVER connections. But first, some scheme-specific enhancements are necessary. With the Troublesome Component approaches, an ingress OXC deciding on a backup route for a GOLD connection would benefit by knowing if the number of protection lambdas on a link is equal to the number of GOLD backups on a link. If this equality holds true, then using this link for another GOLD backup will definitely require an additional lambda. So, as well as advertising troublesome components, a link-state packet should indicate if the number of GOLD backup lightpaths on a link is actually equal to the protection lambdas on the link. Other schemes are unaltered.

A simulation was carried out to test the utility of the new GOLD definition. In this simulation, network B was used and operated at a fixed offered load (760 Erlangs). Two types of connections were permitted into the network: GOLD and SILVER, in varying proportions. This was done using the old and new definitions of the GOLD service class. All schemes were simulated.

The Results
The series of graphs 4.45 to 4.54 depict the results of the simulation. For each scheme, there are two graphs:

1. A graph showing the average resource usage per connection under both GOLD definitions.

2. A graph comparing the connection refusal rate under both definitions.

Both graphs are given as function of the percentage of GOLD type connections.
The graphs show that the new definition helps reduce both the resource consumption and blocking probability of connections. However, when almost all the connections are of type GOLD, there is not much advantage to using the new definition. This is obvious because if there are no SILVER backups in the network, there can be no sharing anyway.

In most networks, not every customer will request GOLD level service. Thus, there is clear advantage to replacing the old definition of GOLD protection (i.e. strict 1:1 protection) with the new one.

**Graph 4.45: Scenario 7; Average Resource Usage Per Connection, as a Function of the Proportion of Type GOLD Requests – SPF Routing**

**Graph 4.46: Scenario 7; Percentage of Connection Requests Refused, as a Function of the Proportion of Type Gold Requests – SPF Routing**
Graph 4.47: Scenario 7; Average Resource Usage Per Connection, as a Function of the Proportion of Type GOLD Requests – Joint Routing

Graph 4.48: Scenario 7; Percentage of Connection Requests Refused, as a Function of the Proportion of Type Gold Requests – Joint Routing

Graph 4.49: Scenario 7; Average Resource Usage Per Connection, as a Function of the Proportion of Type GOLD Requests – Educated_DFS

Graph 4.50: Scenario 7; Percentage of Connection Requests Refused, as a Function of the Proportion of Type Gold Requests – Educated_DFS
Graph 4.51: Scenario 7; Average Resource Usage Per Connection, as a Function of the Proportion of Type GOLD Requests – Unrestricted_TC

Graph 4.52: Scenario 7; Percentage of Connection Requests Refused, as a Function of the Proportion of Type GOLD Requests – Unrestricted_TC

Graph 4.53: Scenario 7; Average Resource Usage Per Connection, as a Function of the Proportion of Type GOLD Requests – Restricted_TC

Graph 4.54: Scenario 7; Percentage of Connection Requests Refused, as a Function of the Proportion of Type GOLD Requests – Restricted_TC
Chapter 5

Pre-assigning Lambdas to Backup Lightpaths

5.1 Motivation and Concerns

Although it has not been mentioned before, the current policy described for assigning protection bandwidth to backup lightpaths over a link may not be conducive to achieving fast recovery times. Under the current policy, a group of protection <fiber,lambda> pairs, \( \Lambda = \{(\lambda_1, f_1), (\lambda_3, f_1), (\lambda_2, f_2), \ldots \} \) is reserved for the set of backup lightpaths \((l_1, l_2, l_3, \ldots l_m)\) residing over the link. When a failure occurs, a subset of these backup lightpaths are activated, and each of these activated lightpaths needs to be assigned a specific <fiber,lambda> pair for that link. These lightpath<->lambda mappings can be determined by the OXC downstream of the link in question, and subsequently communicated to its upstream peer. Traffic cannot flow on the protection path until all optical cross-connects along its route have established a mapping and configured their switching fabrics properly. Communication of port maps can be done using an appropriate label distribution protocol, but the point is that it must be done on the fly, during the recovery period. This fact naturally implies slower recovery times. If this process becomes the bottleneck of the restoration phase, then it is a problem that needs to be corrected.

