ENHANCED FAST REROUTING MECHANISMS FOR PROTECTED TRAFFIC IN MPLS NETWORKS

Lemma Hundessa Gonfa

UPC. Universitat Politècnica de Catalunya

Barcelona (Spain). February, 2003

Thesis Advisor:

Prof. Jordi Domingo-Pascual

A THESIS SUBMITTED IN FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE
Doctor en Informàtica

ENHANCED FAST REROUTING MECHANISMS FOR PROTECTED TRAFFIC IN MPLS NETWORKS

Lemma Hundessa Gonfa

Thesis Advisor:

Prof. Jordi Domingo-Pascual

To my wife Dr. Truwork whose patience, love, support and encouragement enabled me to complete this thesis and was of great help in difficult times. To my parents for their interest and encouragement of my academic success since a very early age. Finally, to great Samson Gobena and Begashaw, in memory.

ABSTRACT

Multiprotocol Label Switching (MPLS) fuses the intelligence of routing with the performance of switching and provides significant benefits to networks with a pure IP architecture as well as those with IP and ATM or a mix of other Layer 2 technologies. MPLS technology is key to scalable virtual private networks (VPNs) and end-to-end quality of service (QoS), enabling efficient utilization of existing networks to meet future growth. The technology also helps to deliver highly scalable, differentiated end-to-end IP services with simpler configuration, management, and provisioning for both Internet providers and end-users. However, MPLS is a connection-oriented architecture. In case of failure MPLS first has to establish a new label switched path (LSP) and then forward the packets to the newly established LSP. For this reason MPLS has a slow restoration response to a link or node failure on the LSP.

The thesis provides a description of MPLS-based architecture as a preferred technology for integrating ATM and IP technologies, followed by a discussion of the motivation for the fast and reliable restoration mechanism in an MPLS network. In this thesis first we address the fast rerouting mechanisms for MPLS networks and then we focus on the problem of packet loss, packet reordering and packet delay for protected LSP in MPLS-based network for a single node/link failure. In order to deliver true service assurance for guaranteed traffic on a protected LSP we use the fast rerouting mechanism with a preplanned alternative LSP. We propose enhancements to current proposals described in extant literature. Our fast rerouting mechanism avoids packet disorder and significantly reduces packet delay during the restoration period.

An extension of the Fast Rerouting proposal, called Reliable and Fast Rerouting (RFR), provides some preventive actions for the protected LSP against packet loss during a failure. RFR maintains the same advantages of Fast Rerouting while elimi-

nating packet losses, including those packet losses due to link or node failure (circulating on the failed links), which were considered to be "inevitable" up to now.

For the purpose of validating and evaluating the behavior of these proposals a simulation tool was developed. It is based on the NS, a well-known network simulator that is being used extensively in research work. An extension featuring the basic functionality of MPLS (MNS) is also available for the NS, and this is the basis of the developed simulation tool.

Simulation results allow the comparison of Fast Rerouting and RFR with previous rerouting proposals.

In addition to this we propose a mechanism for multiple failure recovery in an LSP. This proposal combines the path protection, segment protection and local repair methods. In addition to the multiple link/node failure protection, the multiple fault tolerance proposal provides a significant reduction of delay that the rerouted traffic can experience after a link failure, because the repair action is taken close to the point of failure.

Then we proceed to address an inherent problem of the preplanned alternative LSP. As alternative LSPs are established together with the protected LSP it may happen that the alternative is not the optimal LSP at the time the failure occurs. To overcome this undesired behavior, we propose the Optimal and Guaranteed Alternative Path (OGAP). The proposal uses a hybrid of fast-rerouting and a dynamic approach to establish the optimal alternative LSP while rerouting the affected traffic using the preplanned alternative LSP. This hybrid approach provides the best of the fast rerouting and the dynamic approaches.

At the same time we observed that the protection path becomes in fact unprotected from additional failures after the traffic is rerouted onto it. To address this we propose a guarantee mechanism for protection of the new protected LSP carrying the affected

traffic, by establishing an alternative LSP for the rerouted traffic after a failure, avoiding the vulnerability problem for the protected traffic.

Finally, we present a further optimization mechanism, adaptive LSP, to enhance the existing traffic engineering for Quality of Services (QoS) provision and improve network resource utilization. The adaptive LSP proposal allows more flexibility in network resource allocation and utilization by adapting the LSP to variations in all network loads, resulting in an enhancement of existing MPLS traffic engineering.

The remainder of this thesis is structured as follows. In Chapter 1, we present the explanation of general MPLS concepts, definition and architecture. In Chapter 2, we present the state of art, the problem of rerouting and the different mechanisms to protect networks from link failures. These techniques range from central to distributed, from simple dynamic *local* to preestablished *global* repair mechanisms implemented from the physical layer to the IP layer. In Chapter 3, we propose the enhanced fast rerouting mechanism that reduces the additional packet delay and avoids packet reordering during the restoration period on a single link failure in MPLS networks. In Chapter 4, we present an extension mechanism of fast rerouting called the Reliable and Fast Rerouting (RFR) restoration mechanism. This prevents all packet losses during the restoration period while maintaining the previous advantages by using an additional local buffer in each LSR. In this chapter we present a mathematical model for validation of the recovery time and the buffer size needed for the implementation of the proposed mechanism. In Chapter 5, we show the application of the proposed mechanism (RFR) for TCP traffic as well as the effect of link failure in TCP traffic in spite of the reliable transport protocol. Chapter 6 describes a multiple failures tolerance mechanism in a protected LSP. In Chapter 7, we present the proposal that addresses the drawback of the preplanned technique in terms of optimal protection path. At the same time the proposal handles the vulnerability problem for rerouted traffic. In Chapter 8, we present an experimental proposal and preliminary results for a further optimization mechanism that takes into account dynamic changes of network status. Finally, we conclude presenting the summary of our contributions and direction for future work.

ACKNOWLEDGEMENTS

I wish to thank the many people who have, in one way or the other, made this thesis possible. I apologize that I cannot list all the people that I am thankful to.

I would like to express my sincere thanks to my director of thesis Professor Jordi Domingo-Pascual, for giving me the opportunity to work in his research group, for trusting in my commitment, for his guidance and encouragement, and finally for the financial funding provided. I would also like to acknowledge Professor Mateo Valero for believe in me, for treating me as if I was one of his own PhD students and for all the help and support extended in requesting financial support from UPC. I am deeply indebted to both of them. I also do not forget that my presence here has been made possible by my former institute, "Instituto Superior Politécnico José Antonio Echeverría", ISPJAE, and thanks also to Dr. Juan Chavez Amarro.

I am extremely grateful to Adrián Cristal, with whom I have discussed several issues in this thesis, especially on the simulation. He allowed me to feel part of his family, likewise to Cristina, Ardiana and Leo. The same to Sanjeevan Kanapathipillai whose help, stimulating suggestions, encouragement, and discussion in many topics helped me in all the time of research to face all kind of adversities.

I am grateful to Prof. Eduard Ayugadé, for guiding me to the appropriate line of investigation in the department according to my interests and for his concern of my studies and situation.

I wish to express my gratitude to all members of the Integrated Broadband Communications System Research Group: Xavier, Sergio, Carlos V., Xavi, Rene, Carles K., Albert and especially to Josep Mangues, Jaume Masip, and Dr. Josep Solé-Pareta.

My life in Barcelona (Spain) would have been much harder if it were not for many friends who helped me on different circumstances. Starting from my arrival at Madrid. Wubalem, Tefera, Yirgalem, Tsegaye, Dereje, Yoseph, Dawit, Mesfin, Robel, Denekachew, Teshome, Meaza, Behailu, Wondifraw, Fitala, Tamrat, Elena, Dinberu, Sonia, Kifle, Constansa, Quino, Isabel, Fernando, Rodrigo, Oscar, José Antonio, Vero, Douglas, Nori, Paqui, and my distinguish Friday's football team members deserve special mention. My stay here has served to value helpful and trusted friends, and provided me the opportunity to meet many good friends.

Special thanks go to Nega, Asnake, Daniel Tesfaye, Berhanu Moges, Gashaw, Gosaye, Abebayehu, Gudeta and Kebede for their help, support and friendship.

I appreciate the friendship of my fellow researchers: Alex, Alex Ramirez, Beatriz, Blaise, Carlos, Carlos Molina, Daniel, Dany, Germán, Javier, Jesus, Jesus Acosta, José, Lorenzo, Manel, Oscar Ardaiz, Oscar Camacho, Oscar, Lepe, Pedro, Ramón, Suso, Victor, Jaume, Fernando, Ayose, Oliver, Ruben, Juan Iván, Ruben Alvaro, Raimir, Francisco, Larisa, Olimpia, Fran, Xavi, Marco, Carmelo, Alejandro, Amilcar, and system support people (Alex, Xavi, Victor, Joan, etc).

Special thanks go to Tesfaye Zibello for giving me permission to commence the PhD study and to my apartmentmates throughout these years: Ricardo, Carlos, Hector, Alex, Alirio and Diego.

I have furthermore to thank all staffs members of the Computer Architecture Department (DAC) at UPC for their hospitality. All these people were always nice and considerate, and my awkward feelings as an 'outsider' was minimized by their company. Thanks again for all.

I would express heartfelt thanks to all my friends abroad: Alemayehu, Bisrat, Dilnesahu, Yeshi, Hirut, Tekalign, Tibebe, Yeshi morena, Yitbarek, Mekenen, Elsa, Muazim, Hava, Adanech, Laurie, Elena, Elsa, Abebe, and Yohanis for their invaluable support and encouragement given to my wife in all aspects.

Especial thanks goes to my brother Ayalkibet for covering my main weakness for living alone, he knows what I mean, to my sister Tsedalu, and brothers Assefa and Teshome, and to all my families.

Last but certainly not least, I wish to give a very special thank you and all my love to my wife Truwork. Without her, I doubt that this thesis would ever have been written. I would also like to recognize the unrelenting support my wife afforded me. My deepest gratitude I must reserve for her, whose patience and understanding I am very thankful for.

This work has been supported by a grant of the Agencia Española de Cooperación Internacional (AECI), by CIRIT 2001-SGR-00226, by the Ministry of Science and Technology of Spain under contract MCYT TIC2002-04531-C04-02, and by UPC research support fund.

CONTENTS

LIST OF FIGURES

Chapter 1

Chapter 2

Chapter 3

Chapter 6

Chapter 7

Chapter 8

Chapter 9

LIST OF TABLES

Chapter 8

Chapter 9

Glossary

ACK or Ack	Acknowledgement
APS	Automatic Protection Switch
AS	Autonomous System
ATM	Asynchronous Transfer Mode
ATMARP	ATM Address Resolution Protocol
BGP	Border Gateway Protocol
BUS	Broadcast and Unknown Server
BW	Bandwidth
Bwid	Bandwidth of aggregated initial demand
BWt	Bandwidth threshold
Bwu	Bandwidth usage
CBR	Constant Bit Rate
CPU	Central Processing Unit
CR-LDP	Constraint-based Routing-Label Distribution Protocol
CSPF	Constraint-based Shortest Path First
DCS	Digital Cross-Connects
DiffServ	Differentiated Services
DLCI	Data Link Connection Identifier
ERB	Explicit Routing information Base
ER-LSP	Explicitly Routed Label Switched Path
FEC	Forward Equivalence Class
FR	Frame Relay
FTN	FEC-To-NHLFE
FTP	File Transfer Protocol
GMPLS	Generalized Multi-Protocol Label Switching
IETF	Internet Engineering Task Force
ILM	Incoming Label Map
IntServ	Integrated Services

IP	Internet Protocol
IS-IS	Intermediate System-to-Intermediate System protocol
ISP	Internet Service Provider
LANE	LAN Emulation
LDP	Label Distribution Protocol
LER	Label Edge Router
LIB	Label Information Base
LSP	Label Switched Path
LSPID	Label Switch Path ID
LSR	Label Switched Router
MARS	Multicast Address Resolution Server
MNS	MPLS Network Simulator
MPLS	Multi-Protocol Label Switching
MPOA	Multiprotocol over ATM
NHLFE	Next Hop Label Forwarding Entry
NHRS	Next Hop Resolution Server
OSPF	Open Shortest Path First
PFT	Partial Forwarding Table
PNNI	Private Network-to-Network Interface
QoS	Quality of Service
RSVP	Resource Reservation Protocol
RSVP-TE	Resource Reservation Protocol with Traffic Engineering
RTP	Real-time Transport Protocol
RTSP	Real Time Streaming Protocol
SHN	Self-Healing Network
SHR	Self-Healing Ring
SLA	Service Level Agreement
SONET	Synchronous Optical Network
SPD	Segment Protection Domain
SPF	Shortest Path First

SRLG	Shared Risk Link Group
STM	Synchronous Transfer Mode
TCP	Transmission Control Protocol
TE	Traffic Engineering
UDP	User Datagram Protocol
VC	Virtual Channel/Connection
VCI	Virtual Channel/Connection Identifier
VoIP	Voice over IP
VP	Virtual Path
VPI	Virtual Path Identifier
VPN	Virtual Private Network

1

MULTI-PROTOCOL LABEL SWITCHING

1.1 INTRODUCTION

It is estimated that in the near future, data will account for 80 % of all traffic carried by telecommunications networks. Therefore, the past concept of telephone networks which also carry data will be replaced by the concept of data networks that also carry voice. Lately the telecommunication industry has been highly focused on the leap to IP for telecommunication services. It is foreseen that Multiprotocol Label Switching (MPLS) will be chosen as the bearer of IP in future large backbone networks.

Multi-Protocol Label Switching (MPLS) [RVC01],[CDF+99] has recently been accepted as a new approach for integrating layer 3 routing (IP) with layer 2 switching technology (Asynchronous Transfer Mode (ATM), Frame relay (FR) and the extension Generalized MPLS (GMPLS) for optical networks). It tries to provide the best of both worlds: the efficiency and simplicity of routing together with the high speed of switching. For this reason MPLS is considered to be a promising technology that

addresses the needs of future IP-based networks. It enhances the services that can be provided by IP networks, offering scope for Traffic Engineering (TE), guaranteed Quality of Service (QoS), Virtual Private Networks (VPNs), etc. MPLS does not replace IP routing, but works along with existing and future routing technologies to provide very high-speed data forwarding between Label-Switched Routers(LSRs) together with QoS provision.

1.2 BACKGROUND

One challenge in current network research is how to effectively transport IP traffic over any network layer technology (ATM, FR, Ethernet, Point-to-Point). IP was independently developed on the basis of a connectionless model. In a connectionless network layer protocol when a packet travels from one router to the next, each router looks at the packet header to take the decision to forward the packet to the next corresponding hop according to a network layer routing algorithm based on the longest prefix match forwarding principle. Routers forward each IP packet independently on a hop-by-hop basis. Therefore, IP traffic is usually switched using packet software-forwarding technology, which has a limited forwarding capacity.

On the other hand, connection-oriented networks (ATM, FR) establish a virtual connection from the source to the destination (end-to-end) before forwarding the packets. That is, a connection must be established between two parties before they can send data to each other. Once the connection is set up, all data between them is sent along the connection path.

To relate the ATM and the IP protocol layers, two models have been proposed: the overlay model and the integrated model.

1.2.1 Overlay model

The overlay model considers ATM as a data link layer protocol on top of which IP runs. In the overlay model the ATM network has its own addressing scheme and routing protocol. The ATM addressing space is not logically coupled with the IP addressing space, in consequence direct mapping between them is not possible. Each end system will typically have an ATM address and an unrelated IP address. Since there is no mapping between the two addresses, the only way to resolve one from other is through some address resolution protocol. This involves running two control planes: first ATM Forum signaling and routing and then on top of that, IP routing and address resolution.

Substantial research has been carried out and various standards have been ratified by IETF and the ATM Forum addressing the mapping of IP and ATM, such as: Classical IP over ATM [LH98], Next Hop Resolution Protocol(NHRP)[LKP+98], LAN Emulation(LANE) [lan95], Multi-Protocol Over ATM(MPOA) [mpo97], etc. Furthermore, a rather complex signaling protocol has been developed so that ATM networks can be deployed in the wide area, Private Network-to-Network Interface (P-NNI) [pnn96].

Mapping between IP and ATM involves considerable complexity. Most of the above approaches servers (e.g., ATMARP, MARS, NHRS, and BUS) to handle one of the mapping functions, along with a set of protocols necessary to interact with the server. This server solution to map IP over ATM represents at the same time a single point of failure, and thus there is a desire to implement redundant servers, which then require a synchronization protocol to keep them consistent with each other. In addition to this, none of the above approaches exploit the QoS potential of layer 2 switches, i.e., the connection continues to be best-effort.

1.2.2 Integrated Model

The need for an improved set of protocols for ATM switches than those defined by the ATM Forum and the ITU has been addressed by various label switching approaches. These approaches are in fact attempts to define a set of protocols which can control an ATM switch in such a way that the switch naturally forwards IP packets without the help of servers mapping between IP and ATM.

Several label switching approaches have been proposed toward the integration of IP and ATM, supporting both layer 3 IP routing (software forwarding) and layer 2 ATM hardware switching [DDR98]. Under such names as Cell Switching Router (CSR)[KNE97][KNE96][NKS+97][KNME97], IP switching [NLM96][NEH+96a] [NEH+96b][NEH+98], Tag Switching [DDR98][RDK+97], and Aggregate Route-based IP Switching(ARIS) [AFBW97][FA97], layer 3 routing and label binding/swapping are used as a substitute for layer 2 ATM routing and signaling for the ATM hardware-switched connection setup. These four approaches to label switching are the founding contributors of MPLS technology.

Although label switching tries to solve a wider range of problems than just the integration of IP and ATM, the difficulties associated with mapping between IP and ATM protocol models was a significant driver for the development of label-switching technology. Therefore, these early developments were meant to resolve the challenges presented by overlay models (IP over ATM). All these tagging and label swapping approaches provide data forwarding using labels.

In the evolution of MPLS there are perhaps two key ideas. The first is that there is no reason that an ATM switch can't have a router inside it (or a router have ATM switch functionality inside it). The second is that once the router and ATM switch are integrated, dynamic IP routing can be used to trigger virtual circuit (VC) or path setup. Instead of using management software or manual configuration to drive circuit setup, dynamic IP routing might actually drive the creation of circuits or Label Switch Path (LSP) establishment.

Among the many positive attributes that MPLS brings to internetworking is the ability to provide connection-oriented services to inherently connectionless IP networks. The label switched path (LSP) is the establishment of a unidirectional end-to-end path forwarding data based on fixed size labels.

1.3 MPLS ARCHITECTURE

The basis of MPLS operation is the classification and identification of IP packets at the ingress node with a short, fixed-length, and locally significant identifier called a label, and forwarding the packets to a switch or router that is modified to operate with such labels. The modified routers and switches use only these labels to switch or forward the packets through the network and do not use the network layer addresses.

1.3.1 Separation of Control and Data Planes

A key concept in MPLS is the separation of the IP router's functions into two parts: forwarding (data) and control [CO99]. The separation of the two components enables each to be developed and modified independently.

The original hop-by-hop forwarding architecture has remained unchanged since the invention of Internet architecture; the different forwarding architecture used by connection-oriented link layer technologies does not offer the possibility of a true end-to-end change in the overall forwarding architecture. For that reason, the most important change that MPLS makes to the Internet architecture is to the forwarding architecture. It should be noted that MPLS is not a routing protocol but is a fast forwarding mechanism that is designed to work with existing Internet routing protocols, such as Open Shortest Path First(OSPF) [Moy98], Intermediate System-to-Intermediate System (IS-IS) [Ora90], or the Border Gateway Protocol(BGP) [RL95].

The control plane consists of network layer routing protocols to distribute routing information between routers, and label binding procedures for converting this rout-

ing information into the forwarding table needed for label switching. Some of the functions accomplished by the control plane are to disseminate decision-making information, establish paths and maintain established paths through the MPLS network. The component parts of the control plane and the data plane are illustrated in Figure 1.1.

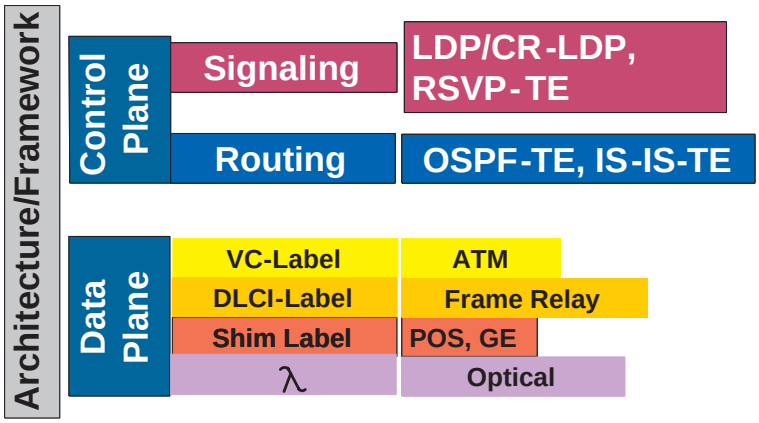

Figure 1.1 Control and Data plane components

The data plane (forwarding plane) is responsible for relaying data packets between routers (LSRs) using label swapping. In other words, a tunnel is created below the IP layer carrying client data. The concept of a tunnel (LSP tunnel) is key because it means the forwarding process is not IP based but label based. Moreover, classification at the ingress, or entry point to the MPLS network, is not based solely on the IP header information, but applies flexible criteria to classify the incoming packets.

1.3.2 Forward Equivalent Class (FEC)

Forward Equivalent Class (FEC) is a set of packets that are treated identically by an LSR. Thus, a FEC is a group of IP packets that are forwarded over the same LSP and treated in the same manner and can be mapped to a single label by an LSR even if the packets differ in their network layer header information. Figure 1.2 shows this behavior. The label minimizes essential information about the packet. This might

include destination, precedence, QoS information, and even the entire route for the packet as chosen by the ingress LSR based on administrative policies. A key result of this arrangement is that forwarding decisions based on some or all of these different sources of information can be achieved by means of a single table lookup from a fixed-length label.

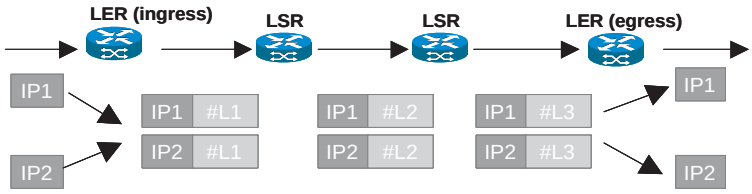

Figure 1.2 Forward Equivalent Class (FEC)

This flexibility is one of the key elements that make MPLS so useful. Moreover, assigning a single label to different flows with the same FEC has advantages derived from "flow aggregation". For example, a set of distinct address prefixes (FECs) might all have the same egress node, and label swapping might be used only to get the traffic to the egress node. In this case, within the MPLS domain, the union of those FECs is itself a FEC [RVC01]. Flow aggregation reduces the number of labels which are needed to handle a particular set of packets, and also reduces the amount of label distribution control traffic needed. This improves scalability and reduces the need for CPU resources.

1.3.3 Label

A label called a "shim label", or an MPLS "shim" header is a short, fixed-length, locally significant FEC identifier. Although the information on the network layer header is consulted for label assignment, the label does not directly encode any information from the network layer header like source or destination addresses [DR00]. The labels are locally significant only, meaning that the label is only useful and rel-

evant on a single link, between adjacent LSRs. Figure 1.3 presents the fields of an MPLS "shim" header.

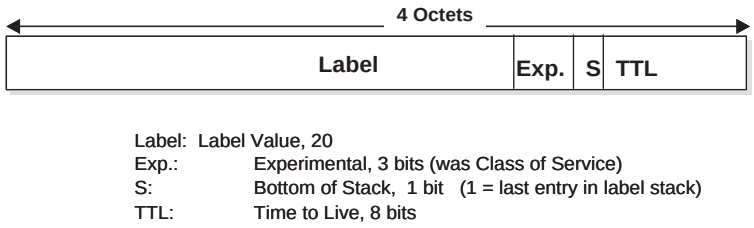

Figure 1.3 MPLS "shim" header format

In MPLS the assignment of a particular packet to a particular flow is done just once, as the packet enters the network. The flow (Forward Equivalence Class) which the packet is assigned to is encoded with a short fixed length value known as a "label" [RTF+01] Figure 1.3. When a packet is forwarded to the next hop, this label is sent along with it, that is, the packets are "labeled". At subsequent hops there is no further analysis of the packet's network layer header. The label itself is used as hop index. This assignment eliminates the need to perform the longest prefix-match computation for each packet at each hop, as shown in Figure 1.4. In this way the computation can be performed just once, as shown in Figure 1.5.

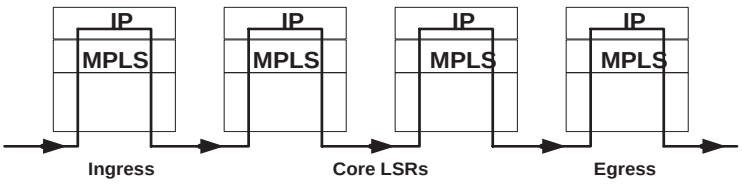

Figure 1.4 IP Forwarding: all LSRs extract information from layer 3 and forward the packets

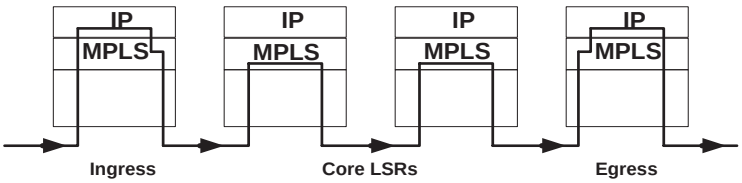

Figure 1.5 MPLS Forwarding: Ingress LSR extracts layer 3 information, assigns packet to FEC, pushes a label and forwards the packet. Core LSRs use label forwarding. Egress LSR pops the label, extracts layer 3 information and forwards the packet accordingly

1.3.4 Label Encapsulations

MPLS is multi protocol because is intended to run over multiple data link layers such as: ATM, Frame Relay, PPP, Ethernet, etc. It is label switching because it is an encapsulation protocol. The label encapsulation in MPLS is specified over various media type [DR00]. The top label on the stack may use the existing formats, lower label(s) use a new shim labels format. For IP-based MPLS, shim labels are inserted prior to the IP header. For ATM, the VPI/VCI addressing is the label. For Frame Relay, the DLCI is the label. Regardless of the technology, if the packet needs additional labels it uses a stack of shim labels. Figure 1.6 illustrates the label encapsulation in MPLS architecture.

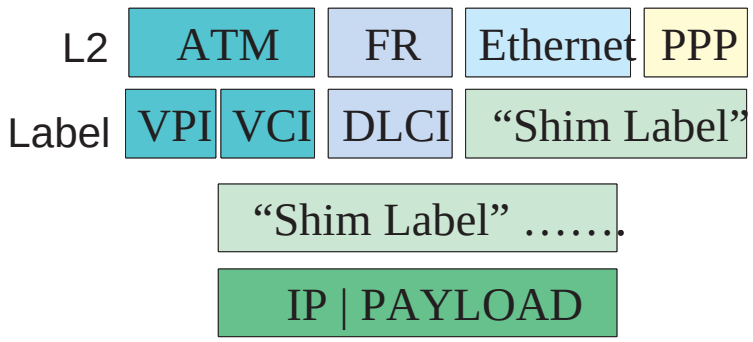

Figure 1.6 Label encapsulation

1.3.5 Label Swapping

Label Swapping is a set of procedures where an LSR looks at the label at the top of the label stack and uses the Incoming Label Map (ILM) to map this label to Next Hop Label Forwarding Entry (NHLFE). Using the information in the NHLFE, The LSR determines where to forward the packet, and performs an operation on the packet's label stack. Finally, it encodes the new label stack into the packet, and forwards the result. This concept is applicable in the conversion process of unlabeled packets to labeled packets in the ingress LSR, because it examines the IP header, consults the NHLFE for the appropriate FEC (FTN), encodes a new label stack into the packet and forwards it.

1.3.6 Label Stacking

A label stack is a sequence of labels on the packet organized as a last-in, first-out stack. A label stack enables a packet to carry information about more than one FEC which allows it to traverse different MPLS domains or LSP segments within a domain using the corresponding LSPs along the end-to-end path. Note that label processing is always based on the top label, without concern that some number of other labels may have been "above it" in the past, or that some number of other labels may be below it at present. The bottom of stack bit "S" in the shim header (see Figure 1.3) indicates the last stack level. The label stack is a key concept used to establish LSP Tunnels and the MPLS Hierarchy. Figure 1.7 illustrates the tunnelling function of MPLS using label stacks.

1.3.7 Label Switch Router (LSR)

A Label Switch Router(LSR) is a device that is capable of forwarding packets at layer 3 and forwarding frames that encapsulate the packet at layer 2. It is both a router and a layer 2 switch that is capable of forwarding packets to and from an MPLS domain. The edge LSRs are also known as Label Edge Routers (LERs).

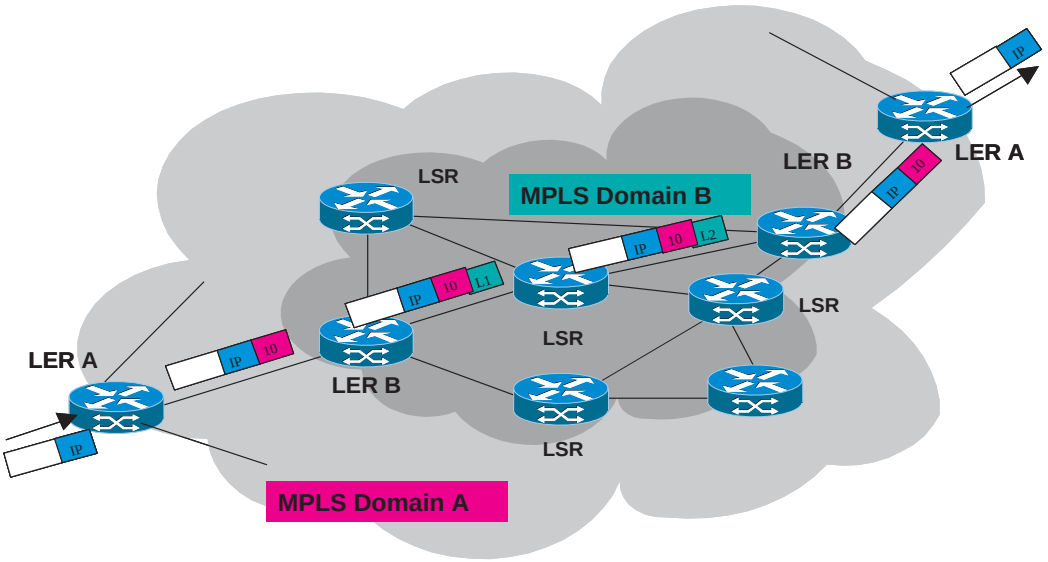

Figure 1.7 Label Stack. LERs A are for MPLS domain A and LERs B are for MPLS domain B

The ingress LSR pushes the label on top of the IP packet and forwards the packet to the next hop. In this phase as the incoming packet is not labeled, the FEC-to-NHLFE (FTN) map module is used.

Each intermediate/transit LSR examines only the label in the received packet, replaces it with the outgoing label present in the label information based forwarding table (LIB) and forwards the packet through the specified port. This phase uses the incoming label map (ILM) and next-hop label forwarding entry (NHLFE) modules in the MPLS architecture.

When the packet reaches the egress LSR, the label is popped and the packet is delivered using the traditional network layer routing module. All the above descriptions are illustrated in Figure 1.8.

If the egress LSR is not capable of handling MPLS traffic, or for the practical advantage of avoiding two lookup times that the egress LSR requires to forward the packet,

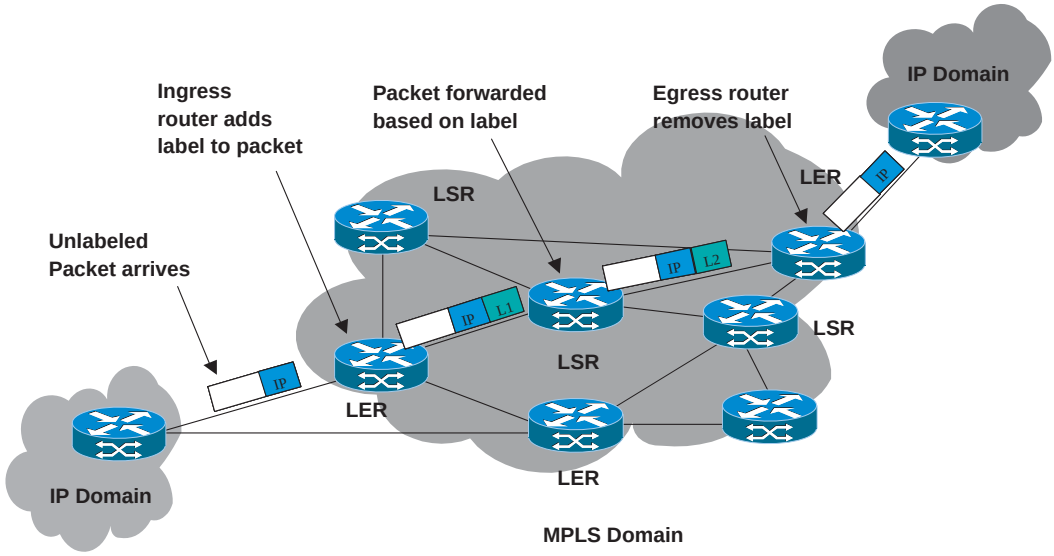

Figure 1.8 MPLS Architecture

the penultimate hop popping method is used. In this method, the LSR whose next hop is the egress LSR, will handle the label stripping process instead of the egress LSR.

1.3.8 Label Switched Path (LSP)

A Label Switched Path (LSP) is an ingress-to-egress switched path built by MPLS capable nodes which an IP packet follows through the network and which is defined by the label (Figure 1.9). The labels may also be stacked, allowing a tunnelling and nesting of LSPs [RVC01] [RTF+01]. An LSP is similar to ATM and FR circuit switched paths, except that it is not dependent on a particular Layer 2 technology.

Label switching relies on the set up of switched paths through the network. The path that data follows through a network is defined by the transition of the label values using a label swapping procedure at each LSR along the LSP. Establishing an LSP involves configuring each intermediate LSR to map a particular input label and

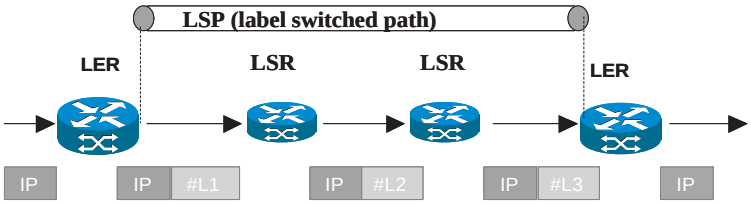

Figure 1.9 Label Switched Path (LSP)

interface to the corresponding output label and interface (label swap). This mapping is stored in the label information based forwarding table (LIB).

There are two kinds of LSP depending on the method used for determining the route: hop-by-hop routed LSPs when the label distribution protocol (LDP) [ADF+01] is used, and explicit routed if the path should take into account certain constraints like available bandwidth, QoS guarantees, and administrative policies; explicit routing uses the constraint routed label distribution protocol (CR-LDP) [JAC+02] or the Resource Reservation Protocol with traffic engineering extensions (RSVP-TE) [ABG+01] as signaling protocols.

1.4 LABEL DISTRIBUTION PROTOCOL

In MPLS two adjacent Label Switching Routers (LSRs) must agree on the meaning of labels used to forward traffic between them and through them. The label distribution protocol (LDP) is a protocol defined by IETF MPLS WG [ADF+01] for distributing labels in MPLS networks. LDP is a set of procedures and messages by which LSRs establish Label Switched Paths(LSPs) through a network by mapping network layer routing information directly to data link layer switched paths, as shown in Figure 1.10.

