FAILURE RECOVERY IN MPLS MULTICAST NETWORKS USING A SEGMENTED BACKUP APPROACH

Rohan Deshmukh

A Thesis

In

The Department

Of

Electrical & Computer Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Applied Science
in Electrical & Computer Engineering
at Concordia University
Montréal, Québec, Canada

September 2004

Your file Votre référence
ISBN: 0-612-94694-0
Our file Notre référence
ISBN: 0-612-94694-0

Abstract

Failure Recovery in MPLS Multicast Network
using Segmented Backup Approach

Rohan Deshmukh

With the diversification of the traffic carried on the Internet, improvement of QoS has been becoming considerable demanding to realize large capacity, high speed and reliable communication in IP networks. Various models have been proposed to address this need, MPLS being one of the main architectures that will likely be deployed mostly on the Internet to achieve these QoS goals. MPLS overlays an IP network to allow resources to be reserved and routes pre-determined. Effectively, MPLS superimposes a connection-oriented framework over the connectionless IP network. It provides virtual links or tunnels through the network to connect nodes that lie at the edge of the network. In the future, it is expected that congestion and faults on a Label Switched Path (LSP) will seriously affect service contents and recovery and restoration of such LSPs would be required to realize a fault-tolerant MPLS net-

work. Researchers have addressed the need in this context in the past with respect to intermediate link and/or node failures mostly for the unicast networks.

In this thesis, we consider multiple failures for MPLS multicast networks. Upon a failure in the primary path, its segmented backup gets activated to transfer the data, making a network real time. Here we propose a new method of providing backup paths in multicast routing tree. The method is based on segmentation cluster formation, in which backup paths are provided by connecting segmentation points (SPs) of the same cluster rather than providing a backup between the receiver label edged routers as suggested by other researchers. The segmented backup path and hence cluster formation aims at minimizing the number of receivers of the multicast routing tree to be dropped from the communication if a failure occurs. Our results show that failure recovery in multicast MPLS network using segmented backup approach is more effective than recent proposals.

Acknowledgments

It has been a great privilege for me to work with Dr. Anjali Agarwal, an exceptional researcher and teacher, who introduced me to real-time systems, fault tolerant systems, Data communication and MPLS Networks and allowed me to pursue research in Networking domain, a subject close to my heart. She has been extraordinarily patient and supportive, have been always available for discussion and responding speedily to research reports. I would like to take this oppurtunity to thank her for her continuous encouragement and guidance throughout the course of my research.

I also take this opportunity to thank OPNET Technical Support Team in clarifying my doubts, helping me out in understanding the concept of OPNET modeler.

I extend my whole-hearted gratitude to my parents for their encouragement and support, without which it would have been impossible to finish this work.

Finally, I would like to thank my colleagues in the research group and my friends for their valuable discussions, support and advice.

In loving memory of my father whose faith, inspiration and strength has and will be with me always ...

Contents

List of Figures

List of Abbreviation

MPLS Multiprotocol Label Switching

QoS Quality of Service

LSP Label Switched Path

LSR Label Switched Router

LER Label Edged Router

IETF Internet Engineering Task Force

RSVP Resource Reservation Protocol

OSPF Open Shortest Path First

FEC Forward Equivalence Class

CRLDP Constraint Route Label Distribution Protocol

TE Traffic Engineering

TCP/IP Transmission Control Protocol/Internet Protocol

FT Fault Tolerance

HA High Availability

RCF Request For Comment

ATM Asynchronous Transfer Mode

PLR Point of Local Recovery

SP Segmentation Point

CoS Class of Service

CPI Control Plane Identity

NM notification Message

FLR Fast and Local Reroute

CL Cluster

CSPF Constrained Shortest Path First

RCI Router Configuration Import

IGP Interior Gateway Protocol

DES Discrete-Event Simulation

PIM-SM Protocol Independent Multicast-Sparse Mode

IGMP Internet Group Management Protocol

PCM Pulse Code Modulation

ToS Type of Service

MP Merge Point

Chapter 1

Introduction

Over the last few years, the Internet has evolved into a ubiquitous network and inspired the development of a variety of new applications in business and consumer markets. These new applications have driven the demand for increased and guaranteed bandwidth requirements in the backbone of the network. In addition to the traditional data services currently provided over the Internet, new voice and multimedia services are being developed and deployed. The Internet has emerged as the network of choice for providing these converged services. However, the demands placed on the network by these new applications and services, in terms of speed and bandwidth, have strained the resources of the existing Internet infrastructure. This transformation of the network toward a packet- and cell-based infrastructure has introduced uncertainty into what has traditionally been a fairly deterministic network.

Quality of Service (QoS) has become an important function demanding large capacity, high speed and reliable service in IP networks especially in multimedia services

for real time applications. Different approaches have been proposed for providing this support within the network layer based on reserving resources for individual data flow or treating differently individual IP data packets at the node based on marking in the IP header. Multiprotocol Label Switching (MPLS) [1, 2, 3, 4] is rapidly becoming a key technology for use in core networks, including converged data and voice networks. MPLS does not replace IP routing, but works alongside existing and future routing technologies to provide very high-speed data forwarding between Label Switched Routers (LSRs) together with reservation of bandwidth for traffic flows with different Quality of Services (QoS) requirement. MPLS enhances the services that can be provided by IP networks, offering scope for traffic engineering and guaranteed QoS.

In sum, despite some initial challenges, MPLS will play an important role in the routing, switching, and forwarding of packets through the next-generation network in order to meet the service demands of the network users.

1.1 MPLS and Its Components

1.1.1 What is MPLS?

MPLS is an Internet Engineering Task Force (IETF)-specified framework that provides for the efficient designation, routing, forwarding, and switching of traffic flows through the network [5]. MPLS performs the following functions:

1. specifies mechanisms to manage traffic flows of various granularities, such as

flows between different hardware, machines, or even flows between different applications.

2. remains independent of the Layer-2 and Layer-3 protocols.
3. provides a means to map IP addresses to simple, fixed-length labels used by different packet-forwarding and packet-switching technologies.
4. interfaces to existing routing protocols such as resource reservation protocol (RSVP) and open shortest path first (OSPF).
5. supports the IP, ATM, and frame-relay Layer-2 protocols.

In MPLS, data transmission occurs on label-switched paths (LSPs). LSPs are a sequence of labels at each and every node along the path from the source to the destination. Each data packet encapsulates and carries the labels during their journey from source to destination. High-speed switching of data is possible because the fixed-length labels are inserted at the very beginning of the packet or cell and can be used by hardware to switch packets quickly between links.

MPLS uses the following terms:

LSRs and LERs:

The devices that participate in the MPLS protocol mechanisms can be classified into label edge routers (LERs) and label switching routers (LSRs). An LSR is a high-speed router device in the core of an MPLS network that participates in the establishment of LSPs using the appropriate label signaling protocol and high-speed switching of the data traffic based on the established paths. An LER is a device that operates at

the edge of the access network and MPLS network.

FEC:

The forward equivalence class (FEC) is a representation of a group of packets that share the same requirements for their transport. All packets in such a group are provided the same treatment en route to the destination.

Labels and Label Bindings:

A label, in its simplest form, identifies the path a packet should traverse. Once a packet has been classified as a new or existing FEC, a label is assigned to the packet. The packets are then forwarded based on their label value. Label assignment decisions may be based on destination unicast routing, traffic engineering, multicast and QoS.

The generic label format is illustrated in Figure 1.1 [5].

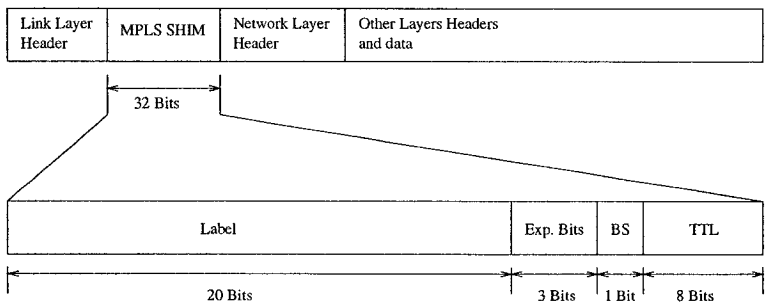

Figure 1.1: MPLS Generic Label Format

Label-Switched Paths (LSPs):

A collection of MPLS enabled devices represents an MPLS domain. Within an MPLS domain, a path is set up for a given packet to travel based on an FEC. The LSP is set up prior to data transmission.

Label Merging:

The incoming streams of traffic from different interfaces can be merged together and switched using a common label if they are traversing the network toward the same final destination. This is known as stream merging or aggregation of flows.

Signaling Mechanisms:

1. Label Request - Using this mechanism, an LSR requests a label from its downstream neighbor so that it can bind to a specific FEC. This mechanism can be employed down the chain of LSRs up until the egress LER (i.e., the point at which the packet exits the MPLS domain).
2. Label Mapping - In response to a label request, a downstream LSR will send a label to the upstream initiator using the label mapping mechanism.

The above concepts for label request and label mapping are explained in Figure 1.2.

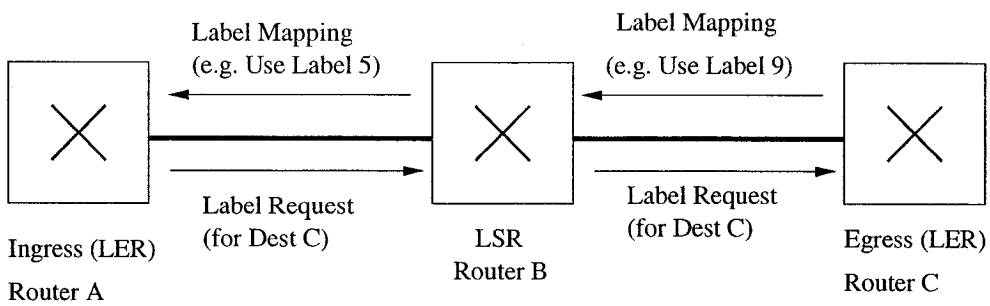

Figure 1.2: Signaling Mechanisms [5]

Traffic Engineering:

Traffic engineering is a process that enhances overall network utilization by attempting to create a uniform or differentiated distribution of traffic throughout the network.

An important result of this process is the avoidance of congestion on any one path. It is important to note that traffic engineering does not necessarily select the shortest path between two devices. It is possible that, for two packet data flow, the packets may traverse completely different paths even though their originating node and the final destination node are the same. This way, the less-exposed or less-used network segments can be used and differentiated services can be provided.

In MPLS, traffic engineering is inherently provided using explicitly routed paths. The LSPs are created independently, specifying different paths that are based on user-defined policies. However, this may require extensive operator intervention. RSVP-TE and CRLDP are two possible approaches to supply dynamic traffic engineering and QoS in MPLS.

1.2 Working of MPLS

The source sends its data to the destination. In an MPLS domain, not all of the source traffic is necessarily transported through the same path. Depending on the traffic characteristics, different LSPs could be created for packets with different QoS requirements.

Figure 1.3 shows a MPLS network that consists of LERs, LSRs as core routers, and path joining these routers (LSP). LER1 is the ingress and LER4 is the egress router. When ingress receives packets from host, it determines the FEC for each packet, deduces the LSP to use and adds a label to the packet. This decision is a local matter

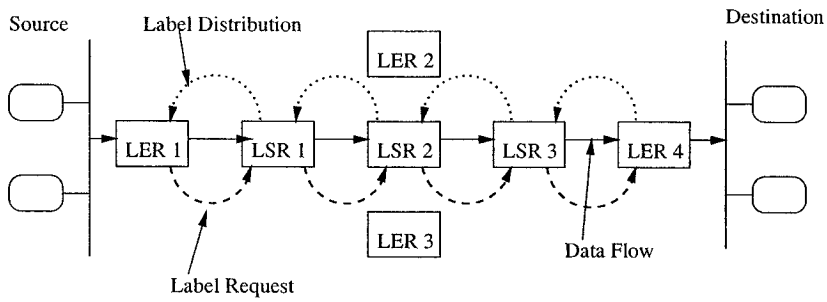

Figure 1.3: Working of MPLS

but is likely to be based on factors including the destination address, the quality of service requirements and the current state of the network. This flexibility is one of the key elements that makes MPLS so useful. Ingress then forwards the packet on the appropriate interface for the LSP. LSR1 is an intermediate LSR in the MPLS network. It simply takes each labeled packet and uses the pairing {incoming interface, label value} to decide the pairing {outgoing interface, label value} with which to forward the packet. The egress LSR performs the same lookup as the intermediate LSRs, but the {outgoing interface, label value} pair marks the packet as it exits the LSP. The egress LSR strips the labels from the packets and forward them using layer 3 routing.

1.3 Multicast

So far, researchers have concentrated on unicast communication, where a single source sends data to single receiver through other intermediate nodes. Many applications however require multicast communication where data are sent simultaneously to multiple receivers. The best example of such requirement is teleconferencing among a group of many people. In multicast communication, when one of the group members

speaks, his voice should be delivered to all other group members. A simple solution to design multicasting is to send the same data in turn to all other members of the group. In other words, we can achieve multicast through unicast as shown in Figure 1.4.

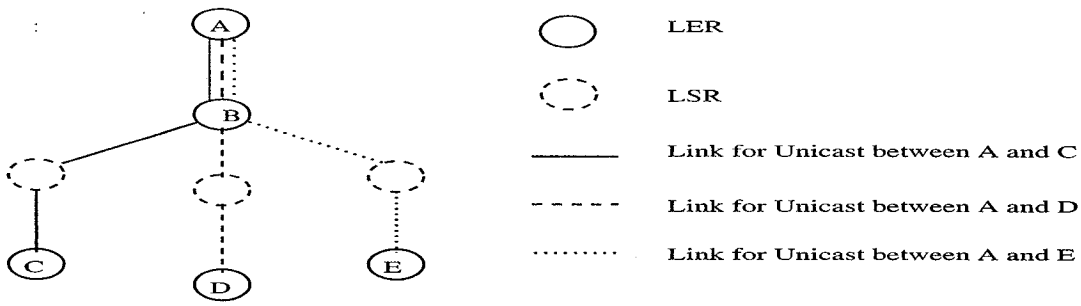

Figure 1.4: Multicast achieved through Unicast

In the multicast Figure 1.4, three different unicast LSPs B-C, B-D and B-E are used. They together constitute the multicast network as LSR B sends data to three different LERs C, D and E whereas LER A sends only one copy of data to LSR B.

1.3.1 Multicast with MPLS

Multicasting in a network is provided by seting up a multicast routing tree [6], where the switches are the nodes of the tree and the links are the edges of the tree. Each muticasting switch of the tree multicasts packets to each outgoing link. By forwarding data on this tree structure and duplicating data at intermediate nodes of the tree, the same information is sent on each link of the tree only once, thereby saving bandwidth.

Currently MPLS multicast is in the primary stage. Multicast and unicast traffic require different types of processing from routers. The packet duplication mechanism that is implemented in IP routers to support IP multicast can be used to duplicate

MPLS packets [7]. MPLS routers at the bifurcation of a multicast routing tree duplicate packets and send copies of the same packet on different outgoing links. Each copy of an incoming MPLS multicast packet is assigned a different label before it is forwarded on an outgoing link. This can be observed in Figure 2.5.

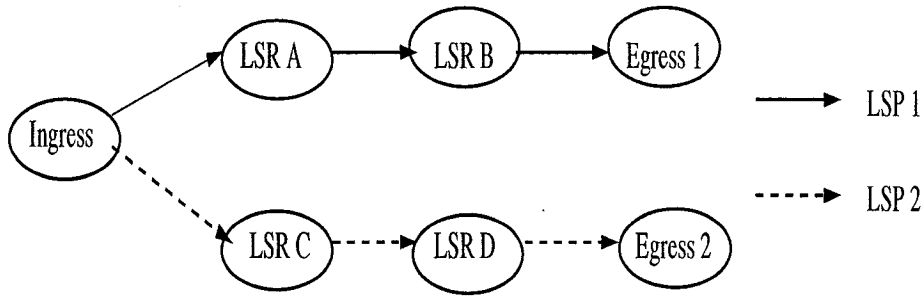

Figure 1.5: Multicast in MPLS network

Here ingress multicasts data on primary and secondary LSPs according to the available bandwidth. Each member of a multicast group can build a shortest path multicast tree to reach all other members. Alternatively, all members of a group can be leaves of a common core based tree whose center can be any node of the network.