As far as recovery times are concerned, it would be more beneficial to assign lightpaths to lambdas beforehand. Then, port maps do not need to be determined and communicated between peer OXCs during restoration. Rather, once a failure occurs, only lightweight notification messages [19] (initiated by the respective ingress OXCs of the failed lightpaths) need to be sent along the backup paths. This will trigger each intermediate OXC to alter its switching matrix based on the pre-assigned port mappings, and eventually the correct end-to-end backup lightpath is formed. Assigning lightpaths to lambdas is a many-to-one mapping. Two backup lightpaths can be assigned to the
Figure 5.1: (a) Five Backup Lightpaths Sharing 3 Protection Lambdas on a Link (b) Individual Lightpaths Multiplexed Onto Specific Lambdas.

same lambda if and only if their corresponding primary paths are non-overlapping (honoring the single component failure guarantee).

It may not be obvious, but pre-assigning lightpaths to lambdas may actually force one to allocate more protection bandwidth than previously. For instance, consider the lightpath placement scenario below (fig 5.2). There are three primary lightpaths shown along with their three corresponding backups. How much protection bandwidth must be reserved on link A-B? Link A-B's worst-case failure hops are shown in bold. There are three such hops, each of which supports two of the three primary lightpaths. Then, based on the explanation given earlier (Section. 2.2), two protection lambdas need to be reserved over link A-B. But, that is only true if one does not pre-assign lightpaths to lambdas. If lambdas were actually pre-assigned in this circumstance, there is no way to map these three lightpaths onto two lambdas, because every pair-wise combination of the three primary lightpaths has some overlap. Thus, there is no other option, but to pre-assign three protection lambdas, violating the principles stated earlier. Still, it is felt that

Figure 5.2: Pathological Case – More Protection Lambdas Required Over Link A-B When Pre-assigning Lambdas to Lightpaths
situations like this are pathological in nature, and occur with small probability. This issue aside, there is still the difficulty of finding the best lambda-to-lightpath mapping on a link, if pre-assigning is done. The “best” mapping is the one that minimizes the number of protection wavelengths needed on the link. This problem is analogous to the famous graph-coloring problem. To model it as a graph-coloring problem, treat the backup lightpaths as nodes, and if two such backup lightpaths have overlapping primary paths, connect the nodes together. Then, obtaining the minimum coloring of the resulting graph actually indicates which lightpaths should be multiplexed over which lambdas, in order to minimize the number of protection lambdas. For the case of fig 5.2, the mapping problem can be modeled by the graph shown below. Here, upon inspection, it is easy to see that a minimum of 3 lambdas is required.

Unfortunately, the problem is not so straightforward when dealing with 10’s or 100’s of lightpaths, with arbitrary contentions. Still, procedures such as the backtracking algorithm in Appendix C, can be used to find the minimum number of colors required to properly color any size graph. However, the graph coloring problem is widely known to be NP-complete, and so the computation time associated with such algorithms is too excessive for a practical setting. This is further compounded by the fact that the algorithm must be run on an OXC, upon every admission control of a backup lightpath, and after the removal of any backup lightpath from a link.
Likely, heuristic means are needed for pre-allocating lambdas for new backup lightpaths. In this work, a simple heuristic procedure was defined. It should be used during the admission control process for a backup lightpath.

5.2 Heuristic Lightpath Multiplexing Algorithm

Phase 1
During the admission control of a candidate backup lightpath, each existing protection lambda on the link is checked sequentially, to see if the lightpath can be multiplexed onto it. (If the link does not have any designated protection lambdas, then a new lambda is required, and should be picked from the remaining pool of free lambdas). The lightpath can only be multiplexed with other backups on a given protection lambda if its primary path does not overlap with theirs. If there is no overlap, then the lightpath can be multiplexed on that lambda and admission control succeeds. If there is some overlap, the OXC records the IDs of all the lightpaths that contend with the backup. This list of contending lightpaths is kept aside for the time being, and the next protection lambda is checked. This procedure continues until a multiplexing attempt succeeds or all designated protection lambdas have been checked, but to no avail. In the latter case, the OXC should have a list of contending lightpaths for each protection lambda. It will use this list in a 2nd phase called reassignment.

