NOTE TO USERS

This reproduction is the best copy available.

UMI[®]

MPLS Protection for Ethernet-Based Network

Submitted by

Yi Lei

A thesis Submitted to

the Faculty of Graduate and Research

in partial fulfillment of the requirements for the degree of

Master of Science

Department of System and Computer Engineering

Carleton University

Ottawa, Ontario, Canada, K1S 5B6

September 9th, 2003

Copyright © 2003 Yi lei

National Library
of Canada

Bibliothèque nationale
du Canada

Acquisitions and
Bibliographic Services

Acquisitions et
services bibliographiques

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 0-612-88774-X

Our file Notre référence

ISBN: 0-612-88774-X

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

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

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

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

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this dissertation.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de ce manuscrit.

While these forms may be included in the document page count, their removal does not represent any loss of content from the dissertation.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

The undersigned hereby recommend to
The Faculty of Graduate Studies and Research

Acceptance of the thesis

MPLS Protection for Ethernet-Based Network

Submitted by

Yi Lei

in partial fulfillment of the requirements
for the degree of Master of Information and System Science

Dr. Chung-Horng Lung, Thesis Supervisor

Dr. Rafik A. Goubran, Chair, Department of System and Computer Engineering

Carleton University

September 9th, 2003

Abstract

With the increasing popularity of the Internet, network reliability has become an important issue. Link failure is one of the main causes of service interruption. Protection at the physical layer is fast, but requires dedicated hardware. On the other hand, IP, although it does not rely on specific topology, is slow.

MPLS protection is the mechanism for rerouting traffic to a backup LSP in the case of failures. This research presents a methodology for addressing MPLS path, link and node protection, with the primary emphasis on the last two. The approach is a fast and efficient mechanism for rerouting packets when there is a link/node failure in an Ethernet-based network. The mechanism proposed in this research introduces a strategy for pre-establishing both the primary LSP and its backup LSP in order to quickly restore the traffic to the backup LSP in case of a failure. A fast and reliable mechanism for detecting a link failure in an Ethernet-based network is also introduced. In addition, this research also presents the design and implementation of the MPLS forwarding engine to support the link/node protection. The experiments executed in this research show that traffic recovery can be achieved in about 30ms.

Keywords -- MPLS, link/node protection, restoration, failure detection, Ethernet

Acknowledgement

I would like to thank Dr.Chung-Horng Lung who gave me supervision of the research and patiently accepted to review the different versions of this thesis. It was a great pleasure to conduct this thesis under his supervision. Also, I would like to thank Dr.Anand Srinivasan, for providing a motivating, enthusiastic, and critical atmosphere during the many discussions we had.

Thanks Dr.Pramod Dhakal for his help and encouragement in the implementation stage of the thesis.

Special thanks to my friend Hossam Khalil for his help during the writing.

My colleagues in EION all gave me the feeling of being at home at work. Many thanks are given to my colleagues.

Lastly, I am very grateful for my husband Xuefeng Zhang, for his love and patience during the research.

Table of Contents

Table of Contents	i
List of Figures	v
List of Tables	ix
Abstract	x
Acronyms	xi
Chapter 1 Introduction	1
1.1 Motivation	1
1.2 Contributions	5
1.3 Thesis Outline	6
Chapter 2 Literature Review	8
2.1 Multi Protocol Label Switching	8
2.1.1 History	8
2.1.2 Label Switched Path	10
2.1.3 Label Routers	12
2.1.4 Advantages	14
2.2 MPLS Connection Management	19
2.2.1 Resource Management	19
2.2.2 Admission Control	19
Chapter 3 Problem Statement	21
3.1 Research Questions	21

3.2 Problem Definition.....	22
3.3 Scope.....	23
Chapter 4 MPLS Protection Methodology	25
4.1 Path Protection.....	27
4.2 Link Protection.....	31
4.2.1 Packets Encapsulation along Backup LSP.....	34
4.3 Node Protection	37
4.4 Two Levels of Label Stack	39
4.5 Working Example	42
Chapter 5 Implementation.....	44
5.1 Framework	45
5.2 Implementation Settings	47
5.2.1 Open IP With MPLS.....	48
5.2.2 Internal Message Queue.....	49
5.2.3 Label Space	50
5.2.4 Design of the FEC.....	50
5.2.5 Label Merging.....	50
5.2.6 Data Structures.....	51
5.2.7 Messages Distinction	56
5.2.8 Command Line Interfaces.....	57
5.3 Failure Detection.....	58
5.3.1 Detection by Signal Protocol	59
5.3.2 Detection by Physical Layer	60

5.3.3 Auto-Negotiation in Ethernet.....	60
5.3.4 Failure Detection.....	61
5.4 Failure Notification.....	62
5.5 Forwarding Engine.....	63
5.5.1 State Machine.....	65
5.5.2 Switchover	67
5.6 Sequence Diagrams.....	69
5.6.1 Forwarding Engine Structure.....	69
5.6.2 Packet Processing Dynamics	73
Chapter 6 Performance Analysis	77
6.1 Experiment Plan.....	79
6.1.1 Ingress Node as the PLR.....	79
6.1.2 Tandem Node as the PLR	81
6.1.3 1:n Protection.....	83
6.1.4 Hardware Used for the Experiments.....	84
6.2 Data Analysis Formula	85
6.3 Failure Detection Time	86
6.4 Failure Notification Time	88
6.5 Switchover Time.....	90
6.6 Recovery Time.....	92
6.7 Summary	95
7.1 Conclusions.....	96
7.2 Future Work.....	97

References.....	99
-----------------	----

List of Figures

Figure 2. 1 MPLS Based Network.....	11
Figure 2. 2 Traffic Engineering Based on MPLS Forwarding.....	16
Figure 2. 3 MPLS Protection Mechanisms: an Illustration.....	18
Figure 4. 1 Path Protection.....	29
Figure 4. 2 Data Structure in NHLFE table before Switchover.....	30
Figure 4. 3 Data Structure in NHLFE Table after Switchover	30
Figure 4. 4 Link Protection	32
Figure 4. 5 Data Structure in FIB Table before Switchover	33
Figure 4. 6 Data Structure in FIB table after Switchover	34
Figure 4. 7 Packets Forwarding along Backup LSP	36
Figure 4. 8 1:n Protection Model in Link Protection.....	37
Figure 4. 9 Node Protection.....	38
Figure 4. 10 1:n Protection.....	41
Figure 4. 11 Packets Forwarding through Backup LSP.....	42
Figure 5. 1 OPEN IP System [OIP-1, OIP-2]	45
Figure 5. 2 Open IP System with MPLS.....	48
Figure 5. 3 Internal Message Process	49
Figure 5. 4 Label Merging	51
Figure 5. 5 Ip2Fec table Data Structure	53

Figure 5. 6 NHLFE table Data Structure	54
Figure 5. 7 Data Dependency between Ip2Fec and NHLFE	55
Figure 5. 8 FIB table Data Structure	56
Figure 5. 9 Packet Distinction Block Diagram	57
Figure 5. 10 FLP Burst Timing.....	61
Figure 5. 11 Link Failure Detection Timing.....	62
Figure 5. 12 MPLS Forwarding Engine Architecture.....	64
Figure 5. 13 MPLS Forwarding Engine State Machine.....	66
Figure 5. 14 Forwarding Engine Switchover upon Failure.....	68
Figure 5. 15 Forwarding Engine UML Class Diagram.....	70
Figure 5. 16 Ingress LSP Setup Sequence Diagram	71
Figure 5. 17 LSR LSP Setup Sequence Diagram	72
Figure 5. 18 Switchover Sequence Diagram.....	72
Figure 5. 19 Sequence Diagram for Ingress Packet Process.....	74
Figure 5. 20 Sequence Diagram for PLR.....	75
Figure 5. 21 Sequence Diagram for MPL	76
Figure 6. 1 MPLS Recovery Cycle Model.....	78
Figure 6. 2 Ingress Node (Link Failure) Experiment.....	80
Figure 6. 3 Ingress Node (Node Failure) Experiment.....	81
Figure 6. 4 Tandem Node (PLR) Link Failure Experiment.....	82
Figure 6. 5 Tandem Node (PLR) Node Failure Experiment.....	83

Figure 6. 6 1:n Protection Experiment.....	84
Figure 6. 7 Distribution of the Failure Detection Time	88
Figure 6. 8 Distribution of the Failure Notification Time	89
Figure 6. 9 Distribution of the Switchover Time.....	91
Figure 6. 10 Distribution of the Total Recovery Time	93
Figure 6. 11 Recovery Time Weight Distribution	94

List of Tables

Table 4.1: NHLFE Table contents before switchover	29
Table 4.2: FIB Table contents before switchover	33
Table 4.3 Label recorded in FIB table	35
Table 4.3 Label recorded in FIB table.....	41
Table 6.1 Switchover Time for Various LSP#	92
Table 6.2 Recovery Time Weight Distributions	95

Acronyms

ARIS – Aggregate Route-based IP Switching

CR-LDP – Constraint-based Label Distribution Protocol

CLI – Command Line Interface

CSR – Cell Switching Router

FE – Forwarding Engine

FEC – Forwarding Equivalence Class

FIB – Forwarding Information Base

FLP – Fast Link Pulse

IGP – Interior Gateway Protocol

IP – Internet Protocol

ISP – Internet Service Provider

LDP – Label Distribution Protocol

LER – Label-Switched Edge Router

LSP – Label Switched Path

LSR – Label Switching Router

MPL – Merge Point LSR

MPLS – Multiple Protocol Label Switch

NHLFE – Next Hop Label Forwarding Entry

NLP – Normal Link Pulse

OSPF – Open Shortest Path First

PLR – Point of Local Repair

PDU – Protocol Data Unit

QoS – Quality of Service

RSVP – Resource Reservation Protocol

RSVP-TE – Resource Reservation Protocol – Traffic Engineering

SNMP – Simple Network Management Protocol

Chapter 1 Introduction

This chapter presents an overview of the research. The chapter begins with a section on motivation, which elaborates the benefits of providing the link/node protection in the MPLS layer. The motivation clarifies some key research implications, including: the choice of MPLS layer for the failure protection, the mechanism used to detect the failure in an Ethernet-based network, the design of the forwarding engine for supporting the MPLS protection, and finally the performance analysis of the proposed mechanism and implementation. The second section identifies the research contributions of the thesis. The third section sets out the organization of this thesis.

1.1 Motivation

With the increasing need for real-time application services, network reliability and survivability have become more important in the Internet. The recovery mechanism used to restore traffic in case of a failure is also getting a lot of attention.

Protection against and restoration after link and node failures can be performed at several layers in a network [Demmester-99]. Given the protocol stack shown in Figure 1.1, the mechanisms for protection and restoration can be placed at the network layer, layer 2 or the physical layer.

One approach is to provide the protection/restoration in the network layer. In this layer, the IP uses IGP protocols [Ayandeh-02] such as OSPF, ISIS and RIP to maintain and update its routing table. In the case of a failure, the IGP protocol takes into account the topology change and re-computes the routing table by using the shortest path algorithm. When all the routing tables in the network are recomputed, the traffic traveling through the original route will be redirected through the new route. Even though this has been proven to be robust and survivable, the recovery coverage of this methodology is slow and can take a long time, from several seconds to minutes [Huang-02].

Figure 1. 1 MPLS and the Simplified Protocol Stack

Another approach is to provide protection/restoration in the physical layer, where SONET is used for optical transmission [Bellcore/GR-1400-95]. Automatic Protection Switching protocol (APS) [Bellcore-90] is used to switch over from a failed fiber to a

protected fiber. APS involves a window of time of 10 ms for fault detection and can achieve a recovery in 50 ms [Demmester-99]. However, this fast recovery is achieved at the expense of inefficient use of bandwidth and is typically limited to SONET/SDH ring-based topologies [Ayandeh-02]. Since failure detection in the SONET/SDH is achieved by the loss of light, it requires dedicated hardware.

