Survivability algorithms in MPLS and WDM optical networks

Zhi Liu
Iowa State University

Follow this and additional works at: https://lib.dr.iastate.edu/rtd

Recommended Citation
https://lib.dr.iastate.edu/rtd/19161

This Thesis is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.
Survivability algorithms in MPLS and WDM optical networks

by

Zhi Liu

A thesis submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE

Major: Computer Science

Program of Study Committee:
Lu Ruan, Major Professor
Ying Cai
Arun Somani

Iowa State University
Ames, Iowa
2005
Copyright © Zhi Liu, 2005. All rights reserved.
Graduate College
Iowa State University

This is to certify that the master's thesis of

Zhi Liu

has met the thesis requirements of Iowa State University

Signatures have been redacted for privacy
TABLE OF CONTENTS

ABSTRACT ................................. v

CHAPTER 1. Introduction .................. 1
  1.1 Introduction of WDM Optical Networks ............... 1
  1.2 Introduction of MPLS networks .................. 3
  1.3 Survivability issues in survivable networks .......... 4
    1.3.1 Protection Schemes ......................... 4
    1.3.2 Restoration Schemes ....................... 5
  1.4 Motivation of our works ..................... 6
    1.4.1 Upstream Node Initiated Fast Restoration in MPLS Networks .... 6
    1.4.2 reducing restoration blocking probability in WDM optical networks 7
  1.5 Outline of The Thesis ..................... 8

CHAPTER 2. Upstream Node Initiated Fast Restoration in MPLS Networks ......................... 9
  2.1 Upstream Node Initiated Fast Restoration (UNIFR) ............ 9
    2.1.1 The Key Idea ............................. 9
    2.1.2 An MPLS Resilience Framework Supporting UNIFR ........... 10
    2.1.3 Computation of Candidate Backup Paths .................. 11
  2.2 ILPs for Spare Capacity Optimization .................. 13
    2.2.1 The Restoration Flow Splitting Allowed (RFSA) Model ....... 14
    2.2.2 The Restoration Flow Non-Splitting (RFNS) Model ........... 16
2.3 Numerical Results ................................................. 17
  2.3.1 Simulation Settings ........................................ 17
  2.3.2 Figure of Merits ............................................ 18
  2.3.3 Results and Discussions ................................... 19
2.4 Conclusion ....................................................... 23

CHAPTER 3. Reducing Restoration Blocking Probability in WDM Optical Networks .................................................. 24
3.1 Background ....................................................... 24
  3.1.1 Previous Work ............................................. 24
  3.1.2 Restoration Blocking Scenarios ............................ 26
  3.1.3 Problem Definition ........................................ 26
3.2 Heuristics to minimize primary holding ......................... 28
  3.2.1 Overview .................................................... 28
  3.2.2 Computation of Candidate Working and Restoration Paths 28
  3.2.3 SEA Routing and Wavelength Assignment Algorithm .......... 30
  3.2.4 Compute_PSCORE() Subroutine ............................ 31
  3.2.5 Least Congestion-Based Compute_RSCORE() Subroutine ...... 33
  3.2.6 Prediction Based Subroutine Compute_RSCORE() .......... 34
3.3 Heuristics to avoid mutual competition ........................ 38
3.4 Numerical results ............................................... 42
  3.4.1 Simulation Settings ........................................ 42
  3.4.2 Results and Discussions ................................... 43
3.5 Conclusion ....................................................... 47

CHAPTER 4. Summary ............................................... 49

REFERENCES ......................................................... 51

ACKNOWLEDGMENTS ................................................ 54
ABSTRACT

In modern ultra-wide bandwidth, high speed and high reliable communication networks, the failure of network components including equipment (such as routers) and transmission media (such as fibers) may cause a huge volume of data loss. Therefore network survivability mechanisms, by which the disrupted traffic upon failures can be restored, are crucial in network design and deserve thorough investigation. In this thesis, we propose some survivability approaches to survive failures in MPLS and WDM optical networks.

MPLS is a promising technology that enables much faster failure recovery than conventional IP rerouting in IP networks. While the traditional MPLS path-based protection scheme is capacity efficient, it is relatively slow in restoration; on the other hand, while traditional MPLS link-based scheme has fast restoration speed, its capacity efficiency is low. In this thesis, we propose a new restoration scheme called UNIFR, which can provide fast restoration as link-based scheme while achieving better capacity efficiency than link-based scheme. We present a MPLS resilience framework that supports UNIFR and give two ILP formulations to solve the spare capacity optimization problem for UNIFR-based restoration model. Simulation study shows that the capacity efficiency of UNIFR-based model is much better than that of link-based model and close to that of path-based model.

In WDM optical networks, although lots of pervious works have been done in both protection and restoration survivability techniques, to our best knowledge, little study focuses on improving the dynamic restoration success ratio. To address this problem, we first identify two restoration blocking types called primary holding and mutual com-
petition. To address primary holding, we propose a dynamic routing and wavelength assignment algorithm for connection establishment that takes the future possible failures into consideration and choose route and wavelength for the working lightpath that could lead to higher chance of successful restoration for the potential failures. To address mutual competition, we present some heuristics ideas to increase restoration success ratio. Simulation shows that our algorithms can clearly reduce the restoration blocking probability while not affecting primary blocking probability and restoration speed much.
CHAPTER 1. Introduction

Our work focuses on the survivability algorithms in MPLS and WDM optical networks. A brief introduction to MPLS and WDM optical network is first presented in 1.1 and 1.2. Then the survivability issues addressed in this thesis will be given in 1.3. Furthermore, the motivations of our works "upstream node initiated fast restoration in MPLS networks" and "reducing restoration blocking probability in WDM optical network" are respectively proposed in 1.4.1 and 1.4.2. Finally, 1.5 gives the outline of this thesis.

1.1 Introduction of WDM Optical Networks

Wavelength-division multiplexing (WDM) optical networks with ultra-high capacity are believed to be the backbone transport networks for the next generation Internet. Wavelength-division multiplexing technique can divide the huge bandwidth of a fiber into many non-overlapping wavelengths channels (currently hundreds of them can be realized within one fiber). Each of the wavelength channel is able to transfer data in parallel at the speed of a few Gbps. Typical WDM optical network equipments include: (1) Optical line terminals (OLTs), which multiplex multiple wavelengths into a single fiber and demultiplex wavelengths on a single fiber into separate wavelengths; (2) Optical add/drop multiplexers (OADMs), which add/drop one or more wavelengths to/from a composite WDM signal and allow the remaining wavelengths to pass through; (3) Optical crossconnects (OXCs), which can switch wavelengths from one port to another. The technique of routing and switching in optical domain (called wavelength routing) that
can independently route wavelengths from an input port to an output port is the key revolution bringing optical networks from SONET to the second generation wavelength routing networks.

In wavelength routing optical networks, to satisfy a connection request, an end to end circuit switched connection (called lightpath) between the source and destination nodes of the connection request need be set up by assigning a dedicated wavelength on each link on its path. All links in one lightpath must be assigned the same wavelength if network nodes are not capable of wavelength conversion. This is known as the wavelength continuity constraint. However, by equipping wavelength converters at network nodes, wavelength continuity constraint can be released so that choosing wavelengths will be much more flexible when satisfying a connection request. Thus reduced connection blocking can be achieved. Since wavelength converters are expensive, in our work, wavelength conversion capacity is not assumed.

The curial problem in wavelength routing optical networks is the routing and wavelength assignment (RWA) problem, which is to set up lightpaths for the connection requests by finding routes and assigning wavelengths to them. The RWA problem can be classified into off-line and on-line scenarios. In the off-line case, complete knowledge of the traffic to be served by the network is known in advance, and objective is to set up lightpaths for as many connection requests as possible using the fixed network resources. Finding a globally optimal solution for the offline problem is NP-hard, which can be formulated as a mixed-integer linear program [1]. In the on-line case, an incoming connection request is dynamically provisioned, i.e., lightpaths are independently established on demand as connection requests arrive at the network. As the trend in backbone transport networks, developing efficient dynamic routing and wavelength assignment algorithms has been widely studied in [2][3][4][5][6][7].
1.2 Introduction of MPLS networks

Multi-Protocol Label Switching (MPLS) [8] is a technology that simplifies the packet forwarding by replacing the standard destination-based hop-by-hop forwarding paradigm with a label-switching forwarding paradigm. MPLS also separates routing and packet forwarding so that it enables one to apply new specialized routing services without requiring changes in the forwarding path [9]. These features of MPLS make it promising for the future Internet and worthy to be widely studied.

An MPLS capable IP network consists of Label Switching Routers (LSRs) interconnected by IP links. To transmit date packets, a Label Switched Path (LSP) is established between the source and destination LSRs, and then data packets will be forwarded along the LSP based on the appended labels. Date packets are partitioned into forwarding equivalency classes (FECs). A set of packets following the same LSP, belonging to the same FEC, are forwarded in a similar manner on each LSR along the path. Each FEC is assigned a fixed-length, locally significant identifier (called label). When a packet needs to be forwarded, at the ingress LSR, the associated FEC with the assigned label is found and the packet is forwarded to its next hop with the assigned label. At subsequent LSRs, the label is used as an index into a table which specifies the new outgoing label and next hop. The old label is replaced with the new one, and the packet is forwarded to the next hop. Finally, the egress LSR will strip the label and forward the packet to final destination based on the IP packet header. This process can eliminate the need for traditional hop by hop slow lookups to routing tables at each router along the path so that the packet forwarding can be simplified.

1.3 Survivability issues in survivable networks

Survivability becomes the key issue in design of modern ultra-wide bandwidth, high speed and high reliable communication networks which usually want high availability for
connections (normally 99.999% availability). The network with capability to continue providing service in the presence of failures (on network link, node, or individual channel) is known as survivable network. The basic idea to achieve survivability is to find a detour around the failure under current available network resources and switch the affected traffic to the detour. In general, there exist two basic approaches to survive the failure in survivable networks: protection[10] and restoration[11], which will be discussed in detail as follow.