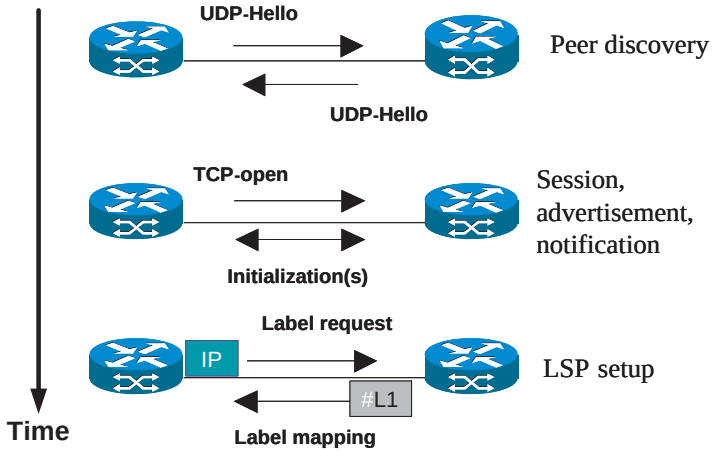

Figure 1.10 Label Distribution Protocol (LDP)

1.5 LABEL DISTRIBUTION MODES

In the MPLS architecture, the decision to bind a label to a FEC is made by the LSR which is downstream with respect to that binding. The downstream LSR informs to the upstream LSR of the label that it has assigned to a particular FEC. Thus labels are "downstream assigned" [RVC01].

The MPLS architecture defines two downstream assignments of label distribution modes for label mapping in LSRs: they are Downstream-on-Demand label distribution mode and Unsolicited Downstream label distribution mode.

1.5.1 Downstream-on-Demand

The MPLS architecture allows an LSR to explicitly request, from its next hop for a particular FEC, a label binding for that FEC. This is known as the "Downstream-on-Demand" label distribution mode, where the upstream LSR is responsible for requesting a label for binding. Figure 1.11 shows this process.

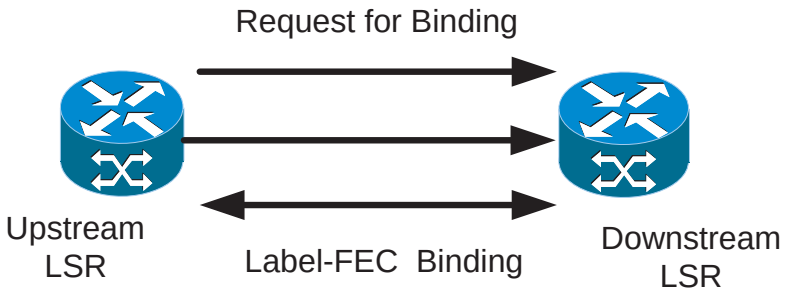

Figure 1.11 Downstream-on-Demand Label Advertisement

1.5.2 Unsolicited Downstream

The MPLS architecture also allows an LSR to distribute label bindings to LSRs that have not explicitly requested them. This is known as the "Unsolicited Downstream" label distribution mode, where the downstream LSR is responsible for advertising a label mapping to upstream LSRs. Figure 1.12 illustrates a downstream LSR delivering a label-FEC binding to an upstream LSR without having been requested for it.

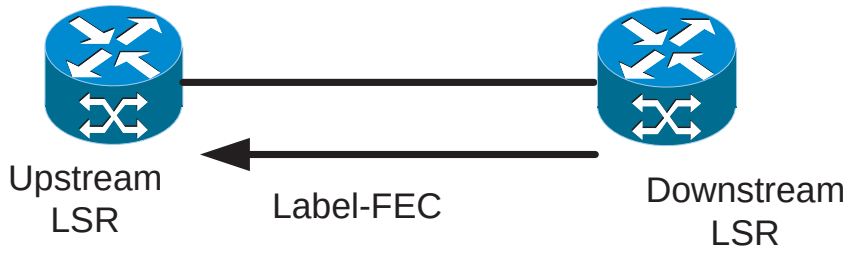

Figure 1.12 Unsolicited Downstream Label Advertisement

1.6 LSP CONTROL MODES

There are two label distribution control modes defined in the MPLS architecture to create (establish) an LSP. They are Independent Label Distribution Mode and Ordered Label Distribution Mode.

1.6.1 Independent Label Distribution control

In the independent label distribution control, each LSR makes an independent decision to bind a label to a particular FEC and to distribute that binding to its label distribution peers (i.e., its neighbors). This corresponds to the way that conventional IP datagram routing works; each node makes an independent decision as to how to treat each packet.

If the independent downstream-on-demand mode is used, the LSR may reply to a request for label binding without waiting to receive the corresponding label binding from the next hop. When the independent unsolicited downstream mode is used, an LSR advertises a label binding for a particular FEC to its label distribution peers whenever the label is ready for that FEC.

1.6.2 Ordered Label Distribution Control

In ordered label distribution control, an LSR only binds a label to a particular FEC in response to a label binding request. The egress LSR sends a label for that FEC directly since it is the last node in the MPLS domain . If the LSR is an intermediate LSR it must have already received a label binding for that FEC from its next hop before it sends its label binding. In this control mode, except the egress LSR, before an LSR can send a label to upstream LSRs, it must wait to receive the label for its request from a downstream LSR.

1.7 LABEL RETENTION MODES

There are two modes to retain labels in an LSR defined in the MPLS architecture. These are Liberal and Conservative label retention modes. These modes specify whether an LSR maintains a label binding or not for a FEC learned from a neighbor that is not its next hop for this FEC according to the routing.

1.7.1 Liberal Label Retention Mode

In liberal label retention mode, every label binding received from label distribution peers in an LSR is retained regardless of whether the LSR is the next hop for the label binding (i.e., whether they are used for packet forwarding or not).

The unsolicited downstream label advertisement mode is an example of when all received labels are retained and maintained by the upstream LSR, as illustrated in Figure 1.13.

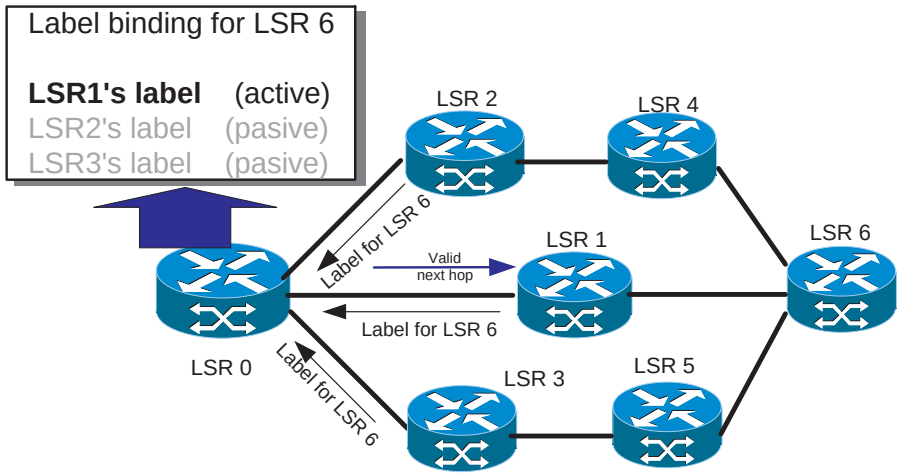

Figure 1.13 Liberal Label Retention Mode

The main advantage of the liberal label retention mode is that the response to routing changes may be fast because the LSR already has spare labels in its LIB. The disadvantage is that it maintains and distributes unnecessary labels.

1.7.2 Conservative Label Retention Mode

In conservative label retention mode the advertised label bindings are retained only if they will be used to forward packets (i.e., if they are received from a valid next hop according the routing), as shown in Figure 1.14. Note that Downstream-on-Demand forces in some way the use of conservative retention mode, rather than the liberal.

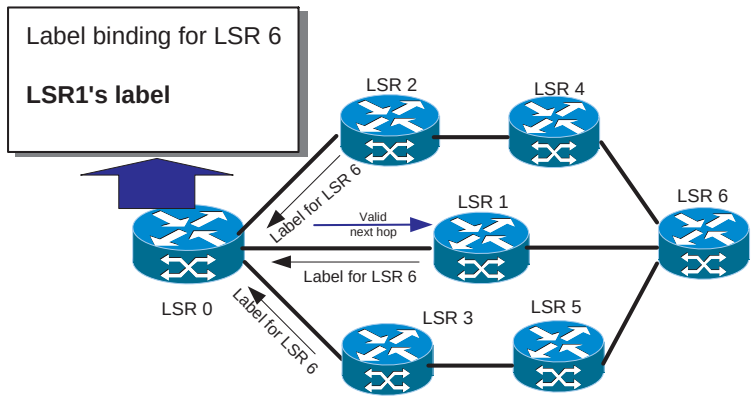

Figure 1.14 Conservative Label Retention Mode

The main advantage of the conservative mode is that only the labels that are required for forwarding of data are retained and maintained. This is very important for scalability in LSRs with limited label space. The disadvantage is well seen when rerouting is needed. In this case a new label must be obtained from the new next hop before labeled packets can be forwarded.

1.8 CONTROL PLANE

1.8.1 Information Dissemination

The link state protocols, specifically OSPF and IS-IS, provide the link state information that details the entire underlying network. This information is crucial to path selection, path establishment and maintenance functions. Further, both OSPF and IS-IS protocols have been extended to include resource information about all links in the specific area. Through these extensions MPLS traffic engineering becomes possible.

1.8.2 Path Selection

The control plane determines the best path through a network using either a hop-by-hop or an explicit route methodology. The hop-by-hop method allows the selection to follow the normal underlying IGP best path. Each node in the path is responsible for determining the best next hop based on the link state database. Alternatively, an explicit route is a path through the network that is specified by the ingress LSR. The explicitly routed path has administratively configured criteria or policies to influence the path selection through the underlying network.

1.8.3 Path Establishment

Once the path has been determined, a signaling protocol (LDP, CR-LDP or RSVP) is used to inform all the routers in the path that a new label switched path (LSP) is required. The signaling protocol is responsible for indicating the specifications of the path, including the session id, resource reservations, and the like, to all other routers in the path. This process also includes the label mapping request for all data that will use the LSP. Following the successful establishment of the path, the signaling protocol is responsible for ensuring the integrity of the peering session.

1.9 DATA PLANE

1.9.1 Packet Forwarding

The data flow into an MPLS network occurs at the ingress LSR, commonly referred to as ingress label edge router, or ingress LER. The ingress LSR classifies a packet or a flow to a specific LSP and pushes the applicable label on the packet. This classification of client data to an LSP occurs only once, at the ingress router, using some policy. Routers along the label switched path perform forwarding based on the top level inbound label. The label switched path terminates at the boundary between an MPLS enabled network and traditional network. The egress label switch router, the egress LER, is responsible for removing the label from the packet and forwarding the packet based on the packet's original contents, using traditional means.

1.10 BENEFIT/APPLICATION OF MPLS

1.10.1 Simple Forwarding

As MPLS uses fixed length label-based forwarding, the forwarding of each packet is entirely determined by a single indexed lookup in a switching table, using the packet's MPLS label. This simplifies the label switching router forwarding function compared to the longest prefix match algorithm required for normal datagram forwarding.

1.10.2 Traffic Engineering

One of the main advantages of MPLS is the ability to do Traffic Engineering (TE) in connectionless IP networks. TE is necessary to ensure that traffic is routed through a given network in the most efficient and reliable manner. Traffic engineering enables ISPs to route network traffic in such a way that they can offer the best service to their users in terms of throughput and delay. MPLS traffic engineering allows traffic to be distributed across the entire network infrastructure.

MPLS traffic engineering provides a way to achieve the same traffic engineering benefits of the overlay model without the need to run a separate network. With MPLS, traffic engineering attempts to control traffic on the network using Constrained Shortest Path First (CSPF) instead of using the Shortest Path First (SPF) only. CSPF creates a path that takes restrictions into account. This path may not always be the shortest path, but, for instance, it will utilize paths that are less congested.

The LSP tunnel is useful for the TE function. LSP tunnels allow operators to characterize traffic flows end-to-end within the MPLS domain by monitoring the traffic on the LSP tunnel. Traffic losses can be estimated by monitoring ingress LSR and egress LSR traffic statistics. Traffic delay can be estimated using by sending probe packets through and measuring the transit time.

One approach to engineering the network is to define a mesh of tunnels from every ingress device to every egress device. IGP, operating at an ingress device, determines which traffic should go to which egress device, and steers that traffic into the tunnel from ingress to egress. The MPLS traffic engineering path calculation and signaling modules determine the path taken by the LSP tunnel, subject to resource availability and the dynamic state of the network.

Sometimes a flow is so large that it cannot fit over a single link, so it cannot be carried by a single tunnel. In this case multiple LSP tunnels between a given ingress and egress can be configured, and the flow load shared among them. This prevents a situation where some parts of a service provider network are over-utilized, while other parts remain under-utilized. The capability to forward packets over arbitrary non-shortest paths and emulate high-speed tunnels within an MPLS domain yield a TE advantage to MPLS technology.

1.10.3 Source based QoS Routing

Source based QoS routing is a routing mechanism under which LSRs are determined in the source node (ingress LSR) based on some knowledge of resource availability in the network as well as the QoS requirements of the flows. In other words, it is a routing protocol that has expanded its path selection criteria to include QoS parameters such as available bandwidth, link and end-to-end path utilization, node resource consumption, delay and latency, including jitter.

MPLS allows decoupling of the information used for forwarding (i.e., label) from the information carried in the IP header. Also the mapping between FEC and an LSP is completely confined to the LER at the head of the LSP: the decision as to which IP packet will take a particular explicit route is totally the responsibility of the LER (ingress LSR) which computes the route. This allows MPLS to support the source based QoS routing function.

1.10.4 Virtual Private Networks

An Internet-based virtual private network (VPN) uses the open, distributed infrastructure of the Internet to transmit data between sites, maintaining privacy through the use of an encapsulation protocol to establish tunnels. A virtual private network can be contrasted with a system of owned or leased lines that can only be used by one company. The main purpose of a VPN is to give the company the same capabilities as private leased lines at much lower cost by using the shared public infrastructure. The MPLS architecture fulfils all the necessary requirements to support VPNs by establishing LSP tunnels using explicit routing. Therefore, MPLS using LSP tunnels allows service providers to deliver this popular service in an integrated manner on the same infrastructure they used to provide Internet services. Moreover, label stacking allows configuring several nested VPNs in the network infrastructure.

1.10.5 Hierarchical Forwarding

The most significant change produced by MPLS in the internet architecture is not in the routing architecture, but in forwarding architecture. This modification in the forwarding architecture has a significant impact in its ability to provide hierarchical forwarding. Hierarchical forwarding allows the encapsulation of an LSP within another LSP (label stacking or multiple level packet control).

Hierarchical forwarding is not new in network technology; ATM provides two level hierarchy forwarding with the notion of virtual path(VP) and virtual circuit(VC) i.e., two levels of packet control. MPLS, however, allows LSPs to be nested arbitrarily, providing multiple level packet control for forwarding.

1.10.6 Scalability

Label switching provides a more complete separation between inter-domain and intra-domain routing, which helps to improve the scalability of routing processes. Furthermore, MPLS scalability also benefits from FEC (flow aggregation), and label stacking for merging LSPs and nesting LSPs. The assignment of a label for each individual flow is not the desired idea for scalability because it increases the usage of labels, which consequently causes the LIB to growth as fast as the number of flows in the network. As FEC allows flow aggregation, this improves MPLS scalability. In addition, multiple LSPs associated to different FECs can be merged in a single LSP, further improving this feature. Some benefits will also be gained from LSP nesting.

1.11 SUMMARY

In conventional network layer protocols, when a packet travels from one router to the next hop an independent forwarding decision is made at each hop. Each router runs a network layer routing algorithm. As a packet travels through a network, each router analyzes the packet header. The choice of next hop for a packet is based on

the header analysis and the result of running the routing algorithm. In conventional IP forwarding, the particular router will typically consider two packets to be in the same flow if they have the same network address prefix, applying the "longest prefix match" for each packet destination address. As a packet traverses the network, each hop in turn re-examines the packets and assigns it to a flow.

Label switching technology enables one to replace conventional packet forwarding based on the standard destination-based hop-by-hop forwarding paradigm with a label swapping forwarding paradigm. This is based on fixed length labels, which improves the performance of layer 3 routing, simplifies packet forwarding and enables easy scaling.

2

REVIEW OF RELATED WORK

2.1 OVERVIEW

Multiprotocol Label Switching (MPLS) fuses the intelligence of routing with the performance of switching and provides significant benefits to networks with a pure IP architecture as well as those with IP and ATM or a mix of other Layer 2 technologies. MPLS technology is key to scalable virtual private networks (VPNs) and end-to-end quality of service (QoS), enabling efficient utilization of existing networks to meet future growth. The technology also helps to deliver highly scalable, differentiated end-to-end IP services with simpler configuration, management, and provisioning for both Internet providers and end-users. However, MPLS is a connection-oriented architecture. In case of failure MPLS first has to establish a new label switched path (LSP) and then forward the packets from the fault point or another node (i.e., in case the fault point is not a candidate to redirect the traffic) to the newly established LSP. For this reason MPLS has a slow restoration response from a link or node failure on the LSP.

In recent years new services and applications were developed with strong real-time connection-oriented characteristics. Such services include Voice-over-IP or the Real Time Streaming Protocol (RTSP)[SRL98]. Also in the transport layer new protocols were developed to support real-time services, like Real Time Protocol (RTP) [SCFJ96]. To meet quality-of-service requirements IETF introduced IntServ [BCS94] [SPG97][Wro97], RSVP [BZB+97] and DiffServ [DCea02] [HBWW99][BB+98][NBBB98] [Bla00] in the Internet service models.

The failure of a major link or backbone router may have severe effects on these services and protocols. After the rerouting is completed the services may experience a degradation of their quality of service, since the alternative route can be longer or more congested. Note that traffic not directly affected by the failure but diverted over an alternative route is also affected by this degradation.

On the other hand, the duration of the interruption due to a link or node failure is in most cases too long for real-time services and multimedia applications to maintain their sessions. At the same time, QoS flows could experience an unacceptable reduction of their QoS on the alternative route, and therefore not be able to be reestablished.

Multimedia applications typically have strict requirements regarding delay, delay jitter, throughput, and reliability bounds. Real-time network services are designed to guarantee these performance parameters to applications that request them. IntServ and DiffServ are added as new Internet services methods to provide these performance guarantees.

For these new services and applications advanced rerouting mechanisms have to be developed in order to provide fast rerouting, so that the sessions will not impaired. Additionally, the design of internet architecture and capacity planning should take alternative routes into account for IP-flows with quality of service guarantees.

From the above consideration one can conclude that resilience is a clear requirement for current and future IP-based networks. Resilience refers to the ability of a network to keep services running despite a failure. Unfortunately, since the Internet was designed for maximum connectivity and robustness, mechanisms for the fast recovery of traffic affected by network failures are not well considered. This is basically due to the limitation of the hop-by-hop destination-based IP routing. Moreover, in IP-based networks some convergence problems may occur when IP routers dynamically update routes to restore connectivity.

One of the challenges of a path-oriented routing protocol such as MPLS is service guarantee during failure. For this reason the ability to quickly reroute traffic around a failure or congestion point in a label switched path (LSP) can be important in mission critical MPLS networks to ensure that guarantees for quality of service to the established LSP will not be violated under failure conditions. In MPLS-based networks when an established label switched path becomes unusable due to a physical link or node failure data may need to be rerouted over an alternative/backup path to minimize these LSP service interruptions.

In this thesis we address the inherent problem of MPLS as connection-oriented architecture to recover from a network component failure.

2.2 SUMMARY OF PREVIOUS WORK ON PATH RECOVERY

Restoration schemes on networks are generally divided into two main categories: Centralized or Distributed schemes. Each of these schemes can be divided into preplanned or dynamic modes of restoration. These repair modes in turn can each use one of two methods of repair activation: Local or Global restoration.

2.2.1 Centralized Recovery

The centralized restoration scheme uses a centralized management system to perform the restoration functions, such as failure detection, selection of alternate route, redirection of flows to the established alternative path, etc. The centralized scheme has the advantage of always getting all network information available, including during failure, so it is easier to optimize restoration paths. As a result, it can make effective utilization of spare resources, and it may decrease network resources required compared to the distributed restoration scheme.

On the other hand, restoration speed is relatively slow with the centralized scheme due to the communication delay between the centralized controller and LSRs, and the concentration of processing load on the centralized controller. Therefore, centralized control may not satisfy the restoration speed requirement.

2.2.2 Distributed Recovery

To alleviate the negative impact of the centralized mode for restoration, some proposals consider the distributed restoration mode. In the distributed restoration scheme, each node in the network is capable of handling failures. The fastest detection occurs at the local end of a link failure using the distributed restoration method.

Grover's Self-Healing network algorithm is the first distributed network restoration algorithm for digital cross-connection system (DCS) based fiber networks proposed in [Gro87] and detailed in his PhD thesis dissertation [W.D89]. Self-healing implies failed path restoration with a distributed network element control mechanism. When a network failure occurs, failed paths are rerouted by processing and message transmission between local network elements without the intervention of a centralized control system. Self-healing schemes can be categorized into self-healing networks (SHN) for mesh networks where no topological restriction exists and self-healing rings (SHR) for ring networks.

Following Grover's publication other distributed network restoration algorithms for DCS-based fiber networks were proposed by Yang and Hasegawa [YH88] and by Chow et al.[CBMS93].

The first method ([YH88]) is called FITNESS, and uses the same relationship principle between adjacent nodes to the fiber cut link as the SHN algorithm ([Gro87])(i.e., sender and chooser relationship). FITNESS, however, reduces the potentially large number of request messages that may be generated in SHN by requesting the aggregate maximum bandwidth that is allowed on a restoration route.

In the second [CBMS93], unlike previous methods, the two nodes adjacent to the fiber cut perform nearly symmetrical (identical) roles during the restoration process. The algorithm is based on a Two-Prong approach. In this approach the restoration is initiated from both nodes with each sending a restoration request message labelled in a different "color". When the intermediate nodes receive a single color labelled requesting message they forward the message on all links which contain spare channels. A node, upon receiving two different color labelled request messages, will make appropriate cross connections between the links over which the two different requests were received. Once the cross connection has been made the request message will be forwarded over the newly connected link to the next node in the restoration path.

All the above proposals start the restoration mechanism after the occurrence of failure. Schemes that try to restore after the presence of failure are known as dynamic restoration schemes. At the same time they activate the repair locally (i.e., use the local repair scheme).

On the other hand, Automatic Protection Switching (APS) and Self-Healing Ring (SHR) [Wu95] use a set of working and backup links to switch traffic from the failed links to pre-assigned/preplanned backup links. These schemes provide high speed restoration of the network.

One of the advantages of the preplanned restoration scheme over the dynamic restoration scheme is the restoration speed. The dynamic restoration scheme uses many messages during the restoration process between restoration pair nodes to locate backup routes, to establish paths, and so on. The preplanned restoration scheme, on the other hand, can complete restoration by passing messages along each pre-established backup link. This simplification of the message transmission process and the reduced number of messages allows higher restoration speeds than the dynamic restoration scheme.

The previous proposals were designed for synchronous transfer mode (STM) networks such as digital cross-connection restoration or self-healing rings. The studies of self-healing concepts at the ATM-layer began in 1990. An extensive survey of work is presented in [Wu95] and [Kaw98]. Restoration mechanisms for ATM networks are presented in [KST94] [KO99] [KT95] [ADH94] [KKT94] and the implementation scheme is presented in [SHT90].

The restoration mechanisms proposed in the MPLS network use the same general protection principles as ATM. In MPLS networks, since an LSP traverses a fixed path in the network, its reliability depends on the links and nodes along the path. Traditionally IP networks have carried only best-effort traffic. However, new applications requiring guarantees are using the IP network infrastructure. This makes it highly desirable to incorporate the faster repair mechanisms.

In [GS00] and [She99] MPLS network restoration mechanisms are proposed. Both address the restoration mechanism using local repair. The fastest detection occurs at the local end of a link failure. Schemes that try to mend connections at the point of failure are known as “local repair” schemes.

In the [GS00] proposal the authors focus on two types of protection: one-to-one (1+1) backup tunnel creating a second separate LSP for every protected LSP tunnel. And one-to-many (1: N) where a single LSP is created which serves to backup a set of protected tunnels using the label stacking advantages.

In [She99] the author considers the problems of engineering reliability of router-router links and fast recovery of MPLS LSPs. Specifically, the problem of fast failure detection and notification of affected MPLS LSPs is addressed.

Local repair has performance advantages in maintaining connectivity but at the expense of efficiency (more hops, more bandwidth, more end-to-end delay).

In [HA00] extensions to CR-LDP and RSVP-TE for setup of pre-established recovery tunnels are proposed. In this proposal after a switchover of traffic to the recovery LSP the authors allow the traffic to merge onto the protected LSP at the merging node downstream of the fault without causing any extra resource reservation.

A path protection mechanism for MPLS networks is proposed in [OSMH01]. The extension of CR-LDP to provide signaling support for establishing protected/working and backup LSPs is proposed in [OSM+01]. In [OSM+01] the authors propose the introduction of an Explicit Route Protection ER-Hop type; the Path Switch LSR (PSL) and the Path Merge LSR (PML) to allow the identification of the end-points of a protected path or path segment; and the Path Protection Type Level Value (TLV) to the Label Request message to help the configuration of a protection domain and Path Protection Error Codes in the CR-LDP. The authors also presented the extension of RSVP-TE for MPLS path protection in [OSM+02].

Several methods have been proposed to reroute traffic in MPLS. There are two schemes for MPLS restoration currently under consideration within IETF giving different approaches to the label switched path (LSP) restoration problem in MPLS-based networks. The first is the fastest MPLS rerouting mechanism available, called the MPLS Fast Rerouting mechanism proposed by Haskin and Krishnan [HK00] and the second is a slower but less complex mechanism proposed by Makam et al. [OSMH01] known as RSVP-based Backup Tunnel. A comparison of different MPLS protection and rerouting mechanisms can be found in [FM01].

2.3 MPLS RECOVERY MODELS

Several IETF drafts and a framework proposal are being discussed in the MPLS working group (MPLS WG) to handle the slow recovery from network component failure as a main disadvantage of MPLS, like any connection-oriented technology. In case of a network failure a new LSP tunnel could be set up for a group of failed LSPs to route the traffic around the failed network element. The IETF MPLS WG defines two recovery models: rerouting, and protection switching or fast rerouting.

Some definitions that will be used throughout the following sections and chapters follows:

Downstream: The direction of data moving from an ingress LSR to an egress LSR. Or, with respect to the flow of data in a communication path: at a specified point, the direction toward which packets are received later than at the specified point.

Upstream: The direction of data moving from an egress LSR to an ingress LSR. Or, the direction from which traffic is expected to arrive.

Primary or protected LSP: The path that carries traffic before the occurrence of a fault.

Backward LSP: The path on which traffic is directed by a recovery mechanism in the upstream direction from the point of failure to a rerouting point.

Alternative LSP: The path by which traffic is rerouted to the destination node after the occurrence of failure.

Protection path: A set of links and nodes traversed by the packet in a protected flow after a failure is detected. During the recovery time the protection path may vary according the recovery scheme used, but after the recovery time the new path is the alternative LSP.

Alert LSR or alert node: The LSR or node that detects a fault.

Recovery period: The duration of time from the detection of the fault until the protected LSP is completely eliminated. In other words, the interval of time between the detection of failure and the time when the last packet sent by the ingress LSR on the protected LSP is rerouted to the alternative LSP.

2.3.1 Rerouting

Rerouting is a technique that can be used in both Label Switching and Packet Switching networks. Rerouting is defined as the establishment of a new path or path segment on demand for traffic restoration after the occurrence of a fault. Thus it is a recovery mechanism in which the recovery path or path segment is created dynamically after the detection of a fault on the working path. For this purpose, an alternative or backup path apart from the primary path used by current traffic is needed. The primary and the backup paths should be totally disjoint. Network components mainly consist of links and nodes. As a node failure causes the failure of the adjacent links connected to the node, we use link failure as a network failure.

When a link on the primary path fails the restoration process starts automatically. A complete rerouting technique is described in the frameworks presented in [SH02][LCJ99] and consists of several steps. The main steps that the rerouting method must accomplish are fault detection, fault notification, alternative path computing, and rerouting of traffic from the primary path to the alternative path.

Fault Detection: The network must be able to detect link failures. Link failure detection can be performed by dedicated hardware or by software in the end nodes of the failed link.

Fault Notification: Nodes that detect a link failure (alert nodes) must notify certain nodes. Which nodes are actually notified depends on the rerouting technique. The alert node initiates the failure restoration process according to the applicable

restoration method to determine the failed paths and create and send a notification message requesting a search for alternative routes to the upstream node.

Alternative Path Computation: The upstream node performs the computation of an alternative path upon the reception of the notification message. If this node is not responsible for redirecting the traffic then it relays the notification message to the corresponding upstream node.

Reroute traffic to alternative/backup path: This process detours the traffic to the backup path instead of sending traffic on the primary, failed path. This process completes the restoration of the network after a link failure.

Traffic reverting: This is the process that returns traffic back from the alternative path to the primary path after the failed link has been repaired. When the traffic reverting mode is used, the mechanism must detect the complete repair of the failed link, notify the related nodes in the network, and reroute the traffic from the backup path to the primary path as soon as the path becomes available.

2.3.2 Fast Rerouting or Protection Switching

The Fast Rerouting or Protection Switching recovery mechanism pre-establishes the alternative protection path before the occurrence of the fault. The criteria to establish the pre-established/pre-planned alternative path are based on network routing policies, the restoration requirements of the protected traffic, and administrative considerations. When a fault occurs the LSR responsible for detouring the traffic switches the protected traffic from the primary path to a pre-established alternative path. Since the protection switching model pre-establishes a recovery path before the occurrence of a fault, the recovery time is shorter than the rerouting model.

We will focus our contribution on fast restoration schemes. Currently there are two schemes for MPLS restoration under consideration within IETF.

2.3.3 Rerouting Strategies

As explained above, fast rerouting uses pre-established alternative LSPs. When a fault is detected, the protected traffic is switched over to the alternative LSP. Setting pre-established alternative paths results in a faster switchover compared to establishing new alternative paths on-demand [HK00][SH02][MSOH99][OSMH01][Swa99]. However, because the fast rerouting alternative LSP is established at the time the protected LSP is setup, it may lead to the use of non-optimal alternative LSPs due to changes in the network. At setup time the alternative LSP was compliant with the QoS requirement and was the best alternative path, but when a failure occurs network conditions may have changed and there may be a different optimal alternative LSP.

Global optimization algorithms that can be computed at the ingress of the LSP have been proposed to alleviate this drawback [Swa99]. The combination of both fast rerouting and optimal path computation would be the best solution for service restoration. Chapter 7 deals with a new proposal that combines both approaches.

There are two possibilities for repair activation: global repair and local repair.

Global repair: Global repair is activated on an end-to-end basis, as shown in Figure 2.1. That is, an alternative LSP is pre-established or computed dynamically from ingress to egress nodes of the path to be protected. Note that when a dynamic approach is used in global repair a failure signal is propagated to the source (ingress LSR) before a new route can be established, which wastes valuable time because the failure notification has to traverse the entire network (MPLS domain).

Local repair: Local repair aims to fix the problem at the point of failure or within a very short distance from the failure, thereby minimizing total packet loss.

The techniques proposed for local repairs in MPLS networks are splicing and stacking [Swa99].

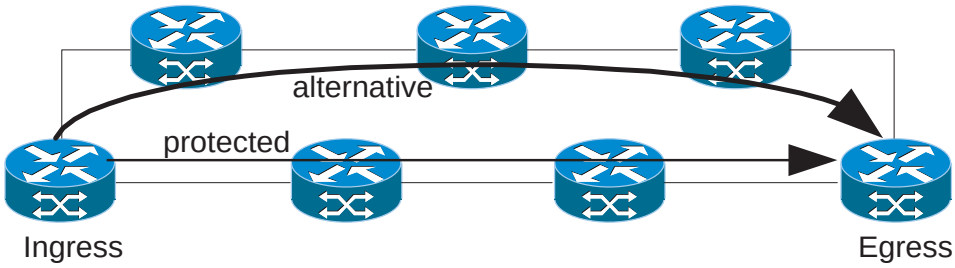

Figure 2.1 Global repair

Splicing: In this case an alternate LSP is pre-established from the point of protection to the egress LSR via an LSP that bypasses the network elements being protected. Upon detection of a failure, the forwarding entry for the protected LSP is updated to use the label and interface of the bypass LSP. Figure 2.2 illustrates the splicing repair technique in an MPLS domain.

The worst case requires as many alternative LSP candidates as the number of LSRs along the protected LSP minus one.

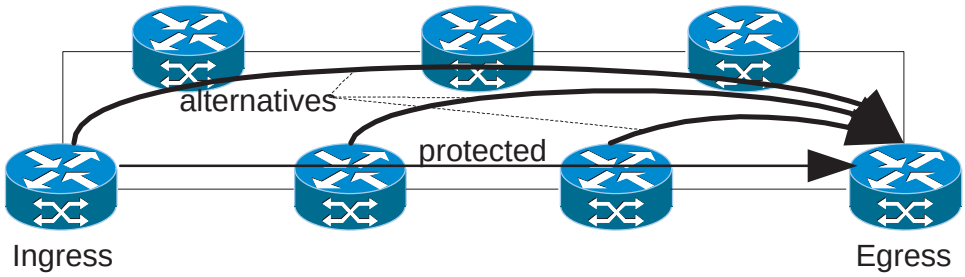

Figure 2.2 Local repair using splicing technique

When we refer dynamic restoration, this corresponds simply to splicing dynamic rerouting as illustrated in Figure 2.3.

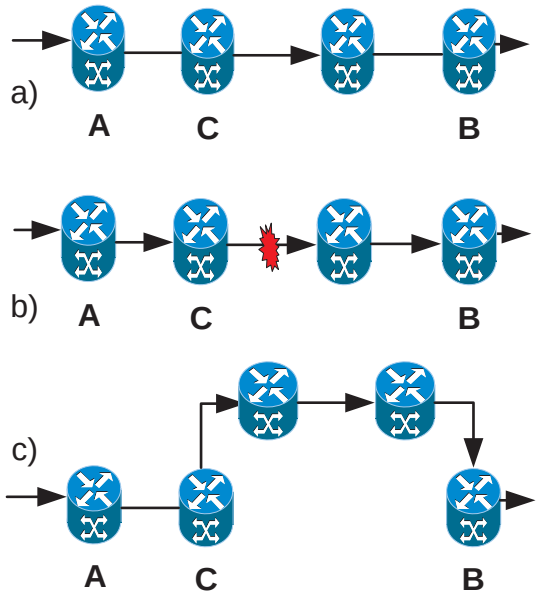

Figure 2.3 Dynamic rerouting steps, using local repair splicing technique

Stacking: In this case a single LSP is created to bypass the protected link; when a fault occurs the bypass LSP is a replacement for the faulty link. This LSP can be used as a hop by another LSP. This is done by pushing the bypass label onto the stack of labels for packets flowing on the rerouted LSP. Figure 2.4 illustrates the stacking repair technique within an MPLS domain.

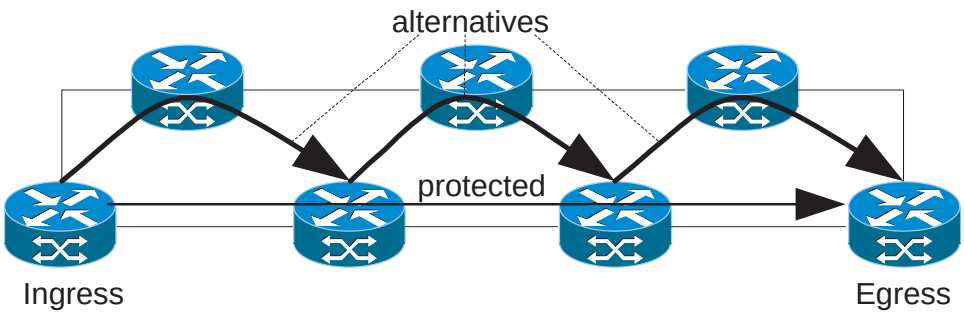

Figure 2.4 Local repair using stacking technique

Restoration and repair Method	Resource Requirement	Speed of Repair	Packet Loss	Packet Re-ordering	Protection Path (length)
Dyn. Local Repair	No	Slow	Minimum	Minimum	Might not be the SP available
Dyn. Global Repair	No	as above + FIS	High	Minimum	Path is shortest available
Fast re-routing Local	Yes, if not shared	Fast	Minimum	Minimum	May not be the optimal
Fast re-routing Global	As above	Fast, depends on FIS	High	Minimum	Better than the above
Fast re-routing with Reversing backup (Haskin's)	As above, plus backward LSP during recovery time	As above	Minimum	High	As above

Table 2.1 Comparison table for repair techniques, SP: shortest path, FIS: failure indication signal

If local repair is attempted to protect an entire LSP, each intermediate LSR must have the capability to initiate alternative, pre-established LSPs. This is because it is impossible to predict where failure may occur within an LSP. A very high cost has to be paid in terms of complex computations and extensive signaling required to establish alternative LSPs from each intermediate LSR to the egress LSR. For this reason, we have chosen the combination of local and global repair strategies with reversing backup (backward) for our mechanism. Our approach is similar to the one adopted in [HK00].