Protection at the physical layer is fast but requires dedicated hardware. Conversely, IP does not rely on specific topology, but it is slow. MPLS, which is implemented between the IP and MAC layers, supports recovery mechanisms that provide a trade-off between recovery speed and deployment cost [Ho-01]. MPLS, being a connection-oriented technology on top of the connectionless IP technology, allows a Label Switch Path (LSP) to be set up before the traffic arrives.

This research focuses on protection provided in the MPLS layer. The intent of this research is to cover link/node failure in Ethernet-based networks. A great deal of research has focused on MPLS protection, but mainly for the SONET domain. The reason for focusing on the MPLS layer is its low cost and fast recovery. On the other hand, Ethernet is currently battling its way out of the enterprise networks by developing new technology along with utilizing the existing ones.

Ethernet-based network failure can be due to many causes; of these, link cut is a common failure. Indeed, in a study conducted between April 1992 and March 1994, Kuhn [Kuhn-97] showed that cable cutting was the cause for 25% of the downtime of telephone networks. In addition, the lower the time taken to detect the failure, the faster the

recovery and the smaller the packet loss. It is very important that the failure be detected in as little time as possible.

In MPLS networks, several different methods have been proposed for detecting link failure. Basically, there are two categories for detecting failure: it can either be detected by the signaling protocols or by the physical layer. In the MPLS, RSVP-TE [Osborne-02] is the signaling protocol used to set up explicit LSP, and LDP is the protocol used to set up the hop-by-hop LSP dynamically. In both protocols, a hello message is sent out periodically to indicate that the neighbor, LSR, is alive. The absence of a hello message can be used as an indication of link failure.

The hello message is an optional functionality in RSVP-TE. It is not necessary for every router to send out such messages. In LDP, it takes 15 seconds to detect the failure [Wu-01].

In SONET/SDH, failure detection is triggered by the loss of a signal, which can be detected in 10ms. However, this method is only used for SONET/SDH. As the Ethernet makes its way towards the core of the network [JUNOS-03], Ethernet-based network technologies are becoming more popular. This research proposes a reliable and efficient mechanism for detecting a link failure in the Ethernet.

In summary, the main objectives of this thesis are to:

- Conduct research on providing protection in the MPLS networks. The research

focuses on the link and node protection and the failure detection in the Ethernet-based network. In addition, in order to support the link/node protection, an MPLS forwarding engine design and implementation is conducted.

- Conduct a performance analysis of the proposed protection mechanism, analyzing different factors that affect its efficiency.

1.2 Contributions

The contributions of the thesis can be summarized as follows:

- A methodology for link/node protection is proposed. The methodology enables the pre-establishment of the primary backup LSP, while the Point of Local Repair (PLR) detects a link/node failure. Moreover, the PLR switches over to the backup LSP.
- An Ethernet network failure detection algorithm is adopted and adapted to MPLS protection problems. The algorithm enables an efficient and reliable detection of the failure. The algorithm is also protocol-independent.
- A performance evaluation of the proposed mechanism and algorithm is presented. The factors that affect the performance are analyzed, including the failure detection time, notification time and switchover time
- The MPLS forwarding engine was designed and implemented to support fast

recovery in case of a failure.

1.3 Thesis Outline

In this dissertation, Chapter 2 contains the literature review. It highlights the history of MPLS and the main components of MPLS, which include label switched paths (LSPs) and label switching routers (LSRs). Section 2.4.1 discusses the advantages provided by MPLS. Section 2.4 discusses MPLS connection management. This section also covers the topic of resource management and admission control.

Chapter 3 discusses the MPLS link/node protection problem and raises some research questions related to failure detection, the MPLS forwarding engine and the data transfer performance in the network.

Chapter 4 describes the mechanism and algorithm for link/node protection. It starts with failure detection in the Ethernet-based network, followed by the introduction of the failure notification to the forwarding engine and the algorithm for the switchover. Section 4.3 discusses the path protection and also mentions its disadvantages. Section 4.4 illustrates the link protection, and section 4.5 illustrates the node protection. Section 4.6 introduces the design MPLS forwarding engine.

Chapter 5 focuses on the MPLS protection implementation rationales, the underlying framework and the environment settings.

Chapter 6 sets out a link/node failure protection performance analysis in an Ethernet-based network. The data was measured in several testing sessions to characterize the MPLS protection performance, and the factors affecting the performance are evaluated.

Chapter 7 concludes this thesis and provides future research suggestions.

Chapter 2 Literature Review

This chapter provides a literature review of the various aspects of the MPLS [Black-01, Rosen-01] technology. The chapter includes the history of MPLS and its associated terminologies and definitions (e.g. Label Switched Path, Label Switching Router). Section 2.1.4 discusses the advantages of MPLS as regards traffic engineering and fault tolerance. Section 2.2 presents the connections management in the MPLS network, including resource management and admission control.

2.1 Multi Protocol Label Switching

The roots of MPLS [Davie-00] go back to numerous efforts in the mid-1990s to combine IP and ATM technologies. Toshiba's Cell Switch Router was the first to be presented to the Internet Engineering Task Force (IETF), while the first such effort to reach the marketplace was IP switching, developed by Ipsilon. To compete with this offering, numerous other companies came up with their own products, notably Cisco Systems (Tag Switching [Alwayn-02]), IBM (aggregate route-based IP switching), and Cascade (IP Navigator [Davie-00]).

2.1.1 History

Toshiba's Cell Switch Router (CSR)

CSR [Davie-00] was first presented to the IETF working group meeting in 1994 and early 1995. CSR's architecture introduced the idea that an ATM-switching fabric could be controlled by IP protocol rather than ATM signaling protocols.

Ipsilon's IP Switching

Announced in 1996, IP Switching made a much larger impact in the marketplace than Toshiba's CSR. Its significant contribution was the specification of a simple switch protocol (GSMP) that enabled virtually any ATM switch to be turned into an "IP switch" with the addition of an external controller.

Cisco's Tag Switching

A few months after Ipsilon's release, Cisco Systems announced its label switching approach – Tag Switching. Tag Switching is a significant technical departure from IP Switching and the CSR approach. Cisco produced an informational RFC describing the approach.

IBM's ARIS

Shortly after Cisco's announcement, IBM described another label switching approach – Aggregated Route-based IP Switching (ARIS). ARIS has a lot in common with Tag Switching in that both use control traffic to set up forwarding tables, but it also differs from Tag Switching in some significant ways.

MPLS

In response to these proprietary initiatives, the IETF set up the MPLS working group in 1997 to develop a common, standardized approach. The working group issued its first set of Proposed Standards in 2001. Meanwhile, however, the market did not stand still. The late 1990s saw the introduction of many routers that were as fast as ATM switches, eliminating the need to provide both ATM and IP technology in the same network.

2.1.2 Label Switched Path

Forwarding Equivalence Class (FEC) is a group of IP packets which are forwarded in the same manner (e.g., over the same path, with the same forwarding treatment). As defined in RFC3031, an LSP (Label Switched Path [Rosen-01]) is a path through one or more LSRs at one level of the hierarchy, followed by packets in a particular FEC. Figure 2.1 shows an LSP setup from Ingress Router to the Egress Router in the MPLS network. The LSP can be seen as traffic trunks [Redford-00], which carry aggregate packets that are classified to the same FEC.

Figure 2. 1 MPLS Based Network

The creation and distribution of a label binding in an LSR is the result of a particular event – either traffic flow is detected or control information received from a control component, or the network topology is changed. There are three fundamentally different approaches to the creation and the distribution of label bindings:

- Data-driven
- Control-driven
- Topology-driven

2.1.2.1 Data-driven

In the data-driven model, whenever data packets for which there is no existing LSP arrive, a label request will be triggered, the label bindings are created and the LSP is set

up. The protocol LDP is used for this purpose.

2.1.2.2 Control-driven

In the control-driven model, the label binding and LSPs are created by the control information. This process happens before the data packets arrive. The LSPs are created and the label bindings are distributed in response to two factors:

- Control information from the signaling protocols such as CR-LDP and RSVP-TE.
- Control information from the Command Line Interface.

2.1.2.3 Topology-driven

In the topology-driven model, whenever the network topology changes because a link is broken, the LSPs along the broken link should be deleted and the information should be sent back to the ingress nodes of the LSPs. At the ingress node, the data packets will be rerouted to an alternative LSP, if it has been pre-setup, or the LDP protocol will be triggered to find a new LSP.

2.1.3 Label Routers

According to the role it plays in the MPLS based network, there are two types of label routers: Label Edge Routers (LER) and Label Switching Routers (LSR).

2.1.3.1 Label Edge Routers

The node at which an LSP starts is called the Ingress Label Edge Router (I-LER). The node where an LSP terminates is called the Egress Label Edge Router (E-LER).

I-LER

Ingress LER is the initial node through which the MPLS packets destined for that particular LSP enter. When an IP packet enters, the destination IP packet is mapped to a unique Forward Equivalence Class (FEC). I-LER receives the IP datagrams and classifies the packets into an FEC, assigns it a label and then builds and attaches a MPLS header to the datagrams. The I-LER performs the PUSH operation.

E-LER

The egress LER is the end-point MPLS router for a LSP. It de-encapsulates the data traffic and removes the MPLS header, then sends out the packets as IP packets. It performs a POP operation.

2.1.3.2 Label Switching Routers

The LSP goes through various intermediate nodes, all named Tandem Label Switching Router (LSR). Tandem LSRs are the intermediate MPLS routers in an LSP.

An LSP receives the MPLS Protocol Data Unit (PDU) and uses the label in the MPLS header to make the forwarding decision, then performs the label swapping through a

SWAP operation.

2.1.4 Advantages

The MPLS forwarding method provides several advantages over the IP routing technique.

Those advantages are described in more detail below:

More Flexibility Using FEC

The label-swapping forwarding algorithm [Alwayn-02] used by MPLS gives an ISP tremendous flexibility in the way in which it assigns packets to FECs. It also has the ability to take any type of user traffic and assign it to the same FEC. It offers ISPs precise control over the flow of traffic in their networks. The FEC for a packet can be decided by one or more parameters, carried in the packets. Primarily, the FEC can be identified by the IP header information in the packets, such as the source address and destination address; it can also be identified by the port number carried in the TCP header.

The identification of the FEC is an issue that can be implementation-dependent. The decision could be based on the requirement of the customer or made by the network manager. It could be based on one parameter, but it could also be based on a combination of several parameters. The following parameters are taken into consideration, in general:

- Source IP Address
- Destination IP Address

- TCP/UDP Source or Destination Port number
- IP protocol ID
- QoS (Differentiated service codepoint).

Easier and Faster Label-Swapping Forwarding Algorithm

The label-swapping forwarding algorithm also simplifies the procedure of forwarding packets. The look up of the IP header and the longest prefix match is just performed at the Ingress LER, and a label will be assigned to each packet. After that, all the forwarding decisions will be based on the label carried in the MPLS headers.

Making Explicit Routes Possible

An explicit route [Black-01] is a pre-configured sequence of hops that the packet will transmit along the service provider's network. Therefore, the construction of the LSPs could be different to the conventional destination-based routing. By using the MPLS, ISPs can set up LSPs by adding control over traffic flows, making it possible to support traffic engineering, QoS and loop prevention.

Constraint-based routing [Lawrence-01] makes use of signaling protocols to provide enhanced capabilities to allow ISPs to set up an explicit route which meets the specific performance requirements, such as bandwidth and delay.

Section 2.1.4.1 introduces the traffic engineering in the MPLS, while section 2.1.4.2

discusses the fault-tolerance of the MPLS.

2.1.4.1 Traffic Engineering

Traffic engineering allows ISPs to move traffic flows from the shortest path calculated by the IGP to the other path, which is less congested, across the network. It also allows the ISPs to set up multiple LSPs for fast re-routing.