1.4 Fault Tolerance in MPLS Network

Any of the resources within a network might fail. In an MPLS multicast network, failures can occur in link, node and/or link-node. MPLS will be used in core networks where system downtime must be kept to an absolute minimum. Many MPLS LSRs may, therefore, exploit Fault Tolerance (FT) [8] hardware or software to provide High Availability (HA) of the core networks. In order to provide HA, an MPLS system needs to be able to survive a variety of faults with minimal disruption to data plane.

In this thesis, we explicitly deal with software faults only [9]. In software, there may be code running on line cards and controller cards. Any distinct element of the software may fail. This could be MPLS signaling code itself, the underlying IP transport stack, the routing software, or the operating system or device drivers.

The aims of FT in an MPLS system are:

1. To preserve established LSPs (in data plane) software so that there is no (or minimal) disruption to data flow
2. To avoid resource leakage especially in the switch when failure happens during the state transitions within the signaling protocol
3. To cause minimal disruption to the processing of new signaling requests both during normal processing and during failover processing.

Underlying FT is the requirement to replicate data from a primary copy of the software to a backup copy. Usually the primary and backup copies run on distinct hardware instances to provide full protection. Thus FT has a prerequisite that the system can operate in a distributed way, passing messages between processor cards.

1.5 Motivation for Thesis

Today's businesses use more internet, groupware, multimedia, and client/server technologies; they need more multicasting applications. Video conferencing, audio conferencing, online training, news distribution, software distribution, and database replication are all good candidates for multicasting, one-to-many communication from a

source to a group of selected destinations. Multicasting applications can minimize the demand for network bandwidth while delivering information from a source to multiple destinations via one stream. With the increasing need to save time and money, multicasting offers a substantial advantage over currently used technologies.

MPLS being one of the main architectures that will likely be deployed mostly on the Internet to achieve QoS goals. In the future, it is expected that congestion and faults on a Label Switched Path (LSP) will seriously affect service contents and recovery and restoration of such LSPs would be required to realize a fault-tolerant MPLS network.

Most of the ongoing research work for MPLS is concerned with unicast failures. Different methods have been proposed to address this problem. But no concrete solution is provided. Not much work has been done for MPLS multicast. Multiple failure recoveries are not provided in the different approaches of MPLS unicast networks as described in chapter 2. RFC 3353 for MPLS multicast proposes initial information. As MPLS is an important architecture deployed in internet and multicast technology is important for communication, we feel it important to work in MPLS multicast for multiple failure recovery. This leads to the development of our scheme for finding a backup path by segmentation approach.

1.6 Thesis Contribution

To transfer the data in MPLS network requires traffic engineering mechanisms to compute backup paths and to perform rerouting after a failure has occured. In this thesis, we address the problem of fast recovery of a multicast routing tree after failures occur. An MPLS network receives traffic directly from multicast hosts attached to MPLS routers, or from networks that simply relay multicast traffic from other multicast hosts. When a link of the multicast routing tree fails, a certain number of multicast hosts accessing the tree directly or through other networks are dropped from the communication. In this thesis, we present an algorithm that aims at selecting a backup path by segmenting the primary path in a given multicast routing tree to improve the resilience of the tree for any type of failure. The backup path selected by the algorithm minimizes the number of group members dropped from a multicast communication on failures as compared to backup between label edged routers. We provide a specification, complexity analysis and simulation of our design.

1.7 Outline of Thesis

The rest of the thesis is organized as follows: In Chapter 2, we describe different failures and its recovery mechanisms for unicast and multicast networks. It also includes different methods and their disadvantages. This is followed by an introduction to our proposed recovery mechanism for multicasting MPLS network. In Chapter 3 we give an overview to our approach of segmented backup for MPLS multicast networks,

an algorithm for backup path and some recovery scenarios. Chapter 4 describes our problem design and graphical results in detail. Finally Chapter 5 concludes with discussions and future research directions.

1.8 Summary

In this chapter, we described in detail about MPLS, its components and its working. Then we focused on multicast, multicast in MPLS and need for multicast in MPLS. Also we described fault tolerance in MPLS networks.

Chapter 2

Failure Recovery in MPLS Network

In this chapter, we focus on different types of failures in MPLS networks. We also discuss in detail different failure recovery methods. We discuss failures in the context of MPLS unicast and multicast networks.

2.1 Objective of Failure Survival

The key objective of failure survival in an MPLS network is to minimize the disruption of data traffic of any failure [10]. Where possible, established LSPs (which may be carrying data) should not be disturbed at all while the failure is recovered. This means that links and cross-connects should stay in place, and data packets should not be discarded. In practice, many failures will require some disruption as new

resources take over from the old. This disturbance should, however, be kept as small as possible. A figure of 60ms is often quoted in the telecommunications world, as the largest disruption to voice traffic that can be managed by the human brain before the effects become noticeable as a break that interrupts understanding or flow. This means that, ideally, any failure should be detected, reported and repaired within a total of 60ms. Even if the repair of an LSP takes longer than 60ms it is still important that the connection is restored automatically. If there is disruption on the data flow, an important consideration is whether data are lost or if so, how much. Neither IP networks nor other networks such as ATM or Frame Relay attempt to provide reliable delivery of data, other than by using higher layer end-to-end protocols such as TCP over network protocols. However, if a substantial amount of data are lost, such protocols may declare the connection failed, and require reconnection.

A slightly lower priority aim is that the signaling service should remain available. That is, that it should continue to be possible to establish new connections for data traffic after the failure. It may be that new connections cannot be signaled while failure is being repaired. Although that is undesirable, it is generally acceptable for a user to retry a connection attempt (e.g. redial at a telephone) if the connection fails to establish for the first time. Given the statistical likelihood of a new connection being attempted during a failure repair, it is often considered acceptable that signaling is temporarily suspended.

The process of repair in one part of the network should, of course, cause as little disruption as possible to other parts of the network. Broadcasting failure information

around the network could seriously disrupt other signaling and data traffic. It is worth noting that the typical requirement is to survive a single network failure. Many network providers and device vendors are not attempting to provide solutions that survive multiple concurrent network failures. While this does reduce complexity, it implies that the recovery time of failed component must be low to ensure that the period of vulnerability is as short as possible.

All of the solutions to these requirements involve forms of redundancy whether within links, as extra links in the network, or through the provision of additional hardware components within a switch. The cost of these solutions imposes an additional requirement: those redundant resources should be kept to a minimum and preferably shared among potential users.

2.2 Types of Failures

Any of the resources within a network might fail. To provide a proper high availability network, the network provider must predict and plan for any of these failures.

As MPLS is connection oriented, it has greater vulnerability towards failures. Failures can occur in link and node. After a fault is detected, the LSRs notify the occurence of fault to all affected LSRs and search for alternate or backup path for an alternate traffic.

Figure 2.1 shows node and link failures in a unicast MPLS network.

When fault occurs in the network at point A, it is treated as node failure, ingress

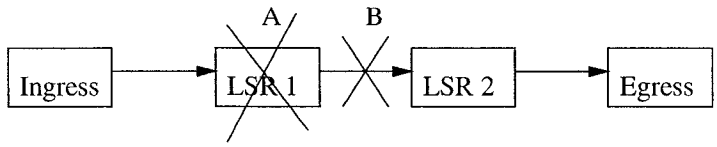

Figure 2.1: Node and Link failures in MPLS Network

will come to know about failure as there will not be any data transfer from LSR 1. When the failure occurs at point B, it is treated as link failure. Here when node fails, links associated with the node also fail. A link failure may be distinguished from a node failure by a method explained in [11]. The link failure is detected when the RSVP "Hello" messages are not reachable via the primary path, the point of local recovery (PLR) uses the RSVP "Hello" to determine whether its neighbor is reachble via another path instead of the failed link. If this is the case, the PLR can conclude that a link failure has occured. If not, the failure is a node failure.

2.3 Fault Recovery Mechanisms

Different types of protection schemes have been deployed in different layers of the network. These schemes can be classified as Dynamic Restoration and Preplanned Restoration.

Dynamic restoration dynamically allocates spare resources for the alternate route. It has the advantage of being cost efficient since none of the resource is allocated before the failure. The drawbacks of this approach are that the restoration may not be guaranteed if the allocation of a new route fails and restoration time could be longer due to searching and deployment of an alternate route. This makes these

schemes only suitable for best effort services, which have been deployed in the IP layer.

Preplanned restoration reserves backup resources at the time of establishing the working paths. Since these schemes do not need the time-consuming connection establishment process, preplanned restoration is capable of restoring traffic within a very short time. However drawback of this scheme is high cost.

For better resource utilization, resource sharing can be used. If two primary paths do not fail at the same time, their backup paths can be shared with each other, and thus cost is reduced.

In the following subsections, we discuss the different mechanisms that deal with fault recovery in general.

2.3.1 Link Failure Recovery

Fault recovery may be done as link rerouting or end-to-end rerouting [12]. In link rerouting an alternate path is found between the two LSRs on the ends of the failed link. For fast recovery alternate path may be pre-stablished, or saught dynamically after fault notification. In end-to-end rerouting, an alternate path between the ingress and Egress LSRs that is completely disjoint from the failed path is found. Recovering based on end-to-end rerouting is more capable of handling node faults or multiple link failures. Again the alternate path may be pre-established and resources may be reserved along the alternate path.

2.3.2 Local Repair

Another method used to recover from faults is based on Local Repair [13]. In this method, when a failure occurs in the network, an LSR upstream of the failure can attempt to re-signal the LSP. LSP signaling relies on IP routing, and therefore can take advantage of the fact that the routing table may be updated with new routes to the downstream nodes. Note, however, that this may take many seconds, and will not necessarily result in a new route being available. This is explained in Figure 2.2. If the failure occurs in link between nodes A and B, LSR A will direct the traffic on the next route to LSR A - LSR D - LSR E - LSR C.

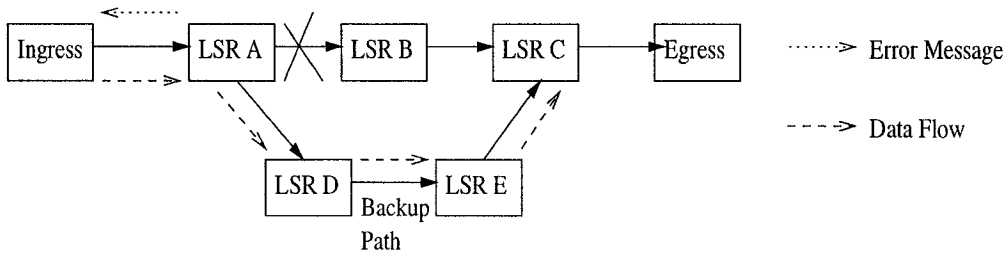

Figure 2.2: Local Re-routing around Link Failure

Data are forwarded based on the (incoming interface, incoming label) to (outgoing interface, outgoing label) mappings and not on the information in the IP routing table. Therefore updates to the IP routing table do not affect the data flow. Data paths can only change once a new LSP has been signaled and devices on the LSP programmed with the new label mappings.

When there is a link node failure within an IP network, the change in topology is distributed by the routing protocol and the routing tables are updated at each node. An LSP can be re-signaled from the ingress and may merge with components

of the old LSP downstream of the fault. Re-routing is, however, usually restricted to the point of failure detection and the ingress—if each LSR on the path attempted to re-route and re-signal the LSP, but failed (e.g., due to inability to find a route that matched the requested constraints), it might take far too long for the error to finally propagate back to the ingress node.

Because this re-signaling is time consuming and may in any case not result in successful re-establishment of the LSP, the signaling protocols impose some restrictions on the extent of local repair that is supported. Since network topologies are rarely full meshes, local repair might not succeed, and re-routing may need to be resolved at the ingress.

Local repair relies on the speed of propagation of routing table updates. This can be slow (up to 30 seconds), which is unacceptable for many MPLS applications. Further, even if the routing table update is quick, this solution requires additional signaling at the time of failure, which will further delay the restoration of a data path.

2.3.3 Protection Switching

Another method called Protection Switching [14] is used to ensure recovery from link or node failure with minimal disruption to the data. Many references to this function include a target failover time of 60ms, which is reputed to be the longest acceptable disruption to voice traffic.

In Protection switching, data are switched from a failed LSP to a backup LSP at the repair point (conventionally at the ingress). The backup LSP is usually pre-

provisioned. This can be considerably quicker than local repair since the backup LSP does not need to be signaled at time of failure.

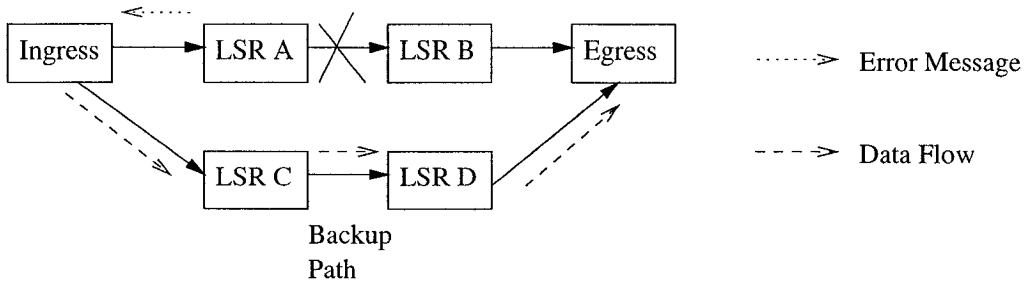

Figure 2.3: Protection Switching

As shown in Figure 2.3 data are switched on the primary path. The backup LSP takes a less favorable path, is ready and is set up, but does not carry any data. When any failure in primary LSP is reported back to the ingress LER (perhaps using notify messages), data are immediately switched to the backup LSP.

It is also possible to consider a scenario where the backup LSP is configured in advance at the ingress, but is not signaled until the failure is reported. This has the advantage that the network resources are not tied up by the backup LSP, but increases the failover time and is subject to the prospect of no resources being available when the backup is needed.

The main concern with protection switching is the speed of repair. The error must be detected and reported to the repair point. The backup LSP must be prepared, and finally the data must be transferred to the back up LSP. The options described above all require some amount of protocol signaling at the time of failover. This varies from simply propagating the error from the point of detection to the point of repair, to the full signaling of the backup LSP.

Obviously, the more signaling is required, the less likely the failover is to be timely. It is generally accepted that significant amounts of signaling (especially if more than one LSP has been broken by the failure) will not provide good enough failover times for most uses of LSPs. In fact, many people are suspicious that simply signaling the failure back to the repair point may compromise the failover.

The cost of providing backup facilities increases as the required speed of the failover increases. Recovery systems that require substantial signaling at the time of failure also take less network resources and are therefore cheaper. Quick mechanisms require more dedicated resources and are therefore more expensive.

Another important concern with the protection switching is that the resources must be pre-allocated to protect each primary LSP. This can be very expensive since resources used for the backups cannot be used to carry revenue-generating traffic.

The simplest has a single LSP providing the backup for more than one primary LSP. Since it is unlikely that more than one primary will fail, this offers a good solution, but it does require that there is more than one primary LSP between ingress and egress—something that may often not be the case.

Another option that works when there are multiple data flow between ingress and egress is to use the backup LSP for low priority data. When the primary fails, the low priority data are dropped or reverts to best effort IP transfer.

In a complex network, the issue may be wider than reducing the number of end-to-end backup LSPs. In this case, there is a need to reduce the amount of resource used on links in the core of the network.

Problems and Remedies

1. If the system has two primary paths, the question is how should LSR 1 behave if both primary LSPs do fail and data start to flow on both backup LSP/LSPs. A probable answer is that the backups are treated as first-come first served so that the data on the second backup are to be used as simply dropped at LSR 1. This is hardly satisfactory, however, if both primary LSPs believe they are protected, and better answer involves signaling to the ingress of the second primary that it is no longer protected.
2. This leads to another question, which is how to detect and indicate to the ingress points that a backup has dropped being used (after restoration of a failed primary) and both primaries can consider themselves protected again.
3. This system can lead to another level of complexity where links and nodes can have both primary and backup resources. Primary resources can be committed only once, but backup resources could be over committed many times, leading to two separate resource spaces to be managed.