1.3.1 Protection Schemes

In the protection scheme, backup path(s) for the working path are established while the incoming connection request is being set up. Redundant spare capacity (backup wavelength) is reserved on the backup path(s) when they are established and before any failure occurs. Normally, data traffic is carried on the working path; once the working path is disrupted by a failure, traffic will be switched to the backup path and escape from the failure. The spare capacity on the backup path can be either dedicated or shared. In the dedicated method, the spare capacity reserved on one backup path will not be shared with other backup paths. In contrast, in the shared method, spare capacity sharing is allowed among backup paths of the connections that will not fail simultaneously. Under single link failure assumption (this is assumed in the whole thesis), several backup paths can share backup capacity as long as their working paths are link disjoint.

Conventional protection schemes are either path-based or link-based. In a path-based protection scheme, an end-to-end backup path is used to protect a working path. The backup path is link-disjoint with the working path so that it can be used to restore traffic upon any link failure on the working path. The source node of the working path is responsible for switching the traffic to the backup path when it receives the failure notification from the node adjacent to the faulty link when the working path fails. In a link-based protection scheme, multiple backup paths are established for a working path
with each backup path protecting one link on the working path. The upstream node adjacent to the faulty link is responsible for locally switching the traffic to the backup path of the failed link and the downstream node adjacent to the faulty link is responsible for merging the traffic back onto the working path.

1.3.2 Restoration Schemes

In restoration, a backup path is dynamically discovered and established using the current available capacity in the network when a failure occurs on the working path. Restoration schemes can also be classified as path-based or link-based. In path-based restoration, similar with path-based protection scheme, when the upstream node adjacent to the faulty link detects the failure, it will send a failure notification message to the source node. The source node stops transferring data packets onto the working path when the failure notification message is received. Then a restoration path to the destination node will be dynamically determined by the source node and the traffic will be switched to the restoration path. In contrast, in link-based restoration, the upstream node adjacent to the failure can locally establish a restoration path from itself to the downstream end node of the faulty link without sending any failure notification to the source node. Then the affected traffic will be forwarded to the local restoration path.

1.4 Motivation of our works

1.4.1 Upstream Node Initiated Fast Restoration in MPLS Networks

The rapid growth of real-time and mission critical traffic carried by the Internet requires survivability to be an key ingredient of the IP networks. Unfortunately, survivability achieved by traditional IP rerouting is too slow (several seconds to several minutes [12]) for some premium services, such as virtual leased line services, and high priority voice and video traffic. However, compared with the IP rerouting, MPLS can support much
faster failure recovery in addition to empowering the best effort IP network with quality of service and traffic engineering capabilities. Fast failure recovery in MPLS networks can be done by pre-establishing one or more backup LSPs for a working LSP so that when a failure occurs on the working LSP, the backup LSP(s) can be used to restore the traffic. Due to the benefits of MPLS, we focus on the survivability in MPLS networks.

Path-based shared protection scheme has received much attention in recent years and various algorithms have been proposed for dynamic routing and spare capacity planning of bandwidth guaranteed restorable connections [13] [14][15] [16][17]. These algorithms compute routes and allocate bandwidth for a pair of working and backup paths for a given connection request with the objective of minimizing the total bandwidth to be reserved for the two paths by exploiting backup sharing. The advantage of path-based scheme over link-based scheme is that it has much higher capacity efficiency [18] due to the more backup capacity sharing opportunities. However, path-based scheme is much slower in restoration than link-based scheme because of the failure notification delay. Before receiving the failure notification, the source node has no knowledge about the failure. Therefore it keeps sending the packets along the working path and all these packets will be lost. Clearly, the farther is the failure from the source node, the more packets will be lost due to the failure notification delay. To address the problems of speeding up the restoration process (reducing the data loss) as well as increasing the capacity efficiency, we propose a new MPLS restoration scheme called Upstream Node Initiated Fast Restoration (UNIFR) in this thesis. The key feature of UNIFR is that it can achieve the same restoration speed as link-based scheme while having capacity efficiency close to that of path-based scheme. The network capacity planning problem based on the UNIFR will also be investigated.
1.4.2 reducing restoration blocking probability in WDM optical networks

In WDM optical networks, compared with protection schemes (discussed in 1.3.1), restoration schemes (discussed in 1.3.2) are more efficient in wavelength utilization due to no backup reservation, however, there is no guarantee that all the disrupted working lightpaths can be restored due to lack of wavelength resources. Furthermore, since routing and wavelength assignment for the backup lightpath need to be done upon a failure, there will be a long latency between the time a failure occurs and the time a backup lightpath is established. To speed up the restoration process, a dynamic multi-initiation restoration protocol (MIP) is proposed in [19]. With MIP, wavelength probing and reservation are done on the pre-computed multi-initiated restoration paths to achieve fast restoration for the disrupted connections. However, to our best knowledge, little work has been done on improving the restoration success ratio (i.e., the number of disrupted connections for which a backup lightpath can be found over the total number of disrupted connections). To address this problem, we propose a dynamic routing and wavelength assignment algorithm for connection establishment that takes the future possible failures into consideration and choose route and wavelength for the working lightpath that could lead to higher chance of successful restoration for the potential failures. We also present some heuristics that can be applied to MIP to increase restoration success ratio.

1.5 Outline of The Thesis

The rest of the thesis is organized as follows. In chapter 2, we propose a new Upstream Node Initiated Fast Restoration (UNIFR) scheme for MPLS networks as well as the MPLS resilience framework supporting UNIFR. The main idea of UNIFR is to let the upstream node adjacent to the faulty link switch the restoration traffic directly to the destination node through some pre-established LSPs. The key advantages of the proposed UNIFR framework include faster failure restoration speed (like link-based scheme) by
eliminating the failure notification delay, high capacity efficiency (close to path-based scheme) due to backup sharing achieved by semi-global protection paths, and simplified dynamic connection provisioning. The UNIFR framework consists of three components: (1) offline computation of primary and candidate backup paths; (2) offline determination of backup LSPs that will be used and spare capacity required on each link of the network to achieve full restoration for any single link failure by solving ILP formulations; and (3) pre-establishment of primary and backup LSPs. We will show the spare capacity efficiency performance of our proposed UNIFR-based scheme compared with link-based and path-based schemes in numerical results.

In chapter 3, we first investigate a dynamic fast restoration protocol called \textit{MIP} and propose two main reasons of the restoration blocking occurrence, which are denoted as \textit{primary holding} and \textit{mutual competition}. Then we present dynamic routing and wavelength assignment algorithms to address \textit{primary holding} by evaluating the restoration blocking performance of the candidate primary paths and wavelengths. We also give some heuristic algorithms applied to \textit{MIP} to address \textit{mutual competition}. The simulation shows that our algorithms can definitely reduce the restoration blocking probability and only have extremely slight influence to the primary blocking probability and restoration speed.

Finally, In chapter 4, the summery of the whole thesis is given.
CHAPTER 2. Upstream Node Initiated Fast Restoration in MPLS Networks

In this chapter, we propose an Upstream Node Initiated Fast Restoration (UNIFR) scheme for MPLS networks. This chapter is organized as follows. In Section 2.1, we describe the idea of UNIFR and how to implement an MPLS resilience framework supporting UNIFR. In Section 2.2, we present two ILP formulations that compute the minimum spare capacity requirement by UNIFR for providing 100% restoration guarantee upon any single link failure. We study the performance of the algorithm in Section 2.3. Finally, we summarize our work in Section 2.4.

2.1 Upstream Node Initiated Fast Restoration (UNIFR)

2.1.1 The Key Idea

The key idea of UNIFR is to let the upstream node adjacent to the failure switch the traffic to a backup path (or a set of backup paths if restoration traffic splitting is allowed) from itself to the destination node immediately after it detects a failure. Like link-based scheme, UNIFR achieves fast restoration by eliminating the failure notification delay incurred in path-based scheme. Unlike link-based scheme, UNIFR uses semi-global backup paths from the failure detecting node directly to the destination node. The semi-global backup paths offer better opportunity for spare capacity sharing among themselves than the local backup paths used in link-based scheme so that more efficient spare capacity
utilization can be achieved.

UNIFR requires that for each demand relation, every node on its working path must have one or more backup paths from itself to the destination node because every node on the working path of a demand relation may detect a failure and therefore be responsible for switching the traffic to the backup path(s). In addition, the backup paths should be pre-established instead of being established in real-time after a failure is detected to allow immediate traffic restoration. Pre-establishing a backup LSP in a MPLS network can be done by creating an entry in the label switching table in every LSR on the backup path.

2.1.2 An MPLS Resilience Framework Supporting UNIFR

To implement a MPLS resilience framework supporting UNIFR, we need to perform two tasks during network planning. The first task is to compute the working and backup paths for each demand relation in the given traffic matrix and compute the working and spare capacities required on each network link to support the working traffic and guarantee 100% restoration upon any single link failure. We will give two ILP formulations for solving this problem in section 2.2. The second task is to dimension the network based on the optimal capacity allocation computed by the ILP and pre-establish the working/backup LSPs for each demand relation as determined by the ILP.

Once the MPLS resilience framework is implemented as described above, dynamic connection provisioning during network operation becomes very simple. When a connection request arrives at the network, the source node simply checks whether the current traffic load between the source and destination nodes plus the requested traffic load exceeds the supportable traffic load. If not, the connection will be admitted, otherwise the connection will be rejected. If a connection is admitted, the source node will send the traffic over the pre-established working LSP. There is no need to reserve spare capacity on the backup LSPs when a connection is admitted since the network dimensioning guarantees that as long as the traffic load between each LSR pair does not exceed the supportable traffic
load, there is enough spare capacity in the network to provide 100% restoration upon any single link failure. When a failure occurs in the network, UNIFR will be used to restore the traffic as follows. Suppose node $i$ detects the failure of link $(i, j)$, it will switch all flows that traverse link $(i, j)$ to the pre-established backup LSPs between node $i$ and the destination nodes of those flows.

In summary, the UNIFR-based MPLS resilience framework offers many attractive features, including fast restoration speed (like link-based scheme), high capacity efficiency (close to path-based scheme), and simplified dynamic connection provisioning. A detailed comparison of the capacity efficiency of UNIFR-based, path-based, and link-based restoration schemes will be given in section 2.3.