By configuring the explicit route, the network administrators have more control over the network.. This results in a network that uses network resources more efficiently, and is growing in demand. It can also offer the flexibility required to meet constantly changing customer expectations, since the network manager can have the full control over the MPLS-based network. This is illustrated in Figure 2.2 below.

Figure 2. 2 Traffic Engineering Based on MPLS Forwarding

As shown in Figure 2.2, for the IP network, since the routing decision is based on the IP destination address, all the packets from router A and router B to router E will go through path 1, which is the shortest path calculated by the IGP [Viswanathan-98] protocol. If there is congestion in path 1, the packets will still go through path 1 in the conventional IP network. In an MPLS network, the network administrator can create an alternative path 2, which is less congested, by assigning a different FEC and label to the traffic flow from router B to router E. That way, the network administrator can force the packets from Router B to Router E to go through path 2.

2.1.4.2 Fault-Tolerance

Reliability is a key issue in the network and can be provided by different fault management mechanisms at different network levels. In the IP network, since each router will examine the packets' destination address and conduct the longest match in the forwarding table, the packets will follow the same shortest path populated by the routing protocol, such as OSPF. So when a fault occurs, the packets will not be forwarded until a new route has been established by the routing protocol.

Being a connection-oriented protocol, MPLS is more robust to failure [Hundessa-01]. LSPs can be set up prior to the traffic. This means that both the primary LSP and the backup LSP can be set up beforehand. In case of a failure, the traffic can quickly be rerouted around a failed link or router; that is, the packets are forwarded along the backup LSP after a failure on the primary LSP.

The backup LSP can be established statically beforehand or dynamically [Hundessa-01] after the occurrence of the failure on the primary LSP. According to how the backup LSP is used upon the failure of the primary LSP, there are two possibilities: path repair and local repair [Kuhn-97]. In path protection, also known as end-to-end protection, the protection is activated at the ingress LER regardless of where the failure occurs. Typically, this requires that the backup LSP be unconnected to the primary LSP. The LSP2 shown in Figure 2.3 demonstrates this concept. The primary path and the backup path are totally separate, except for the ingress and egress routers. In local repair, the protection is activated by the LSR, which detects a failure. There are two types of local repair protection: link protection and node protection. In the link protection, the backup LSP is set up between two LSRs on each end of the broken link. In the node protection, the backup LSP is set up between the LSPs which are upstream and downstream of the failed router. Figure 2.3 illustrates an example of node protection with a LSP4. The node protection segment bypasses the failed node and connects to its downstream node.

Figure 2. 3 MPLS Protection Mechanisms: an Illustration

2.1.4.3 Protection for Ethernet

The Metro Ethernet Forum [Chen-03] is supporting a technique that utilizes MPLS to enhance the resiliency and reliability of the LAN technology, in order to make Ethernet the "carrier-class" for metropolitan-area network deployment. The Metro Ethernet Forum has defined two key implementations for delivering SONET/SDH-like protection for Metro Ethernet Networks; these are OA&M-Based End-to-End Path Protection (EEPP) and Aggregated Line and Node Protection (ALNP).

2.2 MPLS Connection Management

Management of the MPLS connection includes resource management, admission control and protection/restoration. The resource management is responsible for tracking the availability of resources in the network, while the admission control makes decisions based on the information offered by the resource management.

2.2.1 Resource Management

The resource manager provides a method for keeping track of the resource reservation information in a router. The resource management [Gerald-99] deals with bandwidth allocation and reservation in the network to provide good network performance under normal and abnormal operating conditions, for all applications that are sharing the resources in the network. In an MPLS network, the resource management module will keep track of the available bandwidth, the bandwidth that can be reserved and the

bandwidth allocated to each LSP.

2.2.2 Admission Control

In the MPLS network Admission Control is a key aspect in the support of Quality of Service (QoS) and traffic engineering. Admission control is a process used by the network to regulate its network load [Chen-98]. It is used to avoid network congestion [Bosco-03]. According to the current network resources information and the requirements of the LSP, admission control checks the resource availability, and makes the decision as to whether the resource is sufficient or not for setting up the LSP. Admission control discards requests that will employ too many resources to guarantee QoS.

In an MPLS network, LER checks the admission control to compute an explicit path. If the computation succeeds, the LSP is established along the new path through RSVP-TE or CR-LDP signal protocols. During LSP set-up, each node checks the availability of resources for the LSP and eventually either rejects the request, if not enough resources are available, or allows it to continue if there are sufficient resources.

Chapter 3 Problem Statement

This chapter describes the problem statement for the research. The first section states the research questions that are relevant to the thesis. Some of the questions are answered in the subsequent chapters, and some remain for future research. In the third section, the thesis scope prescribes the limitations and boundaries of this study.

3.1 Research Questions

This research proposes a methodology for MPLS protection for Ethernet-based networks. The proposed methodology addresses the following questions:

- How can link/node protection be supported at the MPLS layer?
- How should a simple failure detection mechanism be provided in an Ethernet-based network without the overhead of complex signaling protocols? Is the mechanism efficient and reliable?
- Does the MPLS link/node protection need the support of a forwarding engine?
- How can an MPLS protection/restoration be achieved with relatively satisfactory performance?

3.2 Problem Definition

The main concern of this research is the MPLS protection in Ethernet-based networks. This research addresses the above research questions by focusing on the following problems:

Providing a lightweight MPLS protection scheme

Protection and restoration against link and node failures can be performed at various layers in a network, such as the network layer, link layer or physical layer [Ayandeh-02]. Protection in the network layer has been proven to be robust and survivable. However, the protection coverage in the network layer is slow and can take from several seconds to minutes since the IP is using IGP protocol to exchange the routing information [Huang-02]. Conversely, protection at the physical layer is fast, but requires expensive, dedicated hardware [Bellcore/GR-1230-95].

MPLS is a connection-oriented mechanism over the connectionless IP network. By statically pre-establishing a backup LSP before the occurrence of a failure [Rosen-01, Redford-00], MPLS has the ability to increase network reliability by enabling a faster response to faults than that possible with the traditional Layer 3 (or IP layer) approaches alone. Meanwhile, it still provides the visibility of the network offered by Layer 3. Also, protection in the MPLS layer does not need dedicated hardware.

Failure Detection for an Ethernet-based network

Ethernet is gaining more popularity than ever before, especially with the coming of the G-Ethernet [Spurgeon-00]. Unfortunately, at the time this thesis was written, there was not much research on MPLS protection for Ethernet-based networks in the literature.

Forwarding Engine Support

The forwarding engine is the core module for data forwarding. It receives the packets from the incoming interface, and according to the packet type, sends the data packet out to the outgoing interface and delegates the control packets to the corresponding applications. The MPLS forwarding engine maintains the forwarding information tables, which provides the mapping between the IP addresses and their associated FECs. The forwarding information tables are also responsible for the mapping between the incoming labels and the outgoing labels. If a failure occurs along the primary LSP, the forwarding engine receives a failure notification message. Consequently, the forwarding engine reroutes the traffic from the failed primary LSP to the backup LSP.

3.3 Scope

- The proposed forwarding engine is applied to MPLS unicast. MPLS multicast is beyond the scope of this research.
- The protection methodology proposed in this research is based on the static pre-

establishment of both the primary LSPs and their associated backup LSPs.

- In LSP setup, traffic engineering parameters, such as bandwidth, are not considered in this research.
- The implementation of an MPLS forwarding engine is based on the Eion Open IP framework. Modifications are needed to port to any third party system.
- The failure detection mechanism is restricted to the Ethernet networks.
- Path protection is not covered in this research. Path protection usually has a scalability problem [Osborne-02]. It is assumed that the network operator will detect any path failure along the protected primary LSP and manually configure the switchover via the proposed Command Line Interface (CLI).

Chapter 4 MPLS Protection Methodology

The goal of this research is to introduce a methodology for MPLS protection in Ethernet-based networks. The MPLS protection mechanisms enable an LER or an LSR to reroute the traffic, either because of a link failure or a node failure.

This chapter introduces a methodology for network protection at the MPLS layer. The discussion covers the protection of a path (LSP), a link, or a node. The aim of the research is to develop a lightweight, yet efficient MPLS protection methodology to effectively support the systems, without signaling protocols, which introduce higher overhead and are more expensive.

The protection methodology imposes a strategy of pre-establishing both the primary LSP and its associated backup LSP for path protection. If a primary LSP failure occurs, its pre-established backup LSP takes over the ongoing traffic. Unlike network protection at the IP layer, the implemented mechanism does not involve the re-computation of the routing table update, which is a time-expensive process.

Fundamentally, the protection methodology involves the development of a failure detection module, a failure notification module, and a forwarding engine for network traffic switchover. Once a failure is detected, the forwarding engine is notified to begin updating a data structure (forwarding table) with the information of the backup LSP that was associated with the failed primary LSP earlier, via pre-configuration.

Consequently, the forwarding engine performs a traffic switchover, in a process of a failure recovery, through the backup LSP.

The introduced MPLS protection methodology is similarly applied to link protection and node protection. Although the path, link, and node protection rely on similar bases, the efficiency and implementation differ since each protection process has its own aspects, circumstances and details. The path protection also allows for detection, notification, and switchover. In this research it is assumed that the detection for path protection is achieved by the network operator, who would manually initialize a CLI at the ingress router, in order to provide the required information for a switchover. The main reason for this is to support protection in a constrained environment where signaling protocols are not utilized, due either to cost or execution overhead. The information includes details about the primary LSP identifier and its backup LSP identifier. The path protection can widely recover any failure along the LSP, whether it is a node failure or a link failure. However, path protection is responsible for a wide range (from end-to-end), so it is usually slow and not scalable.

Link protection is only concerned with a short range of responsibilities along the path (LSP). It is limited by a chosen link between two routers. Link failure is typically recovered by a straightforward application of the process's sequence of detection, notification and switchover. Link protection offers higher scalability than path protection, but it is based on the assumption that both routers at each end of the link are still up. If the downstream router along an LSP on the protected link is down, the link protection

will not work. To prevent this situation, node protection is required. Node protection follows the same recovery processes as link protection and it also covers link protection.

Section 4.1 introduces path protection functionality. However, because path protection has less scalability and is slow, it is not supported practically. In this research, path protection is primarily designed to support the complete coverage of protection levels. Section 4.2 presents the link protection mechanism in detail, while section 4.3 introduces node protection. Section 4.4 discusses how to achieve 1:n protection using two levels of label stack, and section 4.5 gives a working example that illustrates the usage of two levels of label stack.

4.1 Path Protection

Path protection is the mechanism which protects against any failures along a primary LSP. In this research, the mechanism proposed for path protection is based on certain assumptions for the reasons stated above; namely, lower overhead in terms of cost and execution. Those assumptions include:

1. Failure detection along the protected LSP is the responsibility of a network operator.
2. Switchover process is triggered after the network operator manually issues a CLI at the ingress router. The implemented CLI passes the required information for

the switch-over process.

3. Failure may occur anywhere along the protected LSP rather than the ingress router and egress router.

In typical path protection, the backup LSP must be disjoint from the primary LSP, which implies that all LSRs along the backup LSP are different from the LSRs allocated along the primary LSP. If there are any shared LSRs, it is impossible to protect such an LSP from a possible failure on the shared link or node. The backup LSP is set up in parallel with the protected primary LSP before the data traffic. The switchover is performed only in the ingress router. The rationale for selecting a backup LSP is beyond the scope of this research.

Figure 4.1 shows an example of a path protection mechanism. The figure contains six routers (A, B, C, D, E and F). The primary LSP is pre-established along routers $A \rightarrow B \rightarrow C \rightarrow D$; labels L_a , L_b and L_c are the labels used to forward the packets along the primary LSP. The designated backup LSP is pre-established along routers $A \rightarrow E \rightarrow F \rightarrow D$, where the labels L_m , L_e and L_f are defined to forward the packets. Both the primary LSP and its backup LSP have the same ingress node and egress node, but different LSRs.