2.3.4 Fast Re-route

Fast re-route [15] is a process where MPLS data can be directed around a link failure without the need to perform any signaling at the time failure is detected. Unlike protection switching, described above, there is typically not even a requirement to propagate the error to the repair point using the signaling protocol—the repair point

is the point of detection.

Most fast re-route protection schemes rely on pre-signaled backup resources [16]. When the failure is reported to the repair point, it simply updates the programming of its switch so that data that were previously sent out of one interface with one label are sent out of a different interface with another label.

The simplest form of fast re-route is called link protection. For link failure fast re-route an LSP tunnel is set up through the network to provide a backup for a vulnerable physical link. The LSP provides a parallel virtual link. When the link fails, the upstream node switches traffic from the physical link to the virtual link so that data continue to flow with minimal disruption. This is explained in Figure 2.4.

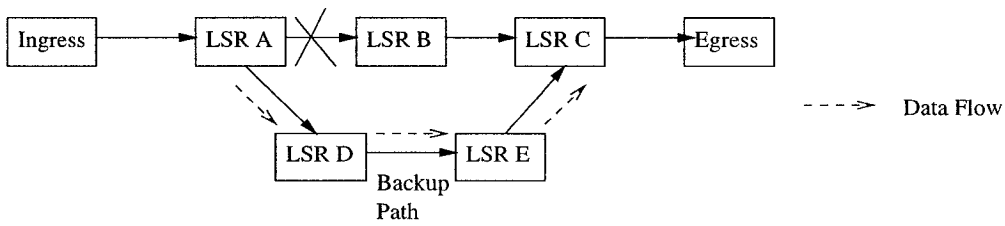

Figure 2.4: Fast Reroute Link Protection

Figure 2.4 shows a backup tunnel that has been set up to protect the link/node between LSRs A and B. When the link and/or node fails, the primary LSP is redirected down the backup tunnel so that the data still flow from A to B. The capacity of the backup LSP should, of course, be sufficient to carry the data from the primary LSP. If all LSPs on a link are to be protected then the capacity should equal the bandwidth of the protected link. This can potentially lead to a huge amount of backup bandwidth being reserved, especially if multiple links must be protected in

this way. But there are some specific limits placed on the use of label spaces when this method of fast re-route is in use. The LSP that provides the backup virtual link is used as an LSP tunnel. That is, the data packets that would have been sent down the physical link have an additional label added and that top label is used to forward the packet along the backup LSP. When the other side of the broken link is reached (the egress of the backup LSP) the top label is stripped from the packet, and the data are forwarded according to the lower label like label stacking.

The main issues with link protection are increased complexity of configuration (each protected link must have a backup tunnel configured) and the amount of resource that must be preserved in the network.

Link protection only handles the case where a single link between two LSRs has failed. However, it is also possible that an entire LSR will fail. Once this information has been passed back upstream, each LSR can determine the correct labels to use in the label stack when it re-routes an LSP after failure. Note that fast re-route path protection requires a considerable investment in network resources as the LSP may need three times the resources to cover forward, reverse and alternate paths. Several proposals have been made to allow resource sharing, but these methods are as complicated as those described in protection switching.

A more sophisticated fast re-route scheme is described in [17]. This provides a protection path for a given LSP that is established at the same time as the primary LSP. The protection path provides an alternate path that can be directly accessed from any point on the primary LSP such that data can be switched over to that

path without communicating the failure to any other devices. This is different from protection switching where the rerouting of the data can only occur at certain repair points, which must be first notified about the failure.

To establish this protection path

1. the primary LSP is signaled from ingress to egress
2. a reverse path is signaled from egress back to ingress using the same route
3. an alternate path is signaled from ingress back to the egress using a path that is disjoint from the primary LSP.
4. the ingress sets up its label mappings so that any data flowing on the reverse path are forwarded to the alternate path.

A major concern with this technique is the length of the data path that results from repairing a failure near the egress. Certain applications (such as voice) are sensitive to the delay in data transfer. If the LSP is long, the delay may already be close to the maximum limit. Tripling the path length may degrade the transfer data too much. This can be mitigated at the ingress, which can be sensitive to data flow on reverse path. If data are detected on the reverse path, the ingress can assume that there has been a failure down stream and can start sending data directly on the alternate route. Some small delay may be possible if packet ordering is important.

To overcome the problem of length of the data path described above, a shortcut path can be set up to connect through the alternate path. A failure upstream of the shortcut path is handled as described above, but a failure occurring downstream

results in data being routed back up the reverse path, along the shortcut path and out along the alternate path.

This shortcut approach however, brings significant additional complexities. Choices must be made as to which short cuts need to be established and which extra functions the signaling protocols are required to do. Furthermore, the device where the short cut meets the alternate path must be capable of handling LSP merge or must be able to detect data arriving on either the shortcut or alternate path and dynamically update its label mappings accordingly.

2.4 MPLS Unicast Network

Until now most of the methods are proposed for unicast failure recovery [12, 14, 18].

Thomas et al [12] examined distributed methods for fast fault recovery using modified Label Distribution Protocol messages. He also focused on link rerouting and end-to-end rerouting for traffic and performance monitoring. He also suggested fast rerouting techniques for traffic monitoring to collect feedback information about network conditions.

Protection switching and fast reroute methods are explained in [14]. Fast rerouting is a method where traffic is reversed at the point of the failure back to the ingress and redirected via an alternative preconfigured LSP. This method is known for low packet loss as it redirects the data near a fault without prior notification to the ingress node, however it may cause long delays not suitable for real-time services.

Shandilya [18] explains the modified OSPF protocol, for finding a secondary fault-tolerant backup path. This also brings forth extension to the OSPF to allow finding local fast rerouting paths on all the nodes along an LSP. He also proposed to run instance of another algorithm on LERs to calculate explicit routes (E-LSPs) satisfying the end-to-end delay QoS criterion.

One scheme is suggested in [19] called adaptive segment path restoration. The basic idea behind this approach is to divide an LSP into several segments. Each segment of the primary path is provided with a backup path. The segmentation of the primary path is adaptive to the topology of the network, allowing for more efficient resource usage whilst yielding restoration times comparable to link restoration. The segmenting principle is that all the LSRs having the same restoration length are put in the same segment. Then in each segment, a backup path is found to cover possible link failures within this segment. The purpose is to make the restoration length and backup hops satisfy the QoS requirement of the different services being transported.

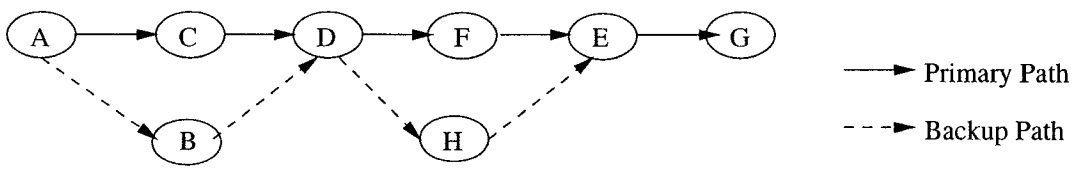

Figure 2.5: Recovery through Adaptive Segment Path Restoration

This is explained in Figure 2.5, where the primary path is divided into different segments and each segment is provided with a backup path. Here the last LSR of each segment is called Segmentation Point. However, if the network is large there will be many backup paths which will cause more resource utilisation. To avoid this,

the concept of sharing resources explained in [20] may be used. In this method, data will flow through a backup path after failure occurs. This backup path is the primary path of another network. But this leads to the problem of traffic engineering and bandwidth utilisation as data will get overflowed. This can be well explained in Figure 2.6. Here if the failure occurs anywhere in the path 1 (Ingress - LSR A - LSR B - Egress), ingress will forward the traffic on the primary of another path 2 (Ingress 1 - LSR C - LSR D - Egress 1). Here at LSR C merging of data takes place and extra label will be stacked and splitting takes place at LSR D. Here we also have to think about sizing of LSP after data flow on the primary LSP of another path or backup path so that LSP does not overflow [21]. One has to decide guranteed bandwidth requirement for Class of Service (CoS) [22].

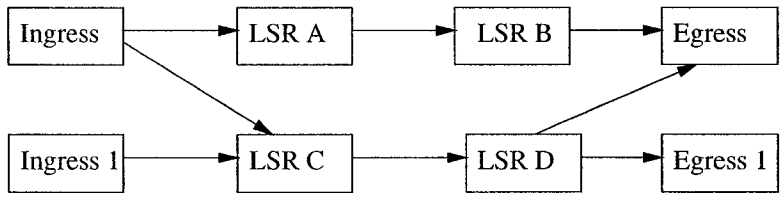

Figure 2.6: Recovery through Sharing Resources

Partial ingress failure recovery is mentioned in [23]. The recovery method proposes a solution of having the redundant image of the control plane(CP) that is the part of the ingress, so that when the failure occurs in the ingress, packets will be forwarded according to the updated CPI (Control Plane Identity). Here in addition to the CP in ingress, CPI is introduced as shown in Figure 2.7. CP operates the software and is responsible for the coordination of the other components. When the CP fails it is treated as an ingress failure. So after the failure, all data will get forwarded

through CPI, which gets updated after each data transfer. When a failure occurs in the ingress, the bandwidth of the flow is distributed among the best possible path giving the priority to the new flow with guranteed service [24, 25, 26]. Bandwidth of each link is partitioned into two fractions, one for low priority data traffic, and one for the high priority stream traffic. This is described in dynamic partitioning [27]. Here partitioning is defined by a partitioning parameter, which changes according to traffic profile and intensity.

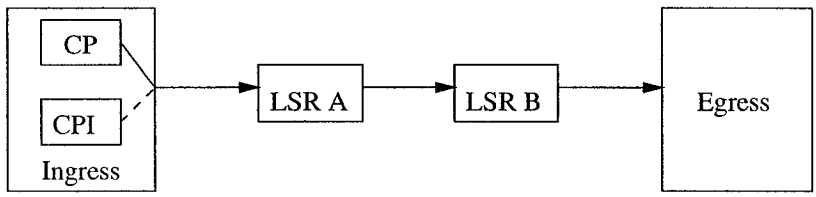

Figure 2.7: Ingress Failure Recovery in MPLS Network

Another approach has been defined in [28]. Here primary path is viewed as made up of small contiguous paths called primary segments. Many backups are provided to protect the small primary segments as shown in Figure 2.8.

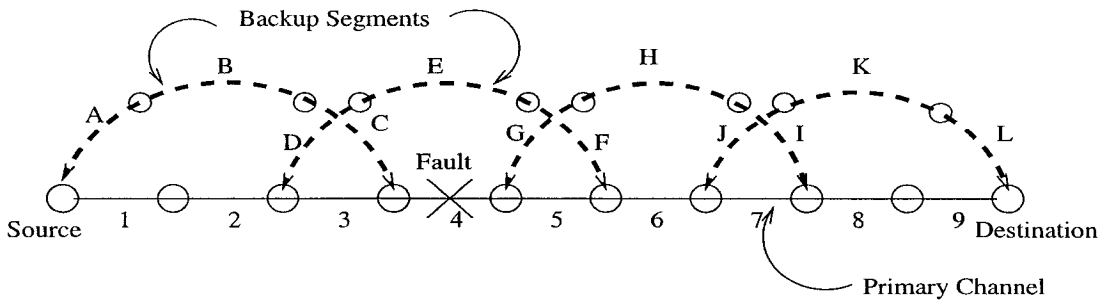

Figure 2.8: Multiple Backup Paths for Multihop Network [28]

Moreover backup consists of different links. As shown in Figure 2.8, 1 to 9 are the links of the primary path with intermediate nodes and backup links are A to L.

The primary segment span links 1-3, 3-5, 5-7, 7-9, while their corresponding backup segments span links A-C, D-F, G-I, J-L, respectively. Links A, B and C constitute a single backup segment. But when there is a big network and there are multiple hops, there will be many primary segments. This may lead to more resource utilization in terms of backup paths for each primary segment.

2.5 MPLS Multicast Network

Failure recovery in MPLS unicast is relatively easy to achieve as compared to multicast network. Not much work has been done for MPLS multicast. Recently published RFC 3353 [29] provides an overview on IP Multicast in MPLS environment and is in the primary stage. In MPLS multicasting, data are sent to different users at the same time using different LSPs according to bandwidth requirements. Analyses of general issues on supporting QoS for multicast applications and review of different ways to implement IP multicasting in different architectures including MPLS based networks are discussed in [30]. In case of failures in such networks, one still needs to deal with bandwidth and traffic engineering. A method that is available for recovery in MPLS multicast networks is proposed in [31] and is based on providing recovery through the receiver end points. In this case a backup path is provided between receiver end routers as shown in Figure 2.9.

After a failure occurs in the link, each edge router node is notified of the failure by Notification Messages (NM). When end points A and B receive the NMs, switchover

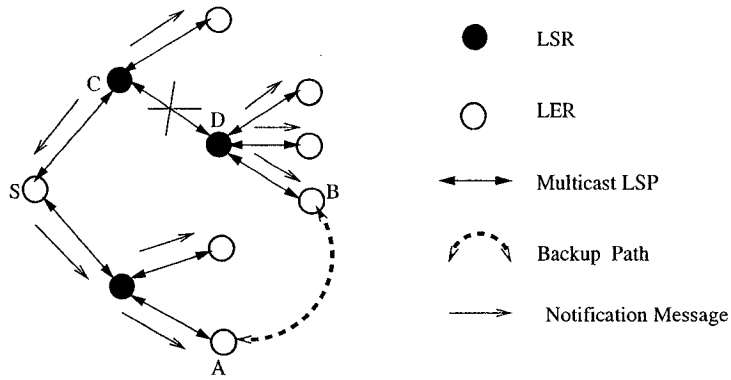

Figure 2.9: Recovery through receiver LERs for MPLS Multicast [31]

will take place. Here no matter where the failure occurs, NM needs to be received by all nodes in the network including end nodes for switchover. If the failure occurs in link C-D, NMs are sent to each node as shown in Figure 2.9. Here we see that NM may unnecessarily be sent to nodes other than A and B. It adds to more traffic in the network than sending NMs to only end nodes where backup exists. Moreover, the recovery mechanism in [31] deals with only link failure recovery but does not deal with node or link-node failure recovery.

2.6 Summary

In this chapter we briefly discussed different types of failures and different methods available to achieve failure recovery. We discussed these different approaches in the context of MPLS unicast and multicast networks. In the next chapter we will discuss our approach for MPLS multicast networks.

Chapter 3

MPLS Multicasting Recovery Mechanism

In this chapter, we discuss in detail our recovery approach for MPLS multicast networks along with proposed algorithm to form a backup path.

In this chapter, we describe MPLS multicast Fast and Local Reroute (FLR), a rerouting mechanism adapted to protect multicast routing trees from failures. MPLS multicast FLR extends the unicast mechanisms presented in Chapter 2. MPLS multicast FLR makes it possible to repair a multicast routing tree if a failure occurs by rerouting traffic on a pre-planned backup path. The local rerouting feature is provided by limiting the rerouting to its multicast root point. This rerouting mechanism uses the same components as unicast rerouting. A backup path must first be established in a multicast routing tree. Then, when a node detects a failure or recovery, it sends a notification message to the Path Switching LERs, which perform either switchover

or switchback. MPLS multicast Fast Reroute assumes that multicast routing trees are core-based trees.

3.1 Overview

Here we introduce multicast LSPs, which are the multicast counterpart to unicast Label Switching Paths. MPLS is able to create virtual circuits that map multicast routing trees without the need of establishing a distinct virtual circuit between a particular node and all multicast hosts of a multicast group. This is not the case with other switching circuits. A multicast LSP is a point-to-multipoint MPLS virtual circuit. Packets are forwarded on multicast LSPs the same way as they are forwarded on unicast LSPs. But here they are multiplexed by MPLS routers and forwarded on several links. Therefore, a multicast LSP must be established before a multicast communication can actually take place and terminated when the communication is over. Such tasks are performed by a signaling protocol. In MPLS networks, LSPs are associated with FECs. Packets that enter an MPLS network are assigned to an FEC and all packets from the same FEC entering in the network via the same ingress LER are forwarded on the same LSP. The FEC associated with a multicast LSP is the IP address of the multicast group whose traffic is carried by the multicast LSP.