2.1.3 Computation of Candidate Backup Paths

The key task in realizing the UNIFR-based MPLS resilience framework is to determine the working/backup paths as well as working/backup capacities required to support 100% restoration upon any single link failure. In this paper, we assume each demand relation uses the shortest path as the working path and two ILPs for computing the optimal backup paths and spare capacity allocation will be presented in section 2.2. The ILPs require a set of candidate backup paths to be given a priori. In this section we describe how to compute the candidate backup paths.

We assume each demand relation uses the shortest path as the working path. The candidate backup paths are computed as follows. For each possible faulty link $l = (i, j)$ and each affected demand relation $r = (s, d)$ ($l$ is on the working path of $r$), a set of paths from $i$ to $d$, denoted by $B_{l,r}$, are computed to be the candidate backup paths for restoring $r$ when $l$ fails. Clearly, all paths in $B_{l,r}$ should not traverse link $l$. To compute $B_{l,r}$ of size $k$, we first remove link $l$ from the network graph and then use the $k$-shortest paths algorithm [20] to compute $k$ simple paths where $k$ is an adjustable parameter. Note that the $k$ candidate backup paths are not required to be link-disjoint; the only requirement
is that they do not use link \( l \). Fig. 2.1 shows an example of the routing of the candidate backup paths. Here \( p_1 \) is the working path of the demand relation \((s, d)\). For the failure of link \( l \) on \( p_1 \), four shortest paths \((b_1, b_2, b_3, b_4)\) from node \( i \) to node \( d \) are computed as the candidate backup paths for restoring the traffic of demand relation \((s, d)\), i.e.,

\[
B_{l,(s,d)} = \{b_1, b_2, b_3, b_4\}.
\]

![Figure 2.1 Routing of candidate backup paths for link failure.](image)

One problem with computing a set of candidate backup paths for each possible link failure and each affected demand relation is that it will create repeated computations of the same set of candidate backup paths. This can be seen from the example in Fig. 2.1. Suppose demand relation \((s', d)\) uses \( p_2 \) as the working path. Since the faulty link \( l \) is on both \( p_1 \) and \( p_2 \) and both of the working paths share the same destination \( d \), the computation of \( B_{l,(s,d)} \) and \( B_{l,(s',d)} \) will give the same set of paths, i.e., \( \{b_1, b_2, b_3, b_4\} \).

To avoid computing the same set of candidate backup paths multiple times, we can use the following algorithm. For each link \( l = (i, j) \), we find the set \( S \) of demand relations that use \( l \) on their working paths and let \( D \) be the union of the destination nodes of the demand relations in \( S \). For each \( d \in D \), we use the \( k \)-shortest paths algorithm to compute \( k \) shortest paths from \( i \) to \( d \) that do not use \( l \). This set of paths then serves as the candidate backup paths for all demand relations in \( S \) with destination \( d \) upon the failure of link \( l \). This way, repeated computation of the same set of candidate backup paths is eliminated.

Node failure can also be handled by UNIFR. To deal with node failure, we need to
compute a set of candidate backup paths for each possible node failure and each affected demand relation. The candidate backup paths should not use any link that is adjacent to the failed node. Fig 2.2 shows an example. $p_1$ is the working path of demand relation $(s,d)$. When node $a$ on $p_1$ fails, all the links adjacent to node $a$ (i.e., $l_1, l_2, l_3, l_4$) also fail. Thus, the candidate backup paths for demand relation $(s,d)$ upon the failure of node $a$ are computed from the upstream node $u$ to the destination node $d$ and are not allowed to use the links $l_1, l_2, l_3, l_4$. In this example, the candidate backup paths are $b_1, b_2, b_3$ and $b_4$.

![Figure 2.2 Routing of candidate backup paths for a node failure.](image.png)

### 2.2 ILPs for Spare Capacity Optimization

In this section we present two ILP formulations for the following spare capacity optimization problem. Given a network topology, a traffic matrix, a set of candidate backup paths for each possible faulty link and each affected demand (computed by the algorithm given in section 2.1.3), determine the backup path(s) for each pair of possible faulty link and affected demand as well as the spare capacity allocation on the backup paths so that the amount of spare capacity required to provide 100% restoration guarantee for any single link failure is minimized. We consider two models of restoration: the first one allows the restoration traffic to be splitted over multiple backup paths, the second one does not allow restoration traffic splitting so that a single backup path must be chosen for each affected demand upon each possible link failure. We will give an ILP formulation for each of the two models. The ILPs are extended from the ILP given in [18], which is designed
for link-based and path-based restoration schemes.

2.2.1 The Restoration Flow Splitting Allowed (RFSA) Model

In this model, the restoration traffic for a demand relation can be sent along multiple backup paths.

The notations used in the ILP are given below.

**SETS:**

- \( S \): Set of all links.
- \( R \): Set of all unidirectional demand relations.
- \( R_i \): Set of unidirectional demand relations affected by the failure of link \( i \).

**PARAMETERS:**

- \( \alpha_i \): the cost of one capacity unit on link \( i \).
- \( d_r \): number of capacity units required by demand relation \( r \).
- \( \beta_{i,r,k,j} \): equals 1 if link \( j \) is on the \( k^{th} \) candidate backup path for the affected demand relation \( r \) upon the failure of link \( i \), equals 0 otherwise.
- \( K \): number of candidate backup paths for each pair of faulty link and affected demand relation.

**VARIABLES:**

- \( s_j \): number of spare capacity units required on link \( j \).
- \( p_{i,r,k} \): number of capacity units of the restoration traffic flow sent on the \( k^{th} \) candidate backup path for the affected demand relation \( r \) upon the failure of link \( i \).

**OBJECTIVE:**
Minimize:

\[ \sum_{j \in S} \alpha_j s_j \quad (2.1) \]

The objective function is to minimize the total cost of the spare capacity required on all links. The cost factor \( \alpha_j \) can be configured for different optimization scenarios. One option is to set \( \alpha_j = 1 \), in which case the total number of spare capacity units required on all links will be minimized. Another option is to set \( \alpha_j \) to be the physical distance of link \( j \), in which case the total millage weighted spare capacity units will be minimized.

The following are the constraints to be satisfied:

\[ \sum_{k=1}^{K} p_{i,r,k} = d_r \quad \forall i \in S, \forall r \in R_i \quad (2.2) \]

Comment: For any link \( i \) failure, the traffic of affected demand relation \( r \) can be fully restored.

\[ s_j - \sum_{r \in R_i} \sum_{k=1}^{K} \beta_{i,r,k,j} \cdot p_{i,r,k} \geq 0 \quad \forall i, j \in S, i \neq j \quad (2.3) \]

Comment: There must be adequate spare capacity on link \( j \) for all simultaneous restoration flows carried on \( j \) due to the failure of any link \( i \).

\[ s_j \in 0, 1, 2, \ldots \quad \forall j \quad (2.4) \]

\[ p_{i,r,k} \in 0, 1, 2, \ldots \quad \forall i, \forall r, \forall k \quad (2.5) \]

This ILP formulation is flexible in that it can be used to optimize spare capacity allocation for our UNIFR-based restoration scheme as well as for the conventional link-based and path-based restoration schemes. When optimization for link-based scheme is desired, all of the affected traffic traversing the faulty link in the same direction can be considered as the traffic of the single demand relation between the two end nodes of the faulty link. Therefore the size of \( R_i \) will be equal to 2 (for the two directions of link \( i \)) and \( d_r \) in constraint (2.2) needs to be replaced by \( w_{i,r} \), which is defined as the number of
unidirectional working capacity units on link \( i \) in direction \( r \) (there are only two directions: \( r = 0 \) or \( r = 1 \)). The computation of candidate backup paths also needs to be changed: a set of candidate backup paths must be specified for each link in the network. For the path-based restoration scheme, the ILP formulation can be used without any changes. The only change from UNIFR-based scheme is that only one set of end-to-end candidate backup paths needs to be computed for each demand relation, which is used for restoring all possible link failures on the working path of the demand relation.

### 2.2.2 The Restoration Flow Non-Splitting (RFNS) Model

A drawback of allowing restoration flow splitting is that the data packets of the restoration flow may arrive at the destination node out of order. This may be undesirable for some applications, such as Internet audio and video streaming. To avoid this problem, we can adopt the Restoration Flow Non-Splitting (RFNS) model, which requires that only one of the candidate backup paths is used for restoration. Some changes to the previous ILP are needed to deal with the RFNS model. First, we introduce a new variable:

\[ x_{i,r,k} : \text{equals 1 if the } k^{th} \text{ candidate backup path is chosen to restore the affected demand relation } r \text{ upon the failure of link } i; \text{ equals 0 otherwise.} \]

Three changes in constraints are needed. First, constrain (2.2) is removed. Second, constraint (2.3) is replaced by constraint (2.6). Third, a new constraint (2.7) is added.

The constraints for the RFNS model are the following:

\[
s_j - \sum_{r \in R_i} \sum_{k=1}^{K} \beta_{i,r,k} \cdot x_{i,r,k} \cdot d_r \geq 0 \quad \forall i, j \in S, i \neq j \quad (2.6)
\]

Comment: There must be adequate spare capacity on link \( j \) for all simultaneous restoration flows carried on \( j \) due to the failure of any link \( i \).

\[
\sum_{k=1}^{K} x_{i,r,k} = 1 \quad \forall i \in S, \forall r \in R_i \quad (2.7)
\]
Comment: Only one of the $K$ candidate backup paths is chosen to carry the restoration traffic for affected demand relation $r$ upon the failure of link $i$.

$$s_j \in \{0, 1, 2, \ldots\} \quad \forall j$$  \hspace{1cm} (2.8)

$$x_{i,r,k} \in \{0, 1\} \quad \forall i, \forall r, \forall k$$  \hspace{1cm} (2.9)

As the ILP for the FRSA model, this ILP can be used to solve the spare capacity optimization problem for path-based and link-based restoration schemes with the necessary modifications.

2.3 Numerical Results

2.3.1 Simulation Settings

We used two test networks shown in Fig 3.8. The first network is a metropolitan area model given in [21], which has 15 nodes and 28 links. The second network is the 14-node 21-link NSFNET.

![Figure 2.3 Topologies of test networks.](image)

Five randomly generated traffic matrices are used for each test network. Each entry $M[i,j]$ of the traffic matrix is a random integer value between 1 and 10, which represents the amount of traffic from node $i$ to node $j$. 
We run the ILPs for six different restoration models, which are specified in Table 2.1. Model 1 and 4 are path-based restoration models that allow and disallow restoration traffic splitting respectively. Model 2 and 5 are link-based restoration models, and model 3 and 6 are UNIFR-based restoration models proposed in this paper.