In table 2.1 we try to summarize the main aspects of different combination of restoration and repairing methods used to protect traffic from network failures.

2.3.4 Haskin's proposal

In Haskin's proposal [HK00] the authors present a method for setting up an alternative LSP to handle fast rerouting of traffic upon a single failure in the primary/protected LSP in an MPLS network. Since the objective of the proposed work is to provide a fast rerouting protection mechanism, the alternative LSPs are established prior to the occurrence of a failure.

For the correct operation of this proposal the complete path during the recovery period is composed of two portions: the path from the egress LSR to ingress LSR in the reverse direction of the primary/protected path (Backward LSP) and the alternative path from the ingress LSR to the egress LSR (Alternative LSP). The alternative LSP must be completely disjoint with the primary LSP (Fig 2.5a).

The main idea of this proposal is to reverse traffic at the point of failure of the protected LSP using the Backward LSP. This provides a quick restoration comparable to the 50 milliseconds provided by a SONET self-healing ring, and at the same time minimizes alternative path computation. Fast protection switching is achieved without signaling since the reversing decision is made using locally available information at the node that detects a downstream link failure (alert LSR).

In this scheme the alert LSR, reroutes the incoming traffic in the reverse direction of the protected path using the backward LSP (Figure 2.5b). When the redirected traffic reaches the ingress LSR, it is switched to the previously established alternative LSP. Furthermore, when the ingress LSR detects traffic in the reverse direction it switches the traffic entering the MPLS domain directly to the alternative LSP (Figure 2.5c). Note that until the ingress LSR receives the first packet from the backward LSP packets continue to be sent via the already broken primary/protected LSP (Figure 2.5b). These packets will experience a two-way delay while traversing the backwards loop from the ingress LSR to the last LSR at the point of failure (alert LSR). Another problem of this scheme is that as packets arriving from the reverse direction are mixed with incoming packets, this results in packet disordering through the alternative LSP

during the restoration period. Finally, the scheme also loses packets circulating in the failed link at the time of failure.

Figure 2.5 illustrates steps followed by Haskin's restoration scheme.

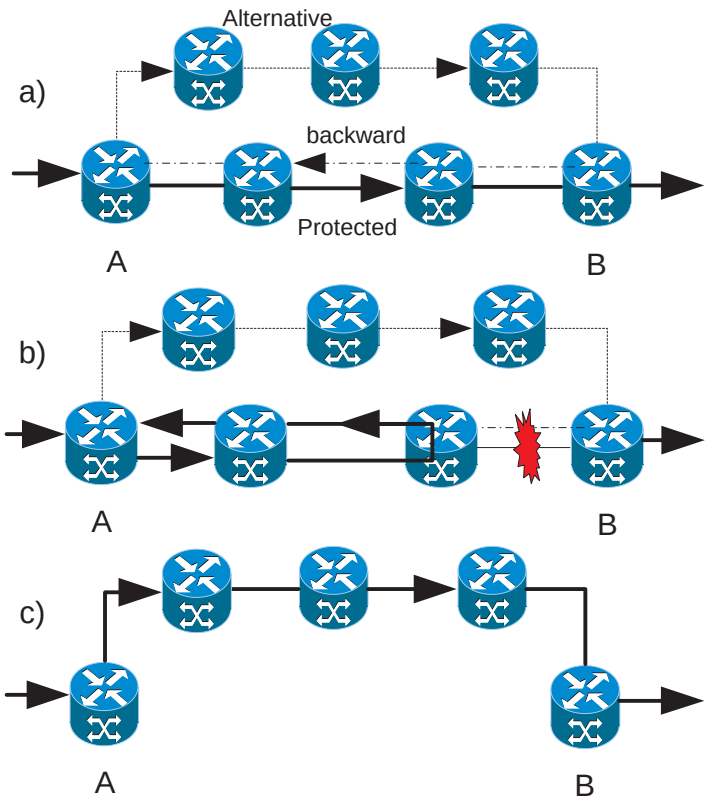

Figure 2.5 Haskin's scheme restoration process

2.3.5 Makam's Proposal

In this proposal [MSOH99] [OSMH01] [SH02] the authors consider the two recovery possibilities for the alternative LSP: pre-established (Figure 2.6) and dynamic recovery (Figure 2.7). The objective is to provide a path protection mechanism in MPLS networks. Unlike Haskin's proposal this scheme uses a fault notification mechanism (FIS) to convey information about the occurrence of a fault to a responsible node in

order to take the appropriate action (e.g., the ingress LSR is notified to switch traffic from the protected path to the alternative path).

Figure 2.6 illustrates steps followed by Makam's restoration scheme using fast rerouting.

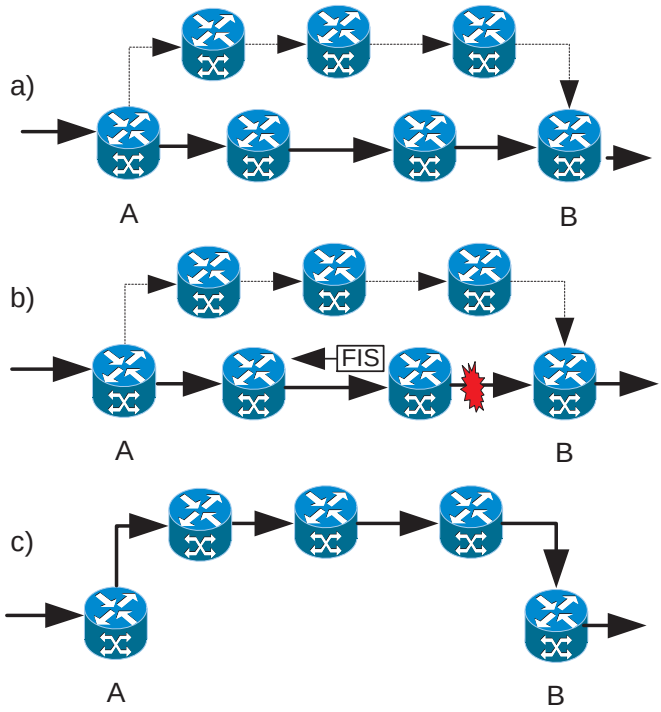

Figure 2.6 Makam's scheme using fast rerouting (preplanned)

When a link failure occurs on the protected path, the alert node signals the failure to the upstream nodes (i.e., the intermediate LSRs on a protected path between the ingress LSR and the alert LSR) as illustrated in Figure 2.6b and Figure 2.7b. The ingress LSR redirects the traffic over a *pre-established or pre-planned alternative LSP* (Fast rerouting method, Figure 2.6c) or *dynamically established alternative LSP* (rerouting method, Figure 2.7c) upon the reception of the failure notification signal.

In the case of using the pre-established alternative LSP, the traffic entering the domain is directly diverted to the pre-established alternative LSP by the ingress LSR after the arrival of the notification signal. This method provides better resource utilization than Haskin's scheme because the length of the protection path used during the recovery period is less than that of Haskin's proposal. However, the traffic that is in transit during the interval of time between the detection of the fault detected and the time the fault notification signal reaches the ingress LSR will be dropped by the alert LSR. Moreover, those packets that were circulating on the failed link at the time of the failure will also be lost.

When the dynamic method is applied, as it takes much longer to establish the alternative LSP, and the amount of dropped packets is larger than with the pre-established alternative LSP. Resource utilization is more efficient than the previously described scheme because updated network information is used. This scheme also provides more flexibility in the establishment of a new alternative LSP.

The main advantage of using a dynamic LSP is that an optimal alternative LSP may be established.

Figure 2.7 illustrates steps followed by Makam's restoration scheme using rerouting (Dynamic).

Table 2.2 shows the restoration and repairing method used by Haskin's, Makam's and the dynamic scheme (Figure 2.3).

2.4 PERFORMANCE EVALUATION METHODOLOGY

2.4.1 Simulation tools

The methodology used for performance evaluation in this thesis is a public domain network simulator version 2 (*ns-2*) originally from Lawrence Berkeley National Labo-

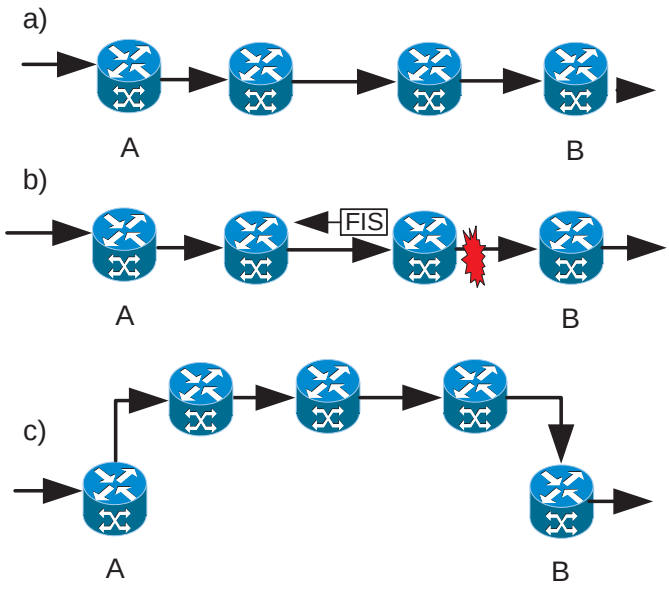

Figure 2.7 Makam's scheme using rerouting (dynamic)

	Haskin's scheme	Makam's scheme	Dynamic scheme
Restoration method	Fast Re-routing (Pre-planned)	Fast Rerouting or Rerouting	Rerouting (Dynamic)
Repairing method	Local	Global	Local

Table 2.2 Comparison of restoration and repairing methods for Haskin's, Makam's and Dynamic scheme

ratory (LBNL) [FVa][FVb] extended for MPLS networks called MPLS Network Simulator (MNS) contributed by Gaeil and Woojik [GW99][GW00][GW01a].

The *ns*-2 is considered the standard simulation tool widely used by the network research community to validate its new proposals. Therefore, the use of *ns*-2 as the evaluation tool has many advantages.

1. It is a well proved standard network simulation with sufficient documentation.
2. It is maintained and updated by contributions from many people from different network research groups.
3. The basic function and parameters in the simulator are calibrated properly. Therefore, the simulation results derived from different proposals using the same simulation conditions are feasible for evaluation. This allows easy and better comparison tools between different proposals for network researchers.

NS-2 is an event-driven simulator designed for IP based networks. In NS-2, a node consists of agents and classifiers. An agent is a sender/receiver object of protocol and a classifier is the object that is responsible for the packet classification used to forward packets to the next node. For the purpose of making a new MPLS node from an IP node, the authors introduce 'MPLS classifier' and 'LDP agent' into the IP node.

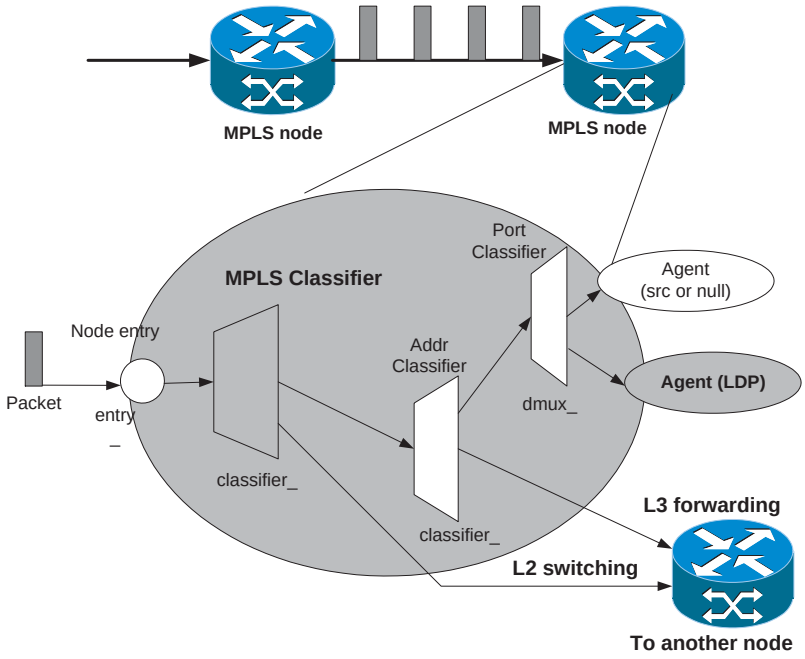

Figure 2.8 Architecture of MPLS node in MNS [GW99]

The simulated MPLS node handles the packets arriving in a three step process. First, it classifies them into labeled and unlabeled packets using the 'MPLS classifier'. Note that this principle is the same that the IP node uses to classify incoming packets into multicast and unicast using a "Multicast classifier". The MPLS classifier is responsible for the label swapping operation for labeled packets, and if it is an unlabeled packet but an LSP for the packet is prepared, the classifier executes a label push operation. Otherwise it sends the packet to the "Addr Classifier". Second, the Addr Classifier executes IP forwarding by examining the packet destination address. Third, if the next hop for the packet is itself, the packet is sent to "Port Classifier". Figure 2.8 shows the sequence of operations that an MPLS node performs on receiving a packet.

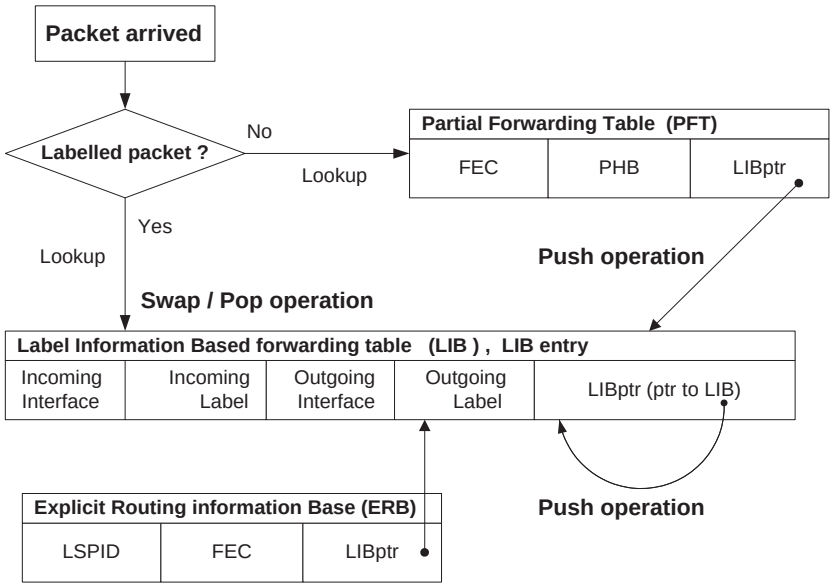

Figure 2.9 Entry tables in an MPLS node for MPLS packet switching

An MPLS node in MNS handles three information tables to forward packets using LSP: Partial Forwarding Table (PFT), Label Information Based forwarding table (LIB) and Explicit Routing information Base (ERB). PFT is a sub-set of the forwarding table and consists of FEC to NHLFE (FTN) mapping. The LIB table has information for LSPs, and ERB has information for Explicit Routing Label Switched

Path (ER-LSP). Figure2.9 shows the structure of these tables and the simple algorithm for forwarding packets [GW99].

Figure 2.10 illustrates the simple switchover mechanism used in MNS using the above tables when a link on the protected LSP fails. Note that the protected LSPs have a pre-established backup LSP using explicit routing.

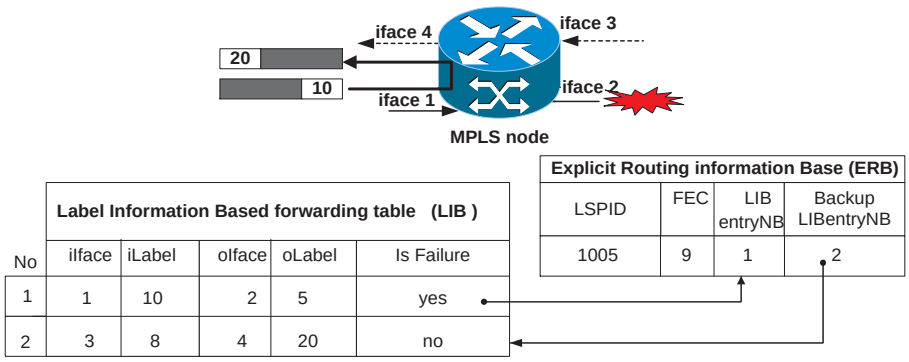

Figure 2.10 LSP restoration using backup LSP with switchover procedure

2.4.2 Performance criteria

Several criteria to compare the performance between different MPLS-based recovery schemes are defined in [SH02]. The most important are: packet loss, additive latency, re-ordering, recovery time, full restoration time, vulnerability, and quality of protection.

Packet loss: Recovery schemes may introduce packet loss during switchover to a recovery path. It is a critical parameter for a restoration mechanism. Throughput rates achieved for the service are seriously affected by packet losses. In real-time applications (e.g., VoIP, Multimedia, etc.) losses may interrupt the connection. Recovery schemes must guarantee minimal or no packet losses during the restoration period.

Latency: Latency represents the amount of time it takes a bit to traverse a network. The latency value is used as an indicator of the quality of the network connection: the lower the latency the better the connection. It is also referred as to end-to-end delay. For real-time applications, such as streaming video and audio, latency variation over time, or delay jitter, is also an important indicator of the network's quality.

Re-ordering of packets: The recovery mechanism may introduce packet disordering. The action of putting traffic back on a preferred path may introduce packet reordering by the ingress node when sending packets through an alternative LSP. This is also not desirable. While data transfers may handle disordered packets, streaming data usually do not.

Recovery time: The time required for an alternative path to be activated and begin carrying traffic after a fault. It is the time between the failure detection and the time when the packets start flowing through the alternative LSP.

Full restoration time: The time between the failure detection and the time all traffic is flowing through the alternative LSP.

Vulnerability: The time that the protected LSP is left unprotected (i.e., without backup) from possible network component failure. Once the alternative LSP becomes the primary LSP new alternative and backward LSPs should be established in order to protect it.

Quality of protection: Upon a failure the probability of a connection to survive the failure determins the quality of protection of the restoration scheme. The quality of protection range can be extended from relative to absolute. Relative survivability guarantee means that it is straightforward to assign different priorities to different connections and restore them based on their relative priority. Absolute means that the survivability of the protected traffic has explicit guarantees and therefore provides a better option for a service level agreement (SLA). The quality of protection of the protected LSP is absolute.

2.4.3 Simulation scenario

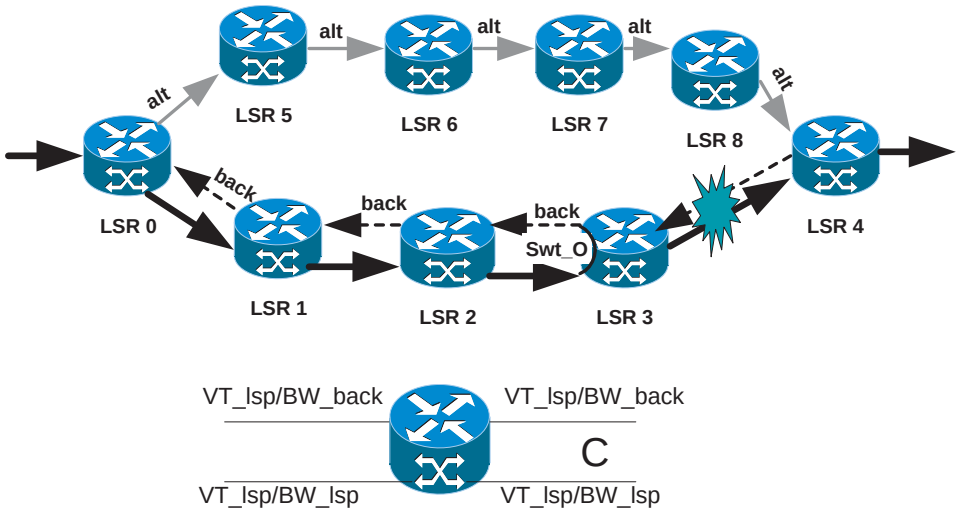

Figure 2.11 Simulation scenario

Figure 2.11 presents the basic simulation scenario used in this thesis, where C is the link capacity, B_{W_lsp} is the protected LSP bandwidth and V_{T_lsp} is aggregated protected flows. For a protected LSP, B_{W_back} is the backward LSP bandwidth and B_{W_alt} is the alternative LSP bandwidth.

The V_{T_lsp}, B_{W_lsp}, B_{W_back}, and B_{W_alt} are subject to:

$$V_{T\_lsp} \leq B_{W\_lsp} \tag{2.1}$$

$$B_{W\_back} \geq B_{W\_lsp} \geq V_{T\_lsp} \tag{2.2}$$

$$B_{W\_alt} \geq B_{W\_lsp} \geq V_{T\_lsp} \tag{2.3}$$

the worst case is when: $V_{T\_lsp} = B_{W\_lsp} = B_{W\_back} = B_{W\_alt}$.

In the simulations we vary the source rate, packet size, LSP length and the bandwidth of protected, backward and alternative LSPs to compare the performance for different restoration schemes.

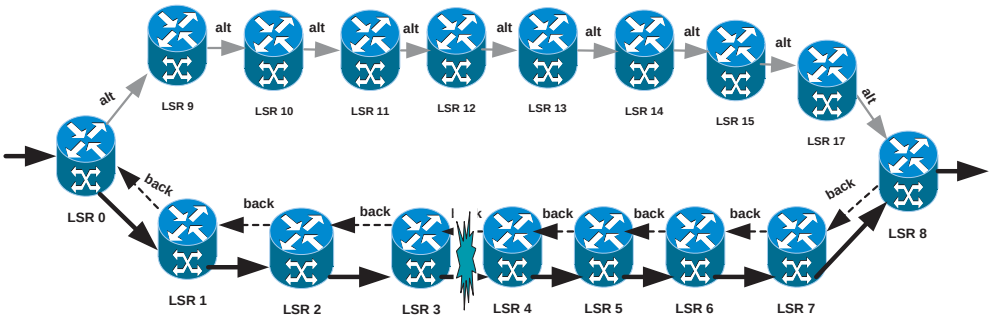

Figure 2.12 Network scenario

We use CBR traffic with a UDP agent generated by the network simulator NS-2 for the simulation. We use UDP traffic for our studies because the main interest is multimedia traffic for real-time requirements. We use CBR traffic due to the behavioral simplicity that it gives the simulation.

2.5 PERFORMANCE EVALUATION OF MPLS RECOVERY SCHEMES

The basic factors that affect the performance of the restoration mechanisms are packet loss, traffic recovery delay (Full Restoration Time) and packet disorder [BR02] [GJW02]. We use these performance measurement parameters to compare the above-mentioned proposals for MPLS restoration schemes for link/node failure. Other parameters will be considered later in other proposals.

Figures 2.13 and 2.14 present the comparison of the behavior of three approaches: Haskin's, Makam's pre-established, and classical dynamic using the local splicing technique (Figure 2.3). Results refer only to the restoration period and show % of packet loss and % of packets out of order due to the restoration mechanisms. The horizontal axis presents the place of the alert LSR within the protected LSP.

Performance evaluations based on the Figure 2.12 for these schemes. Figure 2.13 shows the comparison result for packet losses.

2.5.1 Packet losses

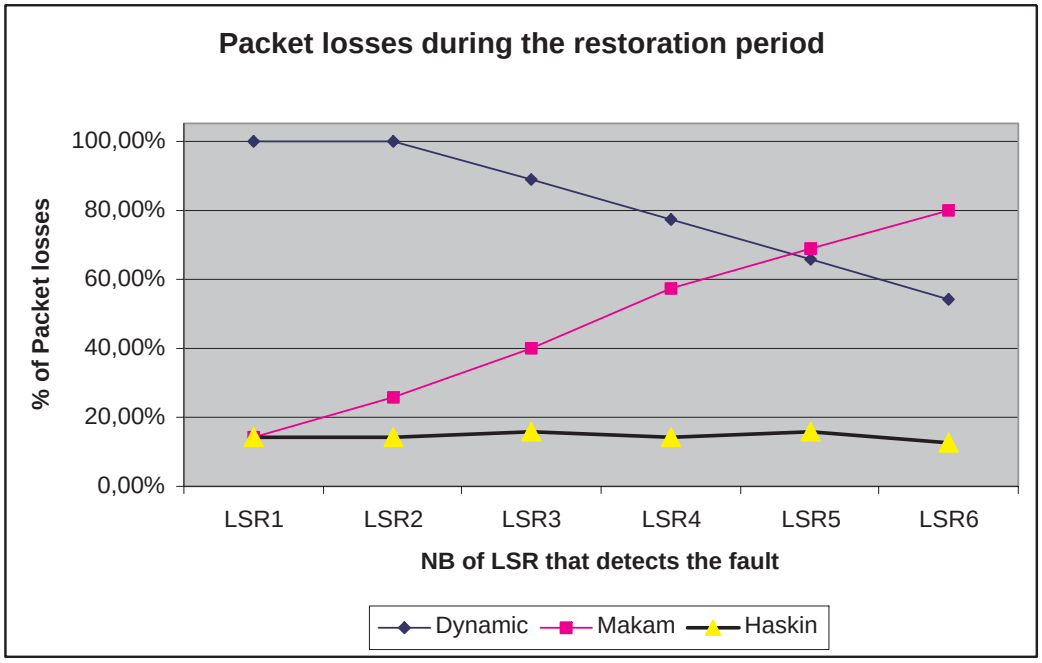

Figure 2.13 Packet loss performance comparison between path protection/restoration schemes in MPLS network

With the dynamic scheme packet losses increase in proportion to the distance between the alert LSR and the egress LSR, because of the set up time of an alternative LSP.

In Makam's scheme [OSMH01] packet losses increase in proportion to the distance between ingress LSR and an alert LSR that detects the failure, because of the delivery time of the fault notification message.

Haskin's scheme [HK00] only loses packets on the failed link or on the link adjacent to the failed LSR.

2.5.2 Packet Disorder

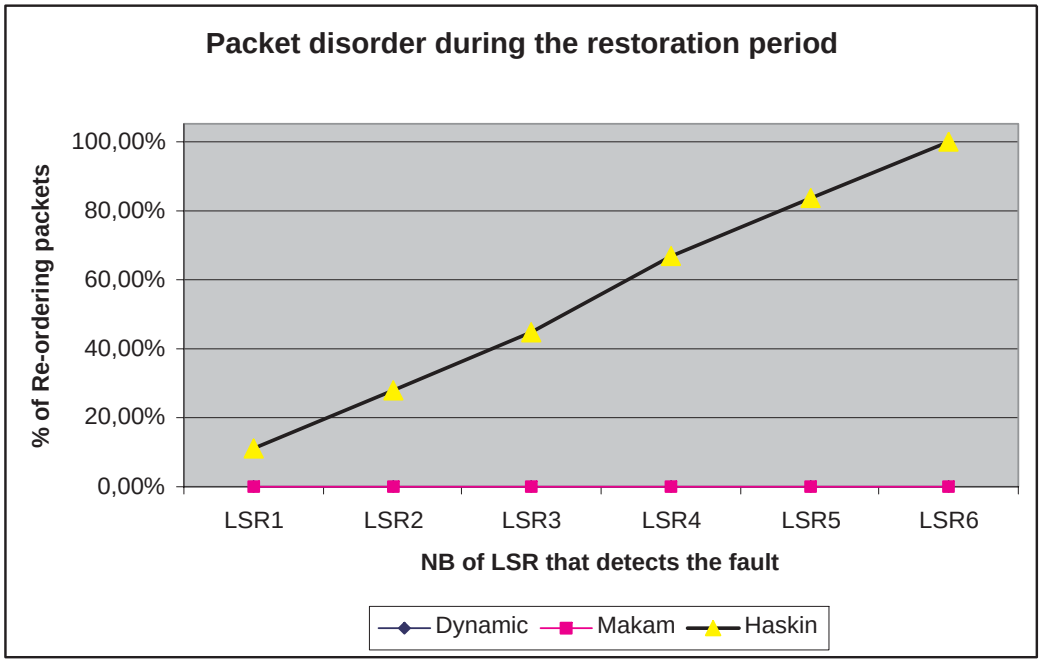

Figure 2.14 Packet disorder performance comparison between path protection/restoration schemes in MPLS network

Figure 2.14 presents the packet disorder result for the these schemes. In Haskin's scheme packet disorder increases in proportion to the distance between ingress LSR and the alert LSR. Note that the packet disorder that we consider here is the disorder produced during the restoration period which does not include the disorder produced by the retransmission of lost packets by a high level protocol (i.e., TCP).

Makam's and dynamic schemes do not introduce packet disorder but cause more packet losses.

Based on the discussion in this chapter we restrict ourselves to the combination of local repair action, reverse, and global restoration schemes with preplanned alternative LSPs. We use local repair action because of its advantage in terms of speed for switchover of traffic from the protected path to the backup path compared to global

repair action. Note that the choice of local restoration may lead to ta higher use of resources due to the length of the resulting protection path. For this reason we use the global restoration scheme, which provides the optimal available path (Table 2.1). We chose the reversing mode because, like local restoration it reports the minimum packet loss. However, unlike local restoration, in the reversing mode the resources are used only during the relatively short recovery period. Note that the reserved resources in the reverse backup path (backward LSP) can be used by low priority traffic. We also exclude the dual-fed path protection technique known as 1+1 because in this system only the transmitting node and receiving node affect recovery, and it consumes excessive network resources.

2.6 MOTIVATION

The effects of packet losses, packet delay and packet reordering on QoS provision are well known phenomena. These parameters are closely related. Chapter 5 provides some detailed explanations of these phenomena.

The main motivation of this thesis is to overcome the drawbacks of the previously proposed schemes for the restoration mechanism in MPLS networks during link/node failure or congestion. We focus mainly on the above problems: packet loss, packet delay and packet disorder.

Proposals in the following two chapters try to improve the performance of recovery schemes on packet loss, packet delay and packet disorder.

3

FAST REROUTING MECHANISM

3.1 INTRODUCTION

Given that network topologies are never stable over time, rapid response to link failures and/or congestion by means of rerouting is critical. There are two basic methods for protected LSP recovery: (1) Dynamic rerouting and (2) Fast rerouting [SH02]. When the primary label switched path (LSP) encounters a problem due to link or node failure, the data that travels it needs to be rerouted over an alternative LSP. This is equivalent to using a new LSP to carry the data. The alternative LSP can be established after a protected LSP failure is detected, or it can be established beforehand in order to reduce the LSP switchover time [SH02]. The former option has a slow response in the rerouting function. The latter has a much better response time. In our proposal we use the fast rerouting technique with pre-established alternative LSPs to protect the packets travelling in the protected LSP.

Fast rerouting uses pre-established alternative LSPs. We focus on improving current mechanisms for fast rerouting proposed by Haskin. The objective is to improve packet delay during the restoration period and minimize packet disorder.

In [HK00], when a failure is detected in the protected LSP, the traffic is sent backwards to the ingress LSR using a pre-established LSP (*backward LSP*). When the ingress LSR receives the first packet from the backward LSP, the traffic flow for the protected LSP is redirected to the alternative LSP that was established previously between ingress and egress LSRs following a global repair strategy (Figure 3.1).

In Figure 3.1, the ingress and egress nodes are LSR0 and LSR4 respectively. The protected LSP is formed by the LSR nodes 0, 1, 2, 3 and 4. If a link failure is detected by LSR3, as shown in the figure, the backward LSP will include the nodes 3, 2, 1 and 0. The *alternative LSP* will be formed by the LSR nodes 0, 5, 6, 7, 8 and 4.

As soon as an LSR node belonging to the protected LSP detects a fault, a switchover is established and packets are sent back through the backward LSP (Figure 3.1). The first packet that is sent back is used as a fault detect notification. Until the fault notification arrives at the ingress LSR, packets are sent via the already broken

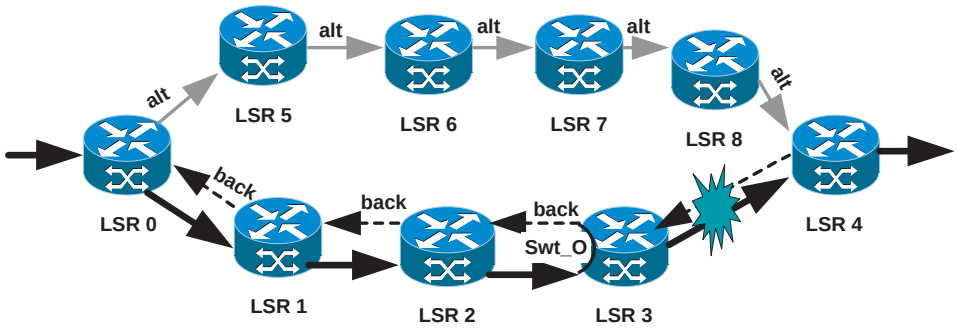

Figure 3.1 Scheme for alternative LSP to handle fast rerouting during the restoration period (back: backward LSP; alt: alternative LSP)

protected LSP. These packets will experience a two-way delay while traversing the backwards loop from the ingress LSR to the alert LSR and back to ingress LSR.

The restoration process ends when the last packet the ingress LSR sent through the already broken protected LSP comes back through the backward LSP. Then the protected and backward LSP are released.

As we explain in Section 2.3.4, an important drawback in this scheme is the delay involved in detecting the first packet that is sent back from the alert LSR to the ingress LSR, plus the delay for the subsequent packets sent along the broken LSP to return to the ingress LSR. Further, this approach also introduces data packet disordering during the LSP rerouting process. This is because once a fault is detected, the ingress node merges the newly incoming traffic and the packets coming back from the point of failure when sending them along the alternative LSP. The problem of this scheme concerning packet loss is left to be addressed in Chapter 4.

3.2 PROPOSED MECHANISM

Our proposal follows the principle described in [HK00] for setting both an alternative LSP and a backward LSP. In this section we address the drawbacks of Haskin's [HK00] proposal with respect to round-trip delay and packet disorder during restoration. In the description, upstream and downstream refer to the direction of traffic in the protected LSP.

In our proposal, when a fault is detected by an LSR, a switchover procedure is initiated and the packets are sent back via the backward LSP. As soon as each upstream node on the backward LSP detects these packets, they start storing the incoming packets (on the primary or protected path) in a local buffer. This avoids the unnecessary forwarding of packets along the broken LSP. Furthermore, the last packet forwarded before initiating storage is tagged in order to be identified on its way back. By doing this, we are able to preserve the ordering of packets when it is time for each

intermediate node to send back its stored packets. We use one of the *Exp* field bits of the MPLS label stack [RTF+01] (see Figure 1.3) for the purpose of tagging and thereby avoid any overheads.

Each LSR on the backward LSP successively sends back its stored packets when it receives the tagged packet. When all packets are returned to the ingress LSR (i.e., the ingress LSR receives its tagged packet) and have been rerouted to the alternative LSP, the restoration period terminates. The packets stored during this time in the ingress LSR, along with all new incoming packets are now sent via the alternative LSP. Note that global ordering of packets is preserved during the whole process.

The detailed algorithm along with the state machine diagram is presented in the next section.

3.3 ALGORITHM DESCRIPTION

Before getting into the details of our algorithm, it is important to take into account the modification we made in the label information base-forwarding table (LIB). We introduce a new field called "*STATUS*" in the LIB, and manage five states in this field. They are: NORMAL, FAULT_DETECT, ALTERNATIVE_DETECT, STORE_BUFFER and SEND_BUFFER.

NORMAL: This state corresponds to the normal operation condition. It means no fault is detected on the protected LSP: the LSR continues working in the normal condition.