In this chapter, we propose and evaluate a new scheme to construct a backup path, called Segmented Backup Path, which is a link between segmentation points (SPs). This is unlike end-to-end receiver edge router based recovery where the backup

is between edge routers. Segmentation approach is followed in [19] for unicast MPLS networks. There the segmentation point is the end point of each segment while in our case SP is the node that multicast data to more than one node and this SP becomes root for the next cluster. More details are given in section 3.3. Using example and studies we show that segmented backup has many advantages over end-to-end receiver edge router backups such as:

1. Better QoS
2. Number of backup paths
3. Number of NMs

In this chapter, we specifically provide an algorithm for segmentation point and cluster formation.

3.2 Segmentation and Cluster Formation

We define a Segmentation Point [33] as a point where traffic is multicast to more than one link. A segmented backup is provided between two or more SPs. Once the SPs and backup is defined, we define a cluster. A cluster consists of root node, SPs and backup path joining these SPs.

Here we propose a new algorithm to provide a backup path in the multicast network. This backup path allows recovery from multiple failures. The flow of the algorithm is explained in Figure 3.1.

The algorithm is described in Figure 3.2.

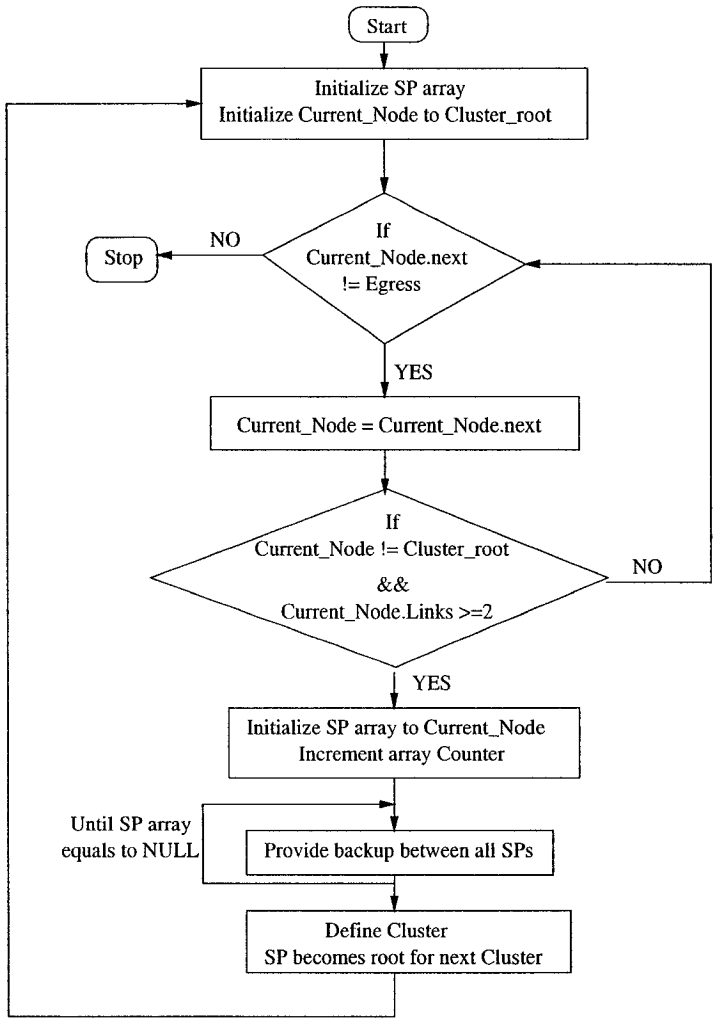

Figure 3.1: Flow to find Segmentation Point and Backup Path

```
1) Initialize Current_Node to be Cluster_Node

2) Initialize segmentation point array SP[ ] to NULL with
        Num_SP = 0 ; Temp_Num_SP = 0 ;

3) Repeat Steps I and II for each link of Cluster_Root

   Step I :
      Initialize Current_Node to be Cluster_Root ;
      If (Current_Node.next is not Egress)
         Current_Node = Current_Node.next ;
   Step II :
      If ((Current_Node is not Cluster_Root) And
            (Current_Node.links is greater than or equal to 2))
        { Initialize SP[Num_SP] to Current_Node ;
          Increment Num_SP ;
          }
       Else
         { If (Current_Node.next is not Egress)
               { Current_Node = Current_Node.next ;
                 Do Step II ;
                }
          }

4) Repeat Step III until SP[Temp_Num_SP+1] is not equal to NULL

   Step III :
        Provide backup between SP[Temp_Num_SP] and SP[Temp_Num_SP+1] ;
        Increment Temp_Num_SP ;

5) Define cluster between all SPs and Cluster_Root.
   Each SP becomes Root for the next Cluster.

6) Repeat Steps 2 to 5 for each Cluster formed in Step 5
```

Figure 3.2: Algorithm to find Segmented Backup and form a Cluster

The algorithm to obtain SPs and backup paths is applied clusterwise. Here we start from the first node (in this case LSR I) of the network as a root node and we check for each multicast LSPs for the root. Then we check for the next available LSR in each LSP. If that LSR sends data to more than one link we treat it as an SP as it is a multicasting point. The same is applied to each available LSP. After finding all SPs on all possible LSPs for that root node, we define the cluster. The cluster is formed of root node, SPs and backup LSPs joining these SPs. Here each SP acts as a root node for next cluster formation. This is explained in Figure 3.3.

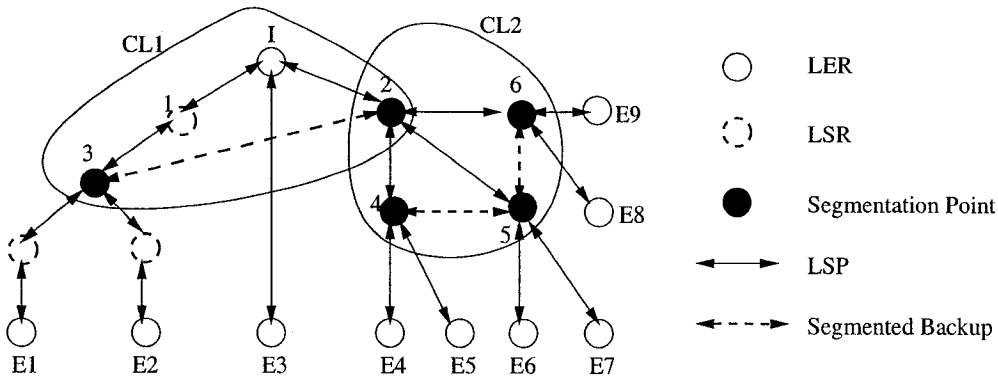

Figure 3.3: Cluster Formation

After applying the algorithm to the network as shown in Figure 3.3, we get LSR 2 and LSR 3 as LSRs that send data to more than one node. So we treat these LSRs as SPs. CL1 represents cluster 1, which is formed by root (Ingress), two SPs (LSR 3 and LSR 2) and backup path joining these SPs. Now LSR 2 (which was one of the SPs for CL1) becomes the root for the next cluster (CL2) formation. Here CL2 is formed by Root (LSR 2), SPs (LSR 4, LSR 5 and LSR 6) and backup path joining these SPs (path between LSR 4 and LSR 5, and LSR 5 and LSR 6). Our algorithm

deals failures only when cluster is formed. The link joining the egress is not recovered as cluster does not include that link. So we have not considered failure of the link joining the egress.

3.3 Recovery Mechanism

Figure 3.3 shows an example of how a Segmentation Point is found and how the backup LSP and hence cluster is defined. Here we will see how failure recovery is achieved with different scenarios.

3.3.1 Failure and Recovery Detection

In this thesis, we took into cosideration that links connect properly when the network is set up. A failure in the link is considered as software failure. Joins of the links to the nodes are firm even during the failure.

To detect failure, nodes regularly send KeepAlive messages on all the links on which they send traffic, and listen for KeepAlive messages on the links from which they receive traffic. In the context of bidirectional multicast LSPs, nodes actually send and listen for KeepAlive messages on each link they are attached to. KeepAlive messages are small messages that take a low percentage of the bandwidth of the link on which they are sent [31].

A failure must be detected as early as possible in order to keep the total repair time low. To do so, KeepAlive messages should be sent at a high frequency. However,

since detecting a failure triggers traffic switchover, failures should not be detected when a link/node has not actually failed. We call Tp the period used by nodes to send KeepAlive messages.

In case of link failure, we consider that a link has failed only when several KeepAlive messages are missing in sequence. Let the beat checking number $n >= 2$ be the number of KeepAlive messages that must be missing before a node reports a link failure. The failure detection time $T_{fdetect}$ is the time between the instant when a link fails and the instant at which a node that receives KeepAlive messages via this link considers that the link has failed. The distribution of the failure detection time is shown in Figure 3.4 [31].

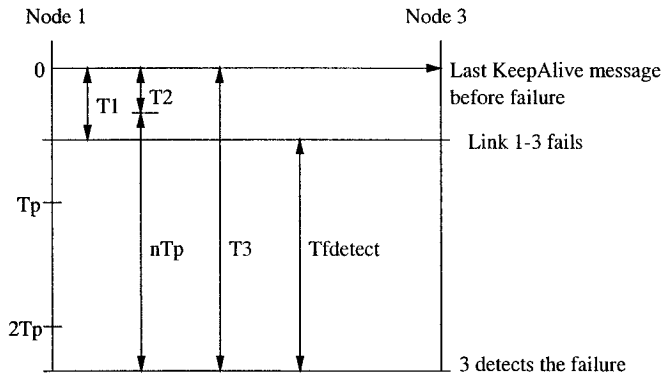

Figure 3.4: Failure Detection Scenario [31]

Suppose Node 1 is sending probes to Node 3 as shown in Figure 3.4. At time t = 0, Node 1 sends the last KeepAlive message before link 1-3 fails. Link 1-3 fails at time T1. Time T1 is uniformly distributed between 0 and Tp. Every period nTp, Node 3 checks whether it received at least one KeepAlive message from Node 1. Since the sender of KeepAlive messages at Node 1 and the receiver of KeepAlive messages

at Node 3 are not synchronized, the time T2 at which Node 3 checks and records the presence of the last KeepAlive message sent by Node 1 is uniformly distributed between 0 and nTp. Node 3 detects the failure at time T2 + nTp since no KeepAlive message is received between t = T2 and t = T2 + nTp. Therefore the time T3 at which the failure is detected is uniformly distributed between nTp and 2nTp. The failure detection time $T_{fdetect}$ is given by T3-T1.

The same mechanism can be used to detect the repair of a link. When a link is reported as failed, its two end nodes keep trying to send KeepAlive messages with period Tp. When one of these two nodes receives such a KeepAlive message, then the link is detected as repaired. Different from link failure detection where a single missing KeepAlive message is not enough for an end node to infer that a failure has occurred, the arrival of the first KeepAlive message on a link previously reported as failed indicates the recovery. For instance, suppose that node 1 sends a KeepAlive message on link 1-3 time T1 after link 1-3 has been repaired as shown in Figure 3.5. Since node 1 tries to send KeepAlive messages every period Tp, T1 is uniformly distributed between 0 and Tp. Node 3 detects the repair as soon as it receives a KeepAlive message, thus the recovery detection time $T_{rdetect}$ to detect the repair is equal to T1 and is uniformly distributed between 0 and Tp.

3.3.2 Failure and Recovery Notification

It is assumed in the proposed failure recovery mechanism that SPs do not fail. If failure occurs in any other point in the network data will be forwarded through the

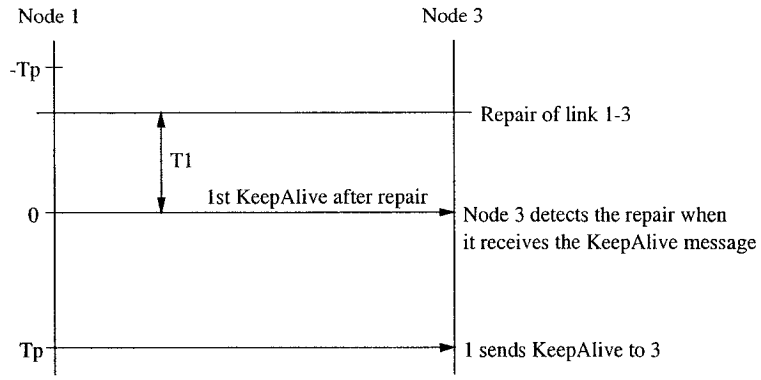

Figure 3.5: Failure Repair Scenario [31]

segmented backup paths rather than using receiver edge router technology as proposed in the literature.

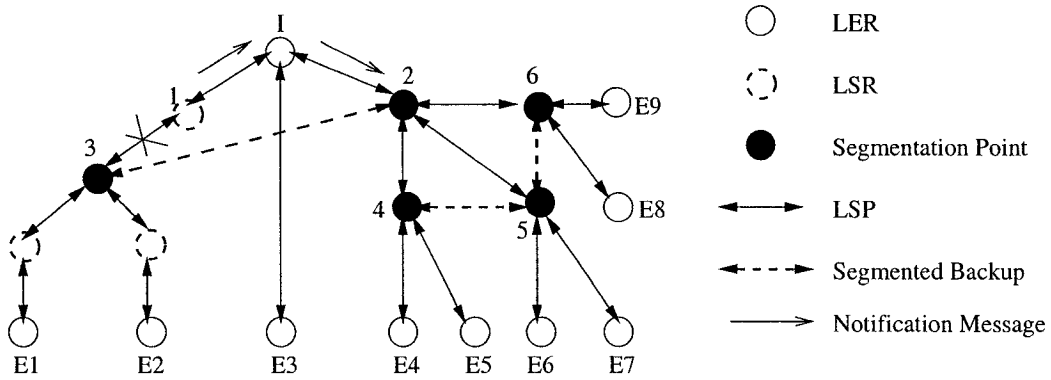

Figure 3.6: MPLS Multicasting and Notification

Consider the tree rooted at ingress I mapped by the bidirectional multicast LSP as shown in Figure 3.6. A segmented backup path SP_{32} has been computed between nodes 3 and 2. There are different primary paths that can be formed like I-E1, I-E4, I-E9 etc. The cluster CL_{I32} consists of links I-1, 1-3 and I-2, and SP_{32}. MPLS multicast can repair the multicast routing tree if any of these links within the cluster fails. Failure is detected by the end nodes for the link failure and by upstream and downstream nodes for node failure as described in section 3.3.1. After link/node

failure is detected, MNs are sent upstream/downstream to the root and SPs of its cluster and the backup path is activated.

When a failure occurs, the multicast routing tree gets split into two trees. For instance, if nodes 1 and 3 detect the failure of link 1-3, the original multicast routing tree gets split into one tree rooted at node 3 and another tree rooted at node 1. These two trees are shown in Figure 3.7. Each of the two nodes that detect the failure sends out a signaling protocol failure notification message to each SP of its cluster. When SPs receive the notification message, switchover will take place.

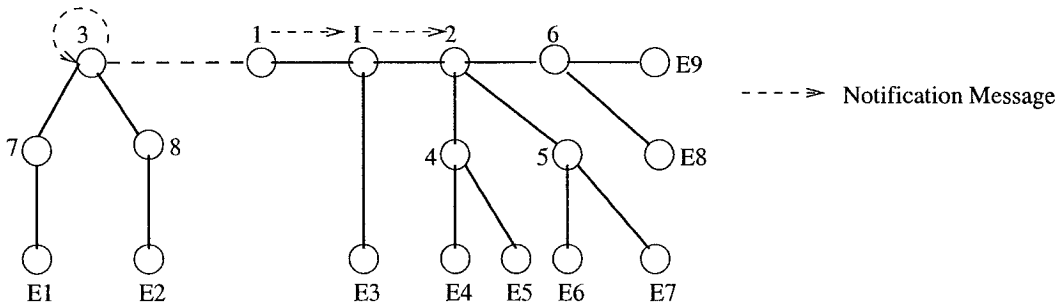

Figure 3.7: Spliting of Multicast Tree after Failure

When a failure is repaired, we use the same mechanism to propagate the repair information. Only the type of message used changes, i.e., signaling protocol recovery notification messages are used. When link 1-3 is repaired, nodes 1 and 3 send the recovery notification messages to the SPs of its cluster. When SPs receive the repair notification message, switchback will take place.