Phase 2 - Reassignment
In the reassignment phase, each protection lambda is revisited sequentially. This time, the OXC will attempt to re-multiplex each the offending lightpaths of a given lambda onto other protection lambdas. In other words, it will try to reassign the mappings of the contending lightpaths of a particular protection lambda to accommodate the new lightpath. If all the contending lightpaths on a given protection lambda are successfully reassigned, then the backup lightpath can be multiplexed on the lambda without problem, and the admission control process succeeds. Otherwise, any reassignments are voided, and the procedure is repeated for the next protection lambda. If the reassignment phase is unsuccessful for all designated protection lambdas, then the decision is made that the backup lightpath cannot be accepted without requiring a new lambda. (Note, that this decision may be wrong, but from a practical viewpoint,
eventually a decision needs to be made, unless the OXC is willing to resort to an exhaustive search such as that from Appendix C.) At the conclusion of this process, if a given lightpath-to-lambda mapping has changed for any old backup lightpath, the new mapping must be communicated (through some label distribution protocol) to its upstream peer.

Note that the number of multiplexing attempts and the number of reassignment attempts is bounded by the number of protection lambdas on the link.

5.3 Simulation – Comparison with Original Approach

It is useful to see what kind of affect enforcing lambda pre-allocation to backup lightpaths may have on any given backup sharing scheme. So, for network 2, all schemes were simulated with and without the constraint of lambda pre-assignment in place. In the case of pre-assignment, the admission control process was modified to include the heuristic lightpath-to-lambda mapping described in Section 5.2.

For either case, nine hundred pre-ordered lightpaths were placed on an initially empty network. Also, link bandwidth on the network was not restricted. Each simulation was repeated 5 times, each time using a different ordered sequence of connection requests.

Results

Reported results are the average of those values obtained in the repeated experiments. Table 5.1 below shows the total amount network resources used in cases where lambda pre-assignment is enforced and not enforced (to reiterate, non pre-assignment had been implicitly used in all previous simulations). The results show pre-assigning lambdas to backup lightpaths will demand only a slight increase in backup resource overhead in comparison to regular non-pre-assignment. For any scheme, less than 0.6 percent extra protection bandwidth was required to support pre-assignment. Indeed, it was observed that for any simulation, a link had to reserve more bandwidth than that indicated by its protection table on just a few occasions. It is not clear whether the extra bandwidth required was due to pathological cases like that described in section 5.1, or simply a liability of the heuristic lightpath-to-lambda mapping (graph coloring) algorithm. Regardless of this issue, the results show that pre-allocating lambdas to backup
lightpaths should not degrade the performance of a backup routing scheme significantly. This confirms what was suggested in [19].

<table>
<thead>
<tr>
<th></th>
<th>Total Primary Resources Used</th>
<th>Total Backup Resources Used</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No Pre-assignment</td>
<td>Pre-assignment</td>
</tr>
<tr>
<td>SPF Routing</td>
<td>2559.0</td>
<td>2559.0</td>
</tr>
<tr>
<td>Joint Routing</td>
<td>2562.8</td>
<td>2562.8</td>
</tr>
<tr>
<td>Educated_DFS</td>
<td>2559.0</td>
<td>2559.0</td>
</tr>
<tr>
<td>Unrestricted_TH</td>
<td>2559.0</td>
<td>2559.0</td>
</tr>
<tr>
<td>Restricted_TH</td>
<td>2559.0</td>
<td>2559.0</td>
</tr>
</tbody>
</table>

Table 5.1: Determining The Resource Penalty of Pre-assigning Lambdas to Lightpaths.
Chapter 6

Conclusions

The rapid growth of the Internet in conjunction with the gaining popularity of bandwidth on demand services has fostered much interest in the area of dynamic wavelength routed optical networks. Furthermore, the demand for reliable connections dictates the need for protection against failures.

This work has focused on the topic of pre-allocated path protection, where a backup lightpath is created with the sole purpose of protecting a working lightpath. It was accepted that backup lightpaths could share resources with others under certain conditions. And so, the objective was to devise ways to exploit sharing opportunities in order to reduce the resource burden backup lightpaths impose on the network. The schemes studied were necessarily heuristic and could be conceivably applied to any online distributed routing environment.