Figure 4. 1 Path Protection

Table 4.1 shows the contents in the NHLFE table after both the primary LSP and LSP have been set up. The table stores records which describe the outgoing label and outgoing interface of the LSPs. Both records have the same FEC value. The corresponding data structure, which tracks such records, is shown in Figure 4.2. The FEC value is the search key of the NHLFE table.

Table 4.1: NHLFE Table contents before switchover

Fec	outgoingLabel	OutgoingI/f	OutgoingCid	nextHopAddress
D	La	Aeth1	1	Beth1
D	Lm	Aeth2	2	Eeth1

Figure 4. 2 Data Structure in NHLFE table before Switchover

When a failure occurs at any link or node along the primary LSP, the network operator issues the CLI at ingress router A. Router A performs the switchover from the primary LSP to the backup LSP after receiving the switchover command from the CLI. Consequently, router A forwards the packets along the backup LSP. Figure 4.3 reflects the changes in the NHLFE table after the switchover.

Figure 4. 3 Data Structure in NHLFE Table after Switchover

Path protection has the advantage of protecting against any link or node failure along the primary LSP. However, it has two main disadvantages:

- Failure recovery in the path protection is a slow process, since the detection of the failure and the switchover are carried out manually.
- The path protection lacks scalability. The relationship between the primary LSP and the backup LSP is 1:1. The backup LSP cannot be reused for another primary LSP. Moreover, in a large network environment, it is difficult to find a disjoint path to which a backup LSP can be assigned.

For these reasons, path protection is usually not supported in practice. For this thesis, path protection is designed to support the complete coverage of various levels of protection. The functionality of path protection is provided in this research, but the rest of the thesis will focus on link and node protection.

4.2 Link Protection

As mentioned previously, the advantage of path protection is that it protects against any link or node failure along the LSP, but it is slow and not scalable. This section focuses on link protection. In link protection, the backup LSP is pre-established around the link that needs to be protected. The backup LSP ingress must be the upstream node of the protected link, while the egress is the downstream node of the protected link.

Figure 4.4 shows an MPLS network example of six routers (A, B, C, D, E and F). The primary LSP is pre-established along the routers $A \rightarrow B \rightarrow C \rightarrow D$. Labels L_a , L_b and L_c are used to forward the packet along the primary LSP. The link between routers B and C is the protected link ($B \rightarrow C$). The backup LSP is set up from router B to router C, through routers E and F ($B \rightarrow E \rightarrow F \rightarrow C$). The labels L_m , L_e and L_f are used to forward the packet along the backup LSP. Router B is assigned as the PLR, while router C is assigned as the MPL.

Figure 4. 4 Link Protection

Table 4.2 shows the contents of the FIB table after both the primary LSP and backup LSP have been set up. The table stores the records which describe the outgoing label and outgoing interface of the LSPs. Both records have the same incoming label value. The

corresponding data structure which tracks such records is shown in Figure 4.5. The incoming label value is the search key of the FIB table.

Table 4.2: FIB Table contents before switchover

inLabel	outLabel	OutgoingI/f	NextHopAddr
La	Lb	Beth1	Ceth1
La	Lm	Beth2	Eeth2

Figure 4. 5 Data Structure in FIB Table before Switchover

If a failure occurs in the protected link (B→C), the implemented detection module in the PLR (router B) detects the failure, builds a failure notification message and delegates the message to the forwarding engine located at this PLR. The forwarding engine switches

over to the backup LSP and sends the traffic along the backup LSP. Figure 4.6 reflects the changes in the data structure of the FIB table after the switchover.

Figure 4. 6 Data Structure of the FIB table after Switchover

In the link protection, the failure detection and the traffic switchover are achieved locally, which implies low time consumption. The network topology changes are transparent to the MPL, which merges the traffic back to the original primary LSP(s).

4.2.1 Packets Encapsulation along Backup LSP

This research imposes one level of label stack to forward the packets along the backup LSP. PLR inserts the label that identifies the backup LSP into the packet. When MPL receives the packets from the backup LSP, it assigns the same outgoing label, which is used for the primary LSP, to the packets. Consequently, the packets are merged as

backups for the original primary LSP.

In the example shown in Figure 4.4, once both the primary LSP and backup LSP have been set up via CLI, the contents of the FIB table in router C are depicted in Table 4.3 below.

Table 4.3: FIB Table contents in MPL

inLabel	outLabel	OutgoingI/f	NextHopAddr
Lb	Lc	Ceth2	Eeth1
Lf	Lc	Ceth2	Eeth1

Figure 4.7 illustrates the packet forwarding along the backup LSP by using one level of label stack (the same example network in Figure 4.4 is used). If the B→C link faces a failure, the PLR (router B) switches the traffic over from the primary LSP to the backup LSP. For the incoming packets with label La, router B swaps the label La to label Lm and sends the packet to router E; similarly, router E swaps the label Lm to label Le; and router F swaps the label Le to label Lf and sends the packets to router C. As an MPL, router C swaps the label Lf to label Lc, which is the same outgoing label for the primary LSP, and thereafter forwards the packets along the primary LSP.

**Figure 4. 7 Packet Forwarding along the Backup LSP
with one Level of Label Stack**

Link protection provides more flexibility and increases the network scalability since it uses fewer resources. It can also protect several LSPs traveling through the same shared, protected link to achieve 1:n protection in the MPLS network. Figure 4.7 illustrates the 1:n protection model by using the link protection methodology. In this figure, three protected LSPs: LSP1, LSP2 and LSP3 share the same link (B→C). The backup LSP, assigned to the shared link, is set up along the path B→G→H→C. If a failure occurs in the protected link B→C, the PLR (router B) switches over the possible traffic, which had originally been along LSP1, LSP2 or LSP3, to the backup LSP. The MPL (router C) merges (resumes) the traffic back to its original LSPs. The 1:n protection is achieved by

using two levels of label stack to forward the packets along the backup LSP. Section 4.4 discusses how the label stack works.

Figure 4. 8 1:n Protection Model in Link Protection

Link protection assumes that the link failure is not caused by a node failure at its downstream node, and that the end routers of this link are always running under all circumstances. In fact, this assumption is not always valid, since node failure along the link is possible and, if it occurred, would cause the link protection methodology to fail.

To overcome an LSP failure due to a possible failure at a critical node, a node protection methodology is mandated for the detection and recovery of such failure.

4.3 Node Protection

This section focuses on the node protection methodology for protecting a critical node (router) along a specific primary LSP. With node protection, the backup LSP is pre-

established around the critical node (the one intended to be protected). The head-end of this backup LSP is the upstream of the concerned node, while the tail end is the next next-hop of the upstream node. Both the head-end and the tail end are located at the primary LSP.

Figure 4.8 shows an MPLS network example with eight routers (A, B, C, D, E, F, G and R). The primary LSP is pre-established along routers $A \rightarrow B \rightarrow C \rightarrow D$. Labels L_a , L_b and L_c are used to forward the packets along the primary LSP. Router C is the critical node, the one intended to be protected, along the primary LSP. Router B is upstream of the protected node and router D is the next next-hop of the upstream node. The backup LSP is set up along routers $B \rightarrow E \rightarrow F \rightarrow D$. Labels L_e , L_f and L_d are used to forward the packets along the backup LSP when a failure occurs on router C.

Figure 4. 9 Node Protection

Node protection uses the same failure detection and notification mechanism used in the link protection methodology. This was observed in Figure 4.8, where Router B works as the PLR. If the B→C link fails, router B switches the traffic over to the backup LSP (B→E→F→D). Apparently, protecting node C prevents any possible failure at its upstream B→C link. It is evident, therefore, that node protection on a specific LSP protects both the node concerned and its upstream link along this LSP.

4.4 Two Levels of Label Stack

This research imposes two levels of label stack to support the 1:n Link/Node protection along a specific LSP where the PLR records the labels expected by the MPL. The first level of label stack expected by the MPL is the label associated with the primary LSP, which is considered to be the inner label stack. The second level is associated with the backup LSP, which is considered to be the outer label stack. In the case of failure, PLR switches over to the backup LSP, uses the recorded label as the inner label and the backup LSP label as the outer label to encapsulate the packets and sends them along the backup LSP.

Figure 4.9 shows a link protection mechanism, which achieves 1:n protection. The example represents an MPLS network of eight routers (A, B, C, D, E, F, G and M). Two primary LSPs are set up: LSP1 (A→B→C→D) and LSP2 (G→B→C→M). Labels L_a , L_{b1} and L_c are used to encapsulate the packets along LSP1, while labels L_g , L_{b2} and L_m are used to encapsulate the packets along LSP2. LSP1 and LSP2 share the common link

B→C, which is the one intended to be protected. A backup LSP (LSP3: B→E→F→C) is set up to protect the link B→C. Router B works as the PLR and router C works as the MPL.

If the link B→C faces a failure, the PLR (router B) switches the traffic, either from LSP1 or LSP2, over to the backup LSP3. To differentiate the packets from the different LSPs originators, the PLR (router B) encapsulates the incoming packets using two levels of label stack. The inner label is the label recorded from the primary LSP, while the outer label is the label associated with the backup LSP. This is illustrated in Table 4.4, which shows the recorded labels in the FIB table.

After the two labels stack has been formed, the PLR forwards the packets along the backup LSP. The MPL (router C) receives the forwarded packets, removes the outer label stack and recognizes the inner label stack. Based on the recognized inner label stack, the MPL decides which LSP originates the packet and forwards the packet along this LSP. Using the two levels of label stack in the link/node protection allows the transparency of the network topology to be changed to the MPL, and achieves 1:n protection.

Figure 4. 10 1:n Protection

Table 4.4 Label recorded in FIB table

IncomingLabel	OutLabel	RecordedLabel
La	Le	L _{b1}
Lg	Le	L _{b2}

4.5 Working Example

In PLR, first, for the packets with incoming label L_a (packets along the LSP1), router B swaps the Label L_a to the recorded label L_{b1} , sets it as the inner label stack and then adds the outer label L_m , which identifies the backup LSP. For the packets with incoming label L_g , router B swaps the label L_g to recorded label L_{b2} and then adds the outer label L_m . As a consequence, the packets that originally go through LSP1 and LSP2 have the same outer label L_e and travel through the same backup LSP.

with two Levels of Label Stack

When the router C receives the packets with two levels of labels: the outer label L_c and a different inner label, router C removes the outer label since it is the egress for the

backup LSP. Router C then makes the forwarding decision based on the inner label, which is the same as the one used before the link was broken.

In summary, the steps taken by each router along the backup LSP are as follows:

- Router B performs a SWAP—PUSH operation. When router B receives the packets with label L_a , it first swaps the label L_a to label L_{b1} and sets the S bit in the label stack to 0, then adds the outer label L_m and sets the S bit in the label stack to 1.
- Routers E and F perform a Swap operation. Router E swaps the label L_e to label L_f and router F swaps the label L_f to label L_c without touching the inner label L_{b1} . As a result, when router C receives the packets, the packets have two levels of labels: outer label L_c and inner label L_{b1} .
- Router C performs a POP—SWAP operation. It receives the packets with the label stack $\{L_f, L_b\}$. It removes the outer label L_f and retrieves the inner label L_b , which is the label used for the primary LSP. Afterwards, the same treatment will be given to the packets as for the primary LSP.

Chapter 5 Implementation

Chapter 4 introduced the methodology for MPLS protection. The protection strategy relies on pre-establishing a backup LSP for specific primary LSP(s). When a primary LSP fails, the PLR detects the failure and notifies the forwarding engine correspondingly. The forwarding engine switches the packets over from the failed primary LSP to its pre-established backup LSP. By using two levels of label stack, 1:n protection, which is more flexible and cheaper for deployment, is also achieved. The proposed protection mechanism involves the development of a failure detection module, a failure notification module and a forwarding engine module. The detection module is responsible for failure detection. The failure notification module is responsible for building a notification message and delegating the message to the forwarding engine. The forwarding engine module is responsible for switching the packets over and forwarding them along the appropriate LSPs.