The failure notification time is the time between the instant at which a failure is detected and the instant at which both SPs are notified of the failure. Likewise, the recovery notification time is the time between the instant at which a repair is

detected and the instant at which both SPs are notified of the recovery. Since the notification mechanism is the same for failure and recovery, failure notification time and recovery notification time are the same.

3.3.3 Switchover and Switchback

Switchover consists of merging the backup path with multicast LSP that maps the original multicast tree (before failure). When both SPs receive the failure notification message, switchover takes place. After switchover is performed data flow through the segmented backup path. When the switchover is done, the cluster will look like as shown in Figure 3.8. When the failure is recovered, switchback is performed and the network returns to its original tree structure.

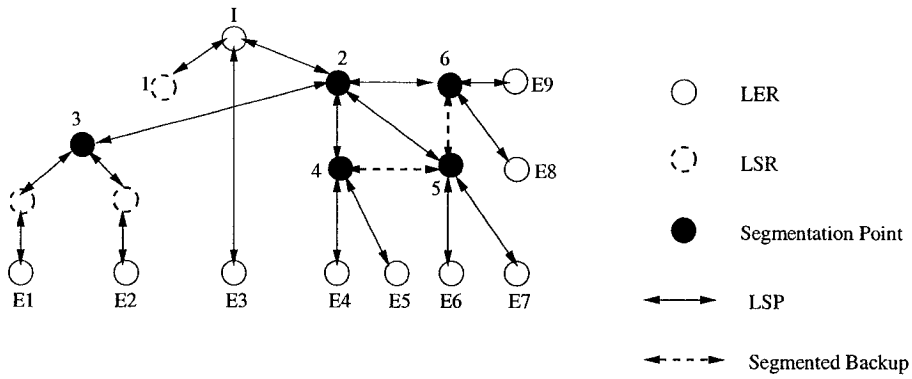

Figure 3.8: MPLS network after Switchover

If the link between nodes 1 and 3 fails, both the nodes 1 and 3 detect the failure of link 1-3. Then, both nodes 1 and 3 send failure notification messages which are propagated through the tree. All nodes of that cluster are notified of the link failure. When the end nodes of the cluster, i.e., SPs 3 and 2 receive failure notification messages,

they perform switchover by merging the backup path in the multicast LSP. A new multicast LSP results from the merging of the backup path and the multicast LSP that mapped the multicast routing tree before the failure. This new multicast LSP maps the multicast routing tree. Nodes now forward packets over the new multicast LSP and no LSR is dropped from the tree.

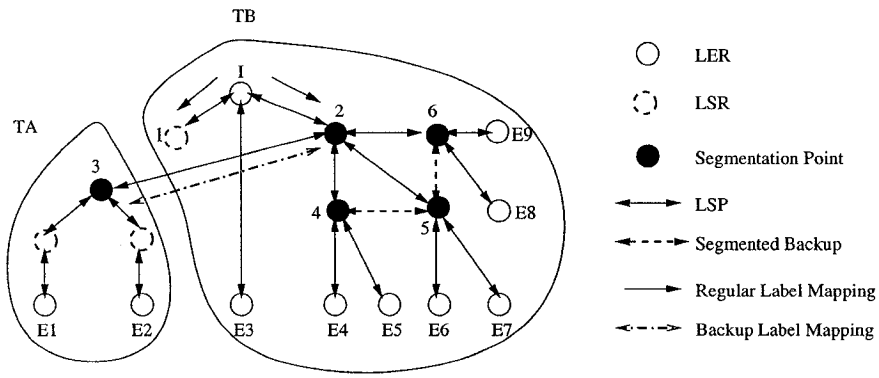

Figure 3.9: Data Transfer by node I after Switchover

So when node I sends any message in the network, it goes to nodes 1 and 2 with regular label mapping. But node 2 will forward it to node 3 with backup label mapping as shown in Figure 3.9.

Both SPs do not perform switchover simultaneously. When a link fails, a multicast routing tree is split into two smaller subtrees TA and TB. Suppose TA is the subtree that contains SP 3 and TB is the subtree which contains SP 2 as shown in Figure 3.9. After the link failure and before nodes 3 and 2 are notified of the failure, traffic sent by node 3 from TA cannot reach node 2 of TB and traffic sent by node 2 from TB cannot reach node 3 of TA. Suppose node 3 is notified of the failure and performs switchover before node 2. After node 3 has performed switchover and before node

2 has performed switchover, traffic sent by node 3 can reach node 2 but conversely traffic sent by node 2 cannot reach node 3. After both SPs have performed switchover, no node is dropped from the multicast routing tree. Switchover consists of a change in the MPLS forwarding table of the LSRs, thus switchover is almost instantaneous. The total time to repair the tree is therefore $T_{repair} = T_{fdetect} + T_{notif}$.

When nodes 1 and 3 resume receiving live messages over the previously failed link 1-3, they detect that the failure has been repaired. Then Nodes 1 and 3 send link recovery notification messages, which are propagated through the multicast routing tree. When nodes 3 and 2 receive those messages, they perform switchback by stopping the forwarding of packets over the backup path. After switchback is completed, the multicast routing tree is exactly the same as the original multicast routing tree that was in use before the failure. Here segmented backup path becomes active only after failure and becomes inactive after recovery.

3.3.4 Multiple Failures

Our approach can also recover from multiple failures like links, link-node or nodes. As explained in Figure 3.10 multiple failures can occur in links 1-3, 2-4, 2-5 and Node 1.

If the failures occur in different links of the same or different clusters, neighbouring nodes will detect the failure and send the NMs. If the failure occurs in any node, then next neighbouring nodes will detect the failure and send the NMs to SPs. As our algorithm treats the failures clusterwise, it recovers from multiple failures. However,

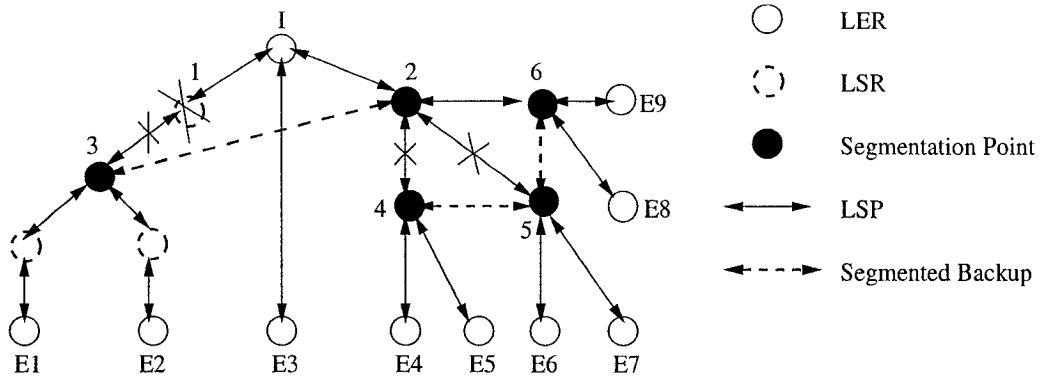

Figure 3.10: Multiple Failure Recovery

the necessary condition for recovery is that there should be at least a path available for NMs to reach SPs and it is possible to create a connected multicast tree.

3.3.5 Complexity Analysis

Assume that it takes time 't' to send NM from one node to another. If failure occurs in link 1-3, nodes 1 and 3 will detect it and send the notification messages to their SPs. But as Node 3 itself is an SP it updates itself for switchover with another SP. Node 1 will forward NM to Node 2 via node I. In this case, therefore it requires time '2t' and switchover will take place between nodes 3 and 2. In case of backup recovery based on receiver edge routers as described in [31] and in section 2.5 of chapter 2, it requires time '2t' for NM from 3 to E1 and time '4t' for 1 to E4, with a maximum of '4t' time. Here we assume backup is between receiver edge routers E1 and E4. The shortest backup between E1 and E3 would require time '2t'.

With our proposed mechnism we can achieve the following advantages over recovery based on receiver label edge routers:

1. Better QoS: As failures are treated cluster wise, it allows faster failure recovery with minimum delay. In our case, as network gets divided into clusters, it is easy to identify and recover failures locally within the cluster. Also failure in one cluster does not affect other clusters.
2. Number of backup paths: Since the number of core routers that are SPs are relatively fewer than the number of edge routers, it is anticipated that the number of backup paths required will be lower than the approach in [31].
3. Number of NMs: Number of NMs are fewer as NMs are confined to clusters as opposed to sending them over the entire multicast tree.

3.4 Summary

In this chapter, we have discussed in detail our approach for recovery in MPLS multicast networks. We discussed failure recoveries in multicast networks along with failure and recovery detection, repair and notification messages. Nodes must perform switchover or switchback when they are notified about failure. We also proposed an algorithm to form backup paths. In the next chapter, we present simulation of our approach for multicast MPLS network and compare different scenarios.

Chapter 4

Simulation and Results

In this chapter, we present the simulation of multicast MPLS using Fast and Local Reroute. At first we will give an overview of MPLS ulticast in OPNET, problem definition and its deployment strategies.

4.1 MPLS Network Modeling

OPNET [32] provides techniques that are well suited to understanding and solving MPLS related problems. Techniques for alternate decisions are documented along each step. In this methodology, we describe a tactical deployment where core and TE routing are combined. LSPs are sized based on traffic parameters. Traffic is associated with LSPs using the automatic binding options. Constrained Shortest Path First (CSPF) is used for routing the LSPs.

4.1.1 Methodology Steps

Figure 4.1 describes in detail about the methodological steps to create a network in OPNET.

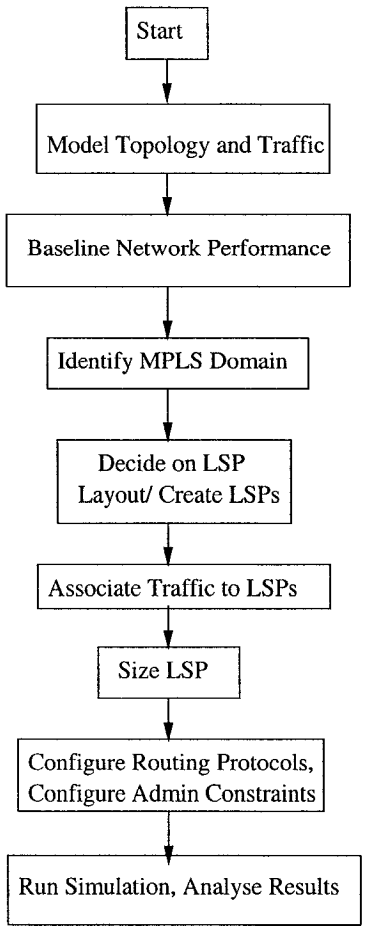

Figure 4.1: Steps for MPLS Network Modeling

These steps are described as follows:

Model Topology and Traffic

The first step in the methodology is to model topology and traffic.

1. Topology: OPNET provides a number of techniques to build or import the network topology. One may also build the topology manually by dragging and dropping objects from an object palette. Import techniques include text file import, Router configuration import (RCI) and import from supported vendor products.
2. Traffic: Traffic can be modeled explicitly using the application models or analytically using the background traffic models. Since MPLS is deployed typically in a service provider environment where there are many customers and hence, large amounts of traffic, it is recommend to use the analytical models to represent traffic flows.

Baseline Current Network

Once topology and traffic are modeled, Flow Analysis can be run to study the current network performance. Additionally, sections of network are identified that are underutilized and are good candidates for traffic engineering. Results and reports generated by Flow Analysis are used to understand the routing behavior of the network in its current state.

Identify MPLS Domain

Identify the section of the network or devices/routers where MPLS needs to be provided. If a tactical deployment is performed, this may be a subset of the network. If a full deployment is performed, all the device interfaces in the network are selected.

MPLS is enabled on the relevant interfaces selected. In general, the number of routers running MPLS determines the memory requirements and the number of LSPs set up.

Create LSPs

LSPs can be created based on bandwidth available, type of traffic sent. A single LSP is created for each demand that exists in the network. The advantage of creating LSPs based on traffic information is that LSPs are created only between points where there is network traffic. Sections of the network that do not carry traffic will not have LSPs. Additionally; traffic is automatically associated or bound to the LSP. The LSP sizes are set to the flow rates.

The MPLS object palette provides a choice of creating static or a dynamic LSP. Static LSPs have their routes and label mappings specified when the LSPs are created. The routes for dynamic LSPs are set up dynamically at runtime using a signaling protocol such as RSVP.

Associate Traffic with LSPs

If LSPs are created based on traffic flows, the LSP are automatically configured to associate traffic using the IGP Shortcuts approach.

Size LSP

For LSPs created based on traffic, the default LSP size is set to the bit rate specified within the flow.

Configure Signaling and Routing Protocols

This step involves selecting a signaling protocol in order to set up the LSP and a routing protocol that will be used for routing the LSP during the simulation.

1. Configure Signaling Protocol Parameters: OPNET uses RSVP signaling to set up LSPs. RSVP will support the bandwidth requirements and will reserve appropriate bandwidth on the links that the LSP traverses.
2. Configure Routing Protocol Parameters: Routing LSPs involves deciding on the path that the LSP should take. Dynamic LSPs use CSPF by default. In this case, constraints can be specified when selecting a path. CSPF selects routes based on the requirements of each LSP. Routes can be calculated offline and configured as explicit routes.

Configure Administrative Constraints

There are a number of constraints that can be configured to control the paths taken by the LSPs.

1. Allocation or subscription factor: The maximum reservable bandwidth specification for RSVP can be set such that the link is over-subscribed or undersubscribed. Typically, links are oversubscribed to improve overall utilization. Links may be under-subscribed in order to guarantee quality of service or allocate a portion of the link to non-MPLS traffic.
2. Resource classes (affinities): Links can be colored and LSPs can be configured

to prefer or not prefer a certain set of colors, thus controlling their routes. A 32-bit string is used to represent color and is referred to as a resource class. LSPs can be configured to include or exclude resource classes.

3. TE parameters: TE constraints such as the bandwidth, hop count and delay can be configured on the LSP.

Run Simulations, Analyze Results

The important statistics to monitor in an MPLS scenario are link utilizations, number of rejected LSPs and traffic statistics. OPNET provides two simulation technologies: Discrete-Event Simulation (DES) and Flow Analysis. In general, the trade-off is that DES provides more detailed simulation, but takes longer to run. The DES environment provides highly detailed models that explicitly simulate packets and protocol messages. Flow Analysis uses analytical techniques and algorithms to model networks. In general, DES is used to study the dynamic effects of protocols, such as TCP windowing or frame relay shaping. Also DES is used if application models are used and response times are to be analyzed to study transient network behavior, such as convergence times for routing protocols. Flow Analysis is appropriate to study networks in a steady state. Routing tables may be analyzed; iterative failure analysis may be performed and resource utilizations are studied as an example.

After following these methodological steps, we create an MPLS model as shown in Figure 4.2.