Five schemes were analyzed:
1) Shortest Path First Routing
2) Joint Routing
3) Educated Depth First Search
4) Unrestricted Troublesome Hop Scheme
5) Restricted Troublesome Hop Scheme

The first two are fairly conventional approaches to backup routing, while the latter three were developed in the course of this research. Although topology characteristics and network loading play an important role, the Educated Depth First Search and Unrestricted Troublesome Hop scheme consistently outperform all others. Simulations show that they could reduce backup resource consumption in typical networks by as much as 50%. In addition, using these schemes generally leads to lower network blocking probabilities over any other scheme. These facts demonstrate a clear
advantage to judiciously picking low cost backup routes for individual connections in order to benefit the network in the long run.

The performance gap between schemes narrows when path choices are limited. This happens if the topology inherently suffers from low connectivity or when network load becomes high for any topology.

The Educated Depth First Search scheme usually needs several search attempts before the lowest cost backup route is obtained. This could potentially lead to long setup latency times. Also, using the Unrestricted Troublesome Hop scheme means link-state packets are not bounded in size. One solution is to limit the number of troublesome hops advertised into a link state packet. This may lead to the proposition of the Restricted Troublesome Hop scheme, in which only fixed number of troublesome hop can be declared. However, when only one troublesome hop is declared, simulations have shown it will result in noticeably higher (as much as 30%) backup resource usage than the unrestricted version. Appendix D illustrates how the performance of the Restricted Troublesome Hop scheme changes when more troublesome hops are advertised. In fact, allowing just one more troublesome hop to be advertised has significantly improves the performance of the Restricted Troublesome Hop scheme.

It is more difficult to install node-disjoint connections compared to their link-disjoint counterparts. In situations where the primary path is chosen independently of the backup, it may be impossible to find a node-disjoint backup route. This is not a problem for the Joint Routing scheme, which can always find a node-disjoint pair when one exists.

Traditional 1:1 protection can be very expensive in WDM networks. But it is possible to define a new service class in which the backup lightpath can actually share resources with other backups and still offer the same reliability as traditional 1:1 protection. This was done in this work, and simulations show it can alleviate much of the resource burden associated with the traditional approach.
Finally, the concept of pre-assigned lightpath-to-lambda mappings is important under the supposition that it can improve fault recovery times. Simulations suggest that pre-assigning lightpaths to lambdas will have little effect on network resource consumption.

Future research is still possible. In particular, it is useful to determine whether network reconfiguration can significantly reduce resource usage. While primary lightpaths cannot be reconfigured, backup lightpaths can, because they are normally not operational most of the time. Perhaps existing placement of backup lightpaths can be periodically optimized to minimize backup resource consumption. In addition to resource consumption, reconfiguring lightpaths will have implications on processing load and overhead for OXCs. Furthermore, procedures need to be developed to enforce synchronization between nodes trying to reconfigure at the same time, and to ensure recovery times remain bounded. These issues need to be studied further.
References


[28] UUNET website:
   “http://www.uu.net/network/maps/northam/?SetLang=en”

Appendix A

Solving for the Optimal Solution

The optimal greedy routing solution is defined as the one finds the primary-backup pair that requires the smallest resource usage. Given full information about the state of the network, the problem can be modeled by an integer programming formulation. The solution to this formulation will in turn lead to the optimal solution. The formulation is given below. It is a slight modification to that given in [21].

Assuming a directed graph of N nodes and E edges, with source 's' and destination 'd', four terms can be defined:

\( x_{ij} \): defined as the working flow on link i-j. It is equal to 1 if the primary lightpath traverses link i-j, and 0 otherwise.

\( y_{ij} \): defined as the backup flow on link i-j. It is equal to 1 if the secondary lightpath traverses link i-j, and 0 otherwise.

\( z_{ij} \): defined as the number of additional protection lambdas (1 or 0) that need to be reserved over link i-j.

\( \theta_{ij}^{uv} \): equal to 1 if the highest number of backup activations over link u-v occurs when link i-j fails.