This chapter focuses on the MPLS protection implementation rationales, the underlying framework and the environment settings. Section 5.1 introduces the framework on which this implementation is based. Next, section 5.2 introduces the implementation setting for this research, which includes adding the designed module to the framework, and design issues with this research. Section 5.3 introduces the implementation of failure detection; section 5.4 introduces the implementation of notification; and section 5.5 introduces the architecture and state-machine of the forwarding engine.

5.1 Framework

The EION Open IP Environment [EION/CSP-00] is a portable, real-time software suite that enables new and traditional network elements to provide high performance interoperability across multiple platforms and products. It is a software solution that provides IP communications building blocks that sit between the operating system and the application/service layer of various Internet-enabled devices. Figure 5.1 elaborates the architecture of the Open IP Environment. There are four main components to the Open IP Environment: common system plane [EION/CSP-00], common control plane [EION/CSP-00], common forwarding plane [EION/FPI-00] and common management plane [EION/NNCLP-00]. Each contains a set of well-defined APIs. This simplifies the integration of the new components into the system. The functionality of each plane is as follows:

Figure 5. 1 OPEN IP System [EION/CSP-00]

Common System Plane

The Common System Plane [EION/CSP-00] is a framework for communication software development. It aids developers of high-performance communication services and applications on UNIX and VxWorks. It simplifies the development of network applications and services that use inter-process communications, event de-multiplexing, explicit dynamic linking, and concurrency. Consequently, the Common System Plane alleviates costly redevelopment by providing extensibility and employing software reuse.

It provides APIs for a wide range of operating systems, such as Linux, Unix, VxWorks, and Windows. Services include thread, fault and memory management, and inter-process communications.

Common Forwarding Plane

The Common Forwarding Engine manages [EION/Ipv4-00] the forwarding table, and receives and forwards PDUs through an egress interface that provides the best route to the packet's destination. The Software Forwarding Engine supports unicast and multicast routing, and is implemented specifically for the hardware platform and operating system on which it runs. The Software Forwarding Engine can be used as is, or as a template for creating a new forwarding engine. It provides APIs to multiple packet forwarding mechanisms, including hardware, software and hybrid platforms.

Common Control Plane

The Common Control Plane [EION/Ipv4-00] is used for the communication of protocol

data units (PDUs) and other control information between the IP stack and protocol modules, route table managers, and a Routing Policy Server. It provides plug-and-play access for all IP modules, such as modules for high-performance implementation of the OSPF routing.

Common Management Plane

The Common Management Plane [EION/NNCLP-00] comprises a set of external interfaces that allow consistent management of the Open IP Environment components, provides open standard APIs for policy-based management, command line interface and network management based on SNMP.

5.2 Implementation Settings

The MPLS protection implementation relies on the open IP system environment. The utilization of the environment is accomplished through harnessing the open IP system with the MPLS-related modules. The communication among the different MPLS protection tasks follows the Internal Message Queue scheme provided by the Open IP system. The LSPs are identified by labels, which are assigned based on the per-platform Label Space approach. Label merging is supported in this implementation in forwarding the packet along their appropriate LSPs. Data structures are provided to manipulate label information and support the different associated processes required for searching, and updating. Throughout the protection process, many messages are exchanged. The manipulation of messages obligates a distinction mechanism to assure appropriate

handling. The implementation allows the external user to configure the routers statically by providing Command Line Interfaces.

5.2.1 Open IP With MPLS

In this research, the implementation relies on and utilizes the Open IP system environment. Figure 5.2 shows how this environment utilization is accomplished. The implementation-fundamental MPLS modules are placed in the environment to support the MPLS protection mechanism. The Common Control Plane hosts the implemented MPLS control package, while the Common Forwarding Plane hosts the proposed MPLS forwarding engine. The interface module CLI, which allows the network operator to configure the routers when necessary, encapsulates the Common Management Plane.

Figure 5. 2 Open IP System with MPLS

5.2.2 Internal Message Queue

The hosting Open IP system is a multi-task environment, where every protocol module is running on its task and has its own message queue. The inter-process communication between the different tasks is achieved by en-queuing the incoming message from a requesting task into the receiving task's Message Queue. The message queuing mechanism is widely applied in the MPLS protection implementation. Message queuing is demonstrated in Figure 5.3, as follows:

1. The detection module en-queues the failure notification messages into the forwarding engine message queue.
2. The CLI en-queues the label information (inserted messages) into the forwarding engine message queue.
3. IP packets (either control packets or Ethernet packets) are viewed as messages and en-queued into the forwarding engine message queue.

Figure 5. 3 Internal Message Process

5.2.3 Label Space

Each LSP is associated with a unique label identifier. Since such labels are locally significant identifiers, LSR uses them to identify LSP(s). There are two types of label space: per interface label space and per platform label space. The per platform label space scheme is used in this implementation. Although the per platform label space contracts the maximum recognizable number of labels at an LSR, the per platform label scheme is easy to implement, and its contracting liability is manageable and reasonably sufficient.

5.2.4 Design of the FEC

In this research, the design decision of the FEC is based on the IP destination addresses carried in the incoming IP packets. In other words, packets with the same IP destination addresses (or same prefixes) are mapped to the same FEC. Designing the FEC based on the IP destination addresses simplifies implementation and is sufficient for this research scope.

5.2.5 Label Merging

Label merging refers to the ability to assign the same outgoing label to incoming packets that have different incoming labels. The MPLS protection implementation supports label merging. The label-merging scheme avoids the maintenance of a large number of labels among routers and can be used to achieve 1:n link/node protection.

Figure 5.4 illustrates a label-merging example which contains four routers (R1, R2, R3

and R4). The incoming packets from R1 carry label L1, while R2 packets carry label L2. R3 assigns the same outgoing label, L3, to the packets coming from either R1 or R2, and forwards the packets to router R4.

Figure 5. 4 Label Merging

5.2.6 Data Structures

Edge LSRs (LERs) are connected to the IP network while core LSRs are connected to the MPLS network. When the packets arrive at the LERs or LSRs, the outgoing labels and the next-hop information need to be resolved. In this research, three tables have been designed for this purpose. The tables are called Ip2Fec, NHLFE and FIB. The Ip2Fec Table is responsible for mapping the IP addresses and the FECs. The NHLFE table is responsible for mapping the FEC to the outgoing label and the next-hop IP addresses. The FIB table is responsible for mapping incoming labels to the outgoing labels and the next-hop IP addresses. The ingress LER uses the Ip2Fec table and the NHLFE table. Initially, the LER consults the Ip2Fec table to extract the FEC. However, the LER extracts the outgoing label and outgoing interface from the NHLFE table. The LSRs get

the outgoing label and outgoing interface from the FIB table.

5.2.6.1 Ip2Fec Table

The Patricia trie is used for the Ip2Fec table. Figure 5.5 shows the data structure for the Ip2Fec table.

Patricia Trie stands for Patricia Algorithm to Retrieve Information Coded In Alphabets [Morrison-68]. D. R. Morrison postulated that the Patricia trie is a more compact form of trie, which retains the same ability to search for a string. He also stated that a Patricia trie is similar to a trie, except that a node with only one child has been removed while the number inside the node represents the depth of the node which is the character position in the key represented by the node [Morrison-68]. This retrieval algorithm and data structure has been introduced to represent the IP routing table in BSD UNIX systems and many existing high-end routers [Skolwer-90].

The Patricia trie has the most efficient and best average performance [Morrison-68]. However, the complexity and operation overhead make it more suitable for a large data table such as an IP2FEC.

```

typedef struct MplsIpnetAddr {
    ipa      ip_add;    /* 32 bit address */
    ipa      ip_mask;    /* subnet mask */
} MPLSIPNA, *mpls_ipna_pt;

typedef struct mpls_iproute_entry{
    struct mpls_iproute_entry *iprt_bwd;
    struct mpls_iproute_entry *iprt_llink;
    struct mpls_iproute_entry *iprt_rlink;
    struct mpls_iproute_entry *iprt_dup;
    byte iprt_skip;      /* bits to skip */
    byte iprt_ltag;
    byte iprt_rtag;
    struct MplsIpnetAddr iprt_ipa;    /* dest */
    unsigned int fec;
} mpls_iproute_ent, *mpls_iproute_ent_pt;

```

Figure 5. 5 Ip2Fec table Data Structure

5.2.5.2 NHLFE Table

Figure 5.6 shows the data structure of the NHLFE table, which uses an AVL tree and a linked list data structure. FEC is the searching key used in the AVL tree. On each node of the tree, the entries are stored in a linked list.

D.E. Knuth stated that an AVL tree is a *balanced binary tree* where the *height* of the two subtrees (*children*) of a node differs by, at most, one. Look-up, insertion, and deletion are $O(\log n)$, where n is the number of *nodes* in the tree.[Knuth-99] An AVL tree is used in

this research since it is efficient and simple to implement.

```
struct Nhlfe {  
    int fec;  
    struct NhlfeEntry* nhlfeEntry;  
};  
struct NhlfeEntry {  
    int outgoingLabel;  
    ipa outgoingIf;  
    ipa nextHopAddress;  
    int cid;  
    bool facilityUsed;  
    int facilityLabel;  
    struct NhlfeEntry* nextNhlfeLink;  
};
```

Figure 5. 6 NHLFE table Data Structure

5.2.6.3 Data Dependency between Ip2Fec and NHLFE

Figure 5.7 shows the data dependency between the Ip2Fec and NHLFE tables.

Figure 5. 7 Data Dependency between Ip2Fec and NHLFE

5.2.6.4 FIB Table

Figure 5.8 shows the data structure of the FIB table, which uses an AVL tree and a linked list data structure. The incoming label is the searching key used in the AVL tree. On each node of the tree, the entries are stored in a linked list.


```

struct Fib {
    int incomingLabel;
    struct FibEntry *fibEntry;
};

struct FibEntry {
    int outLabel;
    ipa inInterface;
    ipa outInterface;
    ipa nextHopAddr;
    int cid;
    bool facilityUsed;
    int facilityLabel;
    struct FibEntry* nextFibEntry;
};

```

Figure 5. 8 FIB table Data Structure

5.2.7 Messages Distinction

The distinction of the incoming messages is necessary in order to decide how each message should be processed. Due to message manipulation, the messages are classified into two categories: control messages and data messages. The control messages carry label information required for LSP identification (e.g. incoming label, outgoing label, outgoing interface and next-hop IP address). They are processed by updating the forwarding tables. The data messages wrap two types of data packets: labeled data packets and unlabeled data packets. The labeled data packets are processed by MPLS forwarding functionality while the unlabeled data packets are processed as follows: if

an LSP exists, it is processed by the MPLS forwarding functionality; otherwise, it is processed by the IP functionality. Figure 5.9 shows the packet distinction procedure.

Figure 5. 9 Packet Distinction Block Diagram

5.2.8 Command Line Interfaces

The Command Line Interface encapsulates the Common Management Plane and provides the network operator with the APIs necessary to configure the routers. The implemented CLI provides two interfacing commands to set up the LSPs statically: ingress node configuration CLI and tandem node configuration CLI. The ingress CLI could be represented with the following command line format:

*Ingress lsp <lspId > <ingressId> <egressId> <outgoing label> <next Hop address>
<FEC>*

where <lspId > is the LSP ID, the<ingressId>is the ingress interface IP address,<egress ID> is the egress interface IP address, <next Hop Address> is the IP address of the next hop and the <FEC> is the FEC value for this LSP.

The tandem CLI could be represented with the following command line format:

*Tandem lsp <lspId > <ingressId> <egressId> <inLabel > <outLabel > <outInterface>
<nextHop>*

Where <lspId > is the LSP ID, the<ingressId>is the ingress interface IP address, <egress ID> is the egress interface IP address, <inLabel> is the incoming label of the LSP, <outLabel> is the outgoing label of the LSP, <outInterface> is the IP address of the outgoing interface and the <nextHop> is the next hop IP address of the LSP.