Here three different LSPs are created for three different egressess. LSP 1 for egress

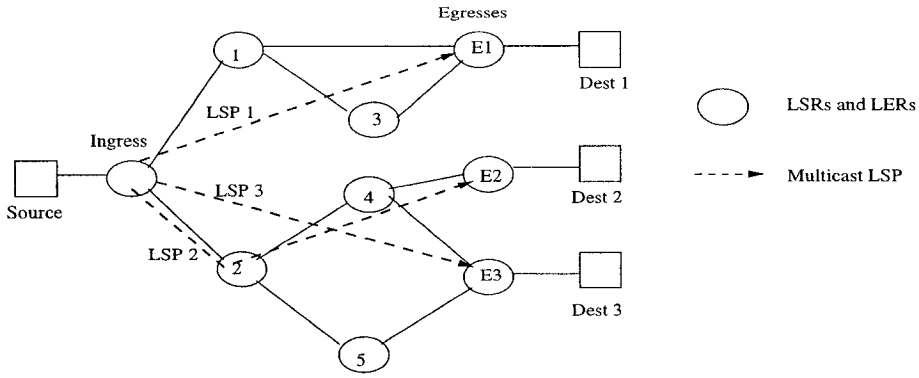

Figure 4.2: Model in OPNET

1 is Ingress - E1. LSP 2 for egress 2 is Ingress - LSR 2 - E2. LSP 3 for egress 3 is Ingress - E3. These are logical LSPs. We will see actual paths configured by different LSPs in section 4.2.2.

4.2 MPLS Multicast-OPNET

In MPLS, multicast trees are built on a per interface basis by combining label value and incoming interface. Multicast routing protocol is used to find one or more network topologies for a given group. One such topology is shared based tree where the source does not matter, e.g. Protocol Independent Multicast, Sparse Mode (PIM-SM) discussed in [35]. In OPNET, Multicasting routing uses the PIM-SM multicast routing protocol.

The OPNET model broadcasts multicast packets at the MAC layer. Multicast packet filtering is done at the IP layer instead of at the MAC layer. (In the real-world implementation of IP multicasting, some filtering occurs at the IP layer, but most filtering occurs at the MAC layer). The model's multicast routers can generate

application layer traffic. Generally, this is not a supported feature of most multicast routers.

4.2.1 Model Architecture

Each node in the network intended to use IP Multicast has an IP module, which spawns IP Multicast processes as child processes. IP Multicasting is implemented through the following process models as shown in Figure 4.3.

Process Model	Description
ip_igmp_host	Implements the Internet Group Management Protocol (IGMP) in Hosts
ip_igmp_rte_intf	Implements the IGMP in routers (with ip_igmp_rte_grp). ip_rte_V4 spawns an instance of this process for each multicast-enabled interface on a router.
ip_igmp_rte_grp	Implements the IGMP in routers (with ip_igmp_rte_intf). ip_igmp_rte_intf spawns a child instance of this process for every multicast group in the subnetwork.
ip_pim_sm	Implements the Protocol-Independant Multicasting-Sparse Mode (PIM-SM) multicast routing protocol in router nodes.

Figure 4.3: IP Multicast Process Model

The following sequence occurs when a host joins an IP multicast group. This is shown in Figure 4.4.

1. Application A joins a multicast group, Group 1, by sending a join request to the ip_igmp_host process using a remote interrupt.
2. The ip_igmp_host process sends an IGMP Membership Report message to the ip_igmp_rte_intf process on the neighboring multicast router. The IGMP Mem-

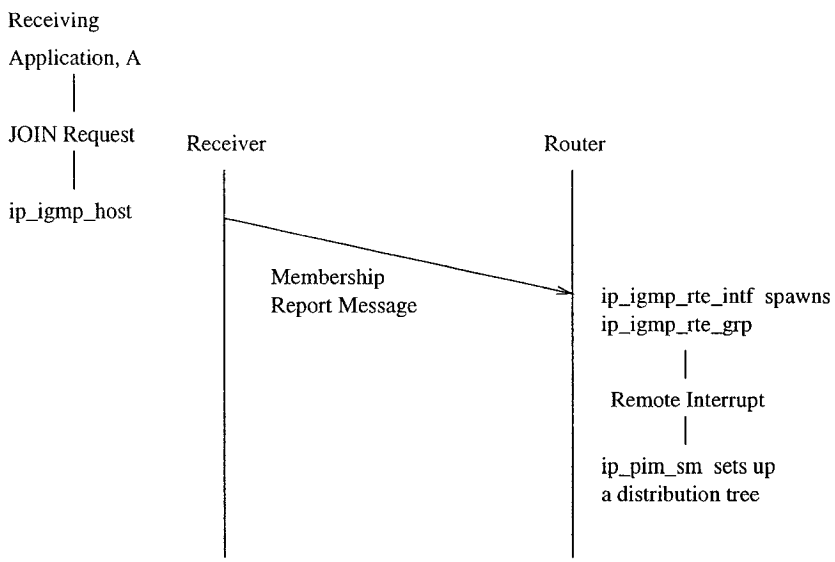

Figure 4.4: Multicasting Operation: Joining a Group

bership Report informs the neighboring multicast router that a local host has joined Group 1.

3. The ip_igmp_rte_intf process (on the router) spawns an ip_igmp_rte_grp process to handle Group 1.
4. The ip_igmp_rte_grp process sends a remote interrupt to the ip_pim_sm process, signifying that a local host has joined Group 1.
5. The ip_pim_sm process sets up a distribution tree for Group 1 so that packets sent to the group can receive Application A.

The following sequence occurs when a host sends packets to an IP multicast group as shown in Figure 4.5:

1. Application B sends a packet to a multicast group by broadcasting the packet on Interface 0. The workstation's IP process forwards the packet to its upper

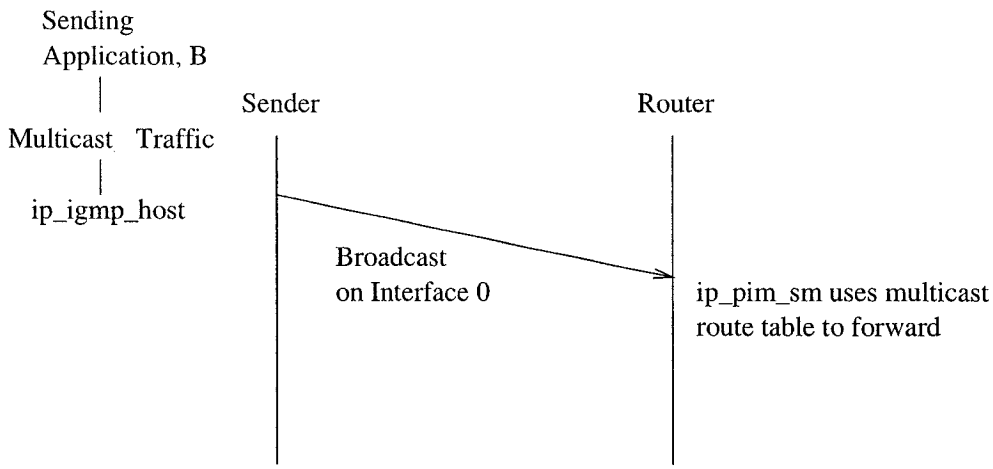

Figure 4.5: Multicasting Operation: Sending Traffic to a Group

layers.

2. The router's IP process forwards the multicast packet to the ip_pim_sm process.
3. The ip_pim_sm process on the router creates and sends one copy of the multicast packet for each out interface specified in the multicast route table.

4.2.2 Design in OPNET

The design of the network as created by OPNET is shown in Figure 4.6.

Here we have created three different LSPs as described in section 4.1.1. For all LSPs we have applied OSPF. So for LSP 1, it takes the path Ingress - LSR 1 - Egress 1. It takes Ingress - LSR 2 - LSR 5 - Egress 2 for LSP 2 where as Ingress - LSR 2 - LSR 5 - Egress 3 for LSP 3. For all these LSPs these are the shortest paths. We have made some changes in the design according to the OPNET functionality.

1. OPNET does not support the creation of an LSP directly between two nodes. The two nodes connect directly and there is an LSP directly going over this

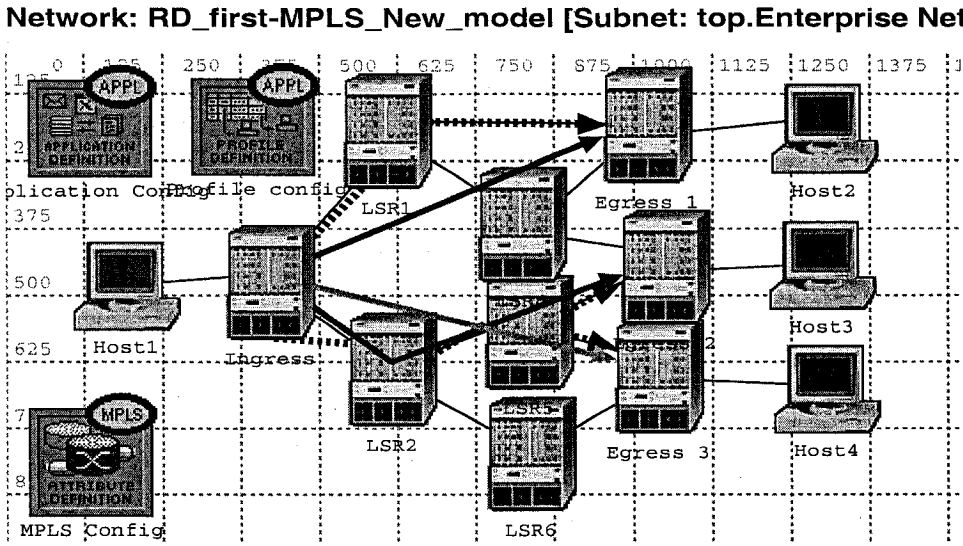

Figure 4.6: Original network in OPNET with its primary paths

link. This is not a valid configuration in real world. To overcome this, we have to add one extra LSR between the two nodes and configure LSP going through the LSR. So instead of adding extra node which will add extra resource, we changed the LSP starting from one of the SPs to root, which is already in the network, passing over the SPs.

Note: This problem is software problem identified as SPR-53642: "Simulation abort with single hop MPLS static LSP". Known workarounds include: Don't use single hop LSPs or add a LSR between the LERs. OPNET developers will investigate this problem and provide a fix in a future release of OPNET.

2. In OPNET, bypass tunnel can not be originated from Ingress. For our segmented backup, we are using bypass tunnel as a backup as that is the only way to create a bypass from intermediate node. So we created a root node(which will act as a dummy node) and originate segmented backup from root for the first cluster formation as Ingress is the root initially.
3. For the design of end-to-end backup LSP, OPNET can not create a backup path flowing over two end nodes. In OPNET one can not use LSP to protect the exit interface from the egress, however, as at that point, the packets are no longer travelling on an LSP, but are being routed via IP. So we had to add extra nodes on the exit interfaces after the original egresses, which will act as new egresses and use the end-to-end backup LSP flowing over the original egresses (e.g., LSR 7, LSR 8 and LSR 9 in the design).
4. For the design of end-egress backup LSP, as OPNET can not create a backup

path between two end nodes (as explained in 3) we have to create a backup going through either of LSR 7, LSR 8, LSR 9. As backup in OPNET should be ingress - engress, we have to create end-egress backup as a bypass tunnel. But as explained in 2, we can not originate bypass tunnel from ingress, we have to create it from root node.

With these changes applied to the network of the Figure 4.6, we get Figures 4.7, 4.8, 4.9 for segmented backup approach, end-egress backup approach and end-to-end backup approach respectively. In these figures, for all these networks, we are dealing with the failure of the link Root - LSR 1.

In Figure 4.7, primary path is Ingress - Root - LSR 1 - LSR 7 - Egress 1. Here segmented backup is Root - LSR 2 - LSR 1.

In Figure 4.8, end-egress backup is Root - LSR 2 - LSR 5 - LSR 8 - LSR 7.

In Figure 4.9, end-to-end backup is Root - LSR 2 - LSR 1 - LSR 7. To compare the results on equal grounds, we have simulated end-to-end backup approach with backup from Root - LSR 7 rather than Ingress - Egress 1. Here backup follows the same path as that of segmented backup path as it is the shortest path. But this is not the case everytime. If there are more nodes on the primary path, it will take other shortest path.

