On Fault-Tolerance and Security in MPLS Networks

Sahel Ahmad Alouneh

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosophy at

Concordia University

Montreal, Quebec, Canada

August 2008

Your file Votre référence
ISBN: 978-0-494-45648-4
Our file Notre référence
ISBN: 978-0-494-45648-4

Abstract

Multi-Protocol Label Switching (MPLS) is an evolving network technology that is used to provide Traffic Engineering (TE) and high speed networking. Internet service providers, which support MPLS technology, are increasingly required to provide high Quality of Service (QoS) guarantees and security.

One of the aspects of QoS is fault tolerance. It is defined as the property of a system to continue operating in the event of failure of some of its parts. Fault tolerance techniques are very useful to maintain the survivability of the network by recovering from failure within acceptable delay and minimum packet-loss while efficiently utilizing network resources. On the other hand, with the increasing deployment of MPLS networks, security concerns have been raised. The basic architecture of MPLS networks does not support security aspects such as data confidentiality, data integrity, and availability. MPLS technology has emerged mainly to provide high speed packet delivery. As a result security considerations have not been discussed thoroughly until recent demands for security have emerged by most providers and researchers.

In this thesis, we propose a new method that has a two-fold objective: to provide fault tolerance and to enhance the security in MPLS networks. Our approach uses a modified *(k, n) threshold sharing scheme* (TSS) combined with multi-path routing. An IP packet entering MPLS network is partitioned into n MPLS packets, which are each assigned to disjoint or maximally disjoint Label Switched Path (LSP) across the MPLS network. Receiving MPLS packets from k out of n LSPs are sufficient to reconstruct the original IP packet.

From the security point of view, the modified TSS provides data confidentiality, integrity, availability and IP spoofing. In addition, fault tolerance in MPLS is supported using reasonable resources. The recovery from node/link failure and/or transmission errors is provided with no delay or packet loss. Packet re-ordering may not be required if packets are lost due to failure. However, sequencing is considered in our approach to identify packets with transmission errors. In order to provide fault tolerance, our scheme requires $n > k$. However, for security purposes, if the target is only to provide data confidentiality, then only a modified (k, k) TSS algorithm is sufficient and consequently no significant redundant bandwidth is required.

To verify that our approach does not require long processing time, we conducted simulations that show the modified TSS processing time does not significantly affect the packet transmission time. RSVP-TE is the MPLS signaling protocol used to establish LSPs. Extensions required to support multi-path routing in RSVP-TE are also studied.

The impact of multi-path routing and modified TSS on MPLS security and fault tolerance is investigated and compared with single routing. The connection intrusion probability and connection failure probability have shown lower values when multi-path routing is used. The application of IPSec security protocol in MPLS networks is also investigated.

Finally, we applied the modified threshold sharing scheme on MPLS multicast networks, where both the source specific tree approach and the group shared tree approach are considered.

To My Late Father

ACKNOWLEDGMENT

I would like to acknowledge and extend my heartfelt gratitude to the following persons who have made the completion of this thesis possible:

My supervisors, Dr. Abdeslam En-Nouaary and Dr. Anjali Agarwal for their support and guidance. They were extremely helpful and understanding during the course of this research work. I have benefited considerably by working under their supervision.

I gratefully acknowledge the financial support of Concordia Univeristy.

I would like to thank the defense committee for offering time to review my thesis.

I would like to thank my mother, brothers and sister for their deep love, encouragement, and consistent support they have given me through my life.

My wife Suhad, for her love and support in every step of my life. To my new born son Rakan, may God bless him.

I would also like to thank my friends in Montreal for their encouragement.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AH	Authentication Header
ATM	Asynchronous Transfer Mode
BGP	Border Gateway Protocol
CA	Certificate Authority
CE	Customer Edge
CR-LDP	Constraint based Label Distribution Protocol
CW	Control Word
DES	Data Encryption Standard
DoS	Denial of Services
ECMP	Equal-cost Multi-path
ESP	Encapsulating Security Payload
FEC	Forward Equivalence Class
FR	Frame Relay
FTN	FEC-to-NHLFE
GF	Galois field
GRE	Generic Routing Encapsulation
IETF	Internet Engineering Task Force
ILM	Incoming Label Map
IP	Internet Protocol
IPSec	IP Security
IS-IS	Intermediate System-to-Intermediate System
ISP	Internet Service Providers

ITU	International Telecommunication Union
IV	Initialization Vector
LDP	Label Distribution Protocol
LER	Label Edge Router
LIB	Label Information Base
LSP	Label Switch Path
LSR	Label Switch Router
MD5	Message-Digest 5
MP-LSP	Multi-Path LSP
MPLS	Multiprotocol Label Switching
MOSFP	Multicast Open Shortest path First
MTU	Maximum Transmission Unit
NHLFE	Next Hop Label Forwarding Entry
NSF	National Science Foundation
OPSF	Open Shortest Path
PE	Provider Edge
PIM	Protocol Independent Multicast
PIM-DM	PIM Dense Mode
PIM–SM	PIM Sparse Mode
PML	Path Merge LSR
PSL	Path Switch LSR
P2MP	Point-to-Multipoint
QoS	Quality of service
RP	Rendezvous Point
RRO	Record Route Object