5.3 Failure Detection

Different mechanisms can be used to detect a link failure. Hello messages, which are sent out periodically by the signal protocols such as LDP [Wu-01] or RSVP-TE [Awduche-01] can be used as an indication of a link failure. SONET/SDH uses the loss of signal as an indication of link failure. Auto-Negotiation [Becker-95] uses Fast Link Pulse (FLP) signals to indicate a link failure. This research makes use of auto-negotiation and a polling mechanism to detect a link failure for Ethernet-based networks. The following

sections introduce each mechanism of the link failure detection in detail.

5.3.1 Detection by Signal Protocol

RFC 3036 defines an LDP [Wu-01] hello message: “Discovery messages provide a mechanism whereby LSRs indicate their presence in a network by sending a Hello message periodically”. “If the timer expires without the receipt of a matching Hello from the peer, LDP concludes that the peer no longer wishes to label switch using that label space for that link or that the peer has failed.” The timeout is 15 seconds, meaning that it takes 15 seconds for the LDP to discover the link failure. The failure detection by the signal protocol is too long to be acceptable.

With regard to RSVP-TE, the hello interval is configurable and can be as small as 5ms. However, there are three main disadvantages with using RSVP-TE: 1) RSVP-TE is running on the IP layer and uses the Ethernet driver to send out packets. The sending of the Hello message can't be greater than the capacity of the driver. For detection, the lower the layer it is, the faster. 2) The bottleneck of the network is neither the computer processor, nor the data transfer. If there is heavy traffic (congestion) in the network, the hello message can't be delivered in anything less than 17.5ms. The RSVP-TE will incorrectly suppose that the link is broken. 3) The hello message is a completely optional parameter in RSVP-TE; not every node in the network supports a hello message. If one of the nodes along the LSP doesn't support the hello message, we can't rely on a hello message for failure detection.

5.3.2 Detection by the Physical Layer

In SONET/SDH [Bellcore/GR-1230-95], failure detection is triggered by the loss of the signal, which can be detected in 10ms. This fast failure detection means the SONET/SDH has the ability to perform the restoring connection in 50ms. However, this methodology can only be used for SONET/SDH.

5.3.3 Auto-Negotiation in Ethernet

Ethernet Auto-Negotiation [Becker-95] is a mechanism that controls the cable when a connection is established to a network device. Auto-Negotiation provides connection interoperability between IEEE 802.3 LANs. It currently supports 10BASE-T, 10BASE-T Full Duplex, 100BASE-TX, 100BASE-TX Full Duplex, and 100BASE-T4. [Becker-95] Moreover, it detects the existing modes of the device at the other end of the wire. Auto-Negotiation uses Fast Link Pulse (FLP) signals, which are a modified version of the Normal Link Pulse (NLP) signals to verify link integrity.

FLP bursts occur at an interval of 16.8ms, with a duration of 2ms. The FLP signal is encoded as a 16-bit word and is composed of 17 to 33 link pulses (identical to the link pulses used in 10BASE-T) to determine if a link has a valid connection. Figure 5.10 shows the FLP bursts.

Figure 5. 10 FLP Burst Timing

5.3.4 Failure Detection

In this research, failure detection is the responsibility of the PLR upstream of the protected link. Link and/or downstream node failures must be detected as early as possible in order to keep the total repair time low. To do so, the failure detection task keeps polling locally for any possible failure. Figure 5.11 illustrates the failure detection mechanism.

The Ethernet interface port, which is attached to the protected link, is periodically checked by this detection task. The periodic validation of the Ethernet interface status is expressed by a TRUE/FALSE Boolean value. A TRUE value implies that the link is still functioning, while a FALSE value implies that the link has failed. The FALSE validation result triggers the failure notification process.

Figure 5. 11 Link Failure Detection Timing

5.4 Failure Notification

Failure notification is manipulated at the PLR of the primary LSP, which travels through the protected link. Failure notification is accomplished by two consecutive steps: the construction of a failure notification message and the delegation of the notification message to the forwarding engine message queue.

The construction of the notification message requires that the message type and message contents be assigned. Apparently, the message type is set to the value `FAILURE_NOTIFICATION`. The message content is identified by the IP address of the Ethernet interface attached to the failed link. Once the notification message has been constructed, the notification module delivers it locally to the forwarding engine as a triggering step for the switchover process.

5.5 Forwarding Engine

Figure 5.12 shows the proposed MPLS forwarding Engine architecture. This architecture contains the IP module, DeviceDriver module, packetProcessor, MplsComLspManager module and the MplsComForwarder. In this architecture, the IP module and DeviceDriver are external modules to the forwarding engine. The IP module sends control messages that contain the labels' information to the PacketProcessor. The packetProcessor performs differently depending on the control packets or the data packets. The PacketProcessor delegates the control messages to the MplsComLspManager module. The MplsComLspManager module decodes the received messages and updates the forwarding tables. The DeviceDriver sends the data packets to the PacketProcessor. The PacketProcessor delegates the data packets to the MplsComForwarder. The MplsComForwarder sends the packets out along the LSP.

Figure 5. 12 MPLS Forwarding Engine Architecture

5.5.1 State Machine

Figure 5.13 shows the state machine for the MPLS Forwarding Engine. The state machine contains three major states: the message waiting state, packet processing state and packet forwarding state. A *system-start* event causes a transition from the Idle state to the Message Waiting state (init state). The incoming message event to the forwarding engine triggers the transition from the Message Waiting state to the Packet Processing state. After the message has been handled properly, a *Process OK* event is issued. This event causes the transition from the Packet Processing state to the Packet Forwarding state. If the handling of the message fails, a Process Failed event is issued, which causes the transition to the Discarding state. The end of the packet forwarding process is accompanied by the generation of a send-out event, which causes the transition from the Packet Forwarding state to the Message Waiting state (init state), an initiate state which waits for incoming packets.

Figure 5. 13 MPLS Forwarding Engine State Machine

5.5.2 Switchover

When the forwarding engine receives a failure notification message, it uses the switchover algorithm depicted in Figure 5.12 to switch from the failed primary LSP to its backup LSP. The algorithm handles the switchover in a sequence of successive processes. Initially, the forwarding engine extracts the interface IP address carried by the failure notification message. The forwarding engine uses the extracted IP address to recognize the different LSPs traveling through this interface attached to the failed link. For each identified LSP, the forwarding engine checks the status of its hosting PLR node regarding this LSP. If the node is an ingress node, the NHLFE table is modified by replacing the primary LSP label with its backup LSP label. If the node is a tandem node, then replacing the primary LSP label with its backup LSP label occurs at the FIB table instead.

```

{
  Get the interface IP address from notification message, Ifa
  For ( every entry in the table)
  {
    if (outgoing interface == Ifa )
    {
      in NHLFE table
      {
        if(backup LSP exists)
        change to backup LSP
      }
      in FIB table
      {
        if(backup LSP exists)
        change to backup LSP
      }
    }
  }
  Else
    continue
}
}

```

Figure 5. 14 Forwarding Engine Switchover upon Failure

5.6 Sequence Diagrams

This section introduces the forwarding engine class diagram and the sequence diagrams of the associated collaborators, in order to clarify the aspects of the design structure and dynamics of the forwarding engine. The sequence diagrams discuss the dynamics of the LSP setup and Switchover process. The dynamics of the packet processing at the different nodes (ingress, PLR and MPL) are elaborated further.

5.6.1 Forwarding Engine Structure

Figure 5.15 shows the UML class diagram of the Forwarding Engine. The forwarding engine has three core modules for managing the forwarding engine services. PacketProcessor, MplsComLspManager and MplsComForward. PacketProcessor have a 1:n relationship with the Ethernet device interface (IfDev). It is responsible for manipulating all received packets from the Ethernet devices or from the applications. MplsComLspManager has three table data structures (MplsComIp2fec, MplsComNhlfe and MplsComForwardingInfoBase), which are required to store the LSPs' routing information. MplsComLspManager maintains and updates those tables. MplsComForward contains MplsComEncoder to pre-pend the MPLS label stack and return the encapsulated packets to the PacketProcessor. PacketProcessor has an MplsComLspManager and an MplsComForward, where processed packets are directed either to the MplsComLspManager, in the case of the control packet, or to the MplsComForward in the case of data packets.

Figure 5. 15 Forwarding Engine UML Class Diagram

5.6.1 LSP Setup and Switch-over Dynamics

As shown in Figure 5.12, the LSP Manager module is the glue between the Control Plane and the Data Plane. It is responsible for setting up the node according to the LSP's information and filling up the associated forwarding tables depending on the status of the node concerned. Figures 5.16 and 5.17 show the dynamics of LSP setup at the Ingress LER and Tandem LSR respectively. Figure 5.18 shows the switchover dynamic if failure occurs.

Figure 5. 16 Ingress LSP Setup Sequence Diagram

Figure 5. 17 LSR LSP Setup Sequence Diagram

Figure 5. 18 Switchover Sequence Diagram

5.6.2 Packet Processing Dynamics

After the switch-over from the primary LSP to the backup LSP, the forwarding engine forwards the packets through the backup LSP. 1:1 protection uses one level of the label stack strategy to forward the data packets, while 1:n protection uses two levels of label stacks. The dynamics of the packet forwarding in the Ingress, PLR and MPL are illustrated in Figures 5.19, 5.20 and 5.21, respectively.

Figure 5. 19 Sequence Diagram for the Ingress Packet Process

Figure 5. 20 Sequence Diagram for PLR

Figure 5. 21 Sequence Diagram for MPL

Chapter 6 Performance Analysis

Chapter 4 discussed the methodology for the design and implementation of the MPLS protection. The chapter covers the implementation rationales when a forwarding engine is required. The protection strategy relies on pre-establishing a backup LSP for specific primary LSP(s). When a primary LSP fails, the PLR detects the failure and notifies the forwarding engine accordingly. The forwarding engine switches the packets over from the failed primary LSP to its pre-established backup LSP. Chapter 5 presented the MPLS protection implementation rationales, the underlying framework and the environment settings.

This chapter provides a performance analysis of the proposed MPLS protection design and implementation. The analysis evaluates the reliability, efficiency and credibility of the implementation. The evaluation includes link failure detection, failure notification and switchover to the backup LSP(s). The evaluation is based on successive experiments in a Linux environment providing fast Ethernet network adaptors. All experiments are processed using a recovery model proposed by IETF RFC 3469. The model considers timing criteria, which measures the elapsed time along the detection recovery process of a primary LSP failure.

Figure 6.1 describes the recovery model, which distinguishes three time zones. The first timing zone is failure detection time ($T_{\text{detection}}$). It ranges from the occurrence of a link failure to the time of the failure detected. The second timing zone is the notification

time ($T_{\text{notification}}$). It ranges from the link failure notification to the beginning of the switchover (from the primary LSP to its backup LSP). The third timing zone is the switchover time ($T_{\text{switchover}}$). This ranges from the beginning of the switchover over to the end of the recovery process.

Figure 6. 1 MPLS Recovery Cycle Model

At the conclusion of this chapter, the implemented MPLS protection will be evaluated for a specific chosen link in a proposed environment. By assuming the Normal distribution scheme and choosing the level of significance as 0.95, the experiments achieve an average overall recovery-processing time of 29.542 ms. with a confidence interval of 28.94 ms ~30.10ms. The experiments are repeated for a large number of times. The resulting data is analyzed for the different sample sizes. The analysis shows that the confidence interval is very close for the different sample sizes. In this research, we report the collected result data for a given sample size of 15, which has half the deviation length

of 0.6 with a precision of 2.2%. A precision of 2.2% is considered to be acceptable for this research.

Compared with other methodologies, which use different layers other than the MPLS layer, this experimental result definitely performs satisfactorily. Based on the evaluation results of the methodology, Chapter 6 concludes that the proposed failure detection methodology is fast and reliable.