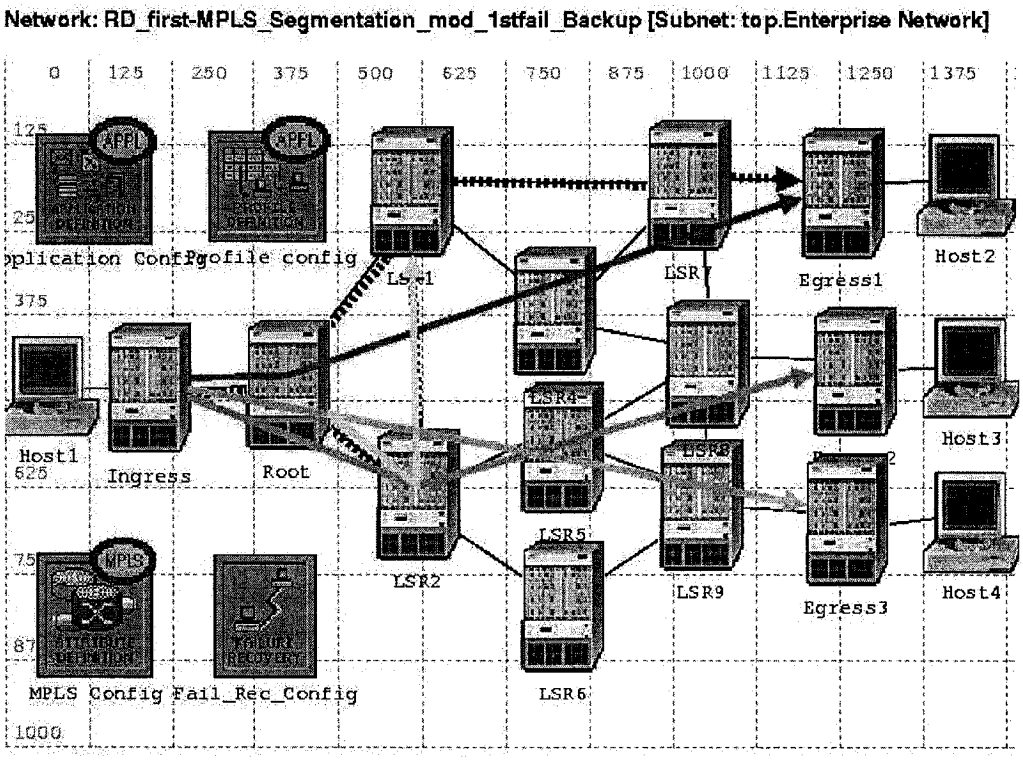

Figure 4.7: Segmented Backup Path in the network

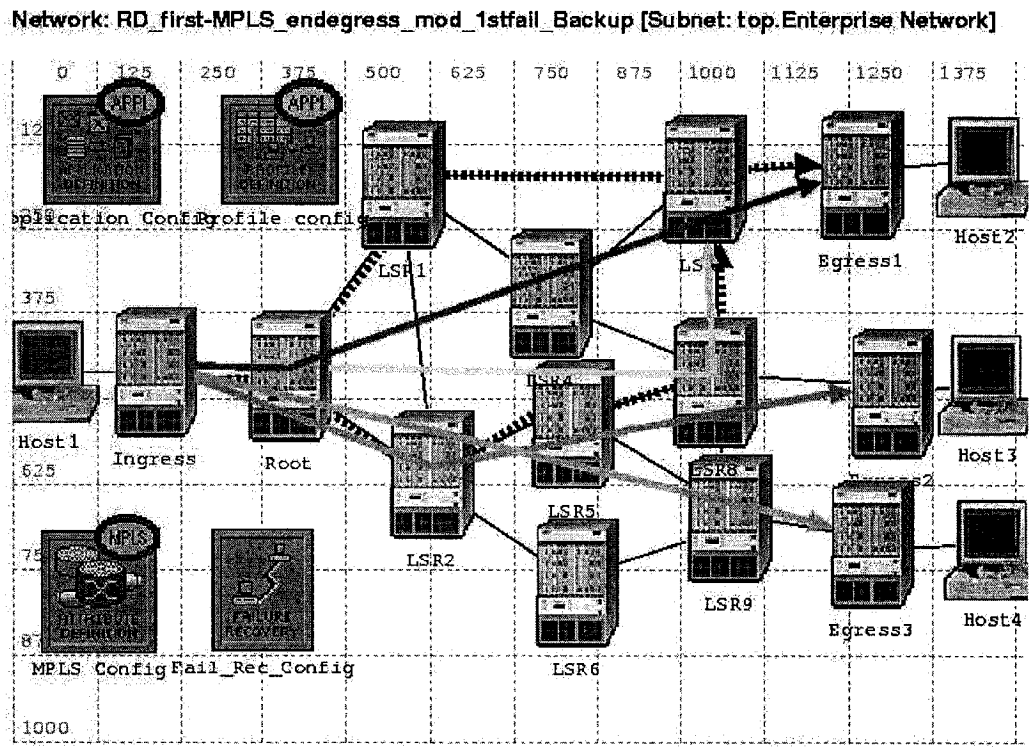

Figure 4.8: End-egress Backup Path in the network

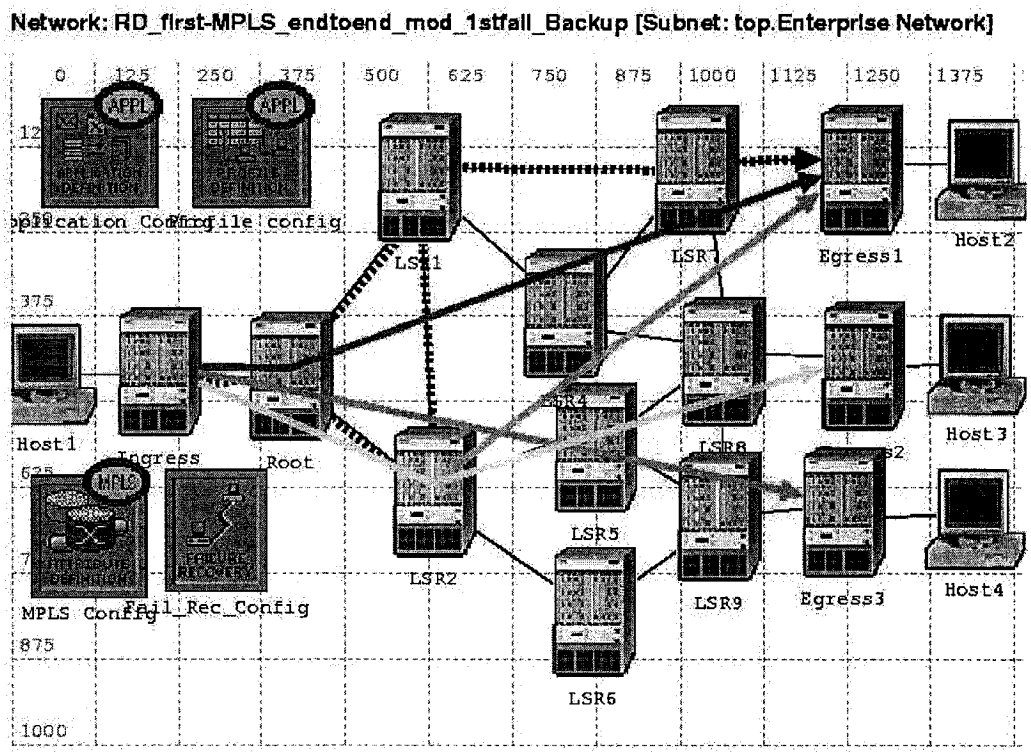

Figure 4.9: End-to-end Backup Path in the network

4.3 Simulation Results

The MPLS network simulator with multicast traffic support has been simulated over OPNET Modeler 10.0 with PIM-SM module. Various kinds of scenarios are used to analyze the simulation results among segmented backup approach, end-egress backup approach and end-to-end backup approach.

4.3.1 Simulation Topology

In our simulation, we use the MPLS Object Palette of OPNET simulator to generate a random network as shown in Figure 4.6. In the MPLS network, there are 17 nodes and 22 different links. In the network, 13 nodes are MPLS nodes and 4 nodes are source and destination nodes. Nodes are connected to each other by various links. In the MPLS domain, one node is ingress LER, 9 nodes are intermediate LSRs and 3 nodes are egress LERs.

4.3.2 Simulation Traffic

The simulation traffic used in this experiment is PCM Quality Speech with one voice frame per packet. The ToS specified is Interactive Voice (6) compatible for PCM quality speech. The traffic is generated in the source node and propagated to different number of destinations (receivers) in the topology. Default traffic trunk is created with 32,000 max bit rate (bits/sec). FEC is created with different destination addresses with respective ToS. Source node is connected to Ingress of the MPLS network and

destination nodes are connected to the Egresses of the MPLS network by point-to-point duplex links.

4.3.3 Experimental Setup

In the experiment, all source and destination nodes are Ethernet workstations. The ethernet_wkstn node model represents a workstation with client-server applications running over TCP/IP and UDP/IP. The workstation supports one underlying Ethernet connection at 10 Mbps, 100 Mbps, or 1000 Mbps. This workstation requires a fixed amount of time to route each packet, as determined by the "IP Forwarding Rate" attribute of the node. Packets are routed on a first-come-first-serve basis and may encounter queuing at the lower protocol layers, depending on the transmission rates of the corresponding output interfaces.

All ingress, egress and intermediate routers are Ethernet gateways. The ethernet2_slip8_gtwy node model represents an IP-based gateway supporting up to two Ethernet interfaces and up to 8 serial line interfaces at a selectable data rate. IP packets arriving on any interface are routed to the appropriate output interface based on their destination IP address. The Open Shortest Path First (OSPF) protocol is used to automatically and dynamically create the gateway's routing tables and select routes in an adaptive manner. This gateway also requires a fixed amount of time to route each packet, as determined by the "IP Forwarding Rate" attribute of the node. Packets are routed on a first-come-first-serve basis and may encounter queuing at the lower protocol layers, depending on the transmission rates of the corresponding

output interfaces.

Multicast LSPs are established using model of dynamic Label Switched Path (LSP). When this model is used, CR-LDP will establish an LSP from the source node of this LSP to the destination node of this LSP. Setup time, reroute time, total flow delays etc. are measured to analyze the performance under different scenarios.

All intermediate LSRs are connected PPP_DS3. This point-to-point duplex link connects two nodes running IP with data rate of 44.736 Mbps.

Here we have implemented OSPF protocol for the data transfer. Data flow on different paths for different scenarios is as shown in Figures 4.6, 4.7, 4.8, 4.9. The difference between end-egress backup and end-to-end backup is that end-egress backup includes two egresses while end-to-end backup has maximum of one egress. It follows through some intermediate LSRs.

4.4 Performation Evaluation

We have run the simulation for 3 mins (180s). We set link failure to occur at time 160s. We have simulated the network design for three different recovery approaches:

1. Segmented backup approach
2. End receiver backup approach
3. End-to-end backup approach

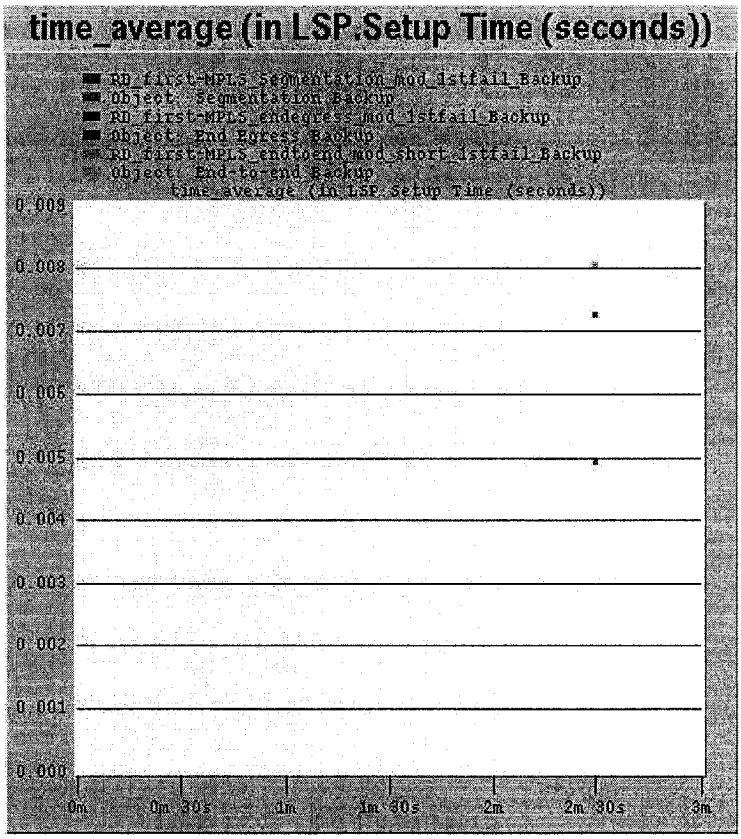

Figure 4.10: Comparison of LSP Setup Times on Segmented Backup, End-egress Backup and End-to-end Backup LSP

4.4.1 LSP Setup Time

Figure 4.10 shows the setup time taken by segmented backup, end-egress backup and end-to-end backup paths. LSP.Setup time is the time that LSP takes to establish itself in the network. In case of segmented backup path, setup time is the time required by the backup path to switch the flow on the backup path after the failure occurs in the network. In this case, segmented backup path takes approximately 0.005 seconds for its setup while end-egress backup takes approximately 0.0072 seconds. For end-to-end backup the setup time is approximately 0.008 seconds. This is explanied in Figure 4.11.

	Segmented Backup	End-egress Backup	End-to-end Backup
LSP Setup Time in seconds	0.005	0.0072	0.008

Figure 4.11: Comparison of LSP Setup Times on Segmented Backup, End-egress Backup and End-to-end Backup LSP

When a LSP fails, LSP teardown NM (RSVP RESV TearDown) is sent upstream to all the nodes for end-egress backup along the LSP path. In case of segmented backup it is sent to all the nodes of that cluster only. Each upstream node checks if it has a bypass tunnel configured.

1. If yes, reroute traffic to bypass
2. If no, send the LSP tear message to the next upstream node.

When the teardown message reaches ingress and there is backup it switches all

the traffic to backup LSP in case of end-to-end backup path.

4.4.2 LSP Delay

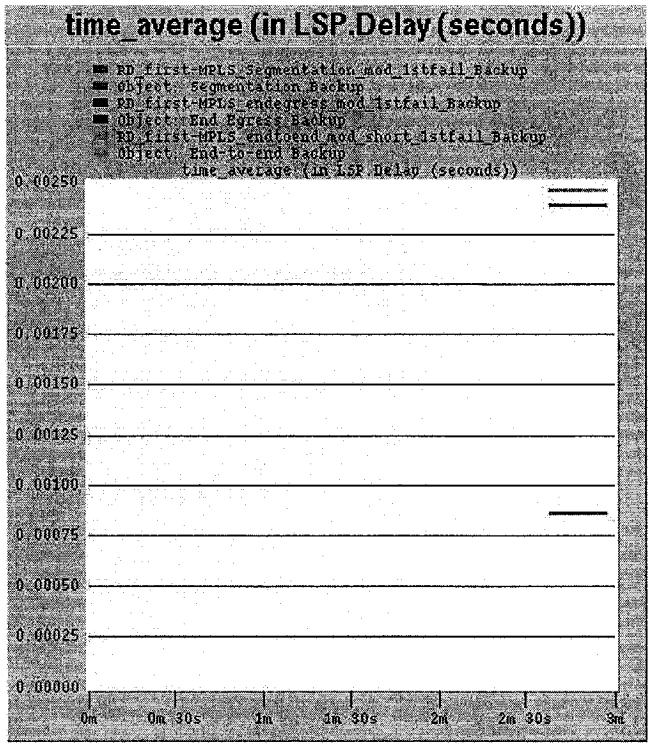

Figure 4.12: LSP Delay on Segmented Backup, End-egress Backup and End-to-end Backup LSP

In Figure 4.12, delay experienced by segmented backup, end-egress backup and end-to-end backup LSP in time_average is shown. LSP.Delay is the delay experienced by packet in the LSP, i.e., time spent by packet within the LSP from ingress node to egress node. Depending upon the size of the network and routing protocol used, traffic will be sent on LSP. LSPs are usually setup around 100s, we wait until around 150s before sending application traffic.

Figure 4.13 shows LSP.Delay on all primary LSP for different scenarios. The delay

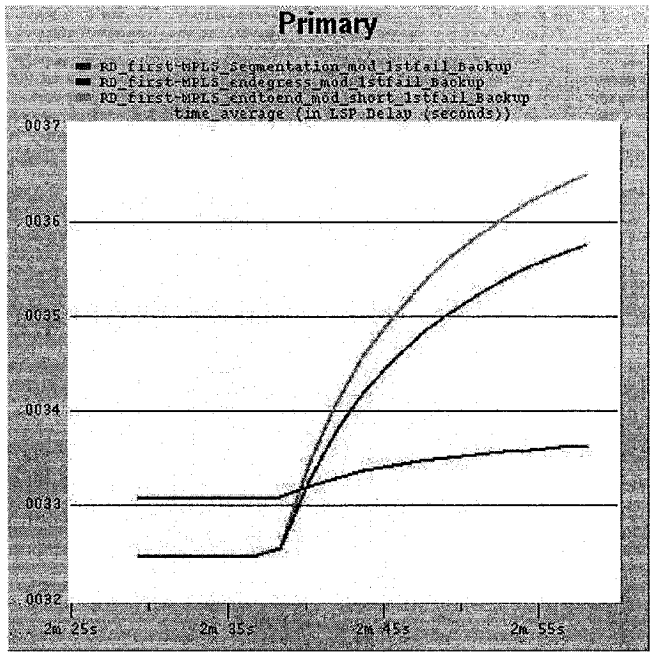

Figure 4.13: LSP Delay on Primary LSPs for different scenarios

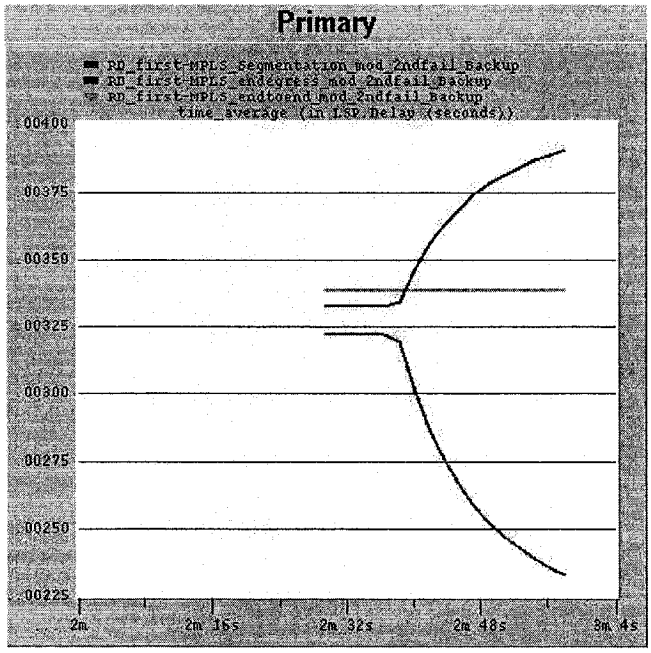

Figure 4.14: LSP Delay on Primary LSPs for different scenarios

on primary LSP includes the delay of the backup path also. Here delay for end-egress backup scenario is smaller than the other two scenarios while delay for segmented backup scenario lies in between other two. Increase or decrease of LSP delay depends on LSP's size, link's size, node's processing rate etc. So in some cases LSP delay for end-egress backup scenario may be higher. This is observed in Figure 4.14. We have failed the link LSR 1 to LSR 7 of the network for this graph result. In this case primary delay for end-egress backup scenario is more than that of segmented backup scenario. Initially it is constant and after some time, it increases while for segmented backup scenario it is constant initially and after some time it decreases. Delay for primary for end-to-end lies in between these two scenarios.

4.4.3 Flow Delay

Figure 4.15 shows the flow delay for primary and segmented backup LSP. Flow.Delay is the delay experienced by packet belonging to a traffic flow in the LSP, i.e., time spent by a packet of a given flow inside the LSP. A flow can be defined as packets going from same source to same destination. These delay statistics include all the delays experienced by a packet from end-to-end, i.e., including transmission and processing delays. Increase or decrease of flow delay depends upon LSP's size, link's size, node's processing rate etc.