Each first 3 terms constitute 2L integer variables ('L' is the number of undirected links in the graph) that can be one of two values: 0 or 1. The last term is part of an array (of size 4L^2) of constants whose values depend on the prevailing placement of lightpaths in the network.
The objective is to minimize the following function:

$$\text{Minimize } \sum_{(i,j) \in E} x_{ij} + \sum_{(i,j) \in E} z_{ij} \quad \text{(objective)}$$

Subject to the following constraints:

1. $$\sum_{j} x_{ij} - \sum_{j} x_{ji} = 0 \quad i \neq s, d$$  
2. $$\sum_{j} x_{sj} - \sum_{j} x_{js} = 1$$  
3. $$\sum_{j} x_{dj} - \sum_{j} x_{jd} = -1$$  
4. $$\sum_{j} y_{ij} - \sum_{j} y_{ji} = 0 \quad i \neq s, d$$  
5. $$\sum_{j} y_{sj} - \sum_{j} y_{js} = 1$$  
6. $$\sum_{j} y_{dj} - \sum_{j} y_{jd} = -1$$  
7. $$z_{uv} \geq \theta_{ij}^{uv} (x_{ij} + y_{uv} - 1) \quad \forall (i, j) \quad \forall (u, v)$$  
8. $$x_{ij}, y_{ij}, z_{ij} \in \{0, 1\}$$  
9. $$x_{ij} + y_{ij} \leq 1 \quad \forall (i, j)$$
Solving Integer Programming problems is typically computationally intensive for even the smallest of networks. Normally software packages such as CPLEX™ are used.
Appendix B

Bandwidth of Individual Links For Scenario 5

In the simulation of scenario 5, the number of fibers on each link of Network C was different. The specific number of fibers allocated to each link is shown below. Nodes are referenced numerically. These nodes represent actual cities in the UUNET network [28].