<table>
<thead>
<tr>
<th>Path-based</th>
<th>Link-based</th>
<th>UNIFR-based</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFSA</td>
<td>RFNS</td>
<td>RFSA</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>6</td>
</tr>
</tbody>
</table>

2.3.2 Figure of Merits

Three figure of merits are used to compare the performance of the six test models

1) Redundancy of model \( x \), denoted by \( r_x \):

\( r_x \) is defined as the ratio of total spare capacity to total working capacity used by model \( x \). It reflects the capacity efficiency of model \( x \).

2) Relative redundancy performance of model \( x \) compared to path-based model, denoted by \( \rho_x \):

\( \rho_x \) is defined as \( \frac{r_x}{r_{path}} \times 100\% \). We are interested in \( \rho_{UNIFR} \) and \( \rho_{link} \), which measure how close are the redundancy of UNIFR-based model and link-based model to the redundancy of path-based model respectively.

3) Average node complexity, denoted by \( \bar{\eta} \):

\( \bar{\eta} \) is defined as the arithmetic mean of all \( \eta_i \), where \( \eta_i \) is the number of backup paths originated from node \( i \) (computed by ILP). \( \bar{\eta} \) reflects the average memory requirement at a node for storing the backup path information.

2.3.3 Results and Discussions

We solved the ILPs for all models on a SUN Ultra 10 workstation with 256MB of RAM. The ILPs for models 4, 5, 6 were solved within 0.08% of optimality (due to their
long running time), and the other models were fully solved. All results presented here are the average values taken over the five demand matrices.

Fig.2.4 and Fig.2.5 show how the redundancy of model 1 – 6 changes as the number of candidate backup paths $K$ increases for the Metropolitan network and NSFNET respectively. For the Metropolitan network, $K$ ranges from 1 to 40. For NSFNET, $K$ ranges from 1 to 7 (since for some node pairs no more than 7 candidate backup paths can be found).

![Figure 2.4 Redundancy vs. K value (Metropolitan Net)](image1)

![Figure 2.5 Redundancy vs. K value (NSFNET)](image2)

The results in Fig. 2.4 and Fig. 2.5 show that the redundancy of all six models
decreases as the number of candidate backup paths increases. This is expected because more candidate backup paths provide more opportunities for backup sharing. When $K > 10$ for the Metropolitan Net and $K > 5$ for the NSFNET, the improvement of redundancy becomes fairly small and the redundancy is approaching some limit. Another finding from the figures is that the RFSA models always have better redundancy than the corresponding RFNS models. However, except for the link-based models (model 2 and 5), the redundancy of the RFNS models (model 4 and 6) is extremely close to the redundancy of the corresponding RFSA models (model 1 and 3) (The redundancy gap is usually less than 0.2%). We believe the reason that the redundancy gap between model 2 and model 5 is fairly large (usually 14%-17%) is that, because the candidate backup paths are local (i.e., for each link in the network), backup sharing opportunities are rather limited when only one backup path is allowed for traffic restoration. Note that the RFNS model has two advantages over the RFSA model: it can keep the packets in the restoration traffic in order and is easy to implement. The simulation results showed that these advantages can be obtained with little sacrifice on capacity efficiency when RFNS is used with path-based and UNIFR-based restoration schemes. Thus, it's desirable to use the RFNS model with path-based or UNIFR-based scheme in practice.

Table 2.2 Relative Redundancy Performance for Metropolitan Net

<table>
<thead>
<tr>
<th>$n$</th>
<th>RFSA $\rho_{UNIFR}$</th>
<th>$\rho_{link}$</th>
<th>RFNS $\rho_{UNIFR}$</th>
<th>$\rho_{link}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>40.7%</td>
<td>86.2%</td>
<td>40.7%</td>
<td>86.2%</td>
</tr>
<tr>
<td>2</td>
<td>37.8%</td>
<td>66.4%</td>
<td>37.6%</td>
<td>72.1%</td>
</tr>
<tr>
<td>3</td>
<td>22.8%</td>
<td>45.0%</td>
<td>22.6%</td>
<td>50.9%</td>
</tr>
<tr>
<td>4</td>
<td>16.9%</td>
<td>36.2%</td>
<td>16.7%</td>
<td>52.2%</td>
</tr>
<tr>
<td>5</td>
<td>17.0%</td>
<td>38.5%</td>
<td>16.9%</td>
<td>55.5%</td>
</tr>
<tr>
<td>10</td>
<td>10.0%</td>
<td>35.4%</td>
<td>10.0%</td>
<td>55.0%</td>
</tr>
<tr>
<td>20</td>
<td>9.1%</td>
<td>33.5%</td>
<td>9.2%</td>
<td>56.2%</td>
</tr>
<tr>
<td>40</td>
<td>7.8%</td>
<td>30.0%</td>
<td>7.8%</td>
<td>50.9%</td>
</tr>
</tbody>
</table>

It can be seen from Fig. 2.4 and Fig. 2.5 that the redundancy of UNIFR-based models (model 3 and 6) is always better than the redundancy of link-based models (model 2 and 5) and worse than the redundancy of path-based models (model 1 and 4). Table 2.2 and 2.3
show the relative redundancy performance of UNIFR-based models and link-based models compared to path-based models for the two test networks. A couple of observations can be made from the two tables. First, $\rho_{UNIFR}$ is always much smaller than the corresponding $\rho_{link}$, which indicates that the capacity efficiency of UNIFR-based model is much better than that of link-based model. Second, $\rho_{UNIFR}$ decreases rapidly as $K$ increases. When $K = 40$ for the Metropolitan network and $K = 5$ for the NSFNET, the redundancy of UNIFR-based model is only 7.8% and 14.8% higher than that of path-based model for the two test networks respectively. These results shows that UNIFR-based scheme provides a promising tradeoff between path-based scheme and link-based scheme: on the one hand, UNIFR-based scheme is as fast as link-based scheme in traffic restoration; on the other hand, the capacity efficiency of UNIFR-based scheme is much better than that of link-based scheme and fairly close to that of path-based scheme.

![Figure 2.6 Average node complexity for Metropolitan Net](image-url)
Fig. 2.6 and Fig. 2.7 show the average node complexity $\bar{\eta}$ of the six models for Metropolitan network and NSFNET respectively. $\bar{\eta}$ for model 4 and 5 is constant because a single backup path is used for each demand relation in model 4 and for each link in model 5. $\bar{\eta}$ for models 1, 2, 3 and 6 tends to grow with $K$, but the growth rate is much slower than the linear growth rate. This indicates the number of backup paths chosen at each node will not increase much as $K$ increases. A main finding from Fig. 2.6 and Fig. 2.7 is that with RFSA, $\bar{\eta}$ of UNIFR-based model (model 3) is very close to that of path-based model (model 1) for NSFNET and even better than that of path-based model for the Metropolitan net. This demonstrates that UNIFR-based scheme is a promising scheme to achieve faster restoration speed without increasing node complexity compared to path-based scheme when restoration traffic splitting is allowed. When restoration traffic splitting is not allowed, $\bar{\eta}$ of UNIFR-based model (model 6) is higher than that of path-based model (model 4) (almost doubled in the worst case) for both test networks. However, this would not create scalability problem because the average node complexity is upper bounded by $K \cdot \Delta \cdot N$ where $\Delta$ is the average node degree and $N$ is the number of nodes in the network.
2.4 Conclusion

In this chapter, we proposed an Upstream Node Initiated Fast Restoration (UNIFR) Scheme for MPLS networks. The key idea of UNIFR is to let upstream node adjacent to the failure use semi-global restoration paths (from itself to the destination nodes) to restore traffic immediately after detecting a failure. This way, fast restoration speed like link-based scheme with more efficient spare capacity utilization than link-based scheme can be achieved.

We introduced an MPLS resilience framework supporting UNIFR, which greatly simplifies both dynamic connection provisioning and traffic restoration by pre-planning working/spare capacities in each network link and pre-establishing working/backup paths for each demand relation in the given traffic matrix. We gave two ILP formulations to solve the network planning problem, which computes the optimal backup paths and spare capacity allocation for UNIFR-based restoration model with and without allowing restoration traffic splitting respectively.

Simulation results showed that UNIFR-based scheme can achieve capacity efficiency much better than link-based scheme and close to path-based scheme. In addition, the average number of backup paths maintained by each node under UNIFR-based model is about the same as under path-based model when restoration traffic splitting is allowed. When restoration traffic splitting is not allowed, the average node complexity of UNIFR based model is upper bounded by $K \cdot \Delta \cdot N$ where $\Delta$ is the average node degree and $N$ is the number of nodes in the network.
CHAPTER 3. Reducing Restoration Blocking Probability in WDM Optical Networks

In this chapter, we give strategies to reduce the restoration blocking probability in WDM optical networks. This chapter is organized as follows. In section 3.1, we first present the previous work and then give two reasons for restoration blocking, which are primary holding and mutual competition, finally, the objective of this paper is presented. In section 3.2, we present dynamic routing and wavelength assignment algorithms to reduce the influence of primary holding on restoration blocking. In section 3.3, we present some heuristic algorithms that try to avoid mutual competition. We study the performance of proposed algorithms in section 3.4. A conclusion is given in section 3.5.