FAULT_DETECT: As the name implies, this is the state to indicate the condition of a faulty link. The node becomes an alert LSR. When a tagged packet is forwarded through the backward LSP, or after a certain time depending on the implementation, the LSR removes the LIB entry.

ALTERNATE_DETECT: This state is in charge of notifying the incoming protected LSP packets of the failure after receiving a packet from the backward LSP. It waits for the first packet coming through the protected LSP, which will be tagged and transmitted.

STORE_BUFFER: This state is in charge of indicating the need to store packets travelling in the protected LSP after detecting the presence of packets through the backward LSP and sending the tagged packet to downstream LSRs. This avoids the unnecessary trip of packets downstream and back again.

SEND_BUFFER: The state in which the stored packets (i.e., packets stored during the STORE_BUFFER time) will be drained from the buffer to the alternative or backward LSP (if it is the ingress LSR or an intermediate LSR, respectively). This state is activated when the tagged packet is received at the ingress or at each intermediate LSR through the backward LSP.

Figure 3.2 presents the state machine diagram of the proposed algorithm. The ingress LSR forwards packets to the alternative LSP while the intermediate LSR forwards packets through the backward LSP. Though the state machine diagram by itself is a formal description, a detailed explanation of the process follows. Given this brief functional explanation of each state, we proceed to describe the whole algorithm of our proposal.

Note that traffic belonging to other LSPs going through the broken link is lost. Only protected LSP traffic is switched-over.

Once a failure along the protected LSP is detected, the protected LSR that detects the fault (alert LSR) performs the switchover procedure (LSR3 in Figure 3.3). This procedure consists of a simple label swapping operation from the protected LSP to the backward LSP for all packets with a label corresponding to the protected LSP. The link status of the label information base forwarding table (LIB) of this LSR for the protected LSP is changed from NORMAL to FAULT_DETECT (Figure 3.2).

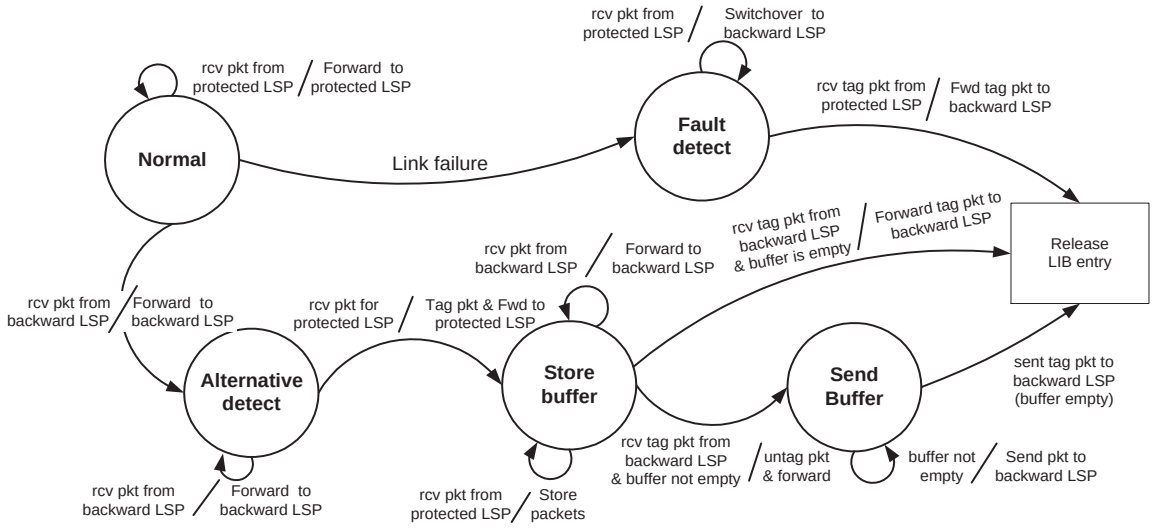

Figure 3.2 State machine diagram for intermediate LSRs

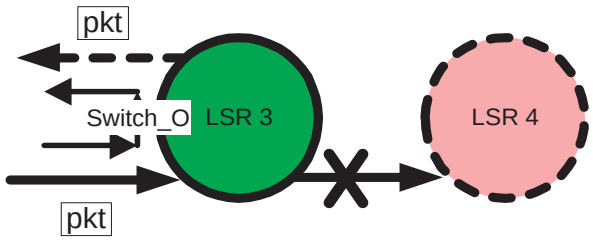

Figure 3.3 FAULT_DETECT and Switchover

The intermediate upstream LSR, in this case LSR2 (Figure 3.4 and Figure 3.1) receives these reversed packets from LSR3 through the backward LSP. When it receives the first packet through the backward LSP, it changes the link status of the LIB entry of the protected LSP corresponding to this backward LSP to ALTERNATIVE_DETECT (Figure 3.2). Then, the first packet received from the protected LSP sees this entry as ALTERNATIVE_DETECT. This indicates that there is a link problem somewhere in the protected LSP. This packet must then be tagged as the last packet from this LSR (LSR2) and forwarded normally downstream and the LIB entry status must be changed from ALTERNATIVE_DETECT to STORE_BUFFER

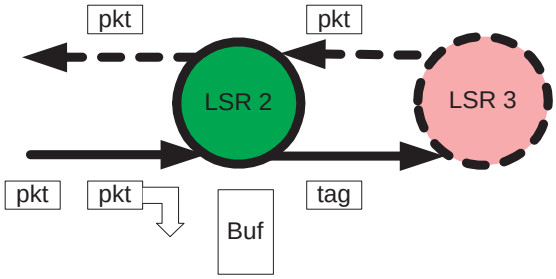

Figure 3.4 Intermediate LSR ALTERNATIVE_DETECT, Tag and STORE_BUFFER

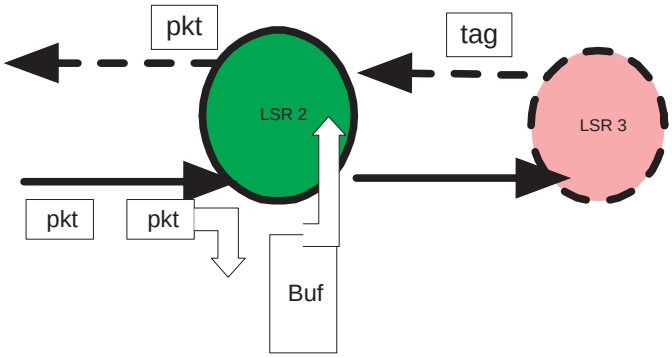

Figure 3.5 In intermediate LSR Tagged packet received and SEND_BUFFR

(Figure 3.2 and Figure 3.4). The next packets in the protected LSP will be stored in the buffer because they will find the link status in LIB as STORE_BUFFER. This continues until the tagged packet is received back through backward LSP.

When an LSR receives a packet from the backward LSP it checks if the received packet is a normal backward packet or a *Tagged* packet. The tagged packet received through the backward LSP is used as a trigger to perform the drain of the stored packets from the local buffer. The LSR has to check if the tag bit of the packet received from the backward LSP is set (1) or not (0). If the comparison result is false (i.e., the tag bit of the packet is set to 0 -normal backward packet-) the packet will

be forwarded using the normal swapping operation. Otherwise, the LSR knows that no more packets are expected from the backward LSP.

At this time, depending on the local buffer condition, the LSR takes one of the following two actions:

i) if the buffer is empty the LSR forwards the tagged packet on the backward LSP without any change in the tag bit. Note that the upstream LSR sends at least one tagged packet to the downstream LSR through the protected LSP after receiving the first packet from the backward LSP, and waits for this tagged packet to return through the backward LSP. Buffer empty means the unique packet sent by the upstream LSR is this tagged packet. For this reason it must be sent directly to the upstream node. Then the LIB entry from the LIB table is released.

ii) if the buffer is not empty the tag bit in the label must be disabled (set to 0) and the packet is sent according to the label swapping result as a normal packet. Moreover, it changes the status from STORE_BUFFER to SEND_BUFFER (Figure 3.5), and then when the buffer is empty, it releases the LIB entry from the LIB table Figure 3.2. Note that SEND_BUFFER finishes its process when it sends the tagged packet through the backward LSP. With this condition, no more packets on the protected LSP can use the output interface corresponding to this LSP before the fault was detected. Finally, the label associated with the protected LSP is removed. This process is repeated at every LSR until reaching the ingress LSR. In the case of the backward LSP, as it can carry other traffic on it (from egress to ingress) the LIB entry release process is done by the normal release procedure. The only thing needed is to release its resources reserved for the protected LSP.

The ingress LSR, unlike the intermediate LSRs, has the responsibility to divert or detour the incoming traffic (i.e., traffic entering the MPLS domain) from the failed primary or protected LSP to the previously established alternative LSP from ingress LSR to egress LSR (end-to-end in the MPLS domain) when it receives the tagged packet. While the intermediate LSR returns the traffic to the backward LSP, when

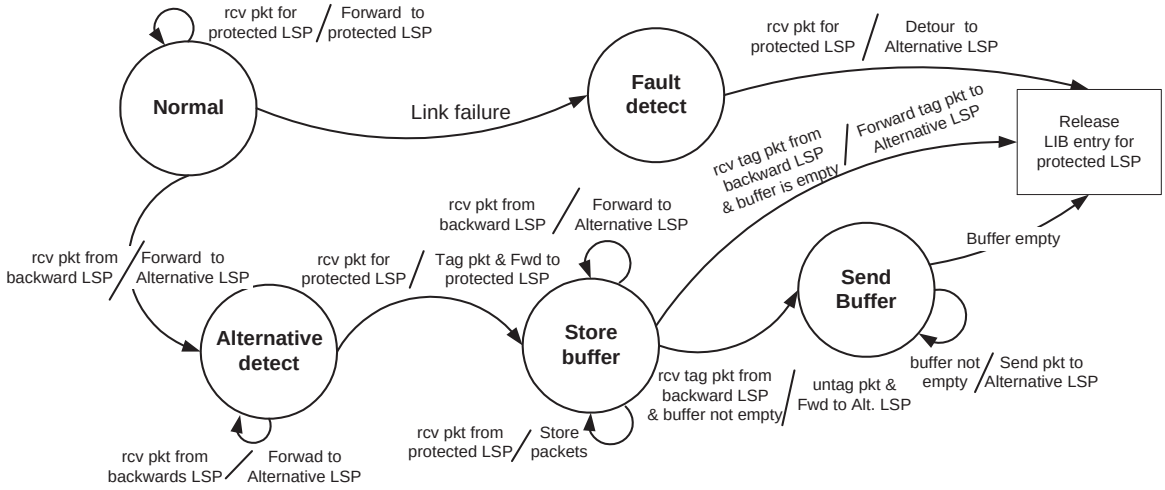

Figure 3.6 State machine diagram for ingress LSR

the ingress LSR receives the tagged packet it drains all its stored packets like any intermediate LSR and when it finishes, starts redirecting the incoming traffic directly to the alternative LSP. Figures 3.7 and 3.8 show the operation at the ingress LSR. When the incoming traffic is transited to the alternative LSP without passing through the buffer, the full restoration process terminates Figure 3.8.

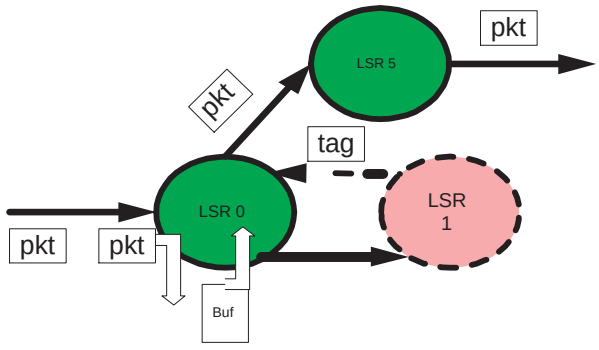

Figure 3.7 In ingress LSR Tagged packet received and SEND_BUFFER

Our proposal avoids sending packets downstream once any intermediate LSR between the ingress LSR and the point of failure detects packets on the backward LSP. This

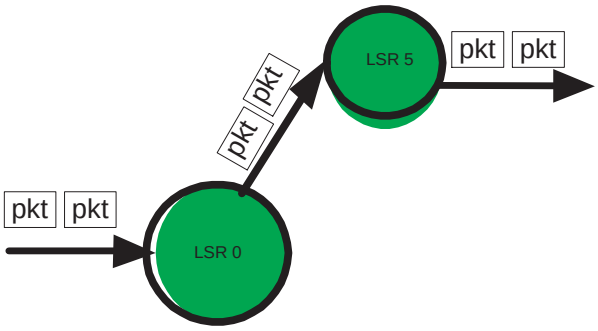

Figure 3.8 Restoration period terminates

reduces considerably the average delay of packets travelling in the protected LSP during the detection of the fault in a distant LSR.

3.3.1 Description of LIB table management

In Figure 3.9 we give a graphical description of the sequence of changes in the label information based forwarding table (LIB) during the recovery period including the field added by our proposal, link status. As you can see, for the purpose of simplicity after receiving the tagged packet we remove the entry corresponding to the backward LSP. This is because we assume that the protected and the backward LSPs belong to the same physical link, or that the backward LSP does not carry other traffic using this node as a merging point. When this is used for other traffic from the egress LSR to the ingress LSR using another physical link or a merging point for other LDP peers, or simply for traffic reverting purposes, it is left for the normal LSP release procedure and the mechanism can only decide whether or not to release the reserved bandwidth for the protected LSP depending on the conditions. Note the link status for this entry remains unchanged (normal).

Although the case when the ingress LSR (LSR0) as alert LSR detecting the failure is not present explicitly, it is easy to infer from Figure 3.9. If the ingress LSR detects the failure, it changes the link state for the protected LSP (LIB_entry 2) to fault_detect,

drains the buffer, and detours the traffic from interface 1 to interface 4 by pushing label 9 on to the packet (LIB_entry 1).

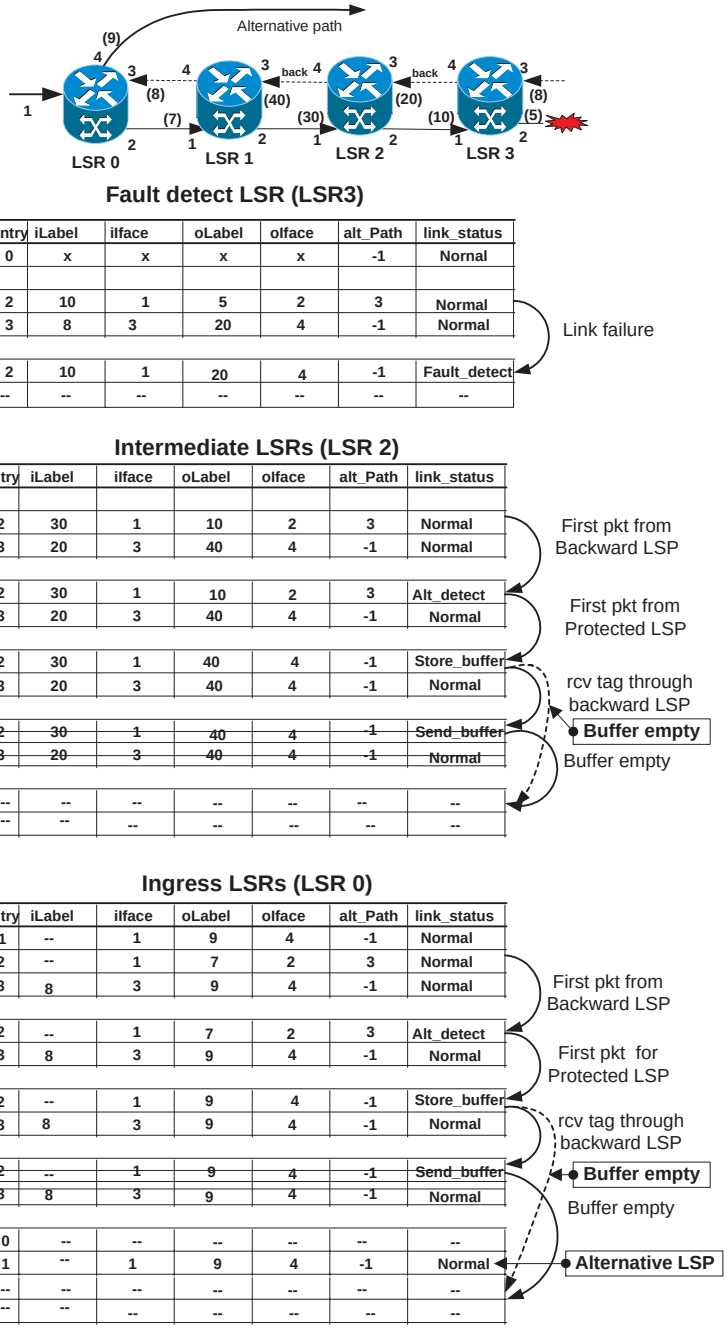

Figure 3.9 LIB entry (label forwarding table), we assume the backward LSP is not carrying other traffic

3.4 RESULTS

The simulated scenario is the one shown in Figure 3.1. The simple network topology with a protected LSP and a pre-established end-to-end alternative LSP is used. We extend the simple network topology for different numbers of intermediate LSRs in the protected LSP, yielding different sizes of LSPs (i.e., LSPs with 5 (e.g., Figure 4.1), 10 and 15 LSRs and with alert LSR at 3rd (Figure 4.1), 8th and 13th respectively) to analyze and compare the behavior of both proposals.

We modified part of the MNS source code to satisfy our particular requirement for the simulation of both mechanisms (ours and Haskin's [HK00]). Note that the simulation platform for these proposals was the same in order to be able to compare the simulation results. We compared our results for the disordered and dropped packets during path restoration with the ones published in [GW01b], thus validating our modified simulator.

In order to compare the results we use CBR traffic flow and a UDP agent with the following characteristics: packet size = 1600 bits, source rate= 400Kbps, burst time=0 and idle time =0, $T_{prop} = 10msec$, and $B_{W\_lsp} = B_{W\_back} = B_{W\_atl} = 1Mbps$ as defaults.

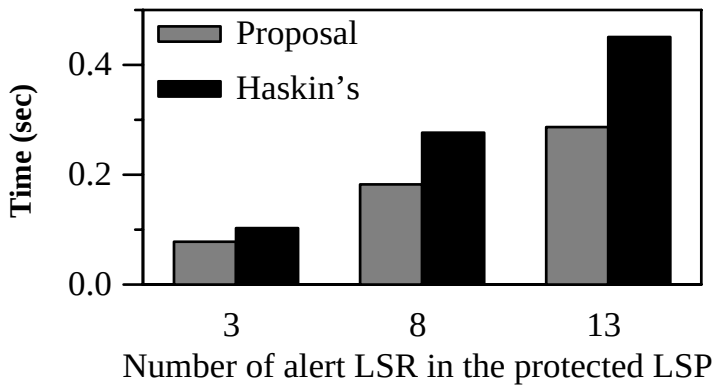

Figure 3.10 Restoration time to alternative LSP

Figure 3.10 shows the overall restoration period for both proposals for different positions of the alert LSR (number of the LSR) within the protected LSP. Note that the position of the alert LSR coincides with the number assigned to the LSR on the protected LSP. We assume the worst case in the sense that the failure occurs in the last link. Time is computed from the detection of the fault until the protected LSP is completely eliminated. The proposed mechanism provides a significant improvement of the path restoration period. Time reductions of 24.12%, 34.05% and 36.37% for the 3rd, 8th, and 13th alert LSRs on the protected LSP respectively are achieved.

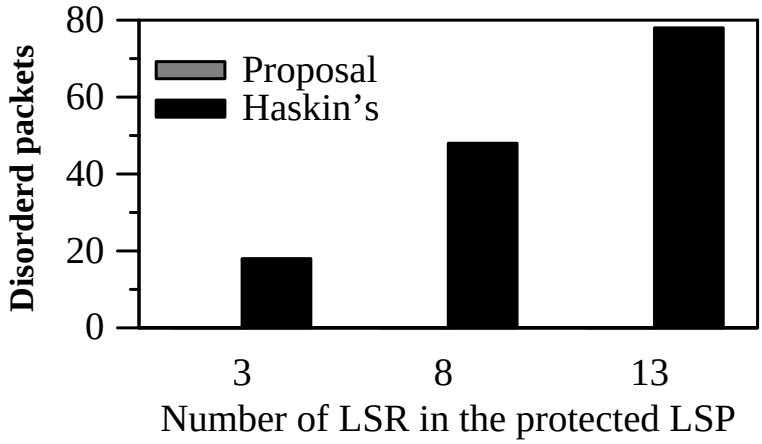

Figure 3.11 Number of disordered packets

Figure 3.11 confirms that the proposed mechanism avoids packet disordering while the restoration is in process.

In Figure 3.12 we can observe the simulation results for the restoration time and different bandwidths given the same traffic. The bigger the BW is, the faster the transmission, and smaller the slope of the line. That means shorter recovery periods are obtained with a faster LSP.

Figure 3.13 varies the packet size for a given bandwidth (5Mbps). In both cases, the number of intermediate LSRs were varied from 1 to 7. As can be seen from both figures, the reduction in restoration time is significantly better for the proposed mechanism for longer protected LSPs (i.e., LSPs with a greater number of nodes).

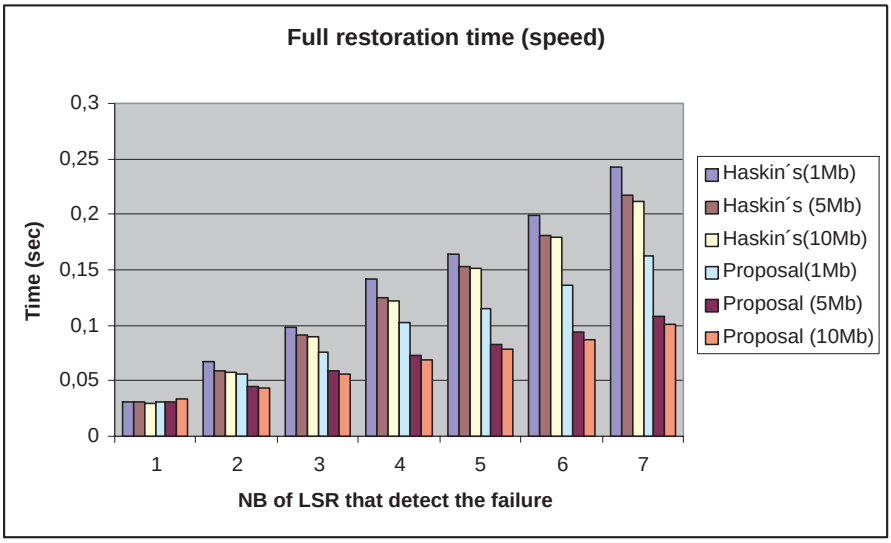

Figure 3.12 Restoration delay for 1600 bits packet size

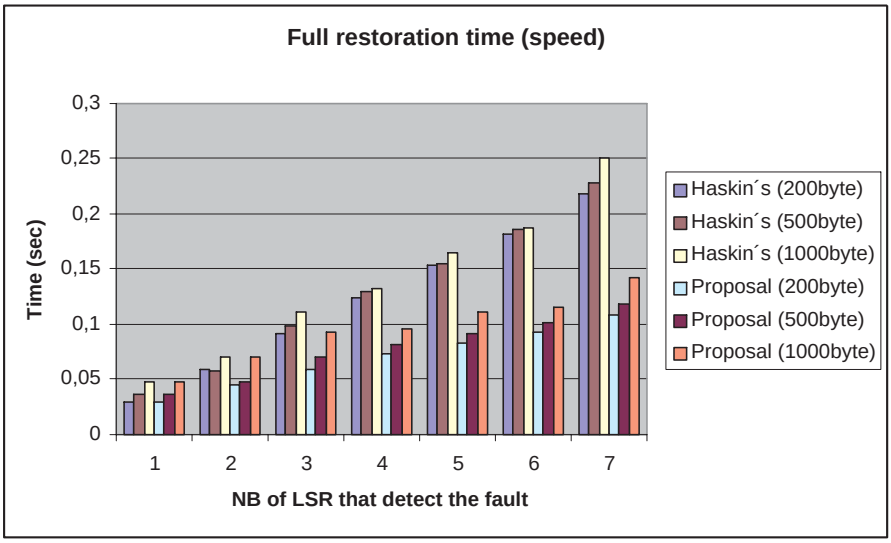

Figure 3.13 Restoration delay for 5Mbps LSP

Shortening the restoration period and the average packet delay during restoration, together with preserving packet sequence, minimizes the effect of a fault and leads to an improvement in the end-to-end performance.

3.5 SUMMARY

This chapter has presented a mechanism to perform fast rerouting of traffic in MPLS networks. We proposed a method to avoid packet disorder and improve the packet average delay time during the restoration period.

A decentralized mechanism involving all LSR from the ingress LSR to the alert LSR is described using a state diagram. Implementation details are also considered in the proposal.

In summary our proposal has the following advantages:

1. Improves the average latency (average packet delay).
2. Avoids packet disorder.
3. Improves end-to-end performance (overall performance).
4. Has a shorter restoration period than Haskin's proposal (i.e., faster network resources release).

In addition to recovery from failures, the proposed mechanism can be used for quality of service (QoS) provision. Once a given LSR detects congestion or a situation that leads to a Service Level Agreement (SLA) or QoS agreement being violated, it may start a fast reroute of a protected LSP that shares the link.

An LSP rerouted due to congestion may experience a slight increase in delay for a short period but no packets will be lost or disordered. Unlike the problem of failure in the link or node, the congestion problem gives administrators time to maneuver the rerouting of packets towards the alternative path. To extend the proposed mechanism to the congestion problem only a guarantee that the LSR is aware of the congestion is needed - just as in the case of a link fault. If this condition is satisfied, the flow can be diverted to the alternative path during a congestion situation.

4

RELIABLE AND FAST REROUTING (RFR)

4.1 INTRODUCTION

Fast rerouting has been recognized as a key component of providing service continuity to end users. We focus on improving current mechanisms for *Reliable and Fast Rerouting* (RFR). Given the function of fast rerouting mechanisms in the previous chapter, it is straightforward to introduce modifications to yield the reliable and fast rerouting mechanism.

In the proposal of Chapter 3, we were able to significantly reduce average delay due to path restoration while eliminating packet disorder for traffic in MPLS networks for a protected LSP. However, critical services (e.g. important traffic from premium customers) will be affected by packet losses and, for TCP traffic, lost packets trigger retransmission requests; hence the gains due to the decrease in restoration time may become negligible. As a consequence, poor performance and degraded service

delivery will be experienced and QoS parameters will be seriously affected during the restoration period.

The main factors that affect the performance of fast rerouting mechanisms are packet loss, traffic recovery delay (Full Restoration Time) and packet disorder. Our previous work has addressed the last two factors. Up to now, packet loss due to node or link failure was considered "inevitable" [SH02][BR02]. It has always been assumed that the transport layer would somehow take care of the retransmission of lost packets - eventually. It is for this reason, we believe, that there has not been any previous work aimed at eliminating packet loss. We have observed that the retransmission process due to packet loss significantly affects the throughput of TCP traffic due to the startup behavior (slow-start) of TCP. This point is briefly addressed in Chapter 5.

In this chapter we propose RFR, a novel recovery algorithm with small local buffers in each LSR node within the protected path in order to eliminate both *packet loss* due to link/node failure and *packet disorder* during the restoration period. This results in a significant throughput improvement for premium traffic.

It is important to note that the objective of this study is to provide and guarantee QoS for critical traffic carried by protected LSPs in MPLS networks and that not all LSPs are protected.

4.2 PROPOSED MECHANISM

The proposed mechanism is based on the mechanism already proposed in Chapter 3. We use the same figure to describe this proposal. In Figure 4.1, the ingress and egress nodes respectively are LSR0 and LSR4. The protected LSP is formed by the LSR nodes 0,1,2,3 and 4. If a link failure is detected by LSR3 - as shown in the figure, the path back to the ingress LSR will consist of the nodes 3,2,1 and 0 (*backward LSP*). The *alternative LSP* will be formed by the LSR nodes 0,5,6,7,8 and 4. We assume

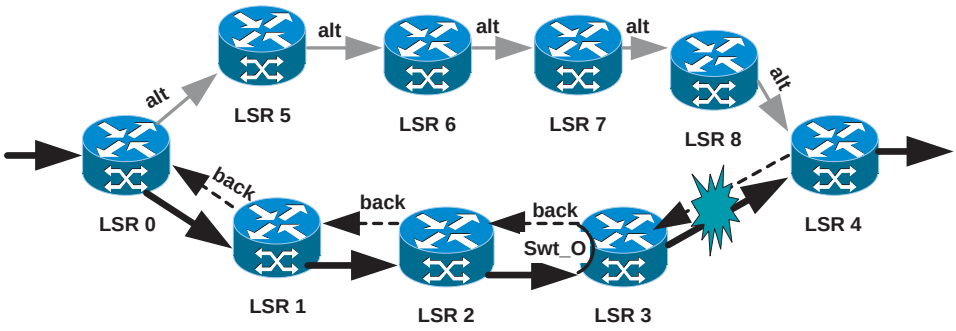

Figure 4.1 Simulation scenario

that the backward and alternative LSPs have already been set-up. As soon as an LSR node belonging to the protected LSP detects a fault, a switchover is established and packets are sent back through the newly activated *backward LSP*. The first packet that is sent back is used as a fault-detect notification.

In our proposal, each LSR in the protected path has a local buffer into which a copy of the incoming packet is saved while it is being forwarded to the next LSR along the protected path. The maximum size this buffer needs to be is about twice the number of packets that can circulate in a given link of the protected LSP. This is because the failure can occur either on a link or at a node. If the link fails, only the packets occupying the link from LSR3 to LSR4 during the failure would potentially be lost (Figure 4.1). If node LSR3 fails, packets on two links will have to be recovered.

There are two possible modes to store the incoming protected packets to the local buffer during the NORMAL condition. The first, called the non-swapped mode, is to store the protected packets before the swapping procedure to the backward/alternative LSP is done. This consists of a simple copy of packets to the local buffer as the packets are received by an LSR. The second is the swapped mode, in which the LSR stores the protected packets to the local buffer after executing the swapping procedure to the backward/alternative LSP. Both modes work well. The main differences between these approaches are the delay and the additional process overhead.

In the non-swapped mode once the fault is detected on the protected LSP, the LSR takes the packet from the local buffer and looks at the packet header ("shim" header or MPLS header) and then proceeds to swap it for the corresponding output label and changes the output interface. Note that this is the second time that the LSR looks at this header. The first was when the packet arrived at the LSR for the first time to be copied to the local buffer. This method introduces delays, processing overhead and additional CPU requirements. On the other hand, in the swapped mode the LSR sends the packets from local buffer directly to the output interface, as it did the swapping process before while in the normal condition, providing better performance than the non-swapped mode. For this reason our proposal uses the swapped mode.

4.2.1 Behavior of the Node that detects the failure

When a fault is detected by an LSR, a switchover procedure is initiated immediately (assuming that the fault-detection-time is zero) and all the packets in its buffer are drained and sent back via the backward LSP. Any subsequent packet coming in on the protected LSP is also sent back. The switchover consists of a simple label swapping operation from the protected LSP to the backward LSP. Note that this node has *copies of packets* that were dropped from the faulty link/node and hence there is *no packet loss*.

4.2.2 Behavior of all other nodes on the backward LSP

As soon as each node of the *backward LSP* detects the first packet coming back (sign of fault or problem downstream), it forwards this packet along the *backward LSP* and invalidates all data that are stored in its buffer for recovery of data from a possible link/node failure associated with this output interface. The next packet coming in from the upstream LSR of the protected LSP will be tagged and forwarded to the downstream LSR via the protected LSP. All subsequent packets that arrive at this node or LSR along the protected path are stored in its buffer without being forwarded (Chapter 3). This contributes significantly to the reduction of the average

packet delay because it avoids the circulation of packets along the loop formed by the already broken protected LSP and the backward LSP.

4.2.3 Role of tagging in eliminating disorder of packets

When a node detects the packet it tagged (the last packet sent downstream before starting to store incoming packets) coming along the *backward LSP*, it knows that all downstream packets have been drained and that it must now send back all its buffered packets. By doing this, it is able to preserve the ordering of packets. Using one of the *Exp* field bits of the MPLS label stack [RTF+01] for the purpose of tagging avoids any overheads.

Each LSR along the *backward LSP* successively sends back its stored packets when it receives its tagged packet. Note that the node responsible for removing the tag is the same node (LSR) which tagged it. When all packets return to the ingress LSR (i.e., the ingress LSR receives its tagged packet) and have been rerouted to the *alternative LSP*, the restoration period terminates. The packets stored during this time in the ingress LSR, along with all new incoming packets (from the source) are now sent via the alternative LSP. Note that at the end of the whole process, global ordering of packets is preserved, packet loss has been eliminated, and the proposal has a shorter restoration period than Haskin's.

4.3 ALGORITHM DESCRIPTION

Figure 4.2 presents the state diagram of the proposed algorithm (RFR). Though the state diagram by itself is a formal description, a detailed explanation of the process follows. We introduce a new field in the label information based-forwarding table (LIB) called status (link state). Five link state identifiers are defined for a protected LSP: NORMAL, FAULT DETECT, ALTERNATE DETECT, STORE BUFFER and SEND BUFFER.

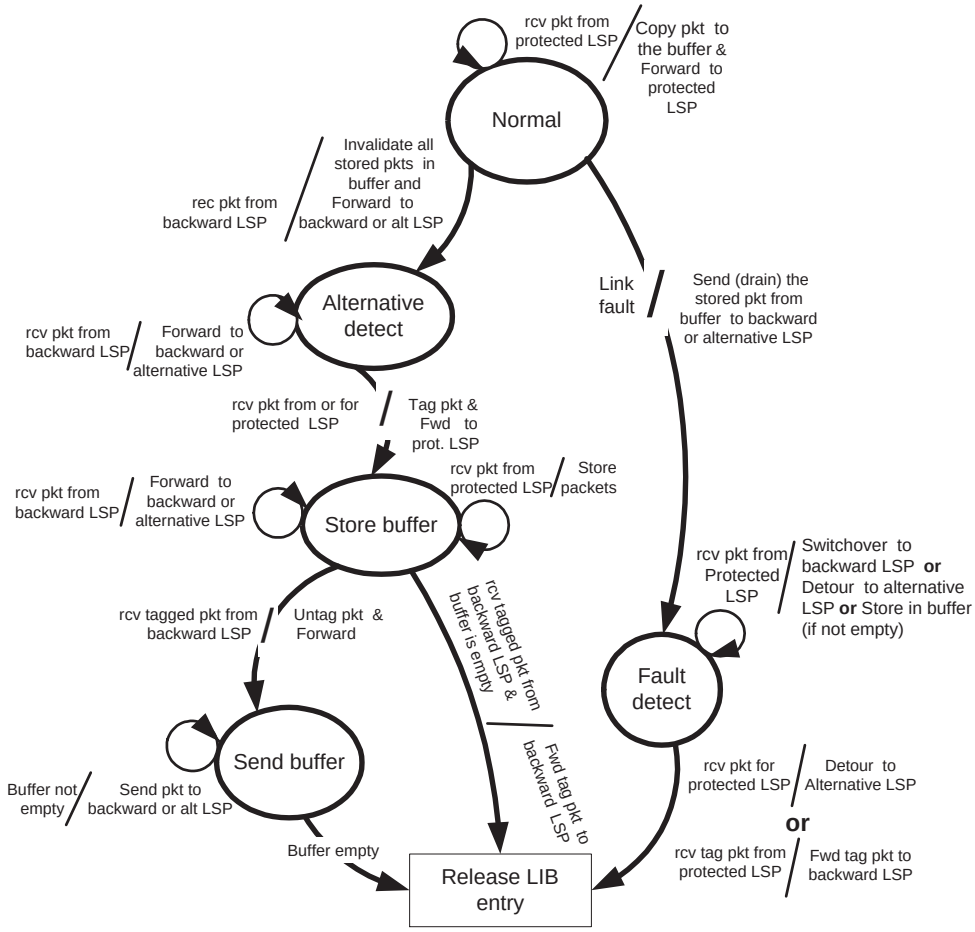

Figure 4.2 RFR state machine diagram

Since this proposal is an extension of the mechanism proposed in Chapter 3, the change introduced in this proposal affects only the NORMAL, ALTERNATE_DETECT and FAULT_DETECT state functions. The functions of STORE_BUFFER and SEND_BUFFER remain unchanged.