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Link:2-0:</td>
<td>111 fibers</td>
<td>Link:26-25:</td>
<td>96 fibers</td>
<td>Link:48-22:</td>
<td>1 fibers</td>
</tr>
<tr>
<td>Link:2-1:</td>
<td>1 fibers</td>
<td>Link:27-0:</td>
<td>55 fibers</td>
<td>Link:48-30:</td>
<td>6 fibers</td>
</tr>
<tr>
<td>Link:3-1:</td>
<td>111 fibers</td>
<td>Link:27-2:</td>
<td>27 fibers</td>
<td>Link:48-32:</td>
<td>13 fibers</td>
</tr>
<tr>
<td>Link:3-2:</td>
<td>14 fibers</td>
<td>Link:27-6:</td>
<td>13 fibers</td>
<td>Link:48-33:</td>
<td>1 fibers</td>
</tr>
<tr>
<td>Link:4-1:</td>
<td>1 fibers</td>
<td>Link:27-11:</td>
<td>55 fibers</td>
<td>Link:48-46:</td>
<td>1 fibers</td>
</tr>
<tr>
<td>Link:4-2:</td>
<td>1 fibers</td>
<td>Link:27-12:</td>
<td>55 fibers</td>
<td>Link:49-46:</td>
<td>14 fibers</td>
</tr>
<tr>
<td>Link:5-1:</td>
<td>27 fibers</td>
<td>Link:27-14:</td>
<td>17 fibers</td>
<td>Link:49-48:</td>
<td>3 fibers</td>
</tr>
<tr>
<td>Link:5-2:</td>
<td>56 fibers</td>
<td>Link:27-16:</td>
<td>1 fibers</td>
<td>Link:50-46:</td>
<td>13 fibers</td>
</tr>
<tr>
<td>Link:5-3:</td>
<td>166 fibers</td>
<td>Link:27-17:</td>
<td>15 fibers</td>
<td>Link:50-48:</td>
<td>55 fibers</td>
</tr>
<tr>
<td>Link:6-5:</td>
<td>27 fibers</td>
<td>Link:29-2:</td>
<td>1 fibers</td>
<td>Link:51-46:</td>
<td>30 fibers</td>
</tr>
<tr>
<td>Link:8-6:</td>
<td>113 fibers</td>
<td>Link:29-27:</td>
<td>18 fibers</td>
<td>Link:52-0:</td>
<td>166 fibers</td>
</tr>
<tr>
<td>Link:9-6:</td>
<td>2 fibers</td>
<td>Link:30-14:</td>
<td>14 fibers</td>
<td>Link:52-17:</td>
<td>27 fibers</td>
</tr>
<tr>
<td>Link:9-8:</td>
<td>2 fibers</td>
<td>Link:30-15:</td>
<td>1 fibers</td>
<td>Link:52-18:</td>
<td>55 fibers</td>
</tr>
<tr>
<td>Link:10-0:</td>
<td>55 fibers</td>
<td>Link:30-27:</td>
<td>34 fibers</td>
<td>Link:52-22:</td>
<td>111 fibers</td>
</tr>
<tr>
<td>Link:10-1:</td>
<td>1 fibers</td>
<td>Link:31-30:</td>
<td>7 fibers</td>
<td>Link:52-27:</td>
<td>444 fibers</td>
</tr>
<tr>
<td>Link:10-6:</td>
<td>55 fibers</td>
<td>Link:32-31:</td>
<td>1 fibers</td>
<td>Link:52-45:</td>
<td>111 fibers</td>
</tr>
<tr>
<td>Link:11-0:</td>
<td>55 fibers</td>
<td>Link:34-30:</td>
<td>1 fibers</td>
<td>Link:52-48:</td>
<td>30 fibers</td>
</tr>
<tr>
<td>Link:12-2:</td>
<td>55 fibers</td>
<td>Link:36-30:</td>
<td>1 fibers</td>
<td>Link:52-50:</td>
<td>13 fibers</td>
</tr>
<tr>
<td>Link:12-7:</td>
<td>249 fibers</td>
<td>Link:36-35:</td>
<td>1 fibers</td>
<td>Link:52-51:</td>
<td>13 fibers</td>
</tr>
<tr>
<td>Link:12-11:</td>
<td>250 fibers</td>
<td>Link:37-36:</td>
<td>1 fibers</td>
<td>Link:53-48:</td>
<td>3 fibers</td>
</tr>
<tr>
<td>Link:13-7:</td>
<td>3 fibers</td>
<td>Link:38-30:</td>
<td>27 fibers</td>
<td>Link:53-51:</td>
<td>27 fibers</td>
</tr>
<tr>
<td>Link:14-7:</td>
<td>20 fibers</td>
<td>Link:38-37:</td>
<td>34 fibers</td>
<td>Link:53-52:</td>
<td>14 fibers</td>
</tr>
<tr>
<td>Link:14-13:</td>
<td>3 fibers</td>
<td>Link:39-38:</td>
<td>5 fibers</td>
<td>Link:54-22:</td>
<td>1 fibers</td>
</tr>
<tr>
<td>Link:15-14:</td>
<td>3 fibers</td>
<td>Link:40-38:</td>
<td>1 fibers</td>
<td>Link:54-48:</td>
<td>4 fibers</td>
</tr>
<tr>
<td>Link:16-10:</td>
<td>29 fibers</td>
<td>Link:40-39:</td>
<td>1 fibers</td>
<td>Link:54-52:</td>
<td>31 fibers</td>
</tr>
<tr>
<td>Link:17-2:</td>
<td>13 fibers</td>
<td>Link:42-33:</td>
<td>2 fibers</td>
<td>Link:54-53:</td>
<td>13 fibers</td>
</tr>
<tr>
<td>Link:17-10:</td>
<td>13 fibers</td>
<td>Link:43-0:</td>
<td>55 fibers</td>
<td>Link:56-29:</td>
<td>2 fibers</td>
</tr>
<tr>
<td>Link:17-16:</td>
<td>1 fibers</td>
<td>Link:43-10:</td>
<td>13 fibers</td>
<td>Link:56-30:</td>
<td>13 fibers</td>
</tr>
<tr>
<td>Link:18-6:</td>
<td>83 fibers</td>
<td>Link:43-21:</td>
<td>55 fibers</td>
<td>Link:56-34:</td>
<td>1 fibers</td>
</tr>
<tr>
<td>Link:18-16:</td>
<td>5 fibers</td>
<td>Link:43-27:</td>
<td>555 fibers</td>
<td>Link:57-28:</td>
<td>14 fibers</td>
</tr>
<tr>
<td>Link:18-17:</td>
<td>15 fibers</td>
<td>Link:43-30:</td>
<td>27 fibers</td>
<td>Link:57-29:</td>
<td>2 fibers</td>
</tr>
<tr>
<td>Link:19-18:</td>
<td>4 fibers</td>
<td>Link:43-33:</td>
<td>15 fibers</td>
<td>Link:57-33:</td>
<td>1 fibers</td>
</tr>
<tr>
<td>Link</td>
<td>Fibers</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20-19</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21-10</td>
<td>124</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21-16</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21-18</td>
<td>69</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22-10</td>
<td>55</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22-18</td>
<td>70</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22-20</td>
<td>69</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23-22</td>
<td>27</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24-22</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25-22</td>
<td>27</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25-23</td>
<td>27</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25-24</td>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>43-41</td>
<td>69</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>43-42</td>
<td>82</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>44-2</td>
<td>27</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>44-3</td>
<td>139</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>45-43</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>45-44</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>46-27</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>46-41</td>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>46-43</td>
<td>27</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>47-43</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>47-44</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>57-56</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>58-1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>58-5</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>59-0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>59-58</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>60-18</td>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>60-20</td>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>59-58</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>60-20</td>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Appendix C