RSVP	Resource Reservation Protocol
RSVP-TE	Resource Reservation Protocol – Traffic Engineering
SCA	Spare Capacity Allocation
SHA	Secure Hash Algorithm
SLA	Service Level Agreement
SLRG	Shared Link Risk Groups
SPM	Spare Provision Matrix
SSL	Secure Socket Layer
SSR	Successive Survivable Routing
TE	Traffic Engineering
TSS	Threshold Sharing Scheme
VoIP	Voice over Internet Protocol
VPN	Virtual Private Network

Chapter 1

Introduction

1.1 Thesis Motivation and Objective

Multi-Protocol Label Switching (MPLS) is a data-carrying mechanism that belongs to the family of packet-switched networks. A number of different technologies were previously deployed with essentially identical goals, such as frame relay and ATM. MPLS technologies have evolved with the strengths and weaknesses of ATM in mind. Many network engineers agree that ATM should be replaced with a protocol that requires less overhead, while providing connection-oriented services for variable-length frames. MPLS is currently replacing some of these technologies in the marketplace. MPLS technology is used to provide Traffic Engineering (TE) and high speed networking. MPLS technology gives network operators a great deal of flexibility to divert and route traffic around link failures, congestion, and bottlenecks [1, 2].

Rapid growth and increasing deployment of MPLS technology have made it an essential consideration in the design and operation of large public Internet backbone networks. Therefore, there has been current demand on Internet Service Providers, which support MPLS technology, to provide Quality of Service (QoS) guarantees and security.

Fault tolerance is an important QoS factor that needs to be considered to maintain network survivability. It is the property of a system that continues to operate the network properly in the event of failure of some of its parts. The recovery from node/link failure

in MPLS networks can be accompanied by many problems and limitations. The main issues of failure recovery in MPLS networks are recovery time, packet loss (i.e., due to failure of node(s)/link(s), re-ordering of packets that can be received out of order at the destination node), and reducing the size of redundant bandwidth that should be used to provide fault tolerance. There are several research approaches that consider these issues in MPLS networks based on two different recovery techniques (i.e., the dynamic protection and the fast protection techniques). In the dynamic protection, there is no bandwidth reservation required to provide fault tolerance. In other words, in dynamic protection, the process to find an alternative path (also called backup path) starts after a node/link failure occurs. Because of that, the recovery time and packet loss ratio (due to node/link failure) can be high. Therefore, for real time applications which do not tolerate long recovery time and large packet loss ratio, this technique may not be suitable. On the other hand, the fast protection technique can provide lower recovery time and packet loss ratio because the backup path is established prior to the occurrence of a node/link failure. However, this technique requires bandwidth reservation for the backup path. In this thesis, our goal is to come up with a new solution for fault tolerance in MPLS networks that takes into consideration the previous fault tolerance issues mentioned above.

Another important issue of MPLS networks is its security. With increasing deployment of MPLS networks, the security of traffic traversing through it has become a crucial concern. In MPLS networks, the forwarding of IP packets is based on labels instead of IP routing lookup (more details on how IP packets are forwarded in MPLS networks are shown in the next chapter). The domain routers in MPLS networks (excluding the edge routers) are not supposed to analyze the content of the IP packet

header and instead they only use the content of the MPLS header. Therefore, to provide security in MPLS networks, any security approach should take into account the characteristic of MPLS networks in order not to affect the performance of packets' forwarding speed. Our goal in this thesis is to provide mechanisms to enhance the security of MPLS networks. The security parameters to be covered in this thesis are: the confidentiality and integrity of data, availability and IP spoofing. The detailed discussion of these security parameters will be covered in the next chapter.

1.2 Contribution of the Thesis

This thesis introduces solutions for two main issues in MPLS networks, which are fault tolerance and security.

We propose a mechanism to enhance the security (confidentiality, integrity, availability) in MPLS networks by using multi-path routing combined with a modified (k, n) Threshold Secret Sharing (TSS) scheme. An IP packet entering MPLS ingress router can be partitioned into n shadow (or share) packets, which are then assigned to disjoint or maximally disjoint paths across the MPLS network. The egress router at the end will be able to reconstruct the original IP packet if it receives any k share packets. The attacker must therefore tap at least k paths to be able to reconstruct the original IP packet that is being transmitted, while receiving k-1 or less of share packets makes it tough or even impossible to reconstruct the original IP packet.

On the other hand, there is no solution until now that can provide a link/node failure recovery with no packet loss and recovery delay. In addition, network resources such as bandwidth have to be significantly utilized. Our proposed mechanism for security

can be easily applied to support fault tolerance in MPLS networks to handle single or multiple path failures. It uses the same idea of the modified (k, n) Threshold Sharing Scheme mentioned above with multi-path routing, wherein k out of n LSPs are required to reconstruct the original message. Our approach guarantees to continue the network operation with no packet loss and recovery delay if there are enough number of LSPs (i.e., k out of n disjoint or maximally LSPs are available at the destination side), and with reasonable network resource utilization.