Flow.delay on LSP after recovery already includes delay incurred by packet on bypass LSP. Before failure, a packet enters from ingress LER, traverses through primary route and exits from egress LER. Let the total end-to-end delay be T1. After

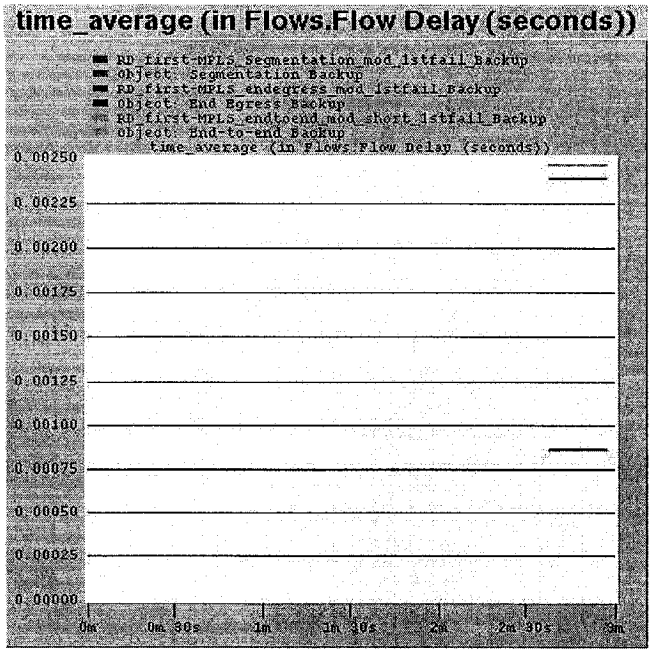

Figure 4.15: Flow Delay on Segmented Backup, End-egress Backup and End-to-end Backup LSP

failure, the packet enters from ingress LER, traverses through another route including backup part and exits from egress LER. Let total end-to-end delay including backup be T2, and delay of the failed segment be T3 and delay of the segmented backup be T4.

T2 = T1 - T3 + T4

This is explained in Figure 4.16.

4.4.4 Traffic In/Out

Figure 4.17 shows traffic In/Out (bits/sec) by segmented backup LSP, end-egress backup LSP and end-to-end backup LSP. In the figure, there are only two traffic flows, one for traffic In (bits/sec) and other for traffic Out (bits/sec). Traffic In

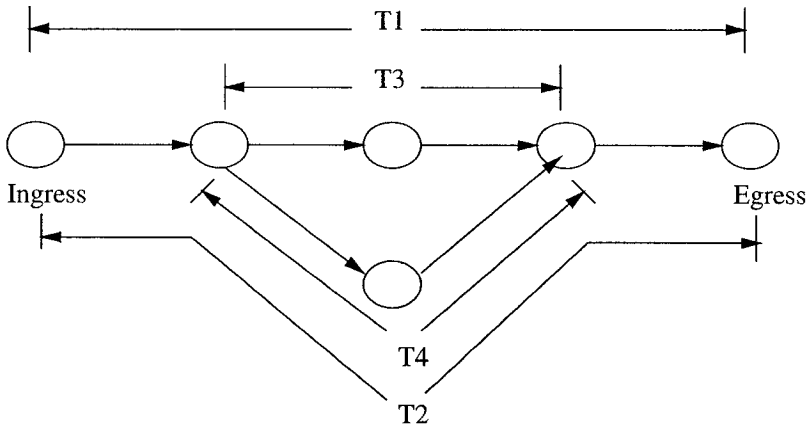

Figure 4.16: Relation between different times

(bits/sec) is the same for all three cases and also Traffic Out for all cases is the same. That is why the graph shows only two traffic flows instead of total six.

We have used segmented backup as a bypass tunnel. The bypass tunnel functions locally, to heal a damaged segment in the primary LSP, by pushing an additional label onto the label stack and switching the frames along the tunnel path. When the frames arrive at the merge point (MP), which is where the two LSPs intersect, the second label is popped, and the MP LSR continues to switch the frames along the primary LSP. So the frames still exit from the primary LSP.

Now, when failed link/node recovers, the local node does not know where to inform about this recovery, as it has already deleted all the LSP related states and information. New mechanisms are required to save the PATH state information for switchback.

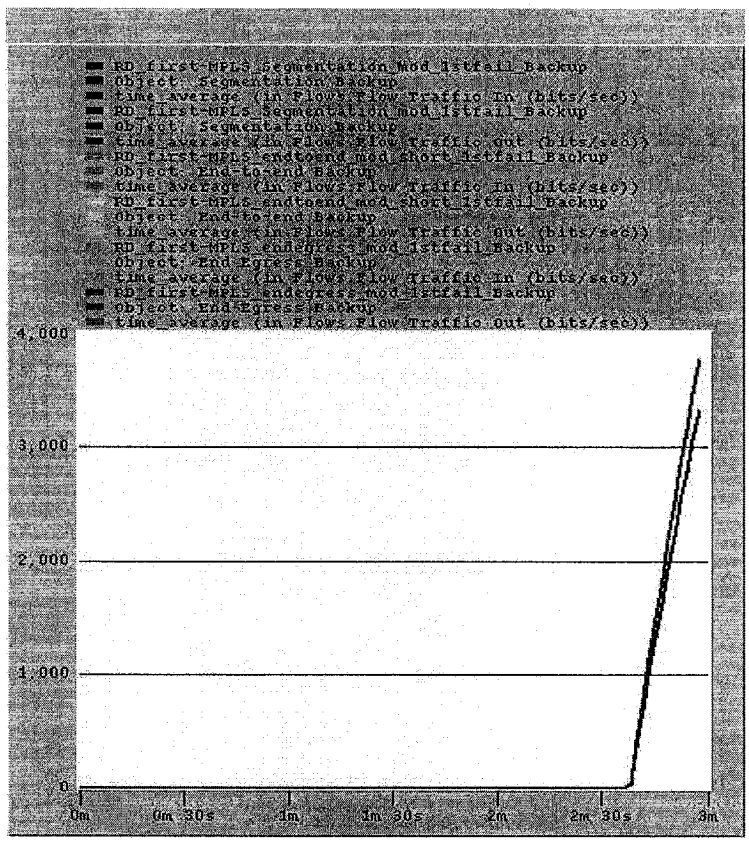

Figure 4.17: Traffic In/Out on Segmented Backup, End-egress Backup and End-to-end Backup LSP

4.5 Summary

In this section we described the modeling of MPLS network in OPNET. We also discussed how MPLS multicast is implemented in OPNET modeler. We also gave details about our experimental setup. We compared in detail simulation results for different recovery approaches with different scenarios.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

The explosive growth of Internet has driven the trend of combining layer 3 routing and layer 2 switching together to improve its performance. In addition to the current data services provided over the Internet, new voice and multimedia services are being developed and deployed. An important delivery mode of the Internet is multicasting, where the information sent by a member of a multicast group is received by all other members of the group. MPLS was introduced mainly for improving packet forwarding and it should also be able to support multicasting in order to meet the requirements of new servies. Because of congestion and failures in the MPLS network, it is necessary to develop a fault tolerant MPLS network for real time services. If a failure occurs, it is crucial to repair the routing tree for multicast communication in a short time. Establishing a backup path to protect a multicast routing tree is a resource consuming

process. Therefore, it is desirable to protect a large number of members of a multicast group with a low number of backup paths. In this thesis, our main approach is to develop a solution for multiple failures in MPLS multicast network.

In this thesis, we presented an algorithm that is able to choose such a backup path to protect a multicast routing tree, and we provided design and simulation of an MPLS-based rerouting mechanism for the protection of multicast routing trees. The backup path is computed after the multicast routing tree establishment and before a link failure occurs, making it suitable for pre-planned rerouting mechanisms. Our approach is based on segmenting the network into clusters after obtaining segmentation points. The backup path is formed by joining these segmentation points, which are also the multicasting points of the same cluster. The segmented backup path and hence cluster formation aims at minimizing the number of receivers of the multicast routing tree that are dropped from the communication if a failure occurs. We showed how a backup path determined by the algorithm could be used to reroute traffic when a path fails.

We have simulated multicast over the MPLS network in the OPNET modeler and evaluated its performance. Our simulation results show significant improvement on the network recovery in terms of LSP setup time, LSP switchover time, and total LSP and flow delay. MPLS multicast Fast and Local Reroute algorithm can repair a multicast routing tree in a few tens of milliseconds, which mostly corresponds to the time to detect the failure.

5.2 Future Work

The mechanism proposed in this thesis deals with failures in the same cluster. Cluster is formed by SPs, which are the multicasting points. Our algorithm stands firm when cluster is formed. But there are chances that in smaller networks one can not form a cluster, or the network itself is one cluster.

Also due to the limitations on OPNET working, we could not create LSP between only two nodes as it treats these LSRs as end nodes whereas the two nodes are directly connected and there is this LSP directly going over this link. This is software problem identified as SPR-53642: "Simulation abort with single hop MPLS static LSP" in OPNET modeler. We had to add an extra node in this segmented backup LSP to work around the problem. We therefore had to include root in the cluster formation. In future release of the OPNET modeler, this problem should be fixed.

When failure is recovered, the local node does not know where to inform about the recovery. This is because it has already deleted all the LSP related states and information. New mechanisms are required to save the PATH state information for switchback.

Finally, we simulated our designs on a small network modeler. We performed our experiments on a multicast LSP and a single flow. An extension to our work includes an implementation in commercial routers and deployment in large scale networks.

Bibliography

[1] E. Rosen, A. Vishwanathan, and R. Callon, "Multiprotocol Label Switching Architecture", IETF RFC 3031, January 2001.

[2] B. Davie, Y. Rekhter, "MPLS Technology and Applications", Morgan Kaufmann Publisher, ISBN: 1-55860-565-4.

[3] F. Le Faucheur, "IETF Multiprotocol Label Switching (MPLS) Architecture", ICATM-98, 1998 1st IEEE International Conference on ATM, pp 6-15.

[4] J Lawrence, "Designing Multiprotocol Label Switching Networks", IEEE Communications Magazine, VOlume: 39, Issue: 7, July 2001, pp 134-142.

[5] International Engineering Consortium, "Multiprotocol Label Switching (MPLS)", http://www.iec.org/online/tutorials/mpls/

[6] Ali Boudani, Bernard Cousin, "A New Approach to Construct Multicast Trees in MPLS Networks", Seventh IEEE Symposium on Computers and Communications (ISCC) , Taormina, Italy, July 2002.

[7] Baijian Yang, Prashant Mohapatra, "Multicasting in MPLS Domains", Computer Communication Journal, Vol. 27, Issue 2, 1 February 2004, pp 145-196.

[8] P. Patrcio, L. Gouveia, A. de Sousa, "MPLS Network Design with Fault-Tolerance and Hop Constraints", Fifth international conference on optimization, OPTI'04, Lisbon, Portugal, July 25-28, 2004.

[9] Adrian Farrel et al, "Fault Tolerance for LDP and CR-LDP", IETF draft-ietf-mpls-ldp-ft-02.txt, November 2002.

[10] Zhou Weihua, Ni Xianle, Ding Wei, "The Study of Service Recovery in MPLS Networks", International Conference on Telecommunications, ICT 2002, Beijing, China, June 23-26, 2002.

[11] Anna Charny, et al, "Distinguish a link from a load failure using RSVP Hellos extensions", IETF draft-vasseur-mpls-linknode-failure-00.txt, October 2002.

[12] Thomas M. Chen and Tae H. Oh, "Reliable Services in MPLS", IEEE Communication Magazine, Dec 1999, Vol. 37, No. 12, pp 58-62.

[13] Surviving Failures in MPLS Networks, White paper from Data Connections www.dataconnection.com

[14] Vishal Sharma, et al, "Framework for MPLS-based Recovery", IETF draft-ietf-mpls-recovery-frmwrk-08.txt, October 2002.

[15] A. Iwata, N. Fujita, T. Nishida, "MPLS Signaling Extensions for Shared Fast Rerouting", IETF draft-iwata-mpls-shared-fastreroute-00.txt, August 2001.

[16] Ping Pan, George Swallow, Alia Atlas, "Fast Reroute Extensions to RSVP-TE for LSP Tunnels", IETF draft-ietf-mpls-rsvp-lsp-fastreroute-07.txt, September 2004.

[17] Dimitry Haskin, et al, "A Method for Setting an alternative Label Switched Paths to Handle Fast Reroute", draft-haskin-mpls-fast-reroute-06.txt, May 2001.

[18] Vaibhav Shandilya, "Fault Tolerant LSP establishment in an MPLS network", IETF draft-shandilya-fault-tolerant-lsp-01.txt, April 2002.

[19] Song Dong and Chris Phillips, "New Service Restoration Scheme in MPLS Networks", International Conference on Telecommunications, Beiging, China, June 2002, C-052.

[20] Ken Owens, et al, "A Path Protection/Restoration Mechanism for MPLS Networks", IETF draft-chang-mpls-path-protection-04.txt, April 2002.

[21] Gustavo B. Figueiredo et al, "Dynamic Sizing of Label Switching Paths in MPLS Networks", IEEE International Telecommunications Symposium, ITS 2002, Brazil, September 8-12, 2002.

[22] Takayoshi Takehara, et al, "Dynamic Logical Path Configuration Method Considering Reliability in MPLS Network", 26th Annual IEEE Conference on Local Computer Networks, LCN 2001, Tampa, Florida, USA, November 14-16, 2001, pp 250-257.

[23] Anjali Agarwal, Rohan Deshmukh "Ingress Failure Recovery Mechanisms in MPLS Network", IEEE MILCOM'2002, Anaheim, CA, USA, October 7-10, 2002.

[24] Mikko Paakkonen, Kimmo Kaario, Timo Hamalainen, "CoS Aware Traffic Flow Balancing in MPLS Networks", Proceedings of 9th IEEE International Conference on Telecommunications, ICT 2002, June 2002, Beijing, China.

[25] Rameshbabu Prabhagaran, Joseph B. Evans, "Experiences with Class of Service (CoS) Translations in IP/MPLS Networks", 26th Annual IEEE Conference on Local Computer Networks, LCN 2001, Tampa, Florida, USA, November 14-16, 2001.

[26] Eric Horlait, Nicolas Rouhana, "Differentiated Services and Integrated Services Use on MPLS", Fifth IEEE Symposium on Computers and Communications, ISCC 2000, Antibes, France, July 4-6, 2000.

[27] Vaselin Rakocevic, John M Griffiths, Graham Cope, "Dynamic Partitioning of Link Bandwidth in IP/MPLS Networks", IEEE International Conference on Communications, ICC 2001, Helsinki, Finland, June 2001.

[28] Krishna Phani Gummadi, Madhavpura Jnana Pradeep and C. Siva Ram Murthi, "An Efficient Primary-Segmented Backup Scheme for Dependable Real-Time Communication in Multihop Networks", IEEE/ACM Transactions on Networking, Vol. 11, No. 1, February 2003.

[29] D. Ooms, et al, "Overview of IP Multicast in a MPLS Environment", RFC 3353, August 2002.

[30] A. Agarwal, K.B. Wang, "Supporting Quality of Service in IP Multicast Networks", Journal of Computer Communications 26, 2003, 1533-1540.

[31] Yvan Pointurier, *Link Failure Recovery for MPLS Networks with Multicasting*, Master's Thesis, University of Virginia, August 2002.

[32] OPNET Modeler 10.0 – Educational Version, OPNET Technologies, Inc, 7255 Woodmont Ave., Bethesda, MD 20814, USA.

[33] R. Deshmukh, A. Agarwal, "Failure Recovery in MPLS Multicast Network using Segmented Backup Approach", ICN'2004, Gosier, Guadeloupe, French Caribbean, March 1-4, 2004.

[34] Feng Huining, Chen Qimei, "A Tree View of the MPLS FEC Strategy", International Conference on Telecommunications, ICT 2002, Beijing, China, June 23-26, 2002.

[35] D. Estrin et al, "Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification", RFC 2362, June 1998.

[36] Kalman Pusztai, Ramona Marfievici, "Traffic Engineered Multicast in MPLS Domains", Second Edition RoEduNet International Conference, Iasi-Romania, June 5-6, 2003.