The proposed algorithm functions as follows. In the normal condition all LSRs store a copy of received packets in the local buffer. The buffer is dimensioned with sufficient capacity to protect against packet losses during a link/node failure in the protected LSP.

Once a failure along the protected LSP is detected, the protected LSR that detects the fault performs the switchover procedure (LSR3 in Figure 4.1). This procedure consists of a simple label swapping operation from the protected LSP to the backward LSP for all packets with a label corresponding to the protected LSP. The link status of the label information base forwarding table (LIB) of this LSR is changed from NORMAL to FAULT_DETECT (Figure 4.2). It then begins to drain all the packets stored in its buffer - i.e., send them back along the backward LSP. Any incoming packets on the protected LSP are also sent back.

The immediate upstream LSR, in this case LSR2 (Figure 4.1) receives these reversed packets from LSR3 through the backward LSP. When it detects the first packet coming on the backward LSP, it changes the link status of the LIB entry of the protected LSP corresponding to this backward LSP to ALTERNATIVE_DETECT (Figure 4.2). Additionally, it invalidates all data in its buffer. Recall that these packets are stored to be used in case the output link fails. When the LSR enters the ALTERNATE_DETECT state the buffer is emptied and will be used to store packets coming in from the protected LSP until they may be forwarded through the backward LSP.

The next, immediate packet received from the protected LSP sees the LIB entry as ALTERNATIVE_DETECT. This indicates that there is a link problem somewhere in the protected LSP. This packet is then tagged as the last packet from this LSR (LSR2) and forwarded normally downstream and the LIB entry status is changed from ALTERNATIVE_DETECT to STORE_BUFFER (Figure 4.2). The subsequent packets coming in on the protected LSP will be stored in the buffer because they will find the link status is STORE_BUFFER. This continues until the tagged packet is received through the backward LSP.

In order to detect the tagged packet coming back on the backward LSP, the LSR has to check if the tag bit of the received packet is *set* or *not*. If the comparison result is false the packet will be forwarded using the normal swapping operation. Otherwise, the LSR knows that no more packets are expected from the backward LSP. Note that

at this point there are two possible actions depending the buffer condition. Here we assume the buffer is "not empty" to describe the complete algorithm. In this case, the tag bit in the label must be disabled (set to 0) and the packet is sent according to the label swapping result as a normal packet. Moreover, it changes the status from STORE_BUFFER to SEND_BUFFER, and then when the buffer is empty, the label is removed from the LIB.

This process is repeated at every LSR up to the ingress LSR. Although in this description we presented the example of link failure, our algorithm can also be used without requiring any additional algorithm for node failure restoration.

4.4 DERIVATION OF THE MODEL

The mathematical formulation of our model is an important step to validate the simulation results. Once we do this, we can study the trade-offs between the cost of using buffers in each LSR within the protected path and the benefits that accrue in terms of performance for high-priority QoS traffic. The size of the buffers required both at the ingress node and at the intermediate nodes between the ingress and the point of failure can be estimated from the derived model and validated by our simulation. The following are the terms used in our derivation with a brief explanation of each:

T_{tran} : − Transmission delay time or packet transmission time,

T_{prop} : − Propagation delay time in a link,

V_{T_lsp} : − Source rate (reference traffic),

B_{W_lsp} : − LSP bandwidth that is the peak rate admitted,

P : − Packet size,

d : − Distance between two adjacent LSRs,

$T_{recovery}$: − Full restoration time,

T_{fault_detect} : − Fault detect time,

$B_{ingress}$: − Buffer size in ingress LSR,

N : − Number of LSR that detects the fault (i.e., number of nodes of the backward LSP excluding the ingress node).

According to our generalized network simulation model (Figure 4.3), after the detection of a failure the total time required by the node detecting the failure to switchover all packets (including buffered packets) and the time for the tagged packet to return to the immediate upstream node must be calculated. This is equal to the link delay for the first packet switched over to reach the next upstream LSR along the backward LSP, plus the round trip link delay for the tagged packet to return to its node (the node which tagged it).

$$T_{switch\_over} = 3 * T_{link} \tag{4.1}$$

Where, T_{link} (link delay) is calculated in our case as the sum of the transmission $\left(\frac{P}{B_{W\_lsp}}\right)$ and propagation $\left(\frac{d}{c}\right)$ delays, assuming that both queuing and processing delays are zero. The reason is that the NS-2 simulator does not account for queueing and processing delays.

$$T_{link} = T_{tran} + T_{prop} \tag{4.2}$$

The rest of the delays up to the point of restoration of traffic along the *alternative LSP* are the sum of the delays for each intermediate LSR to pass back all of its packets to the immediate upstream node. This time can be broken down into two components: (1) time taken to drain all packets from its buffer, and (2) time taken for the last packet (the one that was *tagged*) to reach the next upstream node (T_{link}).

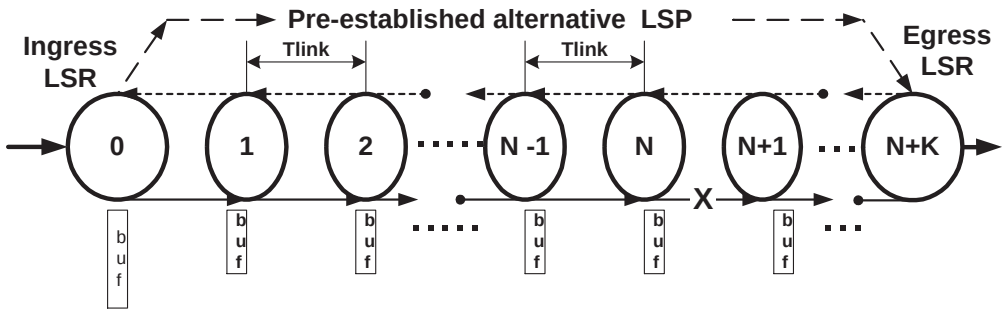

Figure 4.3 Model for equation. Solid line: Protected LSP; Dashed line: Backward LSP

The store period is $2 * T_{link}$ (two-way delay for the tagged packet). Given that the packets that are stored in the buffer arrive at the rate of reference traffic (V_{T_lsp}) during $2 * T_{link}$ and the rate at which the packets are drained from the buffer is equal to the bandwidth (B_{W_lsp}), we have:

$$T_{int\_buffer\_drain\_pkt} = \frac{2 * T_{link} * V_{T\_lsp}}{B_{W\_lsp}} \tag{4.3}$$

and the intermediate LSR delay time (T_{int}),

$$T_{int} = T_{int\_buffer\_drain\_pkt} + T_{link} \tag{4.4}$$

Once we know T_{fault_detect}, T_{switch_over}, T_{int} and N we can calculate the total restoration time starting from the time that the fault was detected. Note that we assume T_{link} over all links is the same (i.e., all links operate at the same rate (B_{W_lsp}) and have the same propagation delay (d)). The sum of delays in the intermediate LSRs is equal to $\sum_{i=1}^{N-1}(T_{int})_i = (N-1) * T_{int}$.

$$T_{recovery} = T_{fault\_detect} + T_{switch\_over} + \sum_{i=1}^{N-1} (T_{int})_i \tag{4.5}$$

applying our previous condition:

$$T_{recovery} = T_{fault\_detect} + T_{switch\_over} + (N-1) * T_{int} \tag{4.6}$$

We assume that the time to detect the fault by an LSR - $T_{fault\_detect} = 0$ since this affects all recovery schemes equally. Then the above equation becomes:

$$\mathbf{T_{recovery}} = \mathbf{T_{link}} \left(\mathbf{N + 2 + 2\,(N - 1)} \frac{\mathbf{V_{T\_lsp}}}{\mathbf{B_{W\_lsp}}} \right) \tag{4.7}$$

Finally, for the **worst case** (i.e., when the $V_{T\_lsp} = B_{W\_lsp}$)

$$\mathbf{T_{recovery}} = \mathbf{3 * N * T_{link}} \tag{4.8}$$

4.4.1 Buffer size requirement calculation for the ingress LSR during the restoration period.

The required buffer size in the ingress LSR is an important factor for the implementation of the proposed mechanism. This node has to store packets from the time it receives the first packet on the backward LSP switched over from the alert node (point-of-failure) until it receives its own tagged packet. The time taken by the former is:

$$T_{ing\_rcv\_first\_pkt} = N * T_{link} \tag{4.9}$$

and the time at which the latter takes place is $T_{recovery}$ (Figure 4.4). Therefore,

$$T_{ing\_store\_pkt} = T_{recovery} - T_{ing\_rcv\_first\_pkt} \tag{4.10}$$

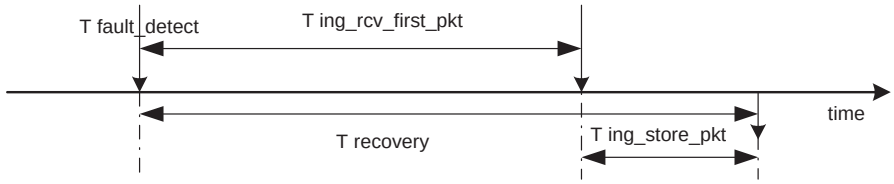

Figure 4.4 Graphical representation of times for ingress buffer calculation

The required buffer size in the ingress LSR during the restoration period is:

$$B_{ingress} = T_{ing\_store\_pkt} * V_{T\_lsp} \tag{4.11}$$

and hence,

$$\mathbf{B_{ingress} = 2 * T_{link} * V_{T\_lsp} * \left(\frac{(N-1)\, V_{T\_lsp}}{B_{W\_lsp}} + 1 \right)} \tag{4.12}$$

For the worst case, when V_{T_lsp}=B_{W_lsp},

$$\mathbf{B_{ingress} = 2 * N * T_{link} * V_{T\_lsp}} \tag{4.13}$$

The required buffer size in each intermediate LSR during the restoration period is:

$$\mathbf{B_{intermediate} = 2 * T_{link} * V_{T\_lsp}} \tag{4.14}$$

4.5 SIMULATIONS AND RESULTS

The objective of the simulation is to validate the formula and to compare the behavior of the proposed mechanism with previous MPLS protection mechanisms [HK00].

The simulated scenario is the one shown in Figure 4.1. The simple network topology with a protected and alternative LSP is used. We extend the simple network topology for different numbers of intermediate LSRs in the protected LSP. We vary the location of the node that detects a fault (alert LSR).

Parts of the MNS source code were modified to simulate both mechanisms (ours and Haskin's [HK00]) and the modified simulator was validated with previously published results for Haskin's method [GW01b] [HD01].

We present the results for CBR traffic flow with the following characteristics: packet size = 200 bytes, source rate= 400Kbps, burst time=0 and idle time =0.

The results based on the derived formula for the proposed model are plotted with the corresponding simulation results, for Full Restoration Time (Figure 4.5) and for the buffer size needed at the ingress LSR (Figure 4.6). These figures show that the analytical results are almost identical to the simulation results, validating our analytical expression of the proposed mechanism (RFR).

Observe that in both cases (Figures 4.5 and 4.6) for the $B_{W\_lsp} = 1Mbps$ the restoration time and the ingress LSR buffer requirement increase due to the fact that the transmission speed of the packets is low compared to 5Mbps, 10Mbps and above. The time required to reach the ingress LSR depends on the speed. The restoration time basically depends on the transmission speed and the same applies for the buffer requirements at the ingress LSR.

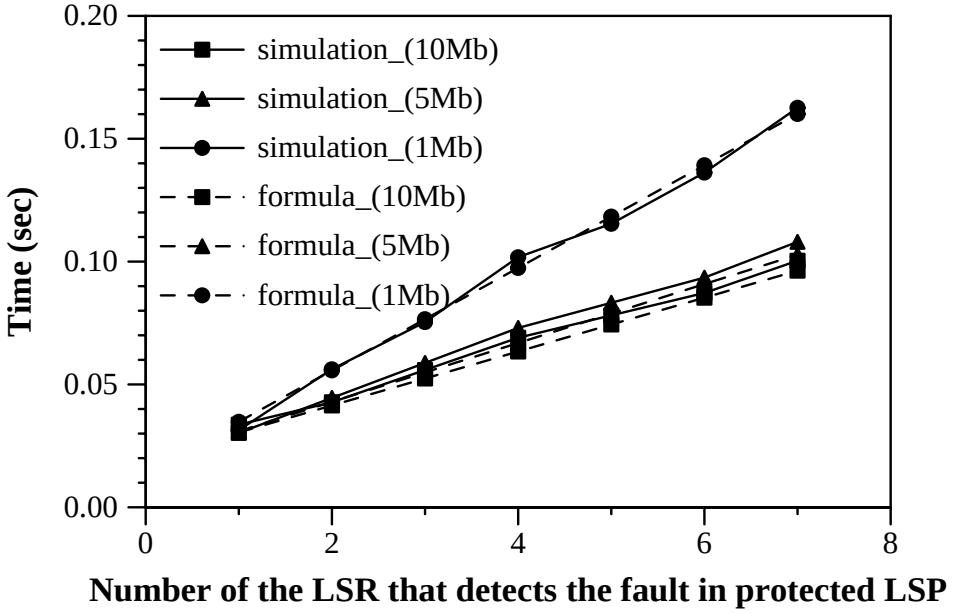

Figure 4.5 Recovery time for different LSP bandwidths

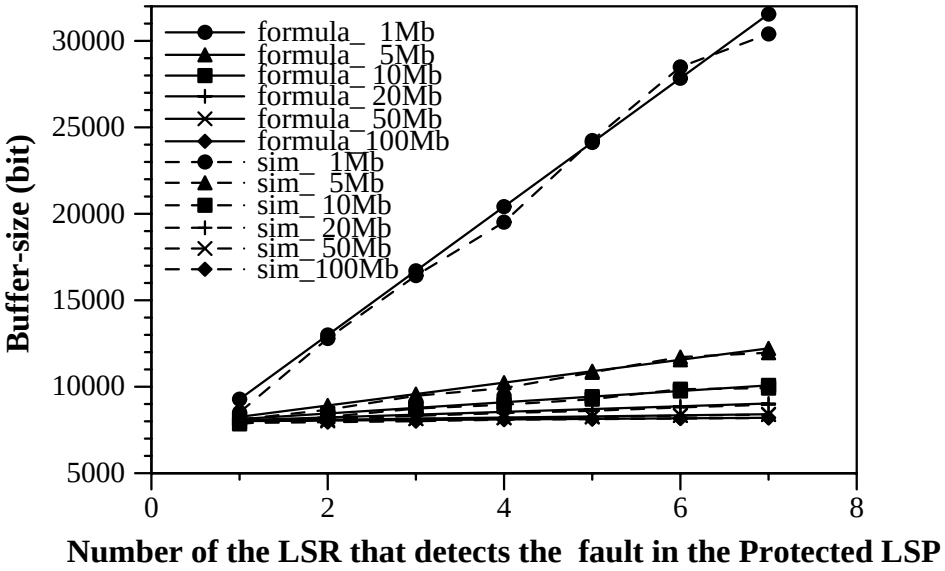

Figure 4.6 Ingress buffer size for Vt_lsp=400k and Pkt_size=200bytes for different LSP bandwidths and numbers of LSR (N)

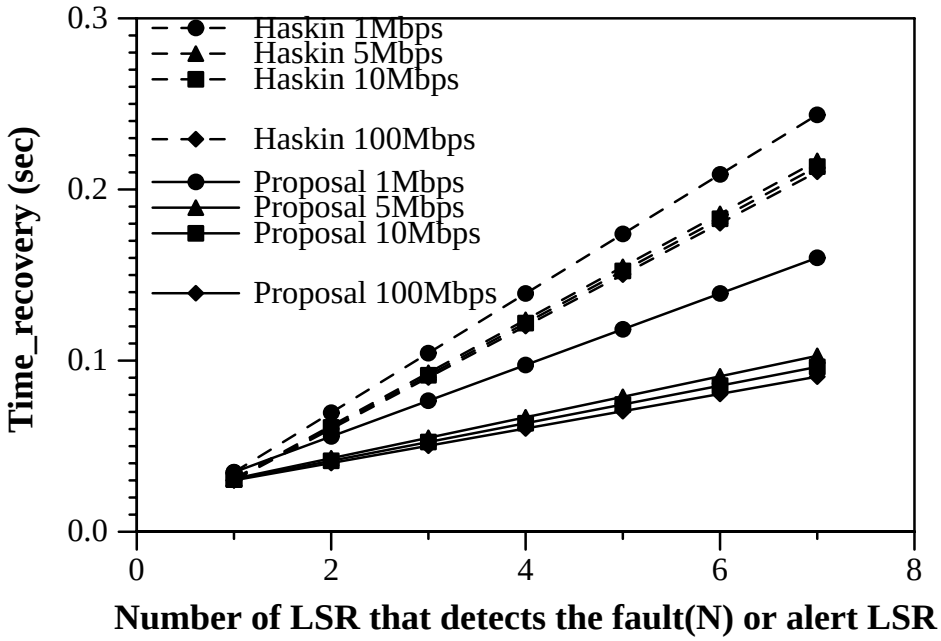

Figure 4.7 Restoration delay for 200 bytes packet size for different LSP bandwidth and number of alert LSR (N) using formula (derived model)

The plots in Figures 4.7 and 4.8 correspond to the comparison of the overall restoration period between Haskin's scheme and RFR for the derived model and the simulation respectively.

We use Figure 4.8 results for comparison of the overall restoration period for both proposals for different points of failure and for different bandwidths. Time is computed from the instant the fault is detected until the protected LSP is completely eliminated. Our proposed mechanism significantly improves the Full Restoration Time. A time reduction of 24.6%, 27.9%, 29.8%, 31.7% and 33% for the 3rd, 4th, 5th, 6th and 7th node of the alert LSR on the protected LSP respectively are achieved. Note that the above percentage values correspond to an LSP bandwidth of 1Mbps. The improvements are greater as the bandwidth increases and the transmission rate of the source remains fixed.

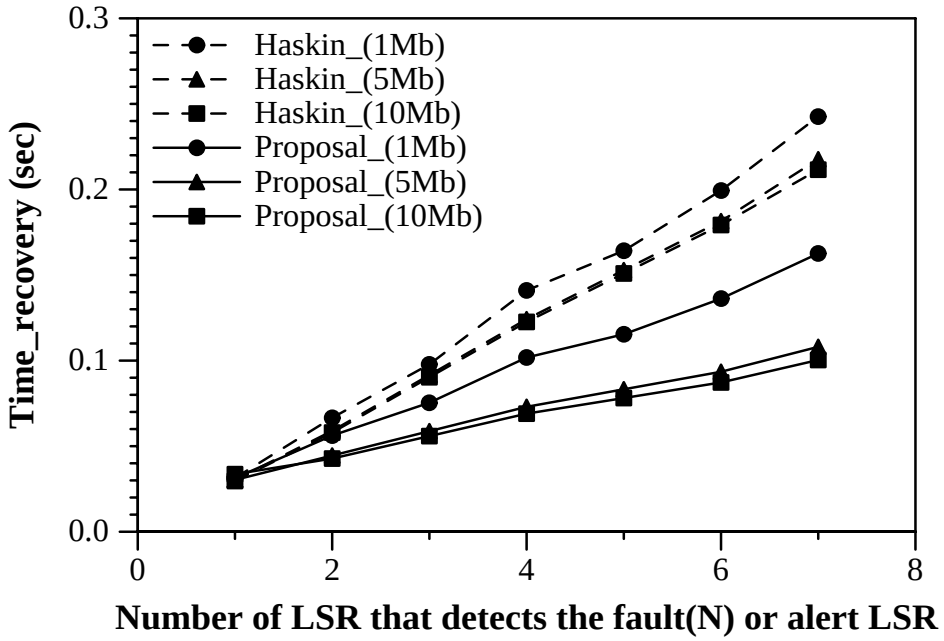

Figure 4.8 Restoration delay for 200 bytes packet size for different LSP bandwidth and number of alert LSR (N) using the simulator

In Figure 4.9 we present the results concerning the ingress node buffer requirement, varying the B_{W_lsp}, distance (d) and N. Results show that even for a long-distance LSP the buffer size required at the ingress node is reasonable compared to the benefits provided by the RFR mechanism.

In Figure 4.10 we maintain the V_{T_lsp} and the distance (d) constant, and vary the B_{W_lsp} and N. In this case, as we increase the B_{W_lsp} the effect of N on the required ingress buffer space becomes negligible. Note that in both cases we maintain the packet size (P) constant. The buffer space (memory) requirements for the implementation of our proposal under different conditions are clearly demonstrated. We have plotted the buffer needs for the ingress node for the longest period it could theoretically have to store packets (waiting for all downstream nodes to drain their packets).

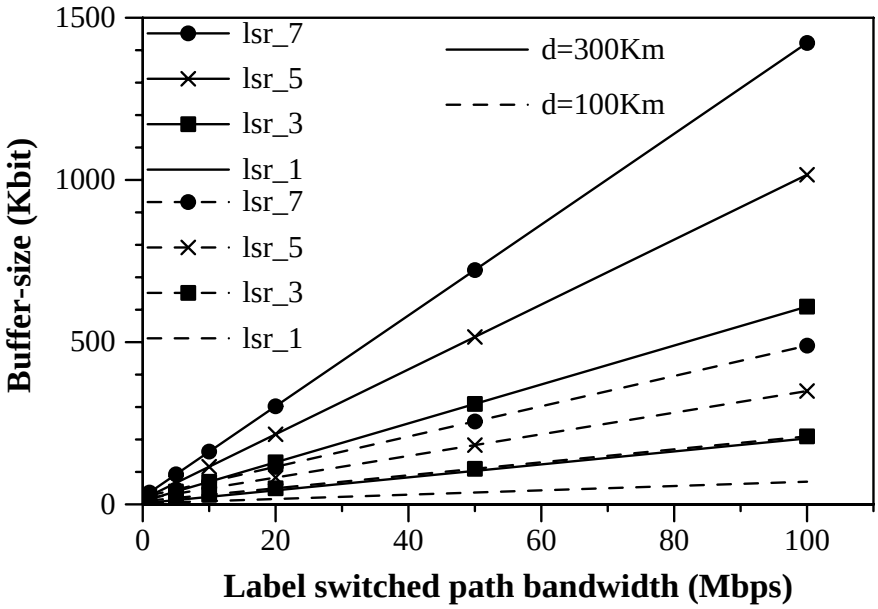

Figure 4.9 Required buffer space for ingress LSR when $Vt\_lsp = Bw\_lsp$ (worst case) and Pkt_size=1600 bits for d=300Km and d=100Km varying the Bw_lsp and N

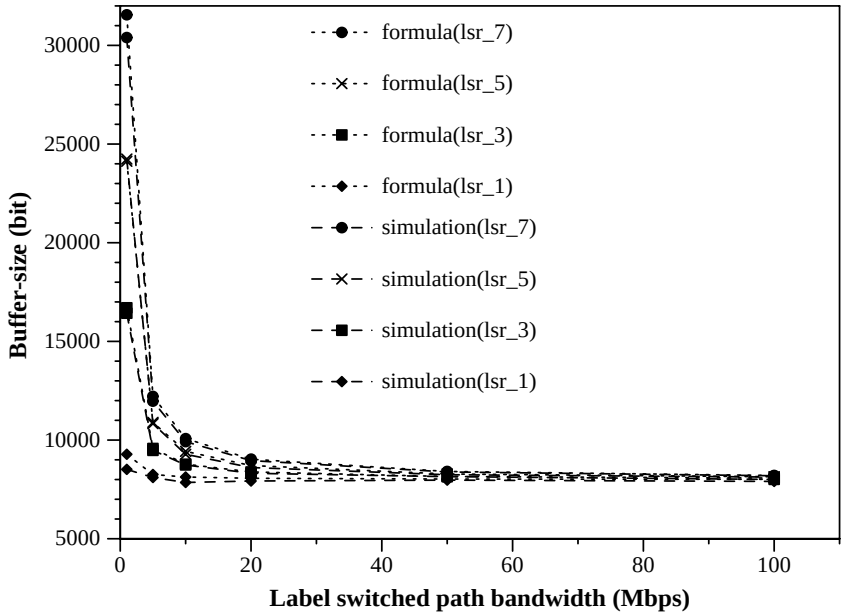

Figure 4.10 Comparison between formula and simulation results for ingress buffer with Vt_lsp=400k and Pkt_size=200bytes for different N and Bw_lsp

4.5.1 Validation of the results and qualitative analysis

The formula for buffer requirements for the ingress buffer (4.12) is:

$$B_{ingress} = 2 * T_{link} * V_{T\_lsp} * \left(\frac{(N-1)\, V_{T\_lsp}}{B_{W\_lsp}} + 1 \right)$$

we define α and β as:

$$\alpha = \frac{B_{W\_lsp}}{V_{T\_lsp}} \tag{4.15}$$

where $\alpha \geq 1$, and

$$\beta = \frac{T_{tran}}{T_{prop}} \tag{4.16}$$

where $\beta > 0$.

substituting the above definitions of α and β, and T_{link} from (4.2) in the formula for the ingress buffer requirements derived from our model, we get,

$$\mathbf{B_{ingress}} = \mathbf{2} \left(\frac{\mathbf{P}}{\alpha} \right) \left(\mathbf{1} + \frac{\mathbf{1}}{\beta} \right) \left(\frac{(\mathbf{N}-\mathbf{1})}{\alpha} + \mathbf{1} \right) \tag{4.17}$$

Considering the scenario used for simulation is: $V_{T\_lsp} = V_{W\_lsp}$, $P = 1600bits$, and T_{Prop} equals 1 msec and 0.1 msec, we get the α and β values to calculate the ingress buffer. The results agree with the simulation.

Note that in this condition Haskin's scheme also stores packets in the ingress LSR equivalent to the amount of packets circulating during the round trip time (i.e., $2 * T_{link} * V_{T\_lsp}$).

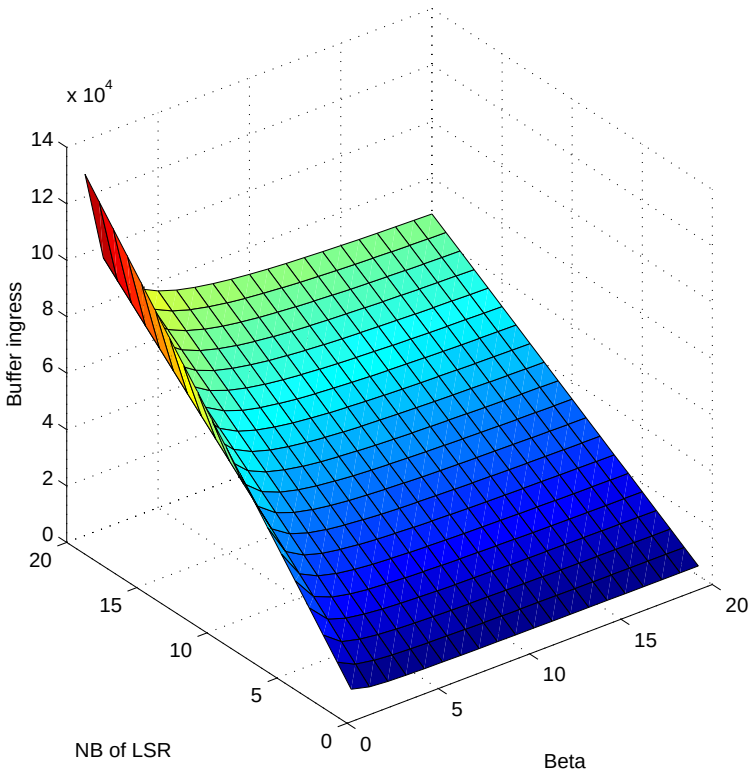

Figure 4.11 Behavior of ingress buffer

For $\alpha = 1$:

$$B_{ingress} = 2PN\left(1 + \frac{1}{\beta}\right) \tag{4.18}$$

When $\beta \to \infty$, it implies the propagation time tends to zero, so,

$$B_{ingress} = 2PN \tag{4.19}$$

The above result proves that with the propagation time zero the nodes are tight and the result depends only on the LSR number (N) and the packet size, which is proportional to T_{tran}.

When $\beta \rightarrow 0$, it implies the propagation time tends to ∞, so the $B_{ingress} \rightarrow \infty$. This is true because there are almost no packets circulating in the network.

The formula for the recovery period from (4.7) is:

$$T_{recovery} = T_{link}\left(N + 2 + 2\left(N-1\right)\frac{V_{T\_lsp}}{B_{W\_lsp}}\right) \tag{4.20}$$

substituting α and β, and T_{link} in $T_{recovery}$, we get,

$$\mathbf{T_{recovery}} = \mathbf{T_{tran}}\left(\mathbf{N + 2 + 2}\left(\frac{\mathbf{N-1}}{\alpha}\right)\right)\left(\mathbf{1} + \frac{\mathbf{1}}{\beta}\right) \tag{4.21}$$

For $\alpha = 1$:

$$T_{recovery} = 3N * T_{tran}\left(1 + \frac{1}{\beta}\right) \tag{4.22}$$

When $\beta \rightarrow \infty$, it implies the propagation time tends to zero, so the recovery time depends on the number of LSRs and the packet transmission time, T_{tran}.

$$T_{recovery} = 3N\frac{P}{B_{W\_lsp}} = 3N * T_{tran} \tag{4.23}$$

When $\beta \rightarrow 0$, it implies the propagation time tends to ∞, so the recovery time tends to ∞.

For Haskin's scheme the recovery period is the sum of the time taken by the first packet switched-over by the alert LSR to arrive at the ingress LSR (T_{link}) and the time taken by the last packet sent before the ingress LSR received the first packet through the backward LSP to travel from the ingress LSR and return back to this LSR, which is equal to $2 * T_{link}$ (i.e., time from ingress-alert-ingress).

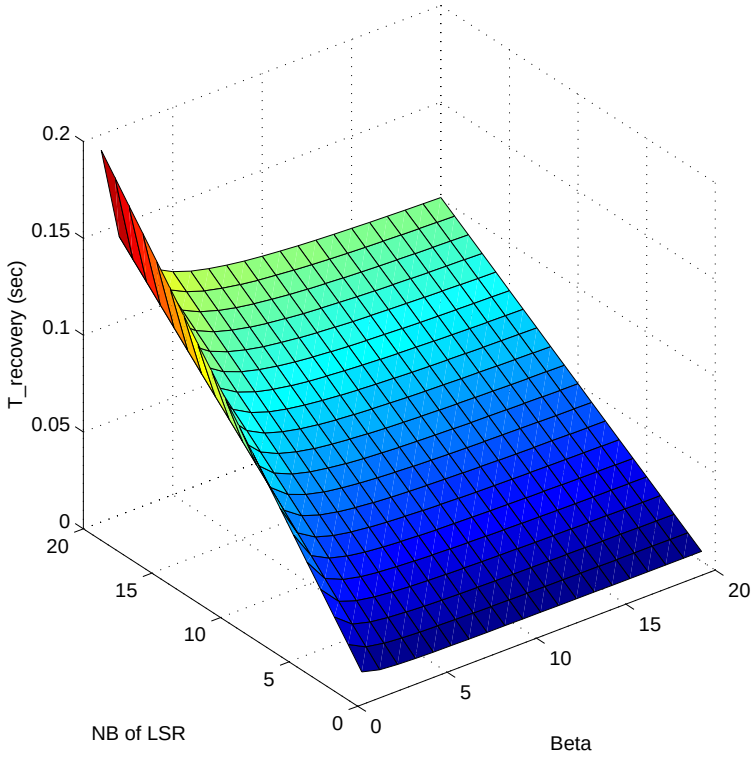

Figure 4.12 Behavior of recovery time

$$T_{recovery\_haskin} = 3 * N * T_{link} \tag{4.24}$$

With α and β,

$$T_{recovery\_haskin} = T_{tran} * 3N \left(1 + \frac{1}{\beta}\right) \tag{4.25}$$

Considering the simulation scenario: $V_{T\_lsp} = 400kbps$, $V_{W\_lsp} = 1Mbps$, $P = 1600bits$, and $T_{prop} = 10msec$ we get the α and β values to calculate the recovery time for Haskin's and our proposal. The results agree with the simulation.

4.6 SUMMARY

This chapter has presented a mechanism to perform Reliable and Fast Rerouting (RFR) of traffic in MPLS networks. Our method eliminates packet loss and packet disorder while improving the average delay time during the restoration period. This is achieved at a minimal cost for additional buffer space (memory) that is far outweighed by the benefits.

Apart from the buffer size, which is not very significant even for the worst case, the most interesting aspect is the linear behavior of our model relating B_{W_lsp}, V_{T_lsp}, P, d and N. This allows easy estimation of the buffer requirements for given bandwidths and QoS constraints.

5

RFR FOR TCP APPLICATIONS

The Transmission Control Protocol (TCP) is the basic transport protocol for Internet applications such as email, web browsing, and file transfer. The majority of data traffic sent over the Internet is transported by TCP. IP networks are designed to be auto-controlled. The end station (host) implementing TCP adjusts its sending rate to the bandwidth of the path to the destination. The routers are in charge of topology changes in the network and of computing new paths based on the new topology. However, this mechanism does not ensure that the network runs in an efficient manner.

Although TCP is a reliable transport mechanism, packet loss, packet delay and packet disorder can seriously affect its performance, and hence application throughput. This is due to the TCP behavior of reducing its window size when congestion is detected or packet losses are detected, giving as a result a lower bandwidth for the application than the optimal one.

In this Chapter we evaluate the benefit for TCP applications of our proposal for a reliable and fast rerouting (RFR) mechanism described in Chapter 4. It is important to note that our work does not introduce any change in TCP. What the RFR does is to introduce a new function in the routers to take care of packet loss, packet disorder and packet delay for protected TCP traffic in an MPLS network (Chapter 4).

In the following section we present a brief explanation of TCP behavior and related algorithms.

5.1 OVERVIEW OF TCP BEHAVIOR

The Transmission Control Protocol (TCP) uses two main flow control mechanisms to transfer data from one end to the other. They are the receiver flow control and the sender flow control [tcp81] [Ste94].

The receiver flow control is based on the receiver using the ACK to advertise the allowable window size. In other words, when the receiver sends the acknowledgment packet to the sender it includes its available buffer size, indicating the amount of data that the sender can send at that time.

On the other hand, sender flow control uses the congestion window to manage its flow. The congestion window defines the maximum number of segments (in bytes) the sender can send without receiving an ACK. The sender's congestion window value is controlled and changed according to its implemented algorithm. Based on the above explanation, the amount of data the sender can send to the receiver without getting an ACK is the minimum of the receiver advertised window and the congestion window value (*cwdn*). Note that at any time during a TCP data transfer, either the receiver or the sender flow control is dominant.

There are four algorithms defined to control the congestion window during a TCP data transfer. The first two algorithms, *Slow Start* and *Congestion Avoidance* are

applied to all data transfers [Jac88]. The other two algorithms, *Fast Retransmit* and *Fast Recovery* are used when packet loss and packet reordering occur [Jac90].

5.1.1 Slow Start and Congestion Avoidance Algorithms

The slow start algorithm defines the way in which the *cwnd* is initially set and increases its value. When the TCP connection is established the congestion window is set to the default value, which is defined to be no more than two segments [APS99]. Then the sender increases its *cwnd* exponentially with each ACK received as an indication of the correct delivery of a sent segment. Thus, the value of the congestion window grows from one segment to two segments when the first segment is ACKed, and sends two segments. When it receives the ACK of these two segments it increases the *cwnd* from two to four. Thus, successively from four to eight, from eight to sixteen, and so on.

This exponential increment is limited either by the receiver advertised window (i.e., the amount of the data that the sender can send can not exceed the available buffer space at the receiver) or the limited capacity in any link along the path to handle the amount of data sent by sender. Observe that in this latter case, as the capacity of the path is not unlimited, if the sender continues increasing the sending rate it will cause packet losses. Packet loss will trigger the slow start algorithm. For this reason it is important to have a mechanism to control this exponential growth of the *cwnd* before the path capacity limit is reached. This is the point at which the congestion avoidance algorithm comes into effect to take the control of the *cwnd*.