3.1 Background

3.1.1 Previous Work

A dynamic multi-initiation path restoration protocol called MI P is proposed in [19] that aims to provide fast restoration to recover from single link failures in WDM networks. It works as follows. A set of candidate restoration paths are pre-computed for each pair of source-destination nodes. When a failure occurs, the upstream node adjacent to the faulty link will detect the failure and send failure notification message (L – FLR) to the source node of each disrupted connection reversely through the working path. Upon receiving L – FLR, each upstream node will immediately initiate restoration by
sending a probe message ($S - REQ$) along a pre-computed restoration path from itself to the destination. With the assumption that the network nodes do not have wavelength conversion capability, $S - REQ$ is to explore the available wavelengths on the restoration path under the wavelength continuity constraint. Note that the restoration initiated from the upstream nodes except the source node can only use the wavelength used by the working path. However, the restoration initiated from the source node can use any available continuous wavelength along the restoration path. If during the probing process, no available continuous wavelengths can be found, the initiating node will probe another restoration path. The destination nodes will receive $S - REQ$ messages from multiple initiating nodes and store them in an FIFO queue. When the destination node receives the first $S - REQ$ message, it will choose one wavelength from the available wavelength information contained in the message and start the backward wavelength reservation process by sending $D - REV$ message to the initiating node reversely along the restoration path taken by the received $S - REQ$ message. If the target wavelength is not available for reservation, another wavelength (if available) will be attempted. If none of the wavelengths contained in the first received $S - REQ$ message can be reserved on the restoration path, the destination node will process the next $S - REQ$ message in the FIFO queue. When a $D - REV$ message successfully arrives at the initiating node, the wavelength reservation process is finished and the restoration path has already been established successfully. The initiating node will start to send data along the restoration path and send a $S - ACK$ message to the destination to inform the destination of the restoration success so that the destination can terminate the restoration process.

Compared with the traditional source-initiated restoration, MIP can achieve faster restoration due to two reasons. First, MIP allows any node upstream to the failed link to initiate restoration before the source node is notified of the failure. Second, the restoration path probe/reservation latency will be shorter if the initiating node is closer to the faulty link.
3.1.2 Restoration Blocking Scenarios

With MIP, unsuccessful restoration may occur in two main scenarios.

First, wavelength occupation by the existing connections causes that no available continuous wavelength can be found on any candidate restoration paths during the probing process of $S - REQ$ messages. In this case, wavelength resource scarcity caused by current existing connections is the reason why restoration path is blocked, so we call this scenario *primary holding*.

Second, wavelength reservations made by the simultaneous disrupted connections cause some of them can not be successfully restored because the wavelength resource has already been reserved by other connections. There are two subcases. The first subcase happens when the restoration paths of disrupted connection $A$ are blocked during $S - REQ$ probing process due to the wavelength reservation already made by some other disrupted connection $B$. The second subcase happens when disrupted connection $A$ tries to reserve target wavelength $\lambda$ by sending $D - REV$ messages, but $\lambda$ is already reserved by some other disrupted connection $B$ even though it is available during the $S - REQ$ probing process of $A$. In both subcases, wavelength resource competition by simultaneous disrupted connections is the reason why restoration path is blocked, so we call this scenario *mutual competition*.

3.1.3 Problem Definition

Consider an optical network with $n$ nodes and $m$ spans. Each span consists of two unidirectional opposite optical links (so there are $2m$ links in the network). In a fiber cut, both two opposite optical links in one span will fail simultaneously. Hence, we define one single span failure as the two simultaneous opposite link failures on this span. We assume $W$ wavelengths are provisioned on each link and no wavelength converters are equipped at the network nodes (so wavelength continuity constraint must be satisfied
when establishing a lightpath).

An incoming connection request $i$ is defined by a triple $(s_i, d_i, t_i)$ in which $s_i$ is the source node, $d_i$ is the destination node and $t_i$ is the holding time of the connection. The source node is responsible for finding a route, assigning a wavelength, and establishing lightpath for the incoming connection request. We define the Working blocking probability as the ratio of the number of unsatisfied incoming connection requests to the number of total arrived requests. In this paper, we'll use MIP to perform restoration when a failure happens. Suppose the faulty span is $(i, j)$. Let $\mathcal{A}(i, j)$ be the set of connections that use link $(i, j)$ on their working paths. Let $\mathcal{B}(i, j) \subseteq \mathcal{A}(i, j)$ be the set of successfully restored disrupted connections upon link $(i, j)$ failure. Define the restoration blocking probability upon single span failure $(i, j)$ as

$$\eta = \frac{|\mathcal{B}(i, j) \cup \mathcal{B}(j, i)|}{|\mathcal{A}(i, j) \cup \mathcal{A}(j, i)|}$$

Our goal is to minimize the restoration blocking probability $\eta$. In order to achieve this goal, the two reasons for restoration blocking will be addressed. There are two key contributions of this paper. First, we develop dynamic routing and wavelength assignment algorithms to decrease the influence of primary holding. Second, we propose heuristic algorithms working together with MIP that are able to reduce the chance of mutual competition. The dynamic RWA algorithms and the heuristic algorithms will be shown in the following two sections.

Here, we define some terminologies that will be used in the following sections. Given a path $p$ and a node $i$ on $p$, we define $UNODE_{i,p}$ to be the set of source-side upstream nodes of $i$ on $p$ including $i$, and define $ULINK_{i,p}$ to be the set of source-side upstream links of $i$ on $p$ (the links only have nodes in $UNODE_{i,p}$ as their end nodes). Given a working path $p$, and a faulty link $l = (i, j)$ on $p$, we define inner restoration path as the restoration path that is initiated from $u \in UNODE_{i,p}$ and $u \neq s$ and define outer restoration path as the restoration path that is initiated from $u$. Clearly, inner restoration path can only
utilize the wavelength used by $p$, however, outer restoration path can utilize any available wavelengths for restoration.

3.2 Heuristics to minimize primary holding

3.2.1 Overview

*primary holding* can cause wavelength scarcity leading to unsuccessful restoration. The key idea to weaken *primary holding* is that for each incoming connection request, we always prefer the working route and wavelength which can provide better restoration resource provision on its corresponding restoration paths when failure happens. Like fixed alternate routing algorithm [22], we first pre-compute a set of candidate working paths between each connection relation, while a set of restoration paths are also pre-computed for each possible failure on every candidate working path. For each candidate working path and each wavelength on it, we can evaluate its restoration potential or even approximately calculate its chance of successful restoration based on the available network state information. We always prefer working path and wavelength with better restoration capability as well as good working blocking performance.

In the following, the detail of how to compute the candidate working and restoration paths is given in 3.2.2; then a score-based routing and wavelength assignment algorithm called SEA will be shown in 3.2.3; finally we concentrate on heuristic algorithms to evaluate the working performance in 3.2.4 and restoration capability in 3.2.5 and 3.2.6.

3.2.2 Computation of Candidate Working and Restoration Paths

We use $K$-shortest paths algorithm [20] to compute $K$ simple shortest paths $P_r$ as the candidate working paths for each connection relation $r = (s, d)$, where $s$ is the source node and $d$ is the destination node.
For \( \forall p_m \in P_r \), a set of restoration paths are computed as follows. For each possible faulty link \( l = (i, j) \) on \( p_m \), a set of paths from each upstream node \( \forall u \in UNODE_{i,p_m} \) to \( d \), denoted by \( B_{u,l,m,r} \), are computed to be the restoration paths initiated from \( u \) for restoring connection \( r \) when \( l \) fails if \( r \) chooses \( p_m \) as its working path. Hence, \( B_{i,m,r} = \bigcup_{u \in UNODE_{i,p}} B_{u,l,m,r} \) is set of total potential restoration paths by taking each possible initiation node into account. Clearly, this computation reflects the multi-initiation restoration idea. When failure \( l \) happens, \( MIP \) (or similar restoration protocol) can be applied to dynamically choose a restoration path from \( B_{i,m,r} \), if applicable. To compute \( B_{u,l,m,r} \) of size \( M \), we first remove link \( l \) from the network graph since all paths in \( B_{u,l,m,r} \) should not traverse the faulty link \( l \). In addition, if \( u \) is not the source node \( s \) (\( B_{u,l,m,r} \) is \textit{inner} restoration paths), all the links in \( ULINK_{i,p} \) should also be removed (the reason is given later). Then we compute \( M \) simple paths from \( u \) to \( d \) to be \( B_{u,l,m,r} \). Note that \( K \) and \( M \) are only the upper bounds of the number of paths, because exact \( K \) and \( M \) simple paths may not be available on the network graph.

Fig. 3.1 is the pseudocode for this algorithm. Fig. 3.2 shows an example of the routing of restoration paths. Here \( p_0 \) is \( 0^{th} \) candidate working path of the connection relation \((s, d)\). For the failure of link \( l \) on \( p_0 \), 2 shortest paths \((b_1, b_2)\) initiated from node \( i \) to node \( d \) are computed as the restoration paths for restoring the traffic of connection relation \((s, d)\), i.e., \( B_{i,l,0,(s,d)} = \{b_1, b_2\} \). Similarly, \( B_{u,l,0,(s,d)} = \{b_3, b_4\} \); \( B_{s,l,0,(s,d)} = \{b_5, b_6\} \) and \( B_{l,0,(s,d)} = \{b_1, b_2, b_3, b_4, b_5, b_6\} \). Fig. 3.2 also shows the reason why we remove all the links in \( ULINK_{i,p} \) when we compute the \textit{inner} restoration paths. Notice that path \( b^* = (u, i, n_3, n_4, d) \) is also the shortest path initiated from \( u \) to \( d \) after removing \( l \). However it is totally overlapped with \( b_1 \). Since both \( b^* \) and \( b_1 \) can only utilize the working wavelength for restoration, actually they can be viewed as exactly coincident. To avoid this potential double counting problem, each \textit{inner} restoration path should be keep from traversing the other upstream node in \( ULINK_{i,p} \). That is why we remove all the links in \( ULINK_{i,p} \). However, when we compute \textit{outer} restoration paths, we do not have this problem because
Computation of Candidate Primary and Restoration Paths

Input:
Network topology graph $G$

Output:
Candidate primary paths: $P_r$, $\forall r$
Candidate restoration paths: $B_{u,l,m,r}$, $\forall u, \forall l, \forall m, \forall r$

Pseudocode:
For each connection relation $r = \{s, d\}$
   $P_r = K$ shortest paths from $s$ to $d$;
   For each $p_m \in P_r$
      For each possible faulty link $l = (i, j)$ on $p_m$
         For each upstream node $u \in UNODE_{i,p_m}$
            remove link $l$ from $G$;
            if ($u! = s$)
               remove all upstream links in $ULINK_{i,p_m}$ from $G$;
               $B_{u,l,m,r} = M$ shortest paths from $u$ to $d$;
      } return $P_r$, $\forall r$ and $B_{u,l,m,r}$, $\forall u, \forall l, \forall m, \forall r$;

Figure 3.1 The code to compute Candidate Primary and Restoration Paths

outer restoration path can utilize any available wavelength, so that even if it is overlapped with other restoration path, they are not identical.