Graph Coloring Using Backtracking

As was mentioned in the main body of this report, the problem of obtaining the lambda-to-lightpath assignment that uses the minimal number of lambdas is akin to solving a graph coloring problem in which the nodes represent the backup lightpaths. A link connects two nodes if the associated backup lightpaths have overlapping primary routes. Once the mapping problem has been translated into a graph-coloring problem, the minimum number of colors can be determined using the backtracking algorithm. The algorithm, described informally in [25] is repeated here:

Label the nodes from 1…N, and the available colors as c1…cM
First pick node 1 and assign it color c1. If node 2 is not adjacent to 1, assign it color c1. Otherwise assign color c2 to node 2 (if M<2, then augment M by 1 and then assign color c2 to node 2). Focus on the next node, node 3. Use c1 if possible for node 3. Otherwise use c2, if this is possible. Only if neither color c1 nor c2 can be used should c3 be used (if M<3, then augment M by 1 and assign color c3 to node 3.). Continue this process as long as it is possible to assign one of the M colors to each additional node always using the first allowable color in the list. If a node is reached that cannot be colored by any of the M colors, backtrack to the last assignment made and change the coloring of the last node colored if possible using the next allowable color in the list (the list is the original one known at the time that assignment was made). If it is not possible to change this coloring, backtrack further to previous assignments, one step back at a time, until it is possible to change a coloring of a node. Then re-initiate the coloring of all nodes from this point, as long as possible. If a coloring using M colors exists, backtracking will produced it. However, if all possibilities have been exhausted, a new color is needed. Augment M by one and assign the new color to the node in question.
The backtracking algorithm is essentially a systematic way of performing an exhaustive search for all possible colorings until a solution is found.

Assuming a link has already set aside $K$ protection lambdas, the objective is to accept any new backup lightpath without requiring an additional lambda. On the other hand, if a backup lightpath is deleted from the link, then port mappings should be re-assigned with the objective of reducing the required number of protection lambdas to $K - 1$. Thus, the graph-coloring algorithm stated before should start with:

$M = K$, when it is being used in the resource reservation for a backup lightpath.
$M = K - 1$, when a backup lightpath is being deleted.
Appendix D

Analyzing the Troublesome Hop Scheme

In the simulations presented in this thesis, two versions of the Troublesome Hop scheme were shown – the first was the Unrestricted_TH scheme, in which all the required number of troublesome hops was advertised. The other was the Restricted_TH scheme, in which only one troublesome hop was advertised.

It is useful to observe the performance of intermediate versions, and compare them to these two extremes. Therefore, the simulation of scenario 2 was repeated for two new versions of the Restricted_TH scheme. In the first version, a maximum of two troublesome hops could be advertised. In the second version, a maximum of three troublesome hops could be advertised. The results are shown below. The curves of the original two versions are also displayed. Also, results at very low network loads have been omitted for the sake of clarity.
The results show that huge performance gains can be achieved by the Restricted_TH scheme even by permitting just one extra troublesome hop to be advertised. In fact, there is not much difference between the Restricted_TH scheme and the Unrestricted_TH scheme when the Restricted_TH scheme can advertise more than one troublesome hop. So, it appears possible to place a bound on the number of advertised troublesome hops without compromising performance.