The variable defined in [Jac88] to control the transition from slow start to congestion avoidance is called the *slow start threshold (ssthl)*. Congestion is assumed when the sender receives repeated ACKs of dropped or disordered packets, or when the time out is reached before receiving an ACK for a segment. When congestion is detected, the value of the *ssthl* is set to half of the current window size ($cwnd/2$) and the *cwnd* is set to one, forcing the sender window to the initial point of the slow start. Then

the sender starts the slow start algorithm, which functions as described before, as long as *cwdn*<= *ssthl*.

When *cwnd*> *ssthl* the congestion avoidance condition takes place: the incremental rate of *cwnd* is $1/cwnd$ for each arriving ACK, resulting a linear growth rate of one segment for each ACK. The congestion avoidance algorithm, as opposed to the exponential growth of the slow start, tries to provide the optimal size of *cwdn* for the delay-bandwidth product. However, despite this linear increment the sender can still reach the maximum capacity limit of any link on the path because the sender has no information about the link status. This condition again triggers the slow start, reducing the amount of data that can be sent through the path. The fast retransmission and fast recovery are proposed to alleviate this effect, using the link's maximum available capacity.

5.1.2 Fast Retransmit and Fast Recovery Algorithms

For the sender, the arrival of three consecutive repeated ACKs is an indication that the packet was actually dropped [APS99], triggering the retransmission of the segment. The retransmit occurs without waiting for a time out. This behavior is defined as fast retransmit [Jac90]. After fast retransmit takes place the sender enters into fast recovery. Note that in fast recovery the sender continues sending data at the optimal rate according the congestion avoidance algorithm until the arrival of the ACK for the retransmitted segment.

Unlike congestion avoidance, which reduces the *cwnd* to one, fast retransmit sets the *cwnd*=*ssthl*+ the number of the duplicated ACK, then continues receiving the ACK of those segments that were sent on the path before the fast retransmit came into action (i.e., retransmits the possibly dropped segment). For each ACK received in this period it increases the *cwnd* by one segment.

When the sender receives the ACK of the retransmitted segment it exits fast recovery mode, and the *cwnd* is set equal to the *ssthl* (i.e., half of the actual congestion window value). At this point congestion avoidance is initiated and the *cwnd* increases linearly according the congestion avoidance algorithm.

Fast recovery improves the throughput for a single packet loss, but multiple packet losses continue to affect the throughput because the window size (*cwnd*) decreases for each dropped packet. When used on high speed networks with a long delay and congestion, TCP becomes slow. When a TCP packet is lost, the sender must retransmit the packet, and the time it waits before retransmission increases according the delay, impairing TCP performance.

5.2 EVALUATION OF RFR FOR TCP CONNECTIONS

In order to evaluate the benefits of RFR for TCP applications during a link/node failure or congestion situation compared to Haskin's scheme we used the same simulation scenario 4.1.

For the throughput comparison for TCP traffic we setup an FTP session over a TCP connection with packet size = 1000 bytes. Figure 5.1 compares the behavior of RFR and Haskin's scheme.

In Figure 5.1 the difference in the sequence number of TCP segments received by the egress LSR is seen clearly for the same simulation time. Figure 5.2 shows a more detailed view of the sequence number during the restoration period.

Additionally, in Figure 5.2 the perturbation caused by the disorder of the packets can be seen (zoomed graph). Note that the time of link failure is 1.51 seconds. Figure 5.1 confirms that the proposed mechanism avoids packet loss and disordering. This benefit is due to the use of the buffer, which avoids the loss of packets and therefore the penalty due to retransmission.

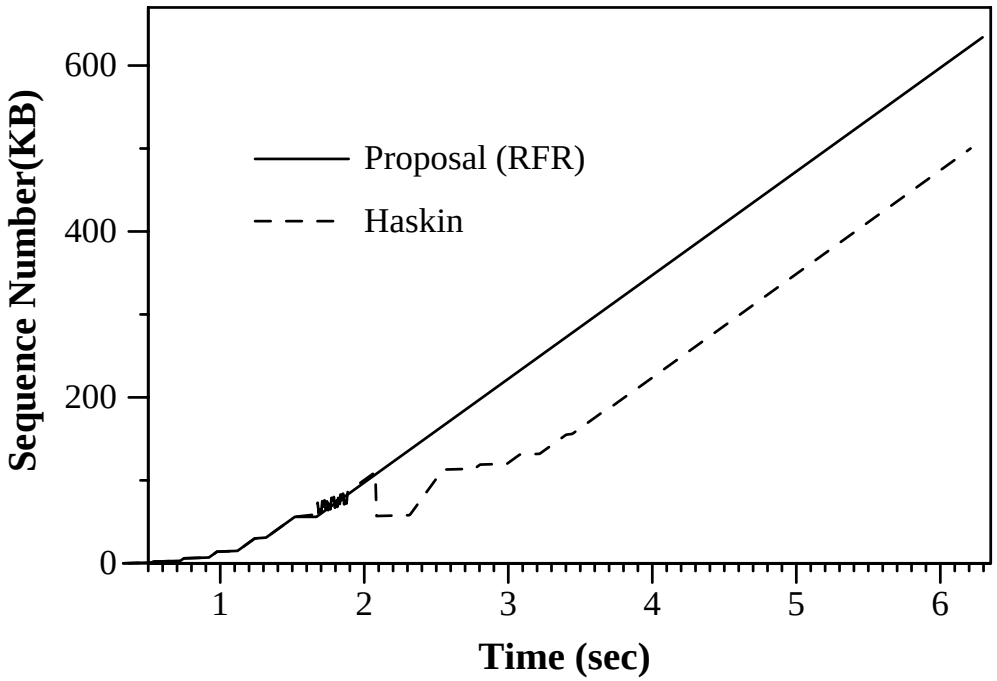

Figure 5.1 Behavior of TCP traffic for MSS of 1000 bytes

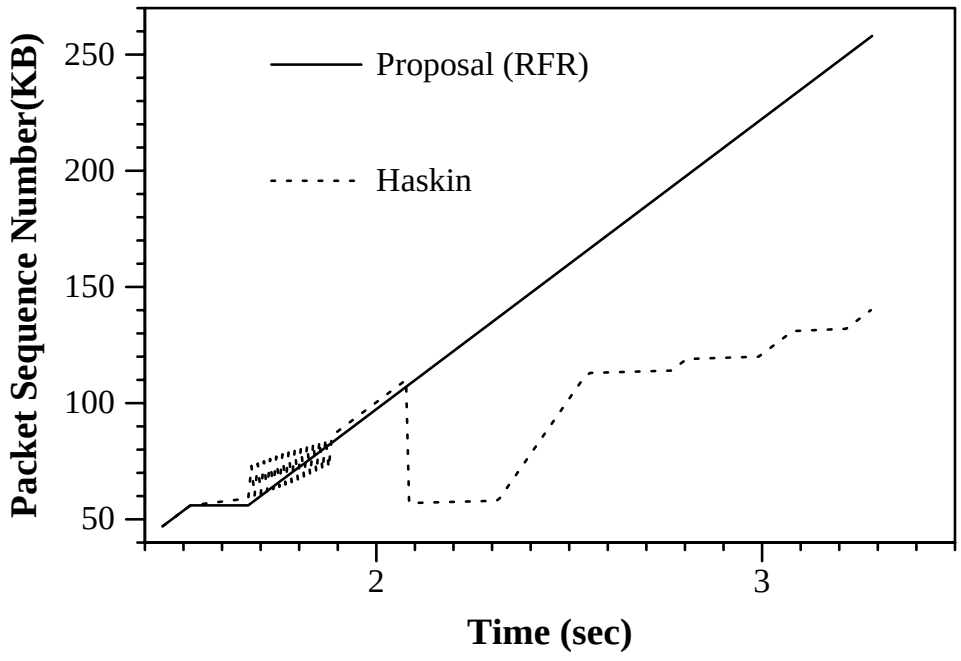

Figure 5.2 Behavior of TCP traffic for MSS of 1000 bytes

5.3 SUMMARY

As RFR avoids packet losses and packet disorder in the protected flows, TCP connections experience neither losses nor disordered packets, and can continue to run at the maximum throughput even during the restoration period of the protected LSP.

6

MULTIPLE FAULT TOLERANCE RECOVERY MECHANISMS

6.1 INTRODUCTION

The recent advances in fiber optic transmission and switched routing techniques dramatically facilitate the increment of link capacity and the provision of several classes of service over the same communication link. The introduction of MPLS as part of the Internet forwarding architecture to address the need of future IP-based Networks [RVC01][CDF+99] will contribute significantly, among other advantages, to the application of traffic engineering (TE) techniques and quality of service (QoS) provision mechanisms.

An adverse consequence of this increase in link capacity is a higher degree of complexity of network survivability. A link failure implies the rerouting of a huge amount of traffic with different QoS classes. In [IG00] the authors assure that fiber cable cuts are surprisingly frequent and serious.

For this reason, the need for rapid restoration mechanisms in an end-to-end label switching technology like MPLS obliged the research community to find different mechanisms to reroute traffic around a failure point in a fast, reliable and efficient way.

Protection schemes in MPLS networks can be classified as link protection, node protection, path protection and segment protection [ACE+02]. Path protection is used to protect a Label Switched Path (LSP) from failure at any point along its routed path except for failures that might occur at the ingress and egress Label Switching Routers (LSRs). The path protection scheme establishes an alternative LSP, either before or after failure detection. Segment protection only needs to protect the portion of the LSP that belongs to a defined segment protection domain. Segment protection will generally be faster than path protection because recovery generally occurs closer to the fault [ACE+02]. Link protection is carried out to protect the link between two adjacent nodes. Node protection addresses the protection of all links connected to the node. For the sake of better understanding of the following sections we will repeat some important concepts in the explanation.

There are two possibilities for establishing an alternative LSP in MPLS-based networks: i) Local repairs using alternative LSPs from point of failure and ii) Global repairs using ingress-to-egress alternative LSPs [OSMH01][Swa99][SH02].

The alternative LSP may be calculated on demand using dynamic restoration or may be pre-calculated and stored for use when the failure is detected using preplanned restoration[SH02] [OSMH01][HK00][CO99]. Usually the alternative LSP is established based on link protection or path protection techniques. The pre-established alternative LSP is better for critical traffic than the alternative LSP established on demand after the occurrence of failure [OSMH01] [HK00].

The dynamic restoration scheme searches, decides, and generates the alternative (backup) LSP dynamically upon failure. When a failure occurs nodes use message flooding to locate the backup routes that can bypass the failed routes. In order to

reduce the number of messages generated and to improve restoration speed, some algorithms restrict message broadcasting to a limited number of hops.

The preplanned algorithm permits many LSPs to be restored at the same time because only one message is generated per LSP. The preplanned restoration scheme preassigns an alternative LSP to each protected LSP before failure occurs. Several schemes have been proposed for selecting the best route(s) from several candidates based on different criteria [KL00][AWK+99][SWW01].

The key concept of the preplanned restoration scheme is the simplification of the restoration process that must be performed after a failure occurs; the goal is rapid and reliable restoration. One more advantage of the preplanned scheme is the ability to efficiently support explicit routing, which provides the basic mechanism for traffic engineering. The major drawback of preplanned alternative LSPs is that they allow less flexibility against multiple or unexpected points of failure. Furthermore, network resource utilization may not be optimal since alternative LSPs are pre-defined.

Our previous proposals for protection mechanisms in Chapter 3 and Chapter 4 assume a single link/node failure addressing basic performance metrics such as packet loss, packet reordering and average packet delay. In this chapter we propose a new protection mechanism for multiple link/node failures within a protected LSP. Multiple link failure on an LSP can be expected to occur during natural and human made disasters on the core networks [CKMO92] [Kuh97]. The cascade effect due to a problem in some part of the network can also be considered as multiple link failure on an LSP in the core networks [THS+94][RM01]. In this work we consider an LSP that goes through several MPLS autonomous systems with different policies or recovery mechanisms. We also consider each segment protection domain as an abstract of an autonomous system.

6.2 RELATED WORK

Published work about multiple link/node failure protection schemes for a particular protected path are practically limited to single link failures that accommodate more than one LSP. Note that any single node or link failure can produce several LSP failures if multiple LSPs have been routed over a failed link or through the node. We consider this a single link failure, but most of the proposals refer to this as multiple failures [CKMO92][KCO+90]. Most of the papers about protection mechanisms refer normally to a single node/link failure. Multiple failures within an LSP can be produced when more than one link, node, or combination of both node and link failure occur.

Using the notion of the Shared Risk Link Group (SRLG), the authors in [BS01] consider as a single failure when all links belonging to that SRLG fail simultaneously. In their proposal they consider multiple failures to be when a single link failure occurs in different LSPs of the MPLS domain (multiple failures in an MPLS domain). Moreover, the main objective of their proposal is to allow the sharing of bandwidth among backup LSPs for restoration mechanisms. The same multiple failures concept is presented in [CJVM01] with a complicated and costly (in terms of time and resources) algorithm called dynamic multilevel MPLS fault management. In this proposal the mechanism starts with global repair and changes to local repair according to the reported failure condition. The improved version of this proposal is presented in [MCSA]. This method periodically updates the network information, in contrast to computing the LSP dynamically on-line. The concept of QoS protection (QoSP) is introduced to select which restoration method is suitable to establish the backup path in the backup decision module (BDM). The most interesting observation in the reported results is that local repair and the reversing method are more suitable in most cases. These results agree with our approach for combining local, global and reversing methods.

In [KL01][KKL+02], the concept of sharing the backup path is used, like the previous proposal [BS01]. But unlike that one, the proposal can be used for multiple link

failure on a protected LSP. The disadvantage of this proposal is that it needs to set up (N-1) bypass tunnels to assure the protection of any combination of link failures on the protected LSP. Despite this, the proposal does not guard the protected LSP from multiple node failures.

In [OSMH01] the authors consider that transferring the protected traffic to the recovery path is enough to take care of multiple failures. This consideration also assumes that no fault can occur in the restored path (alternative LSP) during or after the recovery process.

Using the segment protection domain technique the traffic is rerouted close to the failure point, reducing blocking problems. Local rerouting using a stacking technique in an MPLS domain may produce a backhauling problem, i.e., failure recovery may cause the stream to traverse the same links twice in opposite directions [ADH94][MFB99]. In this case all protected LSP traffic around the failed link is rerouted by pushing the corresponding reroute LSP label onto the stack of labels for packets on the protected LSP without regard to their source and destination nodes. This may result in backhauling because packets can pass through the egress LSR to reach the node at the other end of the failure, and then back from this node to the egress LSR using the primary LSP segment (i.e., the LSP portion from point of failure to egress LSR) increasing the length of the protection path (see Figure 6.1). Note that in MPLS the LSRs see only the label carried by the packet on the top level of stack and this has only a local significance.

In our previous work in Chapter 3 and Chapter 4, we propose methods for path protection and restoration mechanisms using pre-established alternative LSPs setup at the same time as the protected LSP, giving a solution for problems like packet loss, re-ordering and packet delay, which take place during a link/node failure. In this chapter we focus on handling multiple failures in a protected LSP. The motivation of this study is to overcome multiple failure in a protected LSP. Here we propose a new mechanism able to handle a single failure based on Segment Protection Domain

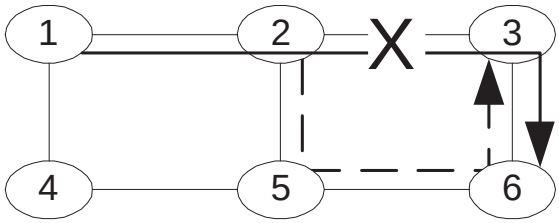

Figure 6.1 Backhauling problem. Ingress LSR is node 1, egress LSR is node 6, protected LSP: 1-2-3-6 (solid line), Local repair LSP (tunnel) for link failure 2-3 is: 2-5-6-3 (dashed line), protection LSP is: 1-2-5-6-3-6, and the arrows indicate the returning direction of the traffic

(SPD), local and global repairing methods; and, an extension of that mechanism to cope with multiple failures on the protected LSP in the MPLS network.

6.3 DESCRIPTION OF THE PROPOSED MECHANISM FOR SINGLE FAILURE

A protection domain is defined as the set of LSRs over which a working path and its corresponding alternative path are routed. Thus, a protection domain is bounded by the ingress and egress LSRs of the domain. The segment protection domain (SPD) is when a protection domain is partitioned into multiple protection domains, where failures are solved within that segment domain. In other words, the entire MPLS domain is the sum of many MPLS segment protection domains. SPDs may be established according to network administration policies, by an autonomous system. The SPD in this chapter is an abstraction of an MPLS autonomous system. In cases where an LSP traverses multiple protection domains, a protection mechanism within a domain only needs to protect the segment of the LSP that lies within the segment protection domain (SPD).

As stated in the former proposals (Chapter 3,4), the capacity reserved for the pre-planned alternative LSPs may be used by low priority LSPs with the caution that any low priority LSPs routed over this link will be preempted if the resource is needed by a high priority LSP as a result of a failure in the protected LSP.

The combination of path protection with segment protection and local repair activation is proposed in this chapter as a solution for multiple fault protection in a protected LSP, and single failures benefit from this proposal as well, in terms of full restoration speed.

In this proposal we combine the main benefits of segment protection (i.e., it is usually faster than path protection because recovery generally occurs closer to the fault) with the benefits of path protection to establish the optimal alternative path from ingress-to-egress in the entire MPLS network domain.

Another advantage of the segment protection scheme is related to blocking problems. Suppose that the failure occurs in a path used by clients with restricted service level agreements (SLA) (i.e., rigorous QoS demands). If the restoration/protection mechanism tries to reroute these important flows to the previously established alternative LSPs far away from the location of the failure, this can produce blocking problems in the other nodes (LSRs), which have not been involved in the failure.

For simplicity in the following example we consider only link failures. However, our proposal can also be used for node failure restoration without any additional modification.

In Figure 6.2 the MPLS domain is divided into three SPDs. Although Figure 6.2 seems to be a simple network topology, it represents the abstraction of a much more complicated concatenation of autonomous systems (AS) represented as segment protection domains (SPD). Note that each link in the figure may traverse one or more LSR, which are not shown in the figure. Border LSRs are in charge of rerouting in case of failure.

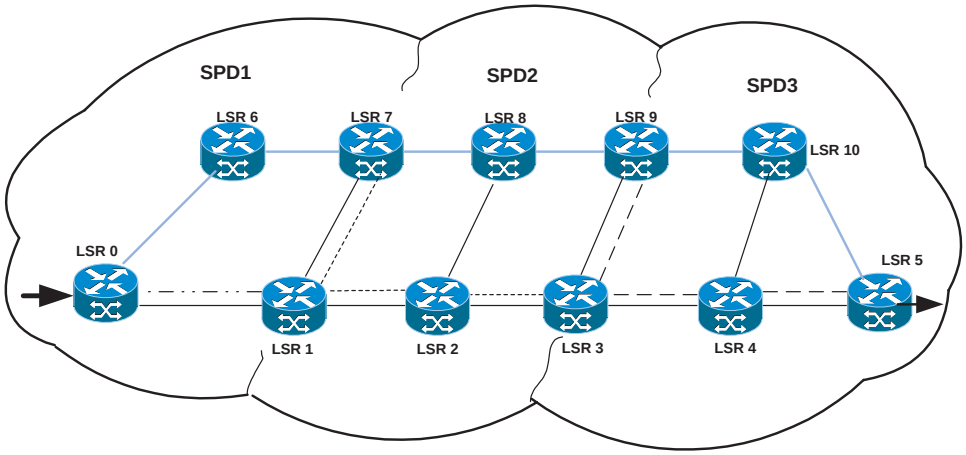

Figure 6.2 MPLS domain

We establish the primary LSP, and using explicit routing we set up the backward and alternative LSPs for path protection in each segment protection domain. The concatenation of the protected LSPs and backward LSPs for the SPDs makes the protected LSP and backward LSP for the entire MPLS domain respectively. The alternative path for the entire MPLS domain is made by concatenation of some portions of SPDs alternative LSPs.

A protection domain is denoted by specifying the protected LSP and the alternative LSP (protected LSP, alternative LSP)[OSMH01]. Using this definition and notation the entire MPLS protection domain (MPD) and all paths in Figure 6.2 are represented as follows.

Ingress LSR 0, Egress LSR 5.

Protected LSP (Primary LSP): the set of LSRs 0-1-2-3-4-5 (solid lines)

Preplanned Alternative LSP: the set of LSRs 0-6-7-8-9-10-5 (dim lines)

MPLS protection domain (MPD): (0-1-2-3-4-5, 0-6-7-8-9-10-5)

Segment Protection Domain 1: (0-1, 0-6-7-1)

Segment Protection Domain 2: (1-2-3, 1-7-8-9-3)

Segment Protection Domain 3: (3-4-5, 3-9-10-5)

Backward LSP for SPD1: the set of LSRs 1-0 (dash-dotted line)

Backward LSP for SPD2: the set of LSRs 3-2-1 (dotted lines)

Backward LSP for SPD3: the set of LSRs 5-4-3 (dashed lines)

During the recovery process the protection LSP is formed by concatenation of the following two portions: the backward LSP starting from the LSR that detects the failure (alert LSR), and the preplanned alternative protection LSP. Note that the use of the backward LSP for protected traffic is transitory (i.e., only during the recovery period). It is used to transport the packets routed on the faulty LSP from the LSR that detects the fault to the LSR responsible for redirecting this traffic. This minimizes packet losses.

Within a segment protection domain any kind of protection technique may be applied independent of other segment domains.

To illustrate the mechanism let us consider the segment protection domain1 (SPD1) in Figure 6.2. Assume a link failure between LSR0 and LSR1. If the link protection scheme is applied the recovery path for entire SPD1 will be formed by the set of LSRs 0-6-7-1. If the path protection scheme is applied, the recovery path for entire SPD1 is formed also by the same LSRs 0-6-7-1.

We apply the same approaches to segment protection domain2 (SPD2) for a link failure between LSR1 and LSR2. In case of link protection, the recovery path for the entire SPD2 (i.e., link protection plus the remaining path segment within SPD2) will

be formed by the set of LSRs 1-7-8-2-3. In case of path protection, the recovery path for entire SPD2 will be formed by the set of LSRs 1-7-8-9-3.

In the case of SPD3 for a link failure between LSR3 and LSR4, applying the link protection scheme the recovery path for the entire segment domain will be formed by the set of LSRs 3-9-10-4-5, while for the path protection scheme the result is the set of LSRs 3-9-10-5. In this case the path protection scheme provides a shorter recovery path than link protection.

As we stated before, our proposal combines the path protection scheme with the segment protection scheme, plus local repair techniques using the preplanned alternative LSP for protected LSPs in the entire MPLS domain. Once the preplanned alternative LSP for entire MPLS domain is setup, the segment protection for each SPD works in combination with this preplanned alternative LSP. This is possible because the alternative path for the entire MPLS domain is made by concatenation of some portions of SPD alternative paths. The first intersection point for both protections (i.e., the path protection for the entire MPLS domain and each segment protection domain) will be the merging point of the traffic rerouted by each SPD into the preplanned alternative LSP. This scheme uses link or path protection within the SPD to forward the packets to the egress LSR (LSR5) of the entire MPLS domain instead of forwarding to the corresponding segment domain egress LSR.

Let us apply the proposal to the previous example, Figure 6.2. If the link between LSR0 and LSR1 fails, the LSR0 reroutes the traffic using the alternative path of SPD1. The first intersection point for the alternative path of SPD1 (0-6-7-1) and the preplanned alternative path for the entire MPLS domain (0-6-7-8-9-10-5) is LSR0. From this merging point the traffic rerouted by SPD1 uses the preplanned alternative LSP. Then, the recovery path for the entire MPLS domain will be formed by LSRs 0-6-7-8-9-10-5. For a failure on link LSR1-LSR2 in SPD2, using the same principle, the first intersection between (1-7-8–9-3) and (0-6-7-8-9-10-5) is LSR7. Then, the recovery path for the entire MPLS domain is formed by LSRs 0-1-7-8-9-10-5. Finally for failure on link LSR3-LSR4 in SPD3, the alternative paths (3-9-10-5) and (0-6-7-

8-9-10-5) coincide on LSR9, and the recovery path will be formed by the set of LSRs 0-1-2-3-9-10-5 (see Table 6.1).

Faulty link	Link protection	Path protection within SPD	Proposal
LSR0-LSR1 in SPD1	0-6-7-1-2-3-4-5 7 links	0-6-7-1-2-3-4-5 7 links	0-6-7-8-9-10-5 6 links
LSR1-LSR2 in SPD2	0-1-7-8-2-3-4-5 7 links	0-1-7-8-9-3-4-5 7 links	0-1-7-8-9-10-5 6 links
LSR3-LSR4 in SPD3	0-1-2-3-9-10-4-5 7 links	0-1-2-3-9-10-5 6 links	0-1-2-3-9-10-5 6 links

Table 6.1 Comparison of restoration path length for single failure for MPLS protection domain (from ingress LSR0 to egress LSR5)

In Figure 6.2, the original end-to-end protected LSP length is 5 links (0-1-2-3-4-5). In Table 6.1, we present the comparison of the recovery path length from the ingress LSR to the egress LSR for single failures. Our proposal provides a shorter recovery path length compared with other approaches. The approach of applying segment protection with global path protection is better than applying segment protection or path protection separately. Moreover, as pointed out by numerous research papers, usually local repair may lead to the use of a non-optimal alternative LSP compared to the possible alternative LSP which can be established from the ingress LSR to egress LSR (Table 2.1). But, using our proposal we reduce the possibility of establishing non-optimal alternative LSPs from the point of failure to the egress LSR because we merge the packets rerouted to the alternative LSP (made by the local repair decision) into the preplanned alternative LSP (calculated by global repair). The use of this label merging technique [RVC01] allows the proposed scheme to avoid the backhauling problem.

6.4 DESCRIPTION OF THE PROPOSED MECHANISM FOR MULTIPLE FAILURES ON AN LSP

In the previous section we described the use of our proposal for a single failure. Here we present the explanation of the proposal for multiple failures. Multiple failures are considered to be the result of multiple single failures in the protected LSP. Applying the same principle used for single failures described in the previous section we are able to extend single failure protection to handle multiple failure protection.

According to the proposal in [OSMH01], the authors offer the possibility of handling multiple failures in an LSP by redirecting traffic from failed LSPs to the alternative LSP, but this approach has the disadvantage of excessive packet losses (i.e., all traffic on the protected path between the ingress node and the far extreme of the failed node/link). The node next to the failed link signals the event to the upstream nodes. Upon the reception of the failure signal the ingress node reroutes the traffic over the pre-established alternative LSP.

To illustrate how our proposal works, we will compare its behavior with Makam's and Haskin's. As an example, we consider a multiple failure on the protected LSP (LSR0-LSR1-LSR2-LSR3-LSR4-LSR5) as a combination of 3 link failures: LSR4-LSR5, LSR2-LSR3 and LSR0-LSR1.

Makam's proposal loses all the packets circulating on the LSP, and the ingress LSR (LSR0) redirects the incoming traffic to the alternative LSP. The same happens with Haskin's proposal in this condition. But, if we consider only the failures between LSR4-LSR5 and LSR2-LSR3 for the MPLS domain formed only by SPD2 and SPD3 (i.e., the LSP formed from LSR1 to LSR5), Haskin's proposal at least recovers packets traversing on the link LSR1- LSR2, while Makam's proposal loses all packets on the LSP plus additional packets sent to the already failed LSP before the notification message reaches the ingress LSR (LSR1).

In our proposal, if RFR is not used we lose only the packets on the failed link because the ingress LSRs in each segment protection domain (LSR0, LSR1 and LSR3) redirect the traffic to the alternative LSP. When link LSR4-LSR5 fails, LSR3 (being the ingress LSR of SPD3) redirects traffic through LSR3-LSR9-LSR10-LSR5. When link LSR2-LSR3 fails, LSR1 redirects traffic to the alternative LSP for SPD2 (LSR1-LSR7-LSR8-LSR9-LSR3). Furthermore, if we apply the proposal presented in Chapter 4 (Reliable and Fast Rerouting), we do not lose any packets.

Note that we assume that the multiple failures are produced at the same time. It is evident this is not the worst condition. The worst condition is produced when the sequence of link failure is LSR4-LSR5, and then LSR2-LSR3 and finally LSR0-LSR1. More precisely, the worst condition occurs as follows. Once the link LSR4-LSR5 has failed and the notification message in case of Makam's scheme or the reverse packet in case of Haskin's proposal is approaching LSR2, just before it reaches LSR2 the link LSR2-LSR3 fails.

Other situations are an intermediate of these extreme conditions. For example, if the link LSR0-LSR1 fails first, and link LSR4-LSR5 fails later, both Haskin's and Makam's schemes behave equally. They lose all packets traveling from LSR1 to LSR4 in addition to the packet losses on the faulty links.

Based on the segment protection approach, if we try to protect the entire protected path (i.e., from LSR0 to LSR5) from a link failure in each SPD (i.e., multiple link failure within the protected path) the recovery path length increases with (repeated link or path protection) within SPDs.

One important observation is that the recovery path length always increases when the link protection scheme is used. On the other hand, the path protection scheme does not always increase the length of the recovery path. The length of the protection path is considered to be a main quantitative measure of the quality of a protection scheme [BR02]. The protection path length can be used as an indication of the delay that the rerouted traffic will experience after a link failure. In addition to the delay,

Faulty links	Link protection	Path protection	Proposal
1-2 and 3-4 in SPD2 and SPD3	0-1-7-8-2-3-9-10-4-5 9 links	0-1-7-8-9-3-9-10-5 8 links	0-1-7-8-9-10-5 6 links
0-1, 1-2 and 3-4 in SPD1, SPD2 and SPD3	0-6-7-1-7-8-2-3-9-10-4-5 11 links	0-6-7-1-7-8-9-3-9-10-5 10 links	0-6-7-8-9-10-5 6 links

Table 6.2 Comparison of restoration path length for multiple failures for MPLS protection domain (from ingress LSR0 to egress LSR5)

the length of the protection path reflects the amount of resources required to protect an LSP.

In Table 6.2 we summarize the restoration path length used by link protection, path protection and our proposal for the entire MPLS domain (end-to-end) for multiple failures based on the network scenario of Figure 6.2. We can observe that our proposal needs only 6 links for a recovery path, performing better than separate link and path protection approaches. The protected LSP length is equal to 5 links. Note that the fact that the proposal recovery path length is one link more than the protected link length is not due to the proposed mechanism. It is simply because the possible alternative LSP found to protect the original protected LSP is one link more than the original (i.e., 6 links).

6.5 SIMULATIONS AND RESULTS

The objective of this simulation is to compare numerically the behavior of this proposal with the reference proposals: Haskin's and Makam's.

The MNS source code was modified to simulate these mechanisms: Haskin's [HK00], Makam's [OSMH01] and our proposal. The failures of links between LSR4-LSR5 and

LSR0-LSR1 are used as the separated single link failures. For multiple failures we use the failures between LSR4-LSR5 and LSR2-LSR3. The simulation scenario is the one shown in Figure 6.2.

We use CBR traffic with the following characteristics: packet size = 1600 bits and source rate= 400Kbps. In all cases path protection is applied for the entire MPLS domain, thus satisfying the requirement of Haskin's and Makam's proposals.

We measured packet loss, packet re-ordering and repeated packets at the egress node (LSR5) for a single failure, multiple failures with path protection, and multiple failures with combined path and segment protection. The figures show all simulation results: packets lost and disordered during the recovery period.

In reference to the simulation results behavior, we use 100% packet loss and packet re-ordering in the the LSR4-LSR5 link failure situation because in this situation there is maximum packet loss for Makam's scheme and maximum packet disorder for Haskin's scheme in the simulation results. The results presented in the figures are proportionally identical when the LSP length, the LSP bandwidth, the packet size and the source rate are varied. Note that both Haskin's and Makam's proposals use path protection schemes establishing the preplanned alternative LSP from the ingress LSR (LSR0).

In the following figures the proposal includes RFR with buffering at the LSR in order to avoid packet losses. It is labelled as "proposal +".

Figure 6.3 shows the results for a single failure without segment protection. Makam's scheme [OSMH01] uses a notification message to the ingress node after a failure to reroute traffic from the ingress LSR to a previously established alternative LSP, resulting in high packet loss and no packet re-ordering . Whereas, Haskin's [HK00] returns packets from the faulty point to the ingress LSR and there reroutes them to the alternative LSP together with the incoming traffic, resulting in minimum packet loss, and maximum packet disorder proportional to the distance (number of LSR) between the ingress LSR and alert LSR.

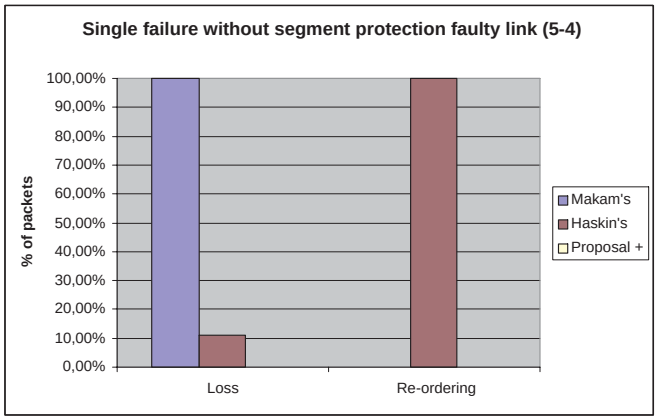

Figure 6.3 Performance comparison results during recovery period for packet losses, packet disorder

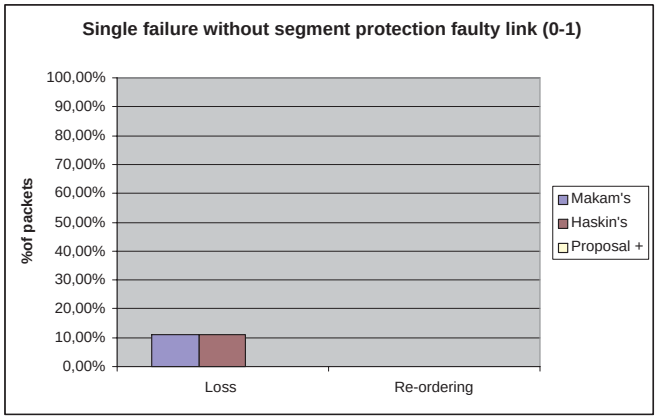

Figure 6.4 Performance comparison results during recovery period for packet losses, packet disorder

Figure 6.4 shows the results for a single failure without segment protection (failed link LSR0-LSR1). Both Haskin's and Makam's behave the same (they lose only the packets on the failed link). Note that in both figures (Figure 6.3 and Figure 6.4) our proposal does not experience packet loss or disorder.

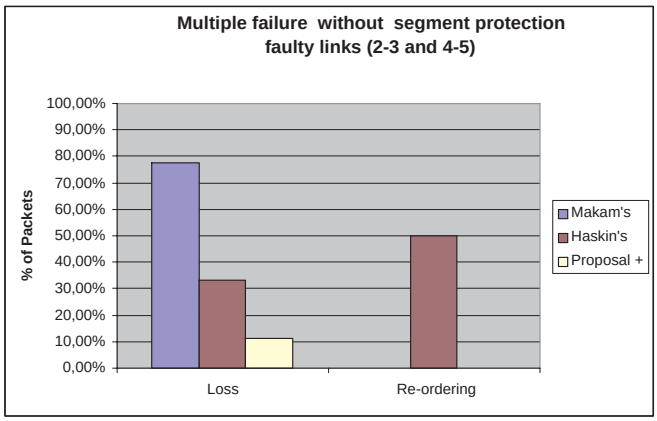

Figure 6.5 Performance comparison results during recovery period for packet losses, packet disorder

In Figure 6.5 the results for multiple failure without segment protection (failed links LSR2-LSR3 and LSR4-LSR5) are depicted. The packet loss for Makam's scheme decreases with respect to the result in Figure 6.3 and increases with respect to the result in Figure 6.4 because the point of failure is closer to and farther from the ingress node (responsible to redirect the traffic) respectively. This is translated as less and more time that the notification signal takes to reach the ingress LSR (LSR0).