3.2.3 SEA Routing and Wavelength Assignment Algorithm

A score-based evaluation algorithm SEA is designed to select route and wavelength according to the evaluation of the restorable capability of candidate routes and wavelengths. Meanwhile, minimizing working blocking probability is also taken into consideration. Let $W$ be the set of wavelengths provisioned on each link. For $\forall w_n \in W$, we compute 2 scores: $PSCORE_{n,m,r}$ and $RSCORE_{n,m,r}$. $PSCORE$ is introduced to show working blocking performance. $RSCORE$ is introduced to show restoration blocking performance. Both of them usually satisfy: $PSCORE_{n,m,r} \in [0, 1]$ and $RSCORE_{n,m,r} \in [0, 1]$, $\forall n, \forall m, \forall r$. Sometimes $PSCORE_{n,m,r} < 0$ may happen, which means $p_m$ will be excluded since no
continuous available wavelength can be found on it. Then the overall score $OSCORE_{n,m,r}$ is defined as:

$$OSCORE_{n,m,r} = PSCORE_{n,m,r} + \alpha \times RSCORE_{n,m,r}$$

$OSCORE$ integrates the consideration for both working and restoration blocking performance, where $\alpha$ is an adjustable parameter (called $PR$ index) to balance these two optimization, since there is usually a tradeoff between them. The path $p_{m^*} \in P_r$ and wavelength $w_{n^*} \in W$ with the highest $OSCORE$ will be selected as the working path and wavelength. Fig. 3.3 is the pseudocode for this algorithm. $Compute\_PSCORE$ and $Compute\_RSCORE$ are the Subroutines respectively to compute $PSCORE_{n,m,r}$ and $RSCORE_{n,m,r}$, which will be shown next.

### 3.2.4 $Compute\_PSCORE()$ Subroutine

The basic rule to choose a working path is that at least one continuous available wavelength should be guaranteed on it as long as every link along it should have as many free wavelengths as possible. Clearly, possessing at least one free wavelength channel (called availability requirement) is the necessary condition for a candidate working path to be selected. Thus, we exclude the candidate path whose availability requirement is not satisfied. Furthermore, using heavily loaded candidate path ought to be avoided to make usage of network resource more balance, which is similar with the idea of Least Congested
SEA Routing and Wavelength Assignment Algorithm

Input:
Candidate paths \( P_r \) and \( B_{l,m,r}, \forall l, \forall m, \forall r \)
Candidate wavelengths \( W \)
Incoming connection request \( r_i = (s_i, d_i, t_i) \)

Output:
Working path: \( p_{m^*} \)
Working wavelength: \( w_{n^*} \)

Pseudocode:
For each candidate working path \( p_m \in P_{r_i} \)
   For each candidate wavelength \( w_n \in W \)
   \{
      \( PSCORE_{n,m,r_i} = \text{Compute}_{PSCORE}() \);
      \( RSCORE_{n,m,r_i} = \text{Compute}_{RSCORE}() \);
      \( OSCORE_{n,m,r_i} = PSCORE_{n,m,r_i} + \alpha * RSCORE_{n,m,r_i} \);
   \}
find \( p_{m^*} \) and \( w_{n^*} \) such that:
\( OSCORE_{n^*,m^*,r_i} = \text{MAX}(OSCORE_{n,m,r_i}), \forall m, \forall n; \)
To break the tie, the shorter \( p_{m^*} \) is firstly preferred, then the \( w_{n^*} \) with smaller label is preferred;
if \( (OSCORE_{n^*,m^*,r_i} > 0) \)
   return \( p_{m^*} \) and \( w_{n^*} \); else the incoming request \( r_i \) will be blocked;

Figure 3.3 Code for SEA Routing and Wavelength Assignment Algorithm
Path Routing [22]. Given path \( p \), define \( MFW(p) \) to be minimum value of the numbers of free wavelengths on all links along \( p \). Then define *congestion ratio* \( \eta_p = \frac{MFW(p)}{|W|} \), where \( W \) is the set of total wavelengths on each link of \( p \). Intuitively, the candidate working path with higher \( \eta \) value will be preferred so as to achieve better load balancing and better working blocking performance.

Given connection request \( r \), candidate primary path \( p_m \in P_r \), and candidate wavelength \( w_n \in W \), \( PSCORE_{n,m,r} \) is formally defined as:

\[
PSCORE_{n,m,r} = \begin{cases} 
-\infty & \text{if availability requirement is not satisfied on } p_m; \\
\eta_{pm} & \text{otherwise.}
\end{cases}
\]

The subroutine \textit{Compute_PSCORE}() is easy to implement based on the definition of \( PSCORE_{n,m,r} \). Notice that \( PSCORE_{n,m,r} \) are identical for different candidate wavelength \( w_n \in W \) on \( p_m \in P_r \). Thus, it does not contribute to wavelength assignment. And \textit{SEA} can be simplified since \textit{Compute_PSCORE}() can be called only once for different wavelength \( w_n \in W \).

### 3.2.5 Least Congestion-Based \textit{Compute_RSCORE}() Subroutine

Like the \( PSCORE \), we use *congestion ratio* \( \eta \) to direct \( RSCORE \). Naturally, the less congestion on the restoration path \( r_p \), the more chance a continuous free wavelength can be found on \( r_p \) when failure happens. Hence, we prefer the candidate primary path whose restoration paths for every possible faulty link have higher \( \eta \) values. The \textit{Least Congestion-Based Compute_RSCORE} subroutine is formally defined in Fig. 3.4. The average *congestion ratio* value of all restoration paths for each possible faulty link \( l \) on candidate working path \( p_m \) is computed as the \( RSCORE \) for \( l \). Then the minimum \( RSCORE \) among all possible faulty links will be chosen to be \( RSCORE_{n,m,r} \). Similar with \( PSCORE_{n,m,r} \), \( RSCORE_{n,m,r} \) defined in this subroutine is unrelated with wavelength.
Least Congestion-Based Compute_RSCORE ()
Input:
Incoming connection request \( r_i = (s_i, d_i, t_i) \)
Candidate working path \( p_m \in P_{r_i} \)
restoration paths \( B(l, m, r_i), \forall l \) on \( p_m \)
Candidate wavelength \( w_n \in W \)

Output: \( RSCORE_{n,m,r}, \forall n, \forall m, \forall r \)

Pseudocode:
For each possible faulty link \( l = (i, j) \) on \( p_m \)
\[
RSCORE_{l,n,m,r} = \sum_{\forall r_p \in B(l,m,r_i)} \frac{MFW_{r_p}}{|W|} \frac{\left| B(l, m, r_i) \right|}{\left| B(l, m, r) \right|}
\]
return \( RSCORE_{n,m,r} = \text{MIN}(RSCORE_{l,n,m,r}), \forall l \) on \( p_m; \)

Figure 3.4 Code for Least Congestion-Based Compute_RSCORE ()

assignment either.

The SEA routing and wavelength assignment algorithm calling Least Congestion-Based Compute_RSCORE() subroutine is denoted by LCB_SEA. Since both \( PSCORE_{n,m,r} \) and \( RSCORE_{n,m,r} \) are identical for different candidate wavelength \( w_n \in W \), actually first fit wavelength assignment algorithm is implied in LCB_SEA. LCB_SEA has \( O(K \times |WP|^2 \times M \times |RP|) \) computation complexity, where \( |WP| \) is the average length of candidate working path and \( |RP| \) is the average length of restoration path.

3.2.6 Prediction Based Subroutine Compute_RSCORE ()

In order to make RSCORE more precise to reflect the restoration capability, we use additional wavelength status information to approximately estimate the successful restoration probability for the incoming connection request when failure happens. Besides the parameter to show whether a wavelength is currently free or not, 3 more wavelength status parameters: \( \text{last-change}, \text{average-active}, \) and \( \text{average-free} \) are needed to be maintained in each node \( v \) for each wavelength \( w \) on each outgoing link \( l \) from \( v \). These parameters are dynamically updated when the wavelength status changes. The wavelength status is:
active if the wavelength is in use for some connection and free if the wavelength is not in use.

- last_change is the most recent time when $w$ on $l$ changes its status (from active to free or from free to active). Initially $last_change = 0$

- average_active is the accumulative average active time of $w$ on $l$. When $w$ changes its status from active to free, it will be updated as: $average_active = (1 - \beta) \times average_active + \beta \times current.active$, where $\beta$ is called accumulative index which is usually a small positive number (0.002 in our simulation). $current.active$ is newly terminated active status period. Initially $average_active = 0$.

- average_free is the accumulative average free time of $w$ on $l$. Similarly, when $w$ changes its status from free to active, it will be updated as: $average_free = (1 - \beta) \times average_free + \beta \times current.free$. $current.free$ is newly terminated free status period. Initially $average_free = 0$.

In order to estimate the restoration successful probability for the incoming connection request, the key is to predict the wavelength available time intervals (called WATI) on its restoration paths by using the current wavelength status parameters. Formally, given the incoming connection request $r = \{s, d, t\}$, candidate working path $p_m \in P_r$, candidate working wavelength $w_n \in W$, any link $e$ on restoration path $r_p \in B(l, m, r)$ upon faulty link $l$, and suppose $w_n$ will be utilized for restoration, we define the following terminologies:

$WATI_{e,b,n,l}$: $w_n$’s free time intervals on link $e$ which are inside $r$’s holding time.

$WATI_{b,n,l}$: Intersection of $WATI_{e,b,n,l}$ for all links $e$ on path $r_p \in B(l, m, r)$.

$WATI_{n,l}$: Union of $WATI_{b,n,l}$ for all paths $r_p \in B(l, m, r)$.