6.1 Experiment Plan

This section presents the experiment plan used to analyze the performance of the proposed mechanism and implementation. The experiment plan includes scenarios such as whether the ingress node or tandem node is acting as a PLR node.

6.1.1 Ingress Node as the PLR

This experiment focuses on a situation where the PLR is the ingress node. The experiment includes two separate tests: one test is a link failure experiment where the link along the primary LSP fails; another experiment is a node failure experiment, where the node along the primary LSP fails. This section also describes the experiment plan for 1:n protection.

Link Failure Experiment

A simple MPLS network is configured for the purposes of this test. It assumes an

Ethernet-based network of 4 routers (R1, R2, R3 and R4). Figure 6.2 illustrates this setup environment, where a primary LSP is set up to include routers R1, R2 and R4 (primary LSP $R1 \rightarrow R2 \rightarrow R4$). The testing assumption includes a segment of the primary LSP as a protected link. It links from Ethernet interface eth1 of router R1 to the Ethernet interface eth3 of router R2. The pre-established backup LSP of the protected link includes routers R1, R3 and R2 consecutively (backup LSP $R1 \rightarrow R3 \rightarrow R2$). A protected link failure is originated manually by unplugging the Ethernet cable between routers R1 and R2. The protected link failure breaks the primary LSP along the routers R2 and R3. Accordingly, the primary LSP traffic is switched over to the pre-established backup LSP by the forwarding engine at R1.

Figure 6. 2 Ingress Node (Link Failure) Experiment

Node Failure Experiment

A similar Ethernet-based MPLS network with 4 routers is configured for the purposes of this test. Figure 6.3 illustrates this setup environment, where a primary LSP is set up to

include routers R1, R2 and R4 (primary LSP $R1 \rightarrow R2 \rightarrow R4$). The testing assumes that the router R2 is the protected node. The pre-established backup LSP for the protected node includes routers R1, R3 and R4 consecutively (backup LSP $R1 \rightarrow R3 \rightarrow R4$). A protected node failure is originated manually by turning off the router R2. The protected node failure breaks the primary LSP. Accordingly, the primary LSP traffic is switched over to the pre-established backup LSP by the forwarding engine at R1.

Figure 6. 3 Ingress Node (Node Failure) Experiment

6.1.2 Tandem Node as the PLR

This experiment focuses on the situation where the PLR is a tandem node. Two separate experiments were carried out: one experiment to test link failure experiment, another to test node failure.

Link Failure Experiment

A simple MPLS network was configured for purposes of this test. It assumes an Ethernet-based network of 5 routers (R1, R2, R3, R4 and R5). Figure 6.4 illustrates this setup environment, where a primary LSP is set up to include routers R1, R2, R4 and R5 (primary LSP $R1 \rightarrow R2 \rightarrow R4 \rightarrow R5$). The testing assumes the link between R2 and R4 is a protected link, which links from the eth2 of router R2 to the eth1 of router R4. The pre-established backup LSP of the protected link includes routers R2, R3 and R4 consecutively (backup LSP $R2 \rightarrow R3 \rightarrow R4$). A protected link failure is originated manually by unplugging the Ethernet cable between routers R2 and R4.

Figure 6. 4 Tandem Node (PLR) Link Failure Experiment

Node Failure Experiment

A similar Ethernet-based MPLS network with 5 routers (as the link failure experiment) was configured for the purposes of this test. Figure 6.5 illustrates this setup environment, where a primary LSP is set up to include routers R1, R2, R4 and R5 (primary LSP $R1 \rightarrow R2 \rightarrow R4 \rightarrow R5$). The testing assumes the node router R4 to be a protected node. The pre-established backup LSP of the protected node includes routers R2, R3 and R5 consecutively (backup LSP $R2 \rightarrow R3 \rightarrow R5$). A protected node failure was originated manually by turning off the router R2.

Figure 6. 5 Tandem Node (PLR) Node Failure Experiment

6.1.3 1:n Protection

This experiment is performed to test the functionality of the 1:n protection. A similar testing environment was set up for this purpose. Figure 6.6 illustrates the setup where three protected LSPs are set up from router R1 to router R5 along $R1 \rightarrow R2 \rightarrow R4 \rightarrow R5$

(because of the environment limitations, the three LSPs travel along the same links). The testing assumes that the link between R2 and R4 is a protected link, which links from the eth2 of router R2 to the eth1 of router R4. The pre-established backup LSP of the protected link includes routers R2, R3 and R4 consecutively (backup LSP R2→R3→R4).

Figure 6.6 1:n Protection Experiment

6.1.4 Hardware Used for the Experiments

The experiments were performed on a testbed environment of five routers running on Linux Operating systems. The Linux machine hosting the PLR node is an Intel (R) Celeron (R) CPU with a CPU speed 1715.189MHZ, while the vendor-id is Genuine Intel and the cache memory size is 8KB. The Fast Ethernet network interface card (NICs)

installed is a RealTek RTL-8139. All links are full-duplex point-to-point 10/100BaseTX Ethernet cables.

6.2 Data Analysis Formula

In this research, Normal Distribution [Patel-82] was used to analyze the result data. There were two reasons for selecting normal distribution for the data analysis in this research:

“1) One reason normal distribution is important is that many psychological and educational variables are distributed approximately normally. 2) A second reason is that it is easy for mathematical statisticians to work with. This means that many kinds of statistical tests can be derived for normal distributions. Almost all statistical tests discussed in this text assume normal distributions. Fortunately, these tests work very well even if the distribution is only approximately normally distributed. Some tests work well even with very wide deviations from normality.” [Patel-82]

Assume $X_1, X_2, \dots, X_N$ is a sample of size n , then the formula for the *mean* of the sample is:

$$\bar{X} = \sum X/n \dots \dots \dots F1$$

and formula for *standard division* :

$$S = \sqrt{\frac{\sum(x - \bar{x})^2}{n - 1}} \dots\dots\dots F2$$

In a case where the sample n is less than 30 or n<30, the small sample procedure [Statistic-03] is used to develop a confidence interval for the mean of a population. The formula for confidence intervals for the population mean based on small samples is:

$$\bar{x} \pm t_{\alpha/2} \cdot (S / \sqrt{n}) \dots\dots\dots F3$$

In this formula $\bar{x}$ is the mean of the sample which can be obtained from formula F1, while $t_{\alpha/2}$ is the interval coefficient providing an area of $\alpha/2$ in the upper tail of a t distribution with n-1 degrees of freedom, which can be found from a t distribution table. S is the standard deviation of the sample, which can be calculated by formula F2, and n is the sample size.

6.3 Failure Detection Time

As per the recovery model definition, the link failure detection time is measured from the time the link is unplugged to the time when the PLR detects the link failure. In the link failure detection test, the failure is originated by manually unplugging the Ethernet cable, which forms the assumed protected link. In the node failure experiment, however, the failure is originated by manually turning off the downstream router, which forms the assumed protected node. Since the implementation assigns a polling thread to detect the

failure of the protected link, the thread issues a timestamp to record the time of failure detection.

In a further elaboration, Figure 6.2 demonstrates a case where a protected link is unplugged (the link between a router R1 and another router R2). The pre-established backup LSP is designed to be the route R1->R3->R2 (the LSP from Router R1 to Router R2 via router R3). Once the protected link is unplugged or the protected node is turned off, the polling thread located at R1 senses the failure and consequently originates a detection timestamp. The time difference between the detection timestamp and the thread polling start time is the failure detection time $T_{\text{detection}}$. The first stage of the experiment ends by recording the failure detection time. The timestamps are recorded by using the system call `gettimeofday()`. The experiment is also repeated for the direct connection between two routers, the result is very close. This research assumes that the failure detection time for both cases is the same.

Since the testing result for the link failure detection time and node failure detection time is very close, this research assumes that the failure detection time for both cases is the same. Figure 6.7 emphasizes the distribution of the failure detection time recorded from the experimental results. Normal Distribution is assumed where the chosen level of significance is 0.95. The graph shows that the time varies between 25 to 30 ms. The average for the $T_{\text{detection}}$ is 27.02ms, with a confidence interval of 26.5ms~ 27.542ms.

Figure 6. 7 Distribution of the Failure Detection Time

6.4 Failure Notification Time

According to the recovery model definition, the failure notification time is measured from the time that the failure detection thread signals the notification, to the time when the forwarding engine receives that notification.

The experiment distinguishes the detection time by a timestamp at Router R1 (by a local failure detection thread). The failure detection thread, located at R1, builds a notification message and delegates it to the forwarding engine. The forwarding engine receives the

notification message, and issues another timestamp to record the notification message arrival time. The time difference between the failure detection timestamp and the notification arrival timestamp is the notification time $T_{\text{notification}}$.

The entire experiment (including detecting all timing zones throughout the recovery process) was repeated 15 times. Figure 6.8 emphasizes the distribution of the failure notification times recorded from the experimental results. Normal Distribution is assumed, where the chosen level of significance is 0.95. The formula introduced in Section 6.2 is used to analyze the resulting data. The graph shows that the time varies between 1.8 and 3.2 ms. The average for $T_{\text{notification}}$ is 2.5ms, where the resultant confidence interval is 2.376ms ~ 2.6244.

Figure 6. 8 Distribution of the Failure Notification Time

6.5 Switchover Time

According to the recovery model definition, the switchover time is the time taken to switch the backup LSP to the primary one. The forwarding engine receives the notification message and begins updating the forwarding information table. At the end of the updating process, the time is recorded by generating an end-of-updating-process timestamp. The switchover time is measured from the arrival time of the notification message to the end-of-updating-process timestamp.

The entire experiment (including detecting all timing zones throughout the recovery process) was repeated 15 times. Figure 6.9 emphasizes the distribution of the switchover times recorded from the experimental results. Normal Distribution is assumed, where the chosen level of significance is 0.95. The formula introduced in Section 6.2 is used to analyze the resulting data. The graph shows that the time varies between 19 and 26 us. The average for the $T_{\text{notification}}$ is 21.467us, where the resulting confidence interval is 20.215us ~ 22.72us.

Figure 6. 9 Distribution of the Switchover Time

Table 6.2 illustrates the switchover time for the different numbers LSPs. The table shows that the switchover time is slightly larger for complicated topology with a large number of LSPs. However, of the total recovery time, it is still the least time-consuming procedure (less than 1ms). In the total recovery time analysis, this research focuses on a situation in which small numbers of LSPs are set up.

Table 6.1 Switchover Time for Various LSP#

Number of LSPs	Switchover Time (us)
1	21.47
10	27.25
100	112.71
200	180.42
300	211.13
400	243.17
500	282.71
600	347.29
700	411.26
800	458.63

6.6 Recovery Time

In the definition of the recovery model, the total recovery time consists of three timing zones: failure detection time ($T_{\text{detection}}$), failure notification time ($T_{\text{notification}}$) and switchover time ($T_{\text{switchover}}$). The observations after testing for each time zone are plotted in Figures 6.7, 6.8 and 6.9. By augmenting the timing zone graphs, the overall recovery time is graphically represented in Figure 6.10. The augmented graph shows an average of

T_{recovery} as 29.542. By assuming Normal Distribution and choosing the level of significance as 0.95, the Confidence Interval for the T_{recovery} is 28.94ms ~ 30.14ms.

Figure 6. 10 Distribution of the Total Recovery Time

Figure 6.11 shows the distribution of the time, where $T_{\text{detection}}$, $T_{\text{notification}}$ and $T_{\text{switchover}}$ and T_{recovery} are depicted.

Figure 6. 11 Recovery Time Weight Distribution

Apparently, the detection time is the major time consumer of the overall recovery time. This is concluded from Table 6.2, which shows the weight of the $T_{\text{detection}}$, $T_{\text{notification}}$ and $T_{\text{switchover}}$ in the total T_{recovery} . $T_{\text{detection}}$ represents 91.47% of the T_{recovery} . Any enhancement of the detection time will affect the performance of the recovery process.