The packet loss increases for Haskin's. This is due to the fact that the LSP segment between the two extreme points of failure in the protected LSP becomes disconnected. Haskin's scheme recovers the packets traversing in the portion of the LSP between the ingress node and the point of failure (LSR0-LSR1-LSR2), and loses packets on the links formed by LSR2-LSR3-LSR4-LSR5. In this case our proposal begins to lose packets. Although we include the RFR proposal, we recover only the lost packets on the links formed by LSR2-LSR3 and LSR3-LSR4 from the LSR2 local buffer. We lose packets circulating on link formed by LSR4-LSR5. This is because we specified the buffer size equivalent to the packets circulating in two downstream links. Note that we can increase the buffer size to avoid the packet losses.

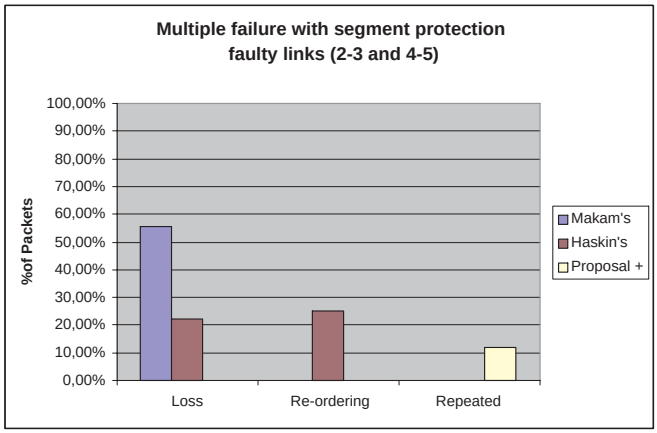

Figure 6.6 Performance comparison results during recovery period for packet losses, packet disorder and repeated packets

Figure 6.6 shows the results for multiple failures applying the combination of path protection with segment protection. The packet loss for Makam's scheme as well as the packet re-ordering for Haskin's experience an important reduction, improving the main drawback of each scheme. This is because the rerouting of traffic is performed close to the failure points, improving their performance. Our proposal using RFR performs better than the others by avoiding both packet loss and packet disorder.

We did extensive simulation with different scenarios and traffic patterns and the results show basically the same behavior. Results presented in the chapter are representative of the behavior of the proposal. Based on these results we believe that the combination of path and segment protection with the local repair method is the best option as a protection mechanism against multiple/single failure for protected traffic on MPLS-based networks. The most complex element of our proposed scheme is to set up all of the alternative LSPs required.

6.6 SUMMARY

The proposed mechanism covers many of the aspects of IP-QoS provision. The proposal provides protection from multiple link/node failure in a protected LSP on an MPLS-based network using a combination of path protection with segment protection and local repair. Rerouting of traffic is performed close to the failure point, increasing the restoration speed and providing a significant reduction of the LSP blocking problem. At the same time it provides better recovery (protection) in terms of path length. As a result, we achieve better network resource utilization and shorter delays for rerouted traffic.

The criteria for partitioning an MPLS domain into several segment protection domains may be established according network administration policies.

The main open issue is how to compute the alternative LSP for each segment protection domain (SPD), and then to identify the merging point in order to select the shortest path. The routing algorithm must establish a global protected LSP and a global alternative LSP for the entire MPLS domain. For each SPD, the algorithm will also establish a global alternative with a merge point with the global alternative for the entire MPLS domain. Several of the proposed routing algorithms might be adopted to find all possible LSPs.

7

MECHANISM FOR OPTIMAL AND GUARANTEED ALTERNATIVE PATH (OGAP)

7.1 INTRODUCTION

As we described in earlier chapters, our proposal uses the preplanned (pre-established) alternative LSP to provide a fast and reliable restoration mechanism for single and multiple failures in MPLS-based networks. However, the preplanned protection scheme can have a risk that the preplanned alternative LSP will become out of date due to changes in the network. By out of date we mean that as network conditions evolve in time the preplanned alternative LSP may cease to be the optimal one. Moreover, after the restoration process, the restored LSP becomes unprotected.

The motivation of this study is to overcome these problems and propose a new mechanism:

i) To establish the updated optimal alternative LSP and

ii) To maintain always at least one alternative LSP for the protected LSP at any time.

7.2 PROPOSED MECHANISM

To overcome this problem we propose to search for a new alternative LSP with updated network information concurrently while rerouting the traffic to the preplanned alternative LSP. Note that a long restoration time is a main problem of a dynamic restoration scheme but this does not apply to our proposal because the protected traffic is rerouted to the alternative LSP using the preplanned alternative LSP. It is worthwhile to consider that inconsistencies in the routing database may exist, which would have a negative effect on the new alternative LSP calculation during the recovery period. To minimize this effect we use the algorithms proposed in [MSSD][MSSD02].

The idea behind this *hybrid approach* is to take advantage of the fast rerouting and the rerouting (dynamic) scheme [SH02]. At the same time our proposal provides a guarantee of an alternative LSP at any time for the protected LSP. It is important to note that, as far as we know, no one before has considered the protection of the alternative LSP once the traffic is rerouted on it. In other words, almost all proposals address a single failure situation. In our case we consider multiple failures on an LSP and also the failure of the new protected LSP (i.e., the old alternative LSP or the newly established optimal path). We also consider the reversion operation. The reversion consists of rerouting the traffic from the alternative LSP to the original protected LSP once the failure has been repaired. To do this, we first wait a certain amount of time before releasing the primary LSP, and then compute a new alternative LSP after a failure, which we can use if and only if the result of the LSP using the repaired link is better than that of the LSP which carries the rerouted traffic (i.e., the new protected LSP). Note that the repaired link announces its link status information as zero bandwidth usage (i.e., advertised cost is zero) [RM01]. Therefore, it is possible that this link may become overloaded if the rerouting point is far from the point of failure. If this is the case, it is impossible to return to the old LSP simply because there isn't sufficient bandwidth to accommodate the traffic.

Moreover, our proposal avoids the update of the alternative LSP each time the information database of the network changes. The update is done only when a failure occurs.

7.3 ALGORITHM DESCRIPTION

Figure 7.1 presents the flow diagram of the proposed mechanism. While no failure is detected in the protected LSP, each LSR continues carrying traffic through the protected LSP. Upon a failure the LSR which detects the failure (alert LSR) or one that receives protected traffic on the backward LSP looks for the preplanned alternative LSP in its label information base forwarding table (LIB). If the LSR is an ingress node for the SPD it should have an alternative LSP available. Otherwise, if the LSR is an intermediate node it must follow the RFR procedure described in Chapter 4. If an alternative LSP is found, then it redirects the traffic from the affected protected LSP to the preplanned alternative LSP and it computes a new alternative path using the network conditions at that time.

If the path discovery and selection algorithm gives us a new alternative LSP we compare it with the one that was established previously as the preplanned alternative LSP. If the new alternative LSP is better than the preplanned one, the traffic will be redirected to the new alternative LSP without disruption of services (using the principle of *make-before-break*). The criteria for considering a path "better" may be based on the length of the path and other QoS parameters. The LSR maintains in its LIB the same preplanned alternative LSP as before, and proceeds to setup the backward LSP for the new protected LSP.

If the result is "not better" (i.e., the previously established preplanned alternative LSP is better than the new alternative LSP computed by the LSR after the failure) we assign the new alternative LSP as the preplanned alternative LSP and proceed to set up the backward LSP for new protected LSP.

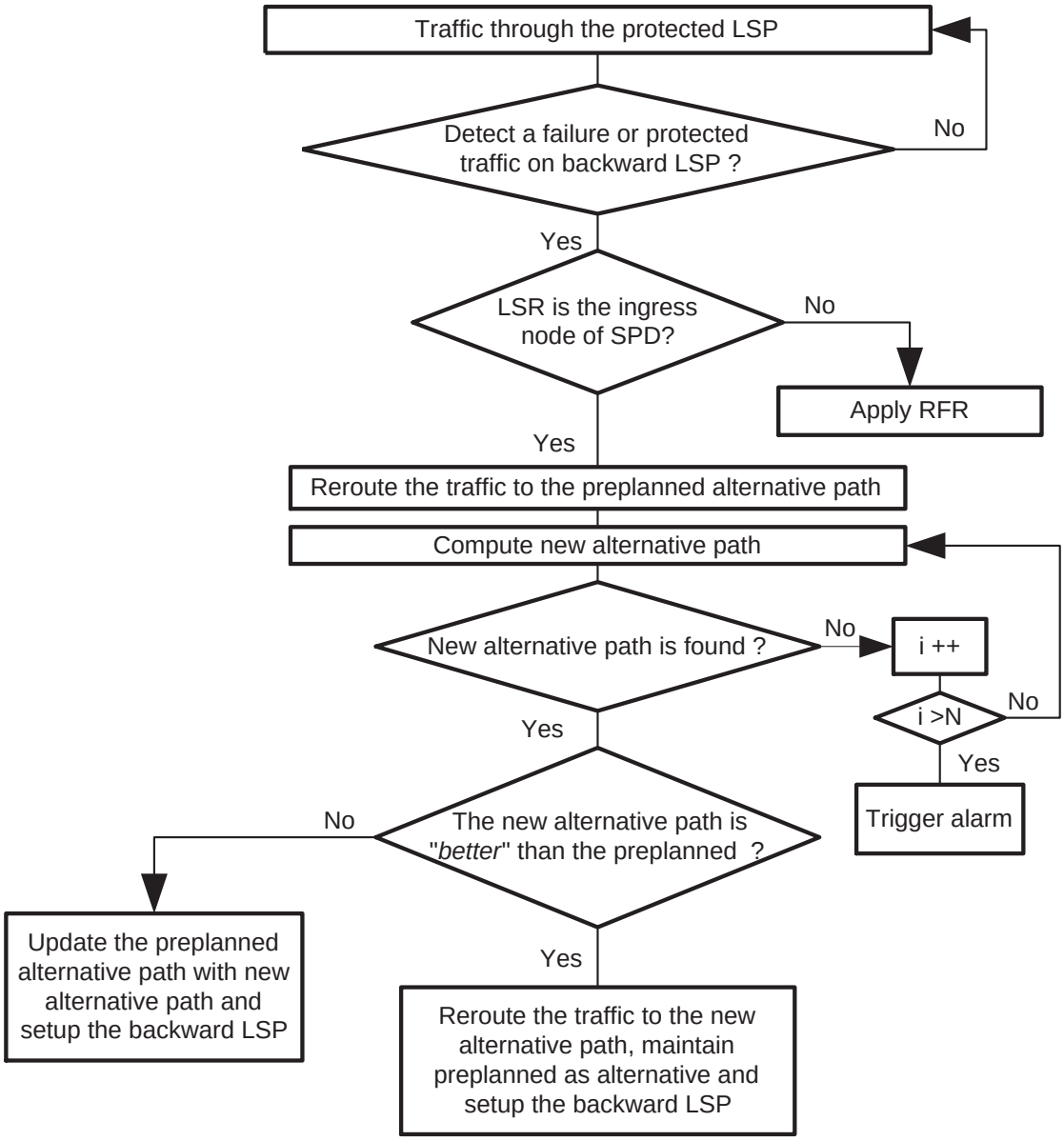

Figure 7.1 Flow diagram

If the routing algorithm is not able to find a new alternative path in the first attempt, we increment the iteration until its value (i) is greater than the control value established previously (N). This value (N) is determined by the network manager and it is a local implementation. If this iteration terminates without finding a new alternative path an alarm is sent to the network control manager to take appropriate measures.

7.4 RESULTS

Table 8.3 summarizes the pros and cons of the different protection schemes for LSPs. Some parameters correspond to QoS provision and others to network resource utilization and feasibility.

The last column refers to the proposal presented in this chapter combined with the previously proposed Reliable and Fast Rerouting mechanism (RFR) presented in Chapter 4.

Performance measurement	Haskin	Makam	OGAP	OGAP + RFR
Complexity	Low	High	Low	Low
Path placement	Restricted	Restricted	Flexible	Flexible
Restoration time	Fast	Slow	Fast	Fast
Packet Loss	Minimum	High	Minimum	None
Packet Re-ordering	High	Minimum	High	None
Resource Requirements	High	Low	Medium	Medium
Optimal path option	No	No	Yes	Yes
Protection for protected LSP	One Alternative	One Alternative	New Alternative Set-up	New Alternative Set-up

Table 7.1 Comparison of MPLS protection schemes

Although most of the concepts shown have been explained already, we would like to clarify some of them.

In the path placement row, unlike others, our proposal is flexible in the sense that the previously established alternative LSP can be changed to a new optimal alternative LSP computed using the rerouting (dynamic) scheme. Other proposals maintain the

same alternative LSP set up during the establishment of the protected LSP to reroute the traffic.

The packet loss and packet reordering values in our case are "none" because we incorporate in this proposal our Reliable and Fast Rerouting mechanism presented in Chapter 4.

Finally, in the last row we try to give the protection range not in terms of the amount of failure points on the protected LSP, but in the ability to handle further failures in the rerouted path. In our case as we establish a new alternative LSP to the rerouted path, our mechanism is able to handle further failures. For Haskin's and Makam's schemes, as they do not establish new alternative LSPs to the rerouted LSP, they only protect the first protected LSP (i.e., they handle only single failures).

7.5 SUMMARY

One of the disadvantages of using a preplanned alternative LSP is that it may not be the optimal one when needed (i.e., at the time of failure). To overcome this disadvantage we propose a hybrid approach OGAP (i.e., preplanned and dynamic rerouting) capable of identifying and using the optimal alternative path based on recent network change information (i.e., after the fault was detected). This avoids the possible use of a non-optimal alternative LSP to reroute the protected traffic and provides the flexibility of alternative route selection and setup as well as better resource utilization. Moreover, our proposal guarantees at least one alternative LSP at any time for the traffic on the protected LSP.

8

ADAPTIVE LSP

8.1 INTRODUCTION

As the Internet has evolved from its research origins into a popular consumer technology, network resource management has become a main problem for service providers. It was thought that the solution to the problem would be new technologies capable of providing sufficient network resources like cheap memory, high-speed links and high-speed processors. Though these improvements contribute significantly to enhancing traffic performance, they do not solve the problem of optimal use of network resources. One of the most important functions performed by the Internet is the routing of traffic from ingress nodes to egress nodes. The most commonly used shortest path routing protocol chooses as a preference the shortest link to forward packets. This ignores performance information which forces communication over excessively long or overloaded links leading to non-optimal path selection or unbalanced network load situations. Therefore, one of the main tasks to be performed by Internet Traffic Engineering (TE) is the control and optimization of routing functions to forward

traffic through the network in the most effective way [ACE+02] [AMA+99]. Thus, the main focus of Internet TE is to facilitate efficient and reliable network operations while simultaneously optimizing network resource utilization and traffic performance.

The optimization objective of Internet traffic engineering should be viewed as a continual and iterative process of network resource utilization improvement. Different networks may have different optimization objectives depending on the network utility models. However, in general, TE optimization focuses on network control regardless of the specific optimization objectives. One major challenge of Internet TE is the realization of automated control capabilities that adapt to significant changes quickly and cost effectively, while still maintaining stability.

MPLS traffic engineering provides an integrated approach to TE. It routes traffic flows across a network based on the resources the traffic flow requires and the resources available in the network. It also employs "constraint-based routing" in which the path or Label Switching Path (LSP) for a traffic flow is the shortest path that meets the resource requirements (constraints) of the traffic flow.

The Label Distribution Protocol (LDP) is in charge of setting up an LSP with a given maximum bandwidth. While the demand bandwidth of the aggregated flows in a particular LSP is less than or equal to the maximum bandwidth assigned to this LSP, the Label Edge Router (LER) continues sending traffic to the established LSP without any problem. The problem arises when the bandwidth demand for the aggregated flows becomes greater than the maximum assigned capacity. Obviously, the traffic demand changes over time but the topology and routing configuration cannot be changed as rapidly. This causes the network topology and routing configuration to become sub-optimal over time, which may result in persistent congestion problems on the LSP. The other issue occurs when the reservable bandwidth in a link on the shortest path (optimal connection) does not meet the bandwidth constraint for the new demands. This situation obliges the routing protocols to select a non-optimal LSP. Note that the reservable bandwidth of a link is equal to its capacity minus the

total bandwidth reserved by LSPs traversing the link. It does not depend on the actual amount of available bandwidth on that link.

In the literature there are different approaches suggested to tackle these network problems. We summarize them as follows.

1. Traffic losses.
2. Create new LSP with more maximum BW (BWmax) and reroute all aggregated traffic on it.
3. Use traffic engineering to split traffic onto a new LSP.
4. Modify the LSP bandwidth, if possible [ALAS+02].

The first option simply decides to drop the excess traffic to control the congestion in the network. This can't be applied any traffic with QoS requirements. The solution is simple, but it is not appropriate for critical traffic.

Options (2) and (3) introduce additional overhead by extra signaling processes to establish the new LSP, but they are transparent to traffic. Option 3 in particular has a problem with regard to network scalability, which is inversely proportional to the number of labels used by an LER to forward the same amount of traffic.

The last option, proposed by Ash et al. [ALAS+02], modifies the bandwidth of an established LSP using CR-LDP. It is transparent to traffic. Here too extra signaling for the additional bandwidth request is required.

All the above mentioned options address the problem of satisfying only the traffic requirement (additional bandwidth demand) triggered by some congestion problems on the network. Other aspects of performance and resource optimization that are not considered are i) to find an optimal LSP and reroute the traffic when there is traffic reduction, and ii) to reroute traffic from a non-optimal LSP to a better LSP when a previously established LSP is released.

The objective of this proposal is to contribute significantly to the improvement of MPLS traffic engineering considering the two performance aspects mentioned above.

8.2 RELATED WORK

In [AMA+99] the authors present a set of requirements for traffic engineering over MPLS, identifying the functional capabilities required to implement policies that facilitate TE in an MPLS domain. They classify the TE performance objectives mainly in two groups: traffic oriented and resource oriented. The first strives to enhance the QoS of traffic streams (packet loss, delay, delay variation, and goodput). The second deals with optimization and efficient network resource allocation and utilization. In [Awd99] the author defines the basic components of the MPLS traffic engineering model: path management, traffic assignment, network state information dissemination and network management.

The MPLS adaptive traffic engineering (MATE) presented in [EJLW01] addresses the network congestion problems using a multipath adaptive traffic engineering mechanism. The mechanism assumes that several explicit LSPs are set between an ingress and an egress node using a standard protocol such as CR-LDP [JAC+02]or RSVP-TE [ABG+01], or configured manually. The proposed adaptive TE mechanism uses probe packets to obtain LSP statistics such as packet delay and packet losses in order to shift traffic among LSPs. Here we clearly see that the MATE mechanism is not capable of modifying the LSP bandwidth to accommodate additional demands when all established LSPs reach their maximum reserved bandwidth. At the same time, the typical range between end nodes proposed in the MATE operational settings (from two to five explicit parallel LSPs) continues reserving the same amount of maximum bandwidth even if the traffic decreases drastically in all LSPs.

In [Swa99] two further optimization strategies are suggested. The first uses multiple LSPs to each destination - like MATE - to balance the load. But, instead of sending a probe packet to monitor the utilization of each LSP, it uses link utilization informa-

tion by extending ISIS or OSPF. The second approach attempts to auto-adjust the bandwidth based on the real usage of an LSP.

In the proposal of Ash et al. [ALAS+02] the authors address the problem related to additional bandwidth requirements for the traffic carried on an LSP. The work presents an approach modifying the bandwidth of an established LSP using CR-LDP without service interruption. The proposed mechanism not only addresses the increase of bandwidth to accommodate the new bandwidth demand, it also includes the possibility to decrease LSP bandwidth when the traffic on the LSP has decreased. In this case, their method releases the delta (difference) bandwidth (ΔBW) and continues using the same LSP.

It is clear that some proposed mechanisms try to solve the congestion problems while others try to accommodate additional bandwidth demands. However, they don't cover the re-optimization of the previously established LSP by rerouting to optimal paths after significant changes in the network occur, such as a reduction of traffic on an LSP or the release of an LSP.

8.3 PROBLEM FORMULATION

We will use an example in a simple scenario in order to illustrate the problem to be solved. In Figure 8.1 we present a simple MPLS network scenario formed by four LERs as edge routers and six intermediate or transit LSRs. We also have four Autonomous Systems (AS) A, B, C and D connected to the MPLS network. In this example we establish the full mesh connection between these four LERs, and we analyze the operation for optimal LSPs and non-optimal LSPs to see the impact on resource utilization.

Building the full mesh of LSPs according to the shortest path (optimal) gives the configuration depicted in Table 8.1.

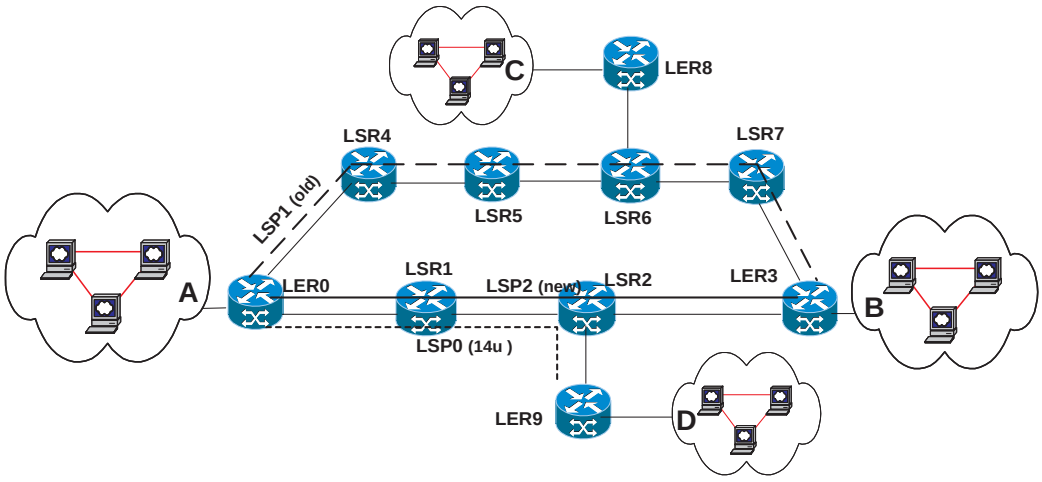

Figure 8.1 Scenario

Note that the maximum number of LSPs that can be established in the network is n*(n-1), where n is the number of LERs. In our case, as there are four LERs the total number of LSPs will be 4*(4-1) =12.

Building the full mesh of LSPs according to the non-optimal (non-shortest) path gives the following configuration. (Table 8.2.)

	A	B	C	D
A	——	0-1-2-3	0-4-5-6-8	0-1-2-9
B	3-2-1-0	——	3-7-6-8	3-2-9
C	8-6-5-4-0	8-6-7-3	——	8-6-7-3-2-9
D	9-2-1-0	9-2-3	9-2-3-7-6-8	——

Table 8.1 Full mesh optimal connection using shortest path algorithm

	A	B	C	D
A	——	0-4-5-6-7-3	0-1-2-3-7-6-8	0-4-5-6-7-3-2-9
B	3-7-6-5-4-0	——	3-2-1-0-4-5-6-8	3-7-6-5-4-0-1-2-9
C	8-6-7-3-2-1-0	8-6-5-4-0-1-2-3	——	8-6-5-4-0-1-2-9
D	9-2-3-7-6-5-4-0	9-2-1-0-4-5-6-7-3	9-2-1-0-4-5-6-8	——

Table 8.2 Full mesh with non-optimal connection

Table 8.3 presents a comparison of the number of LSPs in each link when non-optimal routing is used with respect to the optimal (shortest path) routing.

		Number of LSP per link	
Links	Number of links	Optimal	Non-optimal
0-4, 4-5, 5-6	3	2	10
0-1, 1-2, 6-7, 7-3	4	4	8
2-3, 2-9, 6-8	3	6	6

Table 8.3 Comparison table for fully optimal and non-optimal LSP connection

In the first row of the table we can observe that three network links (0-4, 4-5, 5-6) are shared by 10 LSPs for non-optimal routing, reporting the maximum number of LSPs per link. In the case of optimal routing these three links are shared by only 2 LSPs. The last row reports the maximum LSPs per link for optimal routing, which is 6 LSPs. Those links that are shared by the highest number of LSPs are to be considered to be "critical links" in the network [KL00].

Following this example and Table 8.3 we find that for non-optimal cases there are 10 LSPs in the "critical links", while for the optimal case there are 6 LSPs in the "critical links". Considering all links to be identical, with the bandwidth capacity of

C (link capacity), and all LSP have the same bandwidth assigned, the maximum link bottleneck in the network for optimal routing is equal to C/6 and for the non-optimal is C/10. As C/6 > C/10 we get better network resource utilization for the optimal rerouting. This condition causes us to look for a mechanism for rerouting non-optimal LSPs.

The second aspect we want to illustrate in this section is an example of the operation for increasing the bandwidth of an LSP. The scenario is the one depicted in Figure 8.1. We assume that all links have the same delay and a link capacity of 20 units (C=20).

We define a flow f(i,AS) as the flow number i from autonomous system AS. A Forward Equivalence Class (FEC) Fi corresponds to LERi as the destination node (egress LER) to leave the network.

Consider also that the path LER9-LSR2-LSR1-LER0 is occupied by flows from D to A, with demand for 14 units bandwidth forming the LSP0 for packets classified by LER9 as FEC F0. This path is formed by 3 links, and the cost associated with it is 3.

Now, the available link capacity for the path between LER0 and LER3 through LER0-LSR1-LSR2-LER3, with cost 3, is 6 units. And for the same path through LER0-LSR4-LSR5-LSR6-LSR7-LER3, with cost 5, the available link capacity is 20 units. This situation is common in real networks because the shortest paths (path with less cost) are the preferred paths to be selected by routing protocols.

Suppose that now A sends a flow f(1, A) to B with 10 units bandwidth demand. According to the MPLS architecture the LER0 associates the f(1,A) to the FEC F3 (i.e., LER3 is the destination node to leave the network), and after, it sends the label request message with 10 units of bandwidth to the downstream LSRs. During this process there is not sufficient bandwidth to accommodate the traffic through path LER0-LSR1-LSR2-LER3 (it has only 6 units left). So, the attempt to establish

an LSP using this link for the request is rejected. On the other hand, the downstream LSRs through the path LER0-LSR4-LSR5-LSR6-LSR7-LER3 have 20 units of bandwidth available and accept the request and map the corresponding label to F3. When the label mapping message is received by LER0 the establishment of LSP1 is concluded.

Following the example, suppose now that after the establishment of LSP1 for flows classified by ingress LER0 as F3, A sends a flow f(2,A) to B with 2.5 units of bandwidth demand. This flow has as destination LER3. This implies that LER0 will classify it as F3 and will assign the same label as for f(1,A) and forward it through LSP1. Assume also that after this process A sends a new flow f(3,A) to B with 3 units of bandwidth demand. This flow also receives the same treatment by LER0. This situation increases the bandwidth usage of LSP1 to 15.5 units. The bandwidth of the LSP accommodates the new requirements as new flows are aggregated.

Now we illustrate how non-optimal routing may lead to blocking new requests. Continuing with the previous example, suppose that C attempts to send a flow f(1,C) to D with 6 units. Its request will be rejected by a downstream LSR (LSR6) due to the lack of available bandwidth on the outgoing link (because LSR6-LSR7 has only 4.5 units available), resulting in the rejection of this request producing the blocking problem.

Another aspect of the desired behavior of the network is the ability to reroute current LSPs in order to evolve towards a more optimal routing configuration.

Using the same example situation, assume that after a certain time the flow f(1,A) ceases. Flows f(2,A) with 2.5 units and f(3,A) with 3 units on the LSP1 remain (in total, the used bandwidth is now 5.5 units). Now there is enough available bandwidth through path LER0-LSR1-LSR2-LER3 (6 units) to accommodate these flows (5.5 units). And we believe it is better to forward the remaining aggregated traffic from A to B through LSP2 instead of continuing to do it via LSP1. Doing so, we will be able to dynamically manage network resource utilization, and at the same time reduce the

delay that packets experience by using the path with cost 5 (5 links) instead of cost 3 (3 links). There are many proposals addressing fast rerouting of LSPs without service interruption, so that the rerouting is not a major issue.

Finally, consider the case when all link capacities for low cost paths (optimal LSPs) are occupied by traffic with the same priority. In this case even using Ash's proposal it is impossible to modify (increase) the LSP bandwidth due to the link capacity being fully used. As a result the incoming traffic is forwarded over a high cost LSP. Suppose that after a while, the traffic over the low cost LSP ceases (the associated LSP is released). In this situation if we continue sending the traffic through the non-optimal (high cost) LSP, evidently we are wasting valuable network resources.

For example, suppose that when LSP1 reaches 15.5 units of bandwidth usage, after the aggregation of three flows with 10, 2.5 and 3 units, LSP0 is released. In this condition we are able to reroute the traffic from LSP1 (15.5 units) to an LSP that can be established through path LER0-LSR1-LSR2-LER3 with a capacity of 20 units. Note that although the bandwidth usage (BWu) is not less than the assigned bandwidth threshold (BWt), using this mechanism we are able to reroute the traffic from a non-optimal LSP to the optimal LSP, improving the overall performance of the MPLS network.

Note that the modification of LSP bandwidth proposed in [ALAS+02] also includes the possibility to decrease the LSP bandwidth when the aggregated traffic has decreased. In this case their method releases the bandwidth equal to the difference of bandwidth between current and previous aggregated flows (delta bandwidth) and continues using the same LSP.

8.4 ADAPTIVE LSP ROUTING

In this chapter we propose an additional functionality to the edge LSRs (LERs) introducing new criteria to overcome the problem derived from non-optimal routing

at LSP setup time and provide better performance and network resource utilization to MPLS based networks.

The ingress LER must store bandwidth requests (demand) and dynamically monitor the LSP bandwidth usage compared with the assigned threshold value. At the same time it watches for released LSPs on low cost paths to transfer the same priority traffic from high cost LSPs.

After the establishment of an LSP, the ingress LER continues forwarding packets as per MPLS architecture procedures. Our mechanism starts by storing the information of the LSPs initial aggregated bandwidth demand (BWid) in the LER. Then the LER starts to monitor the aggregate bandwidth usage (BWu). This is possible because an LER both establishes the LSP and forwards the traffic into it, so all information needed for our proposal is readily available within the LER. If BWu remains above the threshold value (BWt) no action will take place. The threshold is defined to be some reasonable percentage of the initially allocated aggregated bandwidth (i.e., BWt=X*BWid, where 0<X<1). When the actual usage (BWu) falls below the threshold value (BWu<BWt), the LER sends a label request message with capacity equal to the actual aggregate bandwidth usage (BWu) to establish a new LSP.

BWid indicates that there is no other available LSP with less cost to accommodate the initial bandwidth demand. On the other hand, it is easy to infer from this affirmation that it is possible to find another LSP with equal or less cost that may satisfy a bandwidth demand smaller than BWid.

Consider that the network status of other links in the network that do not belong to this LSP remain unchanged. Based on this assumption, the probability of finding a new LSP from the same ingress node to egress node for BWid with less cost than the actually established LSP (LSP1) is equal to zero $P_{low\,cost}(BWid) = 0$. And then, the probability of finding an LSP with equal or greater cost is equal to 1. The probability of getting a new LSP for less cost with less bandwidth than the initial bandwidth demand, $P_{low\,cost}$(BW<BWid), increases when we decrease the BW demand with respect

to BWid. For this reason, it is important that the network manager be responsible for attempting the assignment of the appropriate value to BWt and waiting time based on the statistical data of the network. If the margin for triggering our mechanism is set too close to BWid (BWt has a value close to BWid), the LER triggers an LSP setup for slight changes of BWu with respect to BWid. As the probability to establish a new LSP with a high bandwidth demand is low, the probability of finalizing this procedure without success is high. In other words, high bandwidth requests have less probability of establishing a new LSP.

In the proposal the ingress LER not only monitors the decrement of LSP bandwidth usage, but also watches the released low-cost LSPs. When the traffic over the low-cost LSP ceases, and the associated LSP is released, the ingress LER must be capable of transferring traffic on the high cost LSP to the released low cost LSP. This improves network resource utilization and provides better overall performance for the MPLS based networks.

8.5 PROPOSED ALGORITHM

Figure 8.2 presents the flow diagram of the proposed algorithm. Though the flow diagram by itself is a formal description, we describe below our algorithm. It is important to explain the additional tables we include in the LERs. Apart from the normal Label Information Base forwarding table (LIB), we maintain two additional tables.

The first new table corresponds to the first rejected LSP on the optimal path for each LSP that was established using a non-optimal path. The data stored in this table are the LSPID, FEC, bandwidth and attempted output interface (link). Note that we put all LSPs whose optimal path is blocked and that are therefore currently using non-optimal LSPs in the rejected LSP table. Obviously, if the path is impossible to establish at any cost and the request is totally rejected, it implies no LSP was

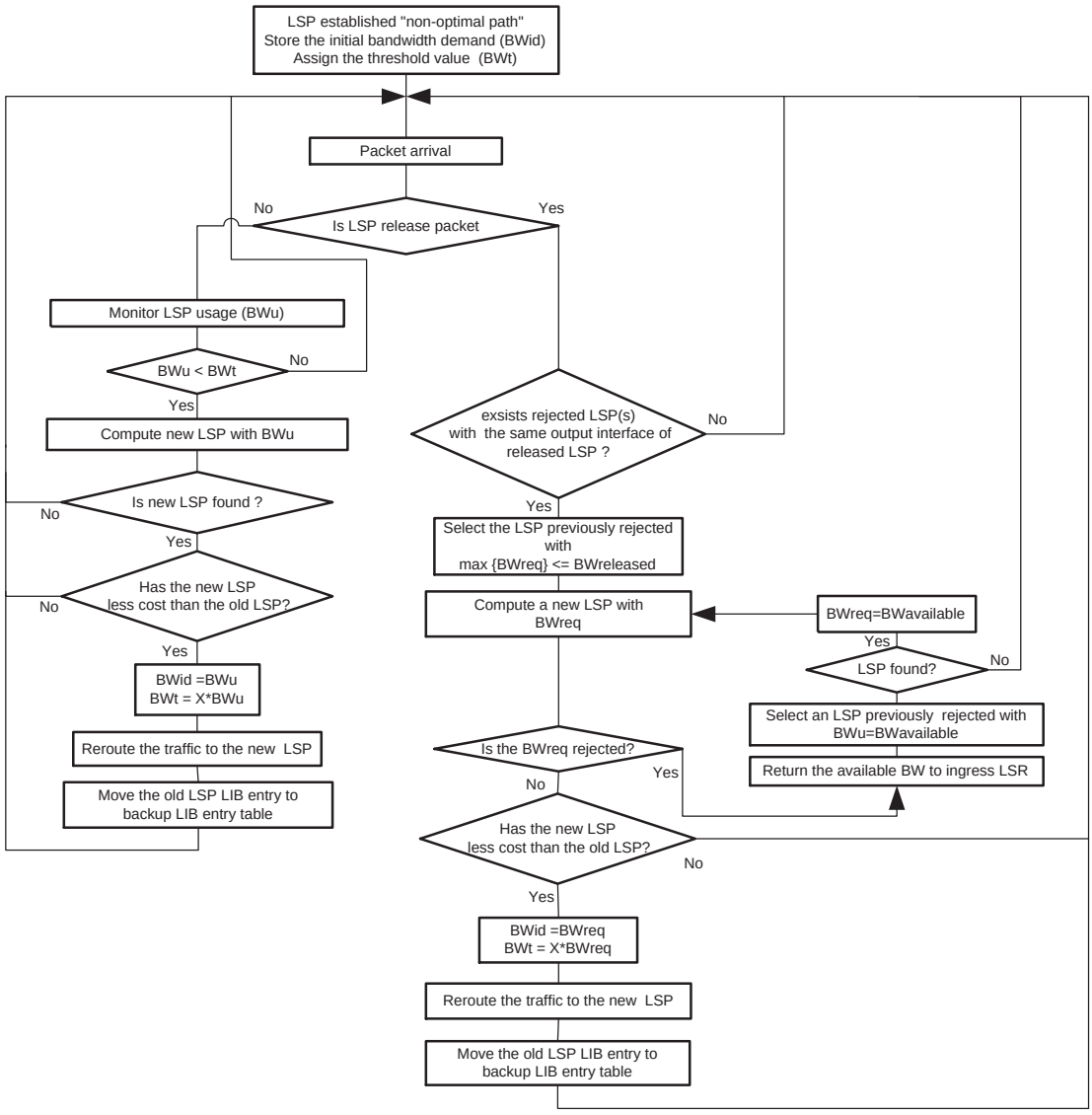

Figure 8.2 Flow diagram for proposed mechanism

established for this request: we ignore this and it is not included in the rejected LSP table.