$SP_{n,l}$: Probability of successful restoration by using $w_n$ to restore failure $l$
Fig. 3.5 shows how to compute $WAT_{e,b,n,l}$ for link $e$ on restoration path $b_1$ in Fig. 3.2. Basically, current time is the incoming connection request $r$'s arrival time and the time interval between arrival and departure is $r$'s holding time $t$. (If $t$ is not available, we can also use $w_n$'s average_busy time to replace $t$.) On link $e$, if currently $w_n$ is free (or active) and $last\_change + average\_free (or \ active) < arrival$, we approximately believe that $w_n$ will change its status from free (active) to active (free) immediately after the arrival time. Otherwise, if $last\_change + average\_free (or \ active) \geq \ arrival$, $w_n$ will change its status at $last\_change + average\_free (or \ last\_change + average\_active)$. Then average_active and average_free will be alternately used to simulate the subsequence active and free period of $w_n$ in a priori. The average_free time intervals laid with $r$'s holding time $t$ will be $WAT_{e,b,n,l}$ in Fig. 3.5 $WAT_{(j,d),b_1,n,l}$ shows the first scenario and $WAT_{(n_4,j),b_1,n,l}$ shows the second scenario. In addition, to compute $WAT_{b,n,l}$, we just do the intersection ($\cap$) operation to find the common intervals of all $WAT_{e,b,n,l}$ for each link $e$ on $b$. To compute $WAT_{n,l}$, union ($\cup$) operation will be used on $WAT_{b,n,l}$ for all restoration path $b$. In Fig. 3.5, $WAT_{b_1,n,l}$ and $WAT_{n,l}$ respectively show $\cap$ and $\cup$ operations.

If the failure $l$ happens inside $WAT_{e,b,n,l}$, wavelength $w_n$ on link $e$ is ready for restoration. If the failure $l$ happens inside $WAT_{b,n,l}$, wavelength $w_n$ on each link $e$ of restoration path $rp_b$ is ready for restoration, so that restoration can be done on $rp_b$ using $w_n$. Furthermore, if the failure $l$ happens inside $WAT_{n,l}$, wavelength $w_n$ on at least one restoration path $rp_b \in B(l,m,r)$ is ready for restoration, so that restoration is guaranteed to be done using $w_n$. Clearly, assuming failure randomly happens within the connection's holding time, $SP_{n,l}$ can be computed as the ratio of $WAT_{n,l}$'s total length to $r$'s holding time $t$. In addition, notice that the wavelength $w_c$ different from the working wavelength $w_n$ is also feasible for restoration only on the outer restoration paths $B(s,l,m,r)$. By the similar computation, we can get $WAT_{c,l}, \forall c \in W - \{w_n\}$. Since all random variables: $SP_{c,l}, \forall w_c \in W$ can be viewed as independent, the probability of successful restoration
Figure 3.5 Compute WATI for candidate working wavelength \( w_n \in W \), candidate primary path \( p_0 \), faulty link \( l = (i, j) \), and restoration paths \( \{b_1, b_3, b_5\} \) shown in Fig. 3.2.

by using any wavelength to handle failure \( l \), denoted by \( SP_l \), can be simply computed to be \( 1 - \prod_{w_c \in W} (1 - SP_{c,l}) \). We prefer the candidate primary path \( p_m \) and wavelength \( w_n \) which can provide better \( SP \) value for any possible faulty link \( l \). The Prediction Based Subroutine Compute_RSCORE () subroutine is formally defined in Fig. 3.6.

The SEA routing and wavelength assignment algorithm calling Prediction Based Compute_RSCORE () subroutine, denoted by PB_SEA, can do the routing and wavelength assignment in conjunction. PB_SEA has \( O(K \cdot |W|^2 \cdot |WP|^2 \cdot M \cdot |RP|) \) computation complexity, where \( W \) is the number of wavelengths on each link, \( |WP| \) is the average length of candidate working path and \( |RP| \) is the average length of restoration path. Notice that \( WATI_{b,c,l} \) for \( b \in B(s,l,m,r), w_c \in W - w_n \) and failure \( l \) is not necessary to be recomputed for each candidate working wavelength \( w_c \in W \), so that computation complexity can be reduced to \( O(K \cdot |W| \cdot |WP|^2 \cdot M \cdot |RP|) \). Furthermore, the PB_SEA can be sim-
plified by just calling $\text{Compute\_RSCORE}()$ for the wavelengths whose $PSCORE > 0$. In reality, especially under high traffic load, the number of continuous free wavelength whose $PSCORE > 0$ on candidate primary path is much smaller than $W$, so that the computational speed is even faster.

### 3.3 Heuristics to avoid mutual competition

In $MIP$’s restoration process, one observation we get from simulation is that usually no less than 60% restoration blocking cases are caused by mutual competition. In the typical $MIP$ restoration failures resulted from mutual competition, before $S\text{-}REQ$ message probes wavelength $\lambda$ or $D\text{-}RSV$ message reserves the target $\lambda$, $\lambda$ has already been reserved in some other simultaneous restoration procedures. Remember that the essentiality of mutual competition is the wavelength resource competition made by simultaneous disrupted connections. Some unavoidable mutual competitions happens due to limited wavelength resources which are not sufficient to restore numerous coincident disrupted connections. However, some other mutual competitions cases may happen due to the flaws of $MIP$ itself. For example, the restoration using wavelength $\lambda$ for disrupted connection $A$ is unsuccessful because $\lambda$ is reserved in the restoration process of disrupted connection $B$, however, $B$ probably can also achieve successful restoration by using some wavelength $\theta$ other than $\lambda$. Clearly, if $B$ originally utilize $\theta$ for its restoration, $A$’s restoration blocking may not happen at all. This kind of mutual competition (called pseudo-mutual competition) can be somehow avoid by carefully designed wavelength reservation mechanisms. Notice that if every simultaneous disrupted connection utilizes its own working wavelength in its restoration, no mutual competition exists any more. That is because all simultaneous disrupted connections definitely have different working wavelengths so that their restoration processes will never interfere with each other any more. However, since the objective of eliminating any mutual competition is impracticable, we change our objective
Prediction Based Compute_RSPCORE ()

Input:
Incoming connection request $r_i = (s_i, d_i, t_i)$
last_change, average_active, and average_free wavelength status
Candidate working path $p_m \in P_i$
restoration paths $B(l, m, r_i), \forall l$ on $p_m$
Candidate wavelength $w_n \in W$

Output: $RSCORE_{n,m,r}, \forall n, \forall m, \forall r$

Pseudocode:
For each possible faulty link $l = (i, j)$ on $p_m$
{
    For each restoration path $rpb \in B(l, m, r_i)$
        For each link $e$ on $rpb$
            compute $WATI_{e,b,n,l}$ based on last_change,
            average_active, and average_free of $w_n$;
            $WATI_{b,n,l} = \bigcap_{e \in rpb} WATI_{e,b,n,l}$;
            $WATI_{n,l} = \bigcup_{rpb \in B(l,m,r_i)} WATI_{b,n,l}$;
            $SP_{n,l} = \frac{|WATI_{b,n,l}|}{t_i}$;

    For each wavelength $w_c \in (W \setminus \{w_n\})$
        For each restoration path $rpb \in B(s_i, l, m, r_i)$
            For each link $e$ on $rpb$
                compute $WATI_{e,b,c,l}$;
                $WATI_{b,c,l} = \bigcap_{e \in rpb} WATI_{e,b,c,l}$;
                $WATI_{c,l} = \bigcup_{rpb \in B(s_i,l,m,r_i)} WATI_{b,c,l}$;
                $SP_{c,l} = \frac{|WATI_{b,c,l}|}{b_i}$;
                $SP_l = 1 - \prod_{w_c \in W} (1 - SP_{c,l})$;
                $RSCORE_{l,n,m,r} = SP_l$;
        }
    return $RSCORE_{n,m,r} = \min(RSCORE_{l,n,m,r}), \forall l$ on $p_m$;

Figure 3.6 Code for Prediction Based Compute_RSPCORE ()
to reducing *pseudo-mutual competition* and the guideline is to give more opportunities to the disrupted connection’s working wavelength to be utilized in the restoration. We will first briefly introduce the wavelength reservation mechanism in *MIP* and several heuristics will be proposed following our guideline.

![Figure 3.7](image)

**Figure 3.7** data structure in each destination node

In *MIP*, there is a *FIFO queue* in the destination node to store all incoming *S-REQ* messages from multiple initiation nodes for one disrupted connection. Furthermore, multiple disrupted connections may share one destination node. So the data structure inside one destination node will be link Fig. 3.7. In Fig. 3.7, There are 2 *FIFO queues*, each of which is related with one disrupted connection and contains several *S-REQ* messages. The earlier arrived *S-REQ* message will be processed earlier. The destination will pick up one wavelength $\lambda$ from the head *S-REQ* message, and sends *D-RSV* message to reserve $\lambda$ backward through the restoration path. Only if the destination node receives *I-NAK* message showing wavelength reservation failure, it will try another wavelength. So *MIP* can guarantee that duplicate restoration paths can be avoided. However, no *mutual competition* is considered here. Hence, serious wavelength reservation interference among simultaneous disrupted connections can be deduced here.

To follow the guideline which is proposed to reducing *pseudo-mutual competition*, 3 heuristics are shown as follows.

- (1). If the head *S-REQ* message contains multiple wavelengths including the working wavelength $\lambda$ of this disrupted connection, $\lambda$ will always be selected first to be
reserved by sending $D-RSV$ message.

- (2). Replace the single $FIFO$ queue for each disrupted connection with 2 $FIFO$ queues. One of these 2 $FIFO$ queues only contains the $S-REQ$ messages coming through the inner restorations paths, so called inner queue, while the other $FIFO$ queues only contains the $S-REQ$ messages coming through the outer restorations paths, so called outer queue. If the inner queue is not empty, the $S-REQ$ messages inside the inner queue will always be processed first. Since each $S-REQ$ message in inner queue only has working wavelength $\lambda$, this kind of $S-REQ$ message will certainly be processed in higher priority than $S-REQ$ message in outer queue, thus pseudo-mutual competition can be reduced.

- (3). If there are only wavelengths $\theta$ different from working wavelength $\lambda$ left, we can wait a short period (called $RSV-DELAY \gamma$) before try to reserve them. There are 2 reasons. The first reason is that during $RSV-DELAY$, it is possible that some $S-REQ$ messages from inner restoration paths may arrive. The second reason is that during $RSV-DELAY$, the other simultaneous disrupted connections may be successfully restored by using their own working wavelengths, so that the probability of pseudo-mutual competition happening will be minimized when wavelengths $\theta$ have to be reserved.