Table 6.2 Recovery Time Weight Distributions

	Average time (ms)	Weight
Failure detection time	27.02ms	91.47%
Notification time	2.5ms	8.45%
Switchover time	0.0215	0.08%
Total recovery time	29.54ms	100%

6.7 Summary

This chapter provided a performance analysis of the proposed MPLS protection design and implementation. The analysis evaluated the reliability, efficiency and credibility of the implementation, in terms of processing time intervals and total failure recovery time. The evaluation includes link failure detection, failure notification and switching over to backup LSP(s). For testing purposes, a simple network environment was set up, equipped with fast Ethernet adaptors. The evaluation of the performance was based on successive experiments, in which testing data records were processed using a recovery model. The model considers timing criteria, which measures the elapsed time in the detection recovery process of a primary LSP failure. Normal Distribution was used to analyze the results of the experiment. The analysis shows that the entire failure recovery process requires an average of 29.54 ms, which is a satisfactory recovery time, compared with SONET's 50ms recovery time [Demmester-99].

Chapter 7 Conclusions and Future Work

This chapter summarizes the work that has been done in this research and draws a conclusion based on the experiments. Section 7.2 identifies the directions for future work.

7.1 Conclusions

Link failure and node failure are common causes of service failures in a network. Protection against link and node failures can be performed at different layers in a network. Protection at the physical layer is fast, but requires dedicated hardware. Conversely, protection at the IP layer does not require specific hardware, but it relies on the routing protocol to re-compute and update the routing tables. Dynamically re-computing the routing table is a relatively slow process, ranging from several seconds to minutes [Huang-02]). The MPLS protection designed and implemented in this research is between the IP layer and the MAC layer. It supports the recovery mechanisms that provide a trade-off between recovery speed and deployment cost.

This research proposes a strategy in which both a primary LSP and its backup LSP are pre-established, and provides an implementation of an MPLS forwarding engine to support link/node protection. In case of a link/node failure on a primary LSP, the MPLS forwarding engine can switch the traffic over to the backup LSP without any additional hardware or network layer protocols being required. Moreover, the recovery does not

require that the routing table be re-computed. The forwarding engine plays a primary role in the recovery process, since it takes over from the moment of failure notification until the end of the switchover process. Nevertheless, the forwarding engine is supported by specific data structures to facilitate the execution of the different tasks required to perform the switchover efficiently.

The implemented MPLS protection was evaluated by setting up a simple Ethernet network environment. The experiments were repeated a number of times. By assuming the Normal Distribution scheme and choosing the level of significance as .95, the experiments achieved an average overall recovery-processing time of 29.542 ms. with a confidence interval of 0.602 ms. Compared with other methodologies, which use different layers other than the MPLS, this research provides an agile scheme, with fast and reliable results.

7.2 Future Work

This research could be extended in several directions. First, Figure 6.11 shows that failure detection is the most time-consuming procedure of the total recovery process, consuming more than 90% of the total recovery time. Any improvement on failure detection would greatly improve performance. This could be a direction for future work. For example, instead of using the polling mechanism proposed in this research, we can use the other, more efficient failure detection mechanism for Ethernet-based networks.

The performance testing was carried out on a small network. It could be expanded to a larger and more complex network. For larger networks, we would also need to study the related data structures and algorithms to improve the efficiency of the MPLS forwarding engine.

References

[Alwayn-02] Alwayn, V., "Advanced MPLS Design and Implementation", Cisco Press, 2002

[Andersson-01] Andersson, L., Doolan, P., Feldman, N., Fredette, A., Thomas, B., "Label Distribution Protocol Specification", RFC 3036, January 2001

[Assi-01] Assi, C., Ye, Y., Shami, A., Dixit, S., Habib, I., Ali, A., "On the merit of IP/MPLS protection/restoration in IP over WDM networks", Global Telecommunications Conference, 2001. GLOBECOM '01. IEEE , Volume: 1 , Nov. 2001 , Page(s): 65 -69 vol.1

[Awduche-01] Awduche, D., Berger, L., Gan, D., Li, T., "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, 2001

[Awduche-99] Awduche, D., Malcolm, J., Agogbua, J., O'Dell, M., J. McManus, "Requirements for Traffic Engineering Over MPLS", RFC 2702, September 1999

[Ayandeh-02] Ayandeh, S., "Convergence of Protection and Restoration in Telecommunication Networks", Photonic Network Communications, 4:3/4, page(s) 237-250, 2002

[Badran-01] Badran, F., "Service provider networking infrastructures with MPLS", Computers and Communications, 2001. , 2001 Page(s): 312 –318

[Becker-95] Becker, D., NWay (TM) Auto-Negotiation, 1995

Last visited Sept. 2003, at URL:

<http://www.scyld.com/expert/NWay.html#1.0>

[Bellcore/GR-1230-95] Bellcore, GR-1230-Core, SONET Bidirectional Line-Switched Ring Equipment Generic Criteria, Issue 2, November 1995.

[Bellcore/GR-1400-95] Bellcore, GR-1400-Core, SONET Dual-Fed Unidirectional Path Switched Ring (UPSR) Equipment Generic Criteria, Issue 1, Revision 1, October 1995

[Bellcore-90] Bellcore, SR-NWT-001756, Automatic Protection Switching for SONET, Issue 1, October 1990

[Black-01] Black, U., MPLS and Label Switching Networks, Prentice Hall PTR, ISBN 0-13-015832-2, 2001

[Bosco-03] Bosco, A., Manconi, M., Ubaldi, F., "Edge Distributed Admission Control in MPLS Networks", IEEE communications, Vol. 7 No.2 Feb. 2003

[Chen-98] Chen, K., "Network Admission Control using Inductive Machine Learning",

1998, Last visited Sept. 2003 at URL:

<http://www.cis.udel.edu/~kchen/projects/admission/admission.html>

[Chen-03] Chen, N., "Defining and Delivering Protection for Metro Ethernet Networks",

Last visited Sept. 2003 at URL

<http://www.supercommnews.com/wednesday/expers/metro.cfm>

[Conte-02] Conte, G., Listanti, M., Settembre, M., Sabella, R., "Strategy for protection and restoration of optical paths in WDM backbone networks for next-generation Internet infrastructures", Lightwave Technology, Volume: 20 Issue: 8 , Aug. 2002 Page(s): 1264–1276

[Davie-00] Davie, B., Rekhter, Y., "MPLS Technology and Applications", Morgan Kaufmann Publishers, 2000

[Demmester-99] Demmester, P. et al., "Resiliency in multiplayer networks", IEEE Communications Magazine, vol. 37, no.8, 1999, pp.70-76

[EION/Ipv4-00] Eion Inc., "Ipv4_OpenIP2_1_Dev_Doc.pdf", Internet Protocol Package, Developer's Guide, Open IP Environment 2.0.2, 2000

[EION/CSP-00] Eion Inc., "CSP-Ace_OpenIP2_1_Dev_Doc.pdf", Common System Plane, Developer's Guide, Open IP Environment 2.0.2, 2000

[EION/FPI-00] Eion Inc., “FPI_OpenIP2_1_Dev_Doc.pdf”, Common Forwarding Plane, Developer’s Guide, Open IP Environment 2.0.2, 2000

[EION/NNCLP-00] Eion Inc., “NNCLP_OpenIP2_1_Cmd_Ref.pdf”, Open IP Environment, Command Reference Manual, ND.CSM.CLP.0001, 2000

[Gerald-99] Gerald, A., Jamoussi, B., Lee, Y., “QoS Resource Management in MPLS-Based Networks”, <draft-ash-qos-routing-00.txt> October, 1999

[Gonnet-84] Gonnet, G., Handbook of Algorithms and Data Structures, Addison-Wesley, International Computer Series, pp 109, 1984.

[Ho-01] Ho, H., Mouftah, T., “SLSP: a new path protection scheme for the optical Internet”, Optical Fiber Communication Conference and Exhibit, 2001. OFC 2001, Volume: 2 , 17-22 March 2001, Page(s): TuO1-1 -TuO1-3 vol.2

[Huang-02] Huang,C. , Sharma, V., Owens, K., Makam, S, “Building reliable MPLS networks using a path protection mechanism”, Communications Magazine, IEEE, Volume: 40 Issue: 3, March 2002, Page(s): 156 –162

[Hundessa-01] Hundessa, L., Pascual, D., “Fast rerouting mechanism for a protected label switched path”, Computer Communications and Networks, 2001. Proceedings. Tenth International Conference, Oct. 2001, Page(s): 527 –530

[JUNOS-03] JUNOS Internet Software Configuration Guide, 2003

Last visited Sept. 2003, at URL:

<https://www.juniper.net/techpubs/software/junos/junos57/swconfig57-mpls-apps/html/rsvp-config7.html>

[Knuth-99] Knuth, E., “The Art of Computer Programming”, Addison-Wesley, Vol. III, 6.2.3, 1999

[Kuhn-97] Kuhn, R., “Sources of failure in the public switched telephone network”, IEEE Computer, 30(4), April 1997.

[Lawrence-01] Lawrence, J. “Designing multiprotocol label switching networks”, IEEE Communications Magazine, Volume: 39 Issue: 7 , Jul 2001 ,Page(s): 134 –142

[Morrison-68] Morrison, R., “PATRICIA - Practical Algorithm to Retrieve Information Coded in Alphanumeric”, ACM, 15(4) pp514-534, Oct 1968

[Okamoto-02] Okamoto, S., Shimano, K., Sahara, A., “Development of the photonic MPLS router to integrate optical and IP networks”, High Performance Switching and Routing, 2002. Merging Optical and IP Technologies, May 2002, Page(s): 24 –28

[Osborne-02] Osborne, O., Simha, A., “Traffic Engineering with MPLS”, Cisco Press, July 2002, ISBN: 1-58705-031-5

[Park-01] Park, J., "validation of the detailed design of the label distribution protocol for the multiprotocol label switching system", Global Telecommunications Conference, 2001. GLOBECOM '01. IEEE, Volume: 1, Nov. 2001, Page(s): 17 -24 vol.1

[Patel-82] Patel, K., Read, B, Handbook of the Normal Distribution, New York: Dekker, 1982

[Redford-00] Redford, R., "Enabling Business IP Services with Multiprotocol Label Switching," Cisco White Paper, July 2000

[Rosen-01] Rosen, E., Viswanathan, A. and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, January 2001

[Sklower-90] Sklower, K., "A Tree-Based Routing Table for Berkeley Unix", Technical Report. Berkeley, California: University of California, 1990

[Spurgeon-00] Spurgeon, C. "Ethernet: The Definitive Guide", Publisher: O'Reilly and Associates, 2000

[Statistic-03] Statistic online tutorial, University of Baltimore, last visited Sept. 2003, at URL: http://www.ubalt.edu/arc/math_resources/stat_confide.html

[Viswanathan-98] Viswanathan, A., Feldman, N., Wang, Z., Callon, R, "Evolution of multiprotocol label switching", IEEE Communications Magazine, Volume: 36 Issue: 5,

May 1998, Page(s): 165 -173

[Wu-01] Wu, J., Montuno, Y., Mouftah, T., Wang, G., Dasylva, C., “Improving the reliability of the label distribution protocol”, Local Computer Networks, 2001. Proceedings. LCN 2001. 26th Annual IEEE Conference, Nov. 2001 Page(s): 236 –242

[Yoon-01] Yoon, S.; Lee, H., Choi, D., Kim, Y., Lee, G., Lee, M., “An efficient recovery mechanism for MPLS-based protection LSP”, ATM (ICATM 2001) and High Speed Intelligent Internet Symposium, 2001. Joint 4th IEEE International Conference, April 2001, Page(s): 75 –79

[Zheng-03] Zheng, Q., Mohan, G., “Protection approaches for dynamic traffic in IP/MPLS-over-WDM networks”, Communications Magazine, IEEE, Volume: 41 Issue: 5, May 2003, Page(s): S24 -S29