The second table corresponds to a backup LSP information table of non-optimal rerouted LSPs, and we call this the "backup LIB entry table". In other words, it is the table entry formed by non-optimal LSPs removed from the LIB after traffic is

rerouted to an optimal LSP, with the only difference being that its bandwidth is set to zero (i.e., the same as reserving an alternative LSP without allocating reserved bandwidth). This backup LSP may be used for fast rerouting in case of failure (Chapter 3 and Chapter 6).

Our algorithm is mainly composed of two procedures: a bandwidth threshold procedure and a released LSP procedure.

8.5.1 Bandwidth threshold (BWt) procedure

The procedure starts when the LER receives any packet except for the LSP release packet. It then starts to monitor the LSP aggregate bandwidth usage (BWu). If the LSP bandwidth usage is less than the bandwidth threshold value (BWt) during a given period of time, the mechanism triggers the LSP compute procedure to compute a new LSP with the BWu request. If found, it compares whether the new LSP has a lower cost (shorter path) than the old one. If the result is yes, the algorithm updates BWid and BWt by BWu and X*BWu respectively for the newly established LSP, reroutes the traffic to the new LSP, and moves the old LSP LIB entry to the backup LIB entry table. If the new LSP has a higher or equal cost, it returns to the initial point in the process. It also returns to the initial point when it does not find a new LSP.

8.5.2 Released LSP procedure

All the necessary information is available in the LER. The procedure starts when the LER receives the LSP release message. After releasing the corresponding LSP, the algorithm looks up the rejected LSP table to verify if there are any rejected LSPs for this output interface. If the result is no, it returns to the initial point. If the result is yes, then the algorithm selects among all rejected LSP candidates (BW $\leq$ BWreleased) the one with maximum bandwidth. Note that this bandwidth corresponds to that being used on the current non-optimal LSP established for that request logged in the

rejected LSP table. Then, the mechanism starts to compute a new LSP with this BWreq. If this request is rejected due to a lack of available bandwidth in some of the segments of the new LSP that did not belong to the released LSP, the notification of reject message returns the amount of available bandwidth to the LER (ingress LSR). This information will be used to select the appropriate rejected LSP from the reject LSP table to establish a new LSP instead of using the bandwidth decremental algorithm. After that it seeks a reject LSP in the list that fulfills this condition. It then updates the BWreq with BW of the selected LSP from the rejected LSP table and starts the process of computing a new LSP. If it is not found, it returns to the initial point.

On the other hand, if the process is able to establish a new LSP, we compare this with the LSP we want to reroute (old LSP) in terms of the LSP length (cost). If the newly established LSP has a lower cost, we reroute the traffic, update the values of BWid=BWreq and BWt=X*BWreq for the newly established LSP, move the old LIB entry to the backup LIB entry table and return to the initial point. If the cost is equal or greater, the process returns to the initial point.

8.6 SUMMARY

Whereas the existing literature deals only with the problem of traffic demand, we have also focused on improving network resource allocation and utilization of MPLS networks in order to optimize the routing of IP traffic. We do this by: 1) dynamically adapting the LSP to the variations of the overall network load and 2) monitoring for released LSPs whose freed bandwidth can be allocated to a non-optimal LSP. We have shown that our enhanced mechanism allows for flexibility in network resource utilization, reduces delay by using the optimal available low cost path, and reduces new LSP request blocking.

Internet service providers generally must pay a fixed fee for the links they use to connect their routers. Obviously, they are interested in taking advantage of this fixed

cost by using optimal network resource utilization. As there is no mechanism for doing this automatically, the operator balances the load using certain criteria (rules) on a daily basis in response to measured link utilization.

Our proposal contributes also to better network resource planning. For example, normally the traffic volume in one direction is higher during the day than during the night. At off-peak hours our proposal plays an important role in rerouting the traffic from high cost paths to low cost paths. In fact, it is extremely likely there is an alternative path that would achieve better utilization and better overall performance. Rerouting traffic to a low-cost (or optimal) LSP reduces delay and delay variation and helps to improve the QoS for delay sensitive and multimedia applications.

The proposal reduces traffic blocking, and the delay that the traffic can experience traversing non-optimal paths. Besides better network utilization, our proposal would give truer figures of network resource utilization information to the network manager for network planning than that obtained by using non-optimal LSPs.

Finally, the specification of the threshold and the period of time to trigger the mechanism is an open issue in this proposal. The number of label request messages to set up a new LSP must be evaluated for different values of BWt and the timer. Moreover, though the proposal has good performance in simple network topologies, we think it needs to be proved in extended network topologies.

9

CONCLUSIONS AND FUTURE WORK

This chapter concludes the thesis with a summary of the contributions of our research and proposes several topics that should be considered by future work.

9.1 CONCLUSIONS

The objectives set out for this thesis have been achieved. This thesis was aimed to develop mechanisms capable of providing a reliable and fast restoration from network component failure in an MPLS-based network for multimedia streaming with strict real time requirements in a better way than existing proposals in order to guarantee service continuity.

The main contributions are summarized in the following paragraphs.

- **Fast rerouting mechanism**. As we discuss throughout this thesis, the main drawback of MPLS technology, as a connection-oriented architecture, is its slow response time from network component failure due to the time needed to establish a new LSP to carry the affected traffic.

  Link failures are a common cause of service disruption in computer networks. Failures in high capacity links or between backbone routers, may seriously affect multimedia streaming and strict real-time application services and protocols. To alleviate this problem the Fast Rerouting approach was adopted as a solution.

  Fast Rerouting relies on pre-planning and requires that a backup LSP be computed, advertised and setup before a link failure is detected. The backup LSP combined with local repair aims to minimize packet losses during the restoration period. We presented an enhanced Fast Rerouting mechanism for MPLS-based networks which reroutes traffic over a backup LSP when a link/node of the protected LSP fails. The goal is to provide quality of service for the traffic carried by the protected LSP, even in case of failure and during recovery, until it is rerouted. Non-protected LSPs may be rerouted but without guarantees (best effort). Our proposal performs better compared to previous proposals in terms of both packet delay and packet disorder. We provide a simple and concise novel algorithm in the intermediate LSRs that operates in a distributed manner, introducing additional functionality to avoid packet re-ordering and to reduce unnecessary additional delay.

  The proposed mechanism can be used for quality of service (QoS) provision. This is possible because the algorithm is capable of handling criteria other than link failure detection for its activation. Once a given LSR detects congestion or a situation that leads to a Service Level Agreement (SLA) or QoS agreement being violated, it may start a fast reroute of a protected LSP. To extend our mechanism to the congestion problem, only a guarantee that the LSR is aware of the congestion problem is needed. Just as in the case of a link fault, the flow can be diverted to an alternative LSP once the congestion situation is detected.

  The proposed algorithm has been evaluated through simulation and the results have shown an improvement in the average latency or average packet delay. The

proposal eliminates packet re-ordering, improving end-to-end performance (overall performance), and has a shorter restoration period (i.e., fast network resources release) compared to Haskin's proposal.

The results of this work were published in the proceedings of the IEEE International Conference on Computer Communications and Networks (I3CN'01), October, 2001 [HD01].

- **Reliable and Fast Rerouting (RFR) mechanism**. This proposal addresses the packet loss issue during network component failure, which remains unsolved and affects the performance of fast rerouting mechanisms as well as our enhanced fast rerouting mechanism presented before. The parameters that affect the performance of any recovery scheme are traffic recovery delay (Full Restoration Time), packet disorder, and packet losses. In the previous proposal we addressed the first two issues.

  The main idea of RFR is to try to find solutions to the problem of packet losses during the failure, more precisely lost packets on the protected LSP. Up to now, packet loss due to node or link failure was considered to be "inevitable". It has always been assumed that the transport layer would somehow take care of the retransmission of lost packets through transmission control protocol (TCP). Our main interest is to protect multimedia and realtime traffic that usually do not benefit from retransmission. Furthermore, we have observed that the retransmission process due to packet loss significantly affects the throughput of TCP traffic due to the startup behavior (slow-start) of TCP. For this reason, critical services (premium traffic) will be affected by packet losses and, for TCP traffic, lost packets trigger retransmission requests, and hence the gains due to the decrease in restoration time achieved by previous the proposal (fast rerouting mechanism) may become negligible. As a consequence, bad performance and degraded service delivery will be experienced and QoS parameters will be seriously affected during the restoration period.

  The RFR mechanism proposes a novel recovery algorithm with small local buffers in each LSR node within the protected path to provide some preventive action against the packet loss problem by storing a copy of the packets in order to elim-

inate *packet loss* due to link/node failure. This buffer is also used to avoid *packet disorder* during the restoration period. This results in a significant throughput improvement for premium traffic.

In this proposal we eliminate packet losses while maintaining the benefits of our previous proposal, making link failures unnoticeable to all end users.

*** Buffer requirement analysis for RFR**. As our mechanism introduces an additional buffer requirement, for the proper operation of the proposal it is important to know the required additional buffer size, especially in the ingress LSR (i.e., ingress buffer). For this purpose and to validate the simulation results, we did an analytical study of buffer requirements and recovery times to justify the additional cost of the buffers that our proposal introduces. The results demonstrate that the buffer requirement is within a justifiable range compared to the benefit gained in network survivability and QoS guarantee for protected traffic.

The results of this work were published in the proceedings of the IEEE GLOBECOM'02, November, 2002 [HD02d].

- Although TCP traffic is not the main aim of this thesis, the RFR proposal was also evaluated for TCP traffic. The simulation results show that the RFR proposal gives support even for traffic using reliable transport protocols (TCP).

  Because RFR avoids packet losses and packet disorder for the protected flows, TCP connections experience neither losses nor disordered packets and may run at the maximum throughput even during the restoration period of the protected LSP.

  The results of this work were published in the proceedings of the IEEE International Conference on Networking (ICN'02), August, 2002 [HD02a].

- **Multiple fault tolerance**. In this work we extend our proposal from single link/node failure tolerance to multiple link/node failures on a protected LSP. The main idea presented in this proposal is the combination of existing proposals: segment protection, path protection and local repair. The significant change is made by the incorporation of the segment protection scheme. This allows the restoration of the failure to take place closer to the point of failure.

As the length of the protection path is a main quantitative measure of the quality of a protection scheme, the protection path length is used as an indication of the delay that the rerouted traffic will experience after a link failure. In addition to the delay, the length of the protection path reflects the amount of resources required to protect an LSP.

The simulation results show a significant reduction of the protection path length by merging the alternative LSP made by the local restoration decision in each segment protection domain, into the alternative LSP used for global restoration in the entire MPLS domain. This improves the main disadvantage of local restoration schemes.

Furthermore, rerouting of traffic is performed close to the failure point, increasing the restoration speed. In consequence, the proposed scheme provides a significant reduction of the LSP blocking problem. At the same time it provides better recovery (protection) in terms of path length. As a result, we achieve better network resource utilization and less delay for the rerouted traffic.

Finally, the proposed mechanism improves the main disadvantage of the previous proposals, packet loss in Makam's scheme and packet reordering in Haskin's scheme. Thereby the combined approach gives a better restoration mechanism than either of the mechanisms applied separately.

- **Optimal and Guaranteed LSP**. One of the main disadvantages of using the fast rerouting (preplanned) schemes is that the preplanned alternative LSP established at the time the protected LSP was set up may become a non-optimal alternative path after the occurrence of failure. For this reason a *dynamic and fast rerouting hybrid* approach was proposed. The proposal gains the advantages of both schemes: fast restoration time from the fast rerouting scheme by rerouting the affected traffic to the preplanned alternative LSP and the use of the possible optimal alternative path, if one exists, by using the dynamic restoration scheme. Note that the time that the dynamic scheme process takes to find the new alternative using the current network information does not affect the restoration time because the protected traffic is immediately rerouted using the fast rerouting mechanism.

The other problem that we address in this proposal is related to vulnerability. This problem appears when the preplanned alternative LSP is converted to the new protected LSP carrying the rerouted traffic on it and it is not protected from further failures. To give a solution to this, we compute a new alternative LSP. Then it is compared with the alternative LSP that is being used currently by the protected traffic. If the new alternative is "better", in terms of path length, then it is considered to be the new protected LSP and the traffic is rerouted through it. The previous alternative LSP remains as alternative LSP.

This method provides a guarantee of an alternative LSP at any time for the protected LSP, avoiding the vulnerability problem for the protection path.

The results of this work have been presented in a paper submitted for an international conference. At the time of writing this dissertation the reviewing process is not yet finished. [HD02c].

- **Adaptive LSP**. MPLS provides an integrated approach to traffic engineering, but it lacks flexibility due to its connection-oriented forwarding behavior. Long duration LSP connections may suffer from non-optimal resource utilization due to the fact that at setup the load of the network forced a non-optimal route. This affects interactive multimedia flows and delay-sensitive applications due to long end-to-end delays along the LSP.

  The proposed adaptive LSP, composed of a bandwidth threshold and released LSP procedures, allows more flexibility in network resource allocation and utilization by adapting the LSP to variations in the overall network load. The release of an LSP frees allocated bandwidth that may be used to update a non-optimal LSP. The mechanism is based on monitoring both a significant decrement of aggregated traffic on the established non-optimal LSP and the release of any LSP in the network. The adaptation is based on two aspects: dynamic bandwidth management for an LSP and rerouting traffic from a non-optimal LSP to an optimal one.

The preliminary results in this mechanism show an improvement of network resource utilization while reducing end-to-end delay and minimizing traffic blocking problems when setting up a new LSP.

The results of this work have been presented in a paper submitted for an international conference. At the time of writing this dissertation the reviewing process is not yet finished. [HD02b].

9.2 FUTURE WORK

There are several open issues related to the proposals presented in the PhD thesis that need further study. In the following paragraphs an outline of the immediate future work to be done is presented.

An aspect that requires more study is related to the definition of Segment Protection Domain, (SPD). Besides the administrative decisions (i.e., autonomous systems), some the criteria to determine and define SPDs are needed. The available buffer size and the maximum delay allowed for protected traffic can be used as additional criteria for SPD setup. From the study for buffer requirements at the ingress LSR presented in Chapter 4 we find a tradeoff between the amount of memory at the ingress LSR and the length of the protected LSP. This gives a first approach to the coverage of an SPD. This needs further study in different topologies and traffic characteristics.

Related to the proposed Reliable and Fast Rerouting mechanism, it seems simple to implement in existing devices; however, to determine the feasibility of this proposal it is important to adapt the Label Distribution Protocol (LDP) and the routing protocols to support our mechanism. For this purpose, the implementation of the proposal in a PC-based platform might be useful to observe the real behavior and determine the modifications needed. Furthermore, an analysis of the number of LDP messages required for the setup of the alternative and backup LSP is needed. For this study a simulation platform based on the MPLS Network Simulator (MNS) would suffice.

As we mentioned in Chapter 8, the open issues in adaptive LSP are to determine the threshold value of the aggregated bandwidth, the waiting time, and the frequency of LSP setup, their dependencies, and their results in providing the optimal protected LSP and the aggregated alternative LSP. This evaluation may be performed via simulation using MNS, running simulations for large networks with many LSR and links. Obviously, the best way to determine these values is the use of real MPLS traffic traces from some authoritative sources as input data for the number of LSPs, inter-arrival time for new LSPs, mean duration of LSPs, etc. Unfortunately, these traces are not available at this time.

Finally, we considered in our proposals all protected LSPs on a link as an LSP (aggregated LSPs) to gain scalability and strictly follow the MPLS architecture. The extension of MPLS for optical networks, Generalized Multi-Protocol Label Switching (GMPLS), gives the possibility to carry individual protected LSP per lambda (DWDM). An interesting topic to be considered in the future is the possibility of handling each protected LSP individually, using the fast rerouting mechanism with preplanned alternative LSPs which are disjoint among them. This approach may allow both the benefit of survivability and the possibility of load balancing in the network. At the same time the approach must establish some regulation to balance the tradeoff of advantages between scalability and load balancing.

REFERENCES

[ABG+01] Daniel O. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, and G. Swallow. RSVP-TE: Extensions to RSVP for LSP Tunnels. *RFC 3209*, December 2001.

[ACE+02] D. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao. Overview and Principles of Internet Traffic Engineering. *RFC3272*, May 2002.

[ADF+01] L. Andersson, P. Doolan, N. Feldman, A. Fredette, and B. Thoma. LDP Specification. *RFC 3036*, January 2001.

[ADH94] J. Anderson, Bharat T. Doshi, and S. Dravida P Harshavardhana. Fast restoration of ATM Networks. *IEEE Jouranal on Selected Areas in Communications, Volume: 12 Issue: 1*, pages 128 –138, January 1994.

[AFBW97] A.Viswanathan, N. Feldman, R. Boivie, and R. Woundy. ARIS: Aggregate Route-Based IP Switching . *Work in progress, Internet draft <draft-viswanathan-aris-overview-00.txt>*, March 1997.

[ALAS+02] J. Ash, Y. Lee, P. Ashwood-Smith, B. Jamoussi, D. Fedyk, D. Skalecki, and L. Li. LSP Modification Using CR-LDP. *RFC 3214*, January 2002.

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

[APS99] M. Allman, V. Paxon, and W. Stevens. TCP Congestion Control. *RFC2581*, April 1999.

[Awd99] D. O. Awduche. MPLS and Traffic Engineering in IP Networks. *IEEE Comunication Magazine, Volume: 37 Issue: 12*, pages 42–47, December 1999.

[AWK+99] G. Apostolopoulos, D. Williams, S. Kamat, R. Guerin, A. Orda, and T. Przygienda and. QoS Routing Mechanisms and OSPF Extensions. *Internet,RFC 2676*, August 1999.

[BB+98] S. Blake, D. Black, , M. Carlson, E. Davies, Z. Wang, and W. Weiss. An Architecture for Differenciated Services. *RFC2475*, December 1998.

[BCS94] R. Barden, D. Clark, and S. Shenker. Integrated Service in the Internet Architecture: an Overview. *RFC1633*, June 1994.

[Bla00] D. Black. Differentiated Services and Tunnels. *RFC2983*, October 2000.

[BR02] R. Bartos and M. Raman. Dynamic issues in MPLS service restoration. *Proc. of the Fourteenth IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS)*, pages 618–623, November 2002.

[BS01] G. Banerjee and D. Sidhu. Label switched path restoration under two random failures. *Proc. of the IEEE Globecom '01*, pages 30–34 Vol.1, November 2001.

[BZB+97] R. Braden, L. Zhang, S. Berson, S. Herzog, and S.Jamin. Resource ReSerVation Protocol (RSVP). *RFC2205*, September 1997.

[CBMS93] C. Edward Chow, J. Bicknell, S. McCaughey, and S. Syed. A fast distributed network restoration algorithm. *proceeding of Twelfth Annual International Phoenix Conference on Computers and Communications*, pages 261–267, March 1993.

[CDF+99] R. Callon, P. Doolan, N. Feldman, A. Fredette, G. Swallow, and A. Viswanathan. A Framework for Multiprotocol Label Switching. *Work in progress, Internet draft <draft-ietf-mpls-framework-05.txt>*, September 1999.

[CJVM01] E. Calle, T. Jové, P. Vilá, and J. L. Marzo. A Dynamic Multilevel MPLS Protection Domain. *Proceeding of 3rd International Workshop on Design of Reliable Communication Netowrks (DRCN)*, pages 7–10, October 2001.

[CKMO92] T. Chujo, H. Komine, K. Miyazaki, and T. Ogura. Spare Capacity Assignment for Multiple-Link Failures. *Proc. of the International Workshop on Advanced Communications and Applications for High Speed Networks*, pages 191–197, March 1992.

[CO99] Thomas M. Chen and Tae H. Oh. Reliable Services in MPLS. *IEEE Communication Megazine, Volume: 37 Issue: 12*, pages 58 –62, December 1999.

[DCea02] B. Davie, A. Charny, and J.C.R. Bennett et al. An Expedited Forwarding PHB (Per-Hop Behavior. *RFC3246*, March 2002.

[DDR98] B. Davie, P. Doolan, and Y. Rekhter. Switching in Ip Networks : Ip Switching, Tag Switching, and Related Technologies. *The Morgan Kaufmann Publishing ,Inc. ISBN 1-55860-505-3*, May 1998.

[DR00] B. Davie and Y. Rekhter. MPLS Technology and Applcations. *Morgan kaufmann publisher Inc. ISBN 1-55860-656-4*, May 2000.

[EJLW01] A. Elwalid, C. Jin, S. Low, and I. Widjaja. MATE: MPLS Adaptive Traffic Engineering. *IEEE INFOCOM 2001*, pages 1300 –1309 vol.3, April 2001.

[FA97] N. Feldman and A.Viswanathan. ARIS Specification. *Work in progress, Internet draft <draft-feldman-aris-spec-00.txt>*, March 1997.

[FM01] A. Farrel and B. Miller. Surviving Failures in MPLS Network. *Technical report, Data Connection Ltd.*, February 2001.

[FVa] K. Fall and K. Varadhan. The network simulator -*ns-2*. *The VINT project. UC Berkeley, LBL, USC/ISI, and Xerox PARC, http://www.isi.edu/nsnam/ns/.*

[FVb] K. Fall and K. Varadhan. The *ns* Manual. *The VINT project. UC Berkeley, LBL, USC/ISI, and Xerox PARC, http://www.isi.edu/nsnam/ns/ns-documentation.html.*

[GJW02] A. Gaeil, J. Jang, and C. Woojik. An Efficient Rerouting Scheme for MPLS-based Recoery and its Performance Evaluation. *Telecommunication Systems, vol.9*, pages 441 –446, March-April 2002.

[Gro87] W. D Grover. The Selfhealing Networks, A Fast Distributed Restoration Technique for Network Using Digital Cross-Connect Machines. *Proceeding of IEEE GLOBECOM '87*, pages 1090–1095, November 1987.

[GS00] R. Goguen and G. Swallow. RSVP Label Allocation for Backup Tunnels. *Work in progress, Internet draft <draft-swallow-rsvp-bypass-label-01.txt>*, November 2000.

[GW99] A. Gaeil and C. Woojik. Overview of MPLS Network Simulator:Design and Implementation. Technical report, Department of Computer Engineerig, Chungnam National University, Korea, December 1999.

[GW00] A. Gaeil and C. Woojik. Design and Implementation of MPLS Network Simulator (MNS) supporting LDP and CR-LDP. *Proceedings of the IEEE International Conference on Networks (ICON'00)*, pages 441 –446, September 2000.

[GW01a] A. Gaeil and C. Woojik. Design and Implementation of MPLS Network Simulator (MNS) supporting QoS. *15th International Conference on Information Networking*, pages 694 –699, January 2001.

[GW01b] A. Gaeil and C. Woojik. Simulator for MPLS Path Restoration and Performance Evaluation. *http://flower.ce.cnu.ac.kr/˜ fog1/mns/index.htm. see path protection/restoration*, April 2001.

[HA00] F. Hellstrand and L. Andersson. Extensions to CR-LDP and RSVP-TE for setup of pre-established recovery tunnels. *Work in progress, Internet draft <draft-hellstrand-mpls-recovery-merge-01.txt>*, November 2000.

[HBWW99] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski. Assured Forward PHB. *RFC2597*, June 1999.

[HD01] L. Hundessa and J. Domingo. Fast rerouting mechanism for a protected Label Switched Path. *Proceedings of the IEEE International Conference on Computer Communications and Networks (I3CN'01)*, October 2001.

[HD02a] L. Hundessa and J. Domingo. A Reliable QoS Provision and Fast Recovery Method for Protected LSP in MPLS-based Networks. *Proceedings of the IEEE International Conference on Networking (ICN02)*, August 2002.

[HD02b] L. Hundessa and J. Domingo. Adaptive LSP in MPLS Networks. *Technical Report UPC-DAC-2002-38*, July 2002.

[HD02c] L. Hundessa and J. Domingo. Multiple Fault Tolerant Protected LSP with Optimal and Guaranteed Alternative LSP in MPLS Networks. *Technical Report UPC-DAC-2002-37*, July 2002.

[HD02d] L. Hundessa and J. Domingo. Reliable and Fast Rerouting Mechanism for a Protected Label Switched Path. *Proceedings of the IEEE GLOBECOM '02*, November 2002.

[HK00] D. Haskin and R. Krishnan. A Method for Setting an Alternative Label Switched Paths to Handle Fast Reroute. *Work in progress, Internet draft <draft-haskin-mpls-fast-reroute-05.txt>*, November 2000.

[IG00] Rainer R. Iraschko and Wayne D. Grover. A highly efficient path-restoration protocol for management of optical network transport integrity. *IEEE Journal on Selected Areas in Communications, Volume: 18 Issue: 5*, pages 779 –794, May 2000.

[Jac88] V. Jacobson. Congestion Avoidance and Control. *Proceeding ACM Symp. Commun. Arch. Protocol, SIGCOMM '88*, August 1988.

[Jac90] V. Jacobson. Modified TCP congestion Avoidance Algorithm. *URL:ftp://ftp.isi.edu/end2end/end2end/interest-1990.mail*, April 1990.

[JAC+02] B. Jamoussi, L. Andersson, R. Callon, R. Dantu, L. Wu, P. Doolan, T. Worster, N. Feldman, A. Fredette, M. Girish, E. Gray, J. Heinanen,

J. Heinanen, T. Kilty, and A. Malis. Constraint-Based LSP Setup using LDP. *RFC 3212*, February 2002.

[Kaw98] R. Kawamura. Architectures for ATM network survivability. *IEEE Communications Survey, Volume: 1 No: 1*, pages 2–11, Fourth Quarter 1998.

[KCO+90] H. Komine, T. Chujo, T. Ogura, K. Miyazaki, and T. Soejima. A distributed restoration algorithm for multiple-link and node failures of transport networks. *Proceeding of IEEE GLOBECOM '90*, pages 459 –463, December 1990.

[KKL+02] S. Kini, M. Kodialam, T. V. Lakshaman, S. Sengupta, and C. Villamizar. Shared Backup Label Switched Path Restoration. *Work in progress, Internet draft <draft-kini-restoration-shared-backup-02.txt>*, April 2002.

[KKT94] Y. Kajiyama, K. Kikuchi, and N. Tokura. An ATM VP-based self-healing ring. *IEEE Journal on Selected Areas in Communications, Volume: 12 Issue: 1*, pages 171–178, January 1994.

[KL00] M. Kodialam and T. V. Lakshaman. Minimum Interface Routing With Applications to MPLS Traffic Engineering. *Proceedings of IEEE INFOCOM'00*, pages 884 –893 vol.2, March 2000.

[KL01] M. Kodialam and T. V. Lakshaman. Dynamic routing of locally restorable bandwidth guaranteed tunnels using aggregated link usage information. *Proceedings of IEEE INFOCOM'01*, pages 376–385 vol.1, April 2001.

[KNE96] Y. Katsube, K. Nagami, and H. Esaki. Cell Switch Router - Basic Concept and Migration Scenario. *Networld+Interop'96 Engineer Conference* , April 1996.

[KNE97] Y. Katsube, K. Nagami, and H. Esaki. Toshiba's Router Architecture Extensions for ATM : Overview. *RFC 2098*, February 1997.

[KNME97] Y. Katsube, K. Nagami, S. Matsuzawa, and H. Esaki. Internetworking based on cell switch router-architecture and protocol overview. *Proceed-*

ings of the IEEE , Volume: 85 Issue: 12, pages 1998 –2006, December 1997.

[KO99] R. Kawamura and H. Ohta. Architectures for ATM network survivability and their field deployment. *IEEE Communications Magazine , Volume: 37 Issue: 8*, pages 88–94, August 1999.

[KST94] R. Kawamura, K. Sato, and I. Tokizawa. Self-Healing ATM Networks Based on Virtual Path Concept. *IEEE Journal on Selected Areas in Communications, Volume: 12 Issue: 1*, pages 120 –127, January 1994.

[KT95] R. Kawamura and I. Tokizawa. Self-healing virtual path architecture in ATM networks. *IEEE Communications Magazine , Volume: 33 Issue: 9*, pages 72–79, September 1995.

[Kuh97] D. R. Kuhn. Sources of failure in the public switched telephone network . *Journal on Computer, Vol: 30 Issue: 4*, pages 31–36, April 1997.

[lan95] LAN Emulation Over ATM Version 1.0. *ATM Forum,af-lane-0021.000*, January 1995.

[LCJ99] L.Andersson, B. Cain, and B. Jamoussi. Requirement framework for fast re-routing with MPLS. *Work in progress, Internet draft <draft-andersson-reroute-frmwrk-00.txt>*, October 1999.

[LH98] M. Laubach and J. Halpern. Classical IP and ARP over ATM. *RFC2225*, April 1998.

[LKP+98] J. Luciani, D. Katz, D. Piscitello, D. Piscitello, and N. Doraswamy. NBMA Next Hop Resolution Protocol (NHRP). *RFC2332*, April 1998.

[MCSA] J. L. Marzo, E. Calle, C. Scolio, and T. Angali. Adding QoS protection in order to Enhance MPLS QoS Routing. *To appear in Proc. of ICC 2003.*

[MFB99] M. Medard, S.G Finn, and R.A. Barry. WDM loop-back recovery in mesh networks. *Proceedings of the IEEE INFOCOM'99*, pages 752–759, March 1999.

[Moy98] J. Moy. OSPF Version 2. *RFC2328*, April 1998.

[mpo97] Multi-Protocol over ATM (MPOA) Version 1.0. *ATM Forum Technical Committee, af-mpoa-0087.0000*, July 1997.

[MSOH99] S. Makam, V. Sharma, K. Owens, and C. Huang. Protection/Restoration of MPLS Networks. *Work in progress, Internet draft <draft-makam-mpls-protection-00.txt>*, October 1999.

[MSSD] X. Masip, S. Sànchez, J. Solé, and J. Domingo. QoS routing Algorithm under Inaccurate Routing Information for Bandwidth Constrained Applications. *to appear in proceeding of ICC '03.*

[MSSD02] X. Masip, S. Sànchez, J. Solé, and J. Domingo. A QoS Routing Mechanism for Reducing the Routing Inaccuacy Effect. *QoSIP'02*, 2002.

[NBBB98] K. Nichols, S. Blake, F. Baker, and D. Black. Definition of the Differenciated Service Field (DS Field) in the IPv4 and IPv6 Headers. *RFC2474*, December 1998.

[NEH+96a] P. Newman, W. L. Edwards, R. Hinden, E. Hoffman, F. Ching Liaw, T. Lyon, and G. Minshall. Ipsilon Flow Management Protocol Specification for IPv4 Version 1.0. *RFC 1953*, May 1996.

[NEH+96b] P. Newman, W. L. Edwards, R. Hinden, E. Hoffman, F. Ching Liaw, T. Lyon, and G. Minshall. Transmission of Flow Labelled IPv4 on ATM Data Links Ipsilon Version 1.0. *RFC1954*, May 1996.

[NEH+98] P. Newman, W. Edwards, R. Hinden, E. Hoffman, F. Ching Liaw, T. Lyon, and G. Minshall. Ipsilon's General Switch Management Protocol Specification Version 2.0. *RFC2297*, March 1998.

[NKS+97] K. Nagami, Y. Katsube, Y. Shobatake, A. Mogi, S. Matsuzawa, T. Jinmei, and H. Esaki. Toshiba's Flow Attribute Notification Protocol (FANP) Specification. *RFC 2129*, April 1997.

[NLM96] P. Newman, T. Lyon, and G. Minshall. Flow labelled IP: a connectionless approach to ATM. *Proceedings IEEE INFOCOM '96, Networking the Next Generation*, pages 1251 –1260 vol.3, March 1996.

[Ora90] D. Oran. OSI IS-IS Intra-domain Routing Protocol. *RFC 1142*, February 1990.

[OSM+01] K. Owens, V. Sharma, S. Makam, C. Huang, and B. Akyol. Extension to CR-LDP for MPLS Path Protection. *Work in progress, Internet draft <draft-owens-crldp-path-protection-ext-01.txt>*, July 2001.

[OSM+02] K. Owens, V. Sharma, S. Makam, C. Huang, and B. Akyol. Extension to RSVP-TE for MPLS Path Protection. *Work in progress, Internet draft <draft-chang-rsvpte-path-protection-ext-03.txt>*, April 2002.

[OSMH01] K. Owens, V. Sharma, S. Makam, and C. Huang. A Path Protection/Restoration Mechanism for MPLS Networks. *Work in progress, Internet draft <draft-chang-mpls-protection-03.txt>*, July 2001.

[pnn96] Private Network Node Interface Specification. *ATM Forum, af-pnni-0055.000*, March 1996.

[RDK+97] Y. Rekhter, B. Davie, D. Katz, E. Rosen, and G. Swallow. Cisco Systems Tag Switching Architecture Overview. *RFC2105*, February 1997.

[RL95] Y. Rekhter and T. Li. A Border Gatway Protocol 4 (BGP-4). *RFC1771*, March 1995.

[RM01] U. Ranadive and D. Medhi. Some observations on the effect of route fluctuation and network link failure on TCP . *Proceedings of the IEEE International Conference on Computer Communications and Networks (I3CN'01)*, pages 460–467, October 2001.

[RTF+01] E. Rosen, D. Tappan, G. Fedorkow, Y. Rekhter, D. Farinacci, T. Li, and A. Conta. MPLS Label Stack Encoding. *RFC 3032*, January 2001.

[RVC01] E. Rosen, A. Viswanathan, and R. Callon. Multiprotocol Label Switching Architecture. *RFC 3031*, January 2001.

[SCFJ96] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A Transport Protocol for Real-Time Applications. *RFC1889*, January 1996.

[SH02] V. Sharma and F. Hellstrand. Framework for MPLS-based Recovery. *Work in progress, Internet draft <draft-ietf-mpls-recovery-frmwrk-08.txt>*, October 2002.

[She99] S. Shew. Fast Restoration of MPLS Label Switched Paths. *Work in progress, Internet draft <draft-shew-lsp-restoration-00.txt>*, October 1999.

[SHT90] K. Sato, H. Hadama, and I. Tokizawa. Network reliability enhancement with virtual path strategy. *Proceeding of IEEE GLOBECOM '90*, pages 464–469, December 1990.

[SPG97] S. Shenker, C. Partridge, and R. Guerin. Specification of Guaranted Quality of Service. *RFC2212*, September 1997.

[SRL98] H. Schulzrinne, A. Rao, and R. Lanphier. Real Time Streaming Protocol (RTSP). *RFC2326*, April 1998.

[Ste94] W.R. Stevens. TCP/IP Illstrated. *Addison Wesley*, 1994.

[Swa99] G. Swallow. MPLS Advantages for Traffic Engineering. *IEEE Comunication Magazine, Volume: 37 Issue: 12*, pages 54–57, December 1999.

[SWW01] S. Suri, M. Waldvogel, and P. R. Warkhede. Profile-Based Routing: A New Frame work for MPLS Traffic Engineering. *In Proc. of Quality of Future Internet Services (QofIS)*, September 2001.

[tcp81] Transmission Control Protocol (TCP). *RFC793*, September 1981.

[THS+94] D. Tipper, J.L Hammond, S. Sharma, A. Khetan, K. Balakrishnan, and S. Menon. An analysis of the congestion effects of link failures in wide area networks. *IEEE Journal on Selected Areas in Communications, Volume: 12 Issue: 1*, pages 172–179, January 1994.

[W.D89] W.D Grover. Selfhealing Networks: A Distributed Algorithm for k-shortest link disjoint paths in a multi-graph with applications in real time network restoration. *PhD thesis, University of Alberta, Department of Electrical Engineering*, Fall 1989.

[Wro97] J. Wroclawski. Specification of the Contrlled-Load Network Element Service. *RFC 2211*, September 1997.

[Wu95] T.-Ho Wu. Emerging Technology for Fiber Network Survivability. *IEEE Communications Magazine , Volume: 33 Issue: 2*, pages 58–59, 62–74, February 1995.

[YH88] C.H Yang and S. Hasegawa. FITNESS-failure immunization technology for network services survivability. *Proceeding of IEEE GLOBECOM '88*, pages 1549–1554, 1988.