Heuristic (2) and (3) will affect the restoration speed due to selecting longer restorations path and introducing wavelength reservation delay, which will be studied in the simulation.
3.4 Numerical results

3.4.1 Simulation Settings

The test network we used in the simulation shown in Fig 3.8 is a metropolitan area model given in [21], which has 15 nodes and 28 links. For each connection relation, at most $K = 5$ simple shortest paths are pre-computed to be the candidate working paths, and at most $M = 2$ link disjoint shortest restoration paths are also pre-computed from each upstream node to the destination node for each possible faulty link on each candidate path. Incoming connection requests arrive to each node following Poisson process with a mean arrival rate $\lambda$. The holding time of each incoming connection is exponentially distributed with mean $1/\mu$. So the traffic load on each node will be measured in $\lambda/\mu$ Erlangs. For each incoming request, the destination is randomly chosen from all the nodes excluding the source node. In this simulation, we assume the number of wavelengths on each link is 10. The faulty link is uniformly distributed among all the links. As mentioned before, we assume span failure, which means the 2 opposite links with the faulty span will fail simultaneously.

![Figure 3.8 Topologies of test networks.](image)

We consider 5 different algorithm assemblies in this simulation.

- *none*: SEA only use $PSCORE$ to guide the primary path routing without any
consideration for restoration blocking.

- **LCB**: Only *LCB_SEA* is employed.

- **PB**: Only *PB_SEA* will be employed.

- **PBH**: *PB_SEA* will be employed to do routing and wavelength assignment. And heuristics to handle *mutual competition* are also implemented to work with *MIP* during failure restoration. However, *RSV-DELAY = 0*.

- **PBHD**: same with *PBH* and *RSV-DELAY = 0.004s*

The working blocking performance, restoration blocking performance, and restoration time will be compared among these 5 algorithm assemblies. The restoration time is defined as the time interval from the instant an end node of the faulty link detects the failure to the instant wavelength reservation of any restoration path is successfully finished. We assume the propagation delay on each link is $5e - 4s$, packet processing time is $1e - 5s$, and wavelength reservation delay is $1e - 5s$. Any data shown in this simulation is the average value taken over 5 different random seeds.

### 3.4.2 Results and Discussions

The results in Fig. 3.9 show the working and restoration blocking probability of the 5 algorithm assemblies under different traffic load. For *none, LCB*, and *PB* algorithm, both working and restoration blocking probability results are given. However, for *PBH* and *PBHD*, we only do the experiment on their restoration blocking probability because their routing and wavelength assignment behaviors are exactly the same with *PB* so that the identical working blocking probability results with *PB* are believed. Both working and restoration blocking probability will increase as the traffic load on each node increases due to more and more serious wavelength resource scarcity. Restoration blocking probability will grow more rapidly than working blocking probability, which is reasonable since finding
free wavelengths for restoration is more difficult due to \textit{mutual competition}. We can find that in most scenario, in terms of the restoration blocking probability, $PBHD < PBH < PB < LCB < \text{none}$ is satisfied. Clearly, \textit{none} has the worst performance merely because no optimization of restoration blocking is provided to it. $PB$ is better than $LCB$ with the benefit from more sophisticate and accurate restoration resource analysis. Furthermore, $PBHD$ and $PBH$ are much better than the other 3 algorithms. That is because in them both \textit{primary holding} and \textit{pseudo mutual competition} are handled respectively by $PB\_SEA$ and heuristics shown in section 3.3. As we mentioned previously, restoration blocking cases due to \textit{mutual competition} happen more frequently than blocking cases due to \textit{primary holding}, which can explain why improvement made by $PBHD$ and $PBH$ is somehow more than $PB$ and $LCB$. Another finding from the figures is that working blocking probabilities of \textit{none}, $LCB$, and $PB$ are fairly close to each other. It shows that our routing and wavelength assignment algorithms $LCB\_SEA$ and $PB\_SEA$ do not affect the working blocking performance a lot, which is really desirable as the satisfaction to our objective.

![Figure 3.9 Blocking probability VS. load per node](image)

Some results about restoration time can be seen from Fig. 3.10. The 2 bottom
curves are for $PB$ and $LCB$ algorithms, which shows that the restoration speeds of $PB$ and $LCB$ are higher than none. We believe this because by employing $PB$ and $LCB$, the restoration wavelength resource scarcity is mitigated especially on the inner restoration paths so that more faster restoration process can be made through these inner restoration paths. Observe that the average restoration time suffered by $PBH$ is longer than none, and $PBHD$ is even worse. In $PBH$, in order to intentionally utilize the working wavelength in restoration process, some restoration paths with shorter latency may be supplanted, which will lead to slower restoration speed. In $PBHD$, besides the reason we just mentioned, a delay for none-working-wavelength reservation introduced by $PBHD$ will also increase the restoration time. Under light traffic loads, we can find that the restoration time increases with the traffic load increasing for all algorithms, which is intuitively because the longer working paths which have more restoration paths options will be preferred. However, when the traffic load further increases, the restoration time tends to decrease because the shorter working paths will be used more frequently.

![Figure 3.10](image)

**Figure 3.10** Restoration time VS. load per node

Fig. 3.11 and Fig. 3.12 shows how $PR$ index $\alpha$ affects the working and restoration blocking probability under traffic load 8 in $PB$ and $LCB$ algorithm. From Fig. 3.11, we can find when $\alpha$ approaches 0, both $PB$ and $LCB$ will work similar as none, since less
effort to improve restoration blocking probability will be made in the algorithms. With \( \alpha \) increasing, the restoration blocking is improved. However when \( \alpha \) becomes further larger (approximately when \( \alpha > 1 \)), restoration blocking is approaching some limit. From Fig. 3.12, we can find the working blocking probability will almost increase monotonously with the increasing of \( \alpha \). So \( \alpha = 1 \) might be nice to balance the tradeoff between working and restoration blocking probability for both PB and LCB algorithms.

Fig. 3.13 shows how RSV-DELAY affects restoration blocking probability under under traffic load 8 in PBHD. we can find when RSV-DELAY = 0, PBHD is exactly same
Figure 3.13 Restoration blocking probability VS. RSV-DELAY

as PBH. From the figure, the restoration blocking is decreasing with RSV-DELAY increasing, which is clear due to higher probability for the disrupted connections using their own working wavelength to do restoration by avoiding pseudo mutual competition. Similar with $\alpha$, when RSV-DELAY keep increasing (approximately when $\alpha > 1$), restoration blocking will also approaching some limit. Since restoration speed will definitely increase with RSV-DELAY increasing, RSV-DELAY should be carefully adjusted by the network operator.

3.5 Conclusion

To address the restoration blocking probability problem in WDM optical networks, several dynamic routing and wavelength assignment algorithms and heuristic algorithms are proposed. First we give 2 main restoration blocking reasons: primary holding and primary holding. To solve primary holding, dynamic routing and wavelength assignment algorithm $LCB.SEA$ and $PB.SEA$ are proposed to evaluate the congestion level and restoration blocking capability of the candidate primary paths and wavelengths for each incoming connection request and choose the path and wavelength with the best evaluation. We also design 3 heuristics to avoid primary holding by encouraging the working
wavelengths to be used in restoration process.

Simulation shows that our algorithms can clearly reduce the restoration blocking probability as well as not affecting primary blocking probability and restoration speed much.
CHAPTER 4. Summary

In chapter 2, we proposed an Upstream Node Initiated Fast Restoration (UNIFR) scheme for MPLS networks. The key idea of UNIFR is to let upstream node adjacent to the failure use semi-global restoration paths (from itself to the destination nodes) to restore traffic immediately after detecting a failure. This way, fast restoration speed like link-based scheme with more efficient spare capacity utilization than link-based scheme can be achieved. We introduced an MPLS resilience framework supporting UNIFR, which greatly simplifies both dynamic connection provisioning and traffic restoration by pre-planning working/spare capacities in each network link and pre-establishing working/backup paths for each demand relation in the given traffic matrix. We gave two ILP formulations to solve the network planning problem, which computes the optimal backup paths and spare capacity allocation for UNIFR-based restoration model with and without allowing restoration traffic splitting respectively. Simulation results showed that UNIFR-based scheme can achieve capacity efficiency much better than link-based scheme and close to path-based scheme. In addition, the average number of backup paths maintained by each node under UNIFR-based model is about the same as under path-based model when restoration traffic splitting is allowed. When restoration traffic splitting is not allowed, the average node complexity of UNIFR based model is upper bounded by $K \cdot \Delta \cdot N$ where $\Delta$ is the average node degree and $N$ is the number of nodes in the network.

We also designed dynamic routing and wavelength assignment algorithms and heuristics to reducing restoration blocking probability in WDM optical networks. First we give 2 main restoration blocking reasons of the current dynamic fast restoration protocol $MIP$.
which are primary holding and mutual competition. To solve primary holding, dynamic routing and wavelength assignment algorithm LCB_SEA and PB_SEA are proposed to evaluate the primary and restoration blocking performance for the candidate primary paths and wavelengths for each incoming connection request and choose the primary path and wavelength with the best evaluation. We also design 3 heuristics to reducing mutual competition by encouraging the working wavelengths to be used by the simultaneous disrupted connections in restoration process so that the wavelength reservation confliction can be avoided as much as possible. Simulation shows that our algorithms can clearly reduce the restoration blocking probability than the traditional restoration schemes with little influence to the primary blocking probability and restoration speed.
REFERENCES


ACKNOWLEDGMENTS

I would like to take this opportunity to express my thanks to those who helped me with various aspects of conducting research and the writing of this thesis.

First and foremost, I am indebted to my major advisor: Dr. Lu Ruan. I am grateful to her for her instruct, patience, support and concern during these years. I learned not only a lot research and writing skills, but also advice and morality as the estates for life from Dr.Ruan whose vision and impeccable insights have been and will continue to be an inspiration to me.

I would also like to thank my committee members, Dr. Arun Somani and Dr. Ying Cai for their valuable comments and suggestions during my oral examination and thesis writing.

I would like to thank other members in my research group, Chang Liu and Fangcheng Tang. Without them, this work would not have been completed as smoothly and successfully as it has been.

Finally, I would also like to thank my family for the support they provided me through my entire life and in particular, I appreciate the love and support provided by my girlfriend, Huan, in the past years.