It is important to distinguish whether the paths in a multi-path connection are node-disjoint or link-disjoint. The paths between a source and a destination are said to be node disjoint if there are no shared node(s) and link(s) between any of these paths. The paths between a source and a destination are said to be link-disjoint if there are no shared links between any of these paths, however there may be shared node(s) between them. In this thesis, we will refer to both cases of node or link disjoint paths by the term "disjoint LSPS or paths", however, if we want to identify either one of them (i.e., node-disjoint or link disjoint) then this will be specified. When the paths are said to be link-disjoint and an assumption is made that shared node(s) do not fail or are attacked, then both of the cases of link and node disjoint paths have the same significance, and therefore we call both cases as disjoint paths. On the other hand, when the LSPs in a multi-path connection have link(s) in common, then we refer to this case by the term "maximally disjoint". More discussion on the effect of having a maximally disjoint multi-path connection on MPLS fault tolerance and security is covered later in the thesis.

It is also important to distinguish that multiple paths between a source and a destination can appear to be disjoint at the LSP level while at the physical level they may

not be. Therefore, we assume the LSPs are disjoint at the physical level, i.e., they belong to different shared risk link groups (SLRG) in order for the modified (k, n) TSS to work properly in real and practical network applications.

MPLS networks were initially designed to serve unicast traffic, but as point to multipoint applications, such as multicast applications have emerged, extensions to MPLS configuration and signaling protocols are needed. Therefore in this thesis we cover the application of the modified (k, n) TSS scheme for the MPLS multicast traffic.

Because the threshold sharing scheme is used in combination with multi-path routing, a performance evaluation is conducted to show the impact of using the multi-path routing on MPLS security and fault tolerance and compare it with the unicast routing.

1.3 Organization of the Thesis

The organization of the thesis is as follows:

In Chapter 2, we introduce a background overview on MPLS networks. After that, a background on MPLS failure recovery methods is discussed. The main recovery factors in MPLS fault tolerance are discussed. Related work on MPLS fault tolerance are presented and discussed. The second part of this chapter presents a background on MPLS security and the main security issues that will be addressed in this thesis, and followed by related work on MPLS security.

Chapter 3 is dedicated to introduce the original *threshold sharing scheme* that is adopted and modified by our approach to provide MPLS fault tolerance and security. Some of the main applications of the threshold sharing scheme are presented.

Our approach for MPLS security and fault tolerance is discussed in Chapter 4. In this chapter, we present our method and modify the threshold sharing scheme to provide fault tolerance and security in MPLS networks. The distribution and reconstruction processes are explained with theoretical examples. Moreover, measurement results for the processing time of threshold sharing scheme and its impact on the total IP packet transmission time are discussed. The impact of using our approach on network bandwidth utilization is also explained. It is worth to note that the application of our approach can be accompanied by some related issues such as the need for packets sequencing and Quality of Service issues. Therefore, these issues are also discussed.

Chapter 5 discusses the deployment of the threshold sharing scheme on MPLS fault tolerance and security. Simulation results for the deployment of the TSS approach on MPLS fault tolerance have been conducted. On the other hand, the deployment of the TSS approach on MPLS security has also been shown. In this chapter we also discuss the case when paths in a multi-path connection are not fully node\link disjoint. In this case, the paths are considered to be maximally disjoint, which means that the paths may share some link(s) between them. In this chapter, we also studied the impact of using the threshold sharing scheme combined with multi-path routing on fault tolerance and security and compare it with single path routing. Finally, a comparison between the threshold sharing scheme with other recovery mechanisms with respect to fault tolerance and with IPSec with respect to security has also been discussed.

In Chapter 6 we studied the feasibility of applying our approach for multicast traffic in MPLS networks. The discussion was divided into two parts, one part for source specific trees and another part for shared group trees. The discussion takes into

consideration the fault tolerance and security issues within the scope of MPLS multicast applications. The discussion in this chapter highlights the limitations of the application of our approach in real and practical multicast networks. Finally, in Chapter 7 we conclude the thesis and present our future plan.

Chapter 2

Literature Review

This chapter introduces MPLS technology and its main components. A background on MPLS failure recovery techniques is presented followed by related work on failure recovery approaches used to support fault tolerance in MPLS networks. Besides, a background on MPLS security and the main security issues are also presented along with related work on the MPLS security approaches.

2.1 Overview of MPLS Networks

One challenge in current network research is how to effectively transport IP traffic over any network layer technology (ATM, FR, and Ethernet). Therefore, MPLS technology has been introduced as an effective solution for traffic engineering and increasing the speed of packet forwarding. MPLS technology belongs to the family of packet-switched networks. MPLS operates at an OSI Model layer that is generally considered to lie between traditional definitions of Layer 2 (data link layer) and Layer 3 (network layer), and thus is often referred to as a "Layer 2.5" protocol. It can be used to carry many different kinds of traffic, including IP packets, as well as native ATM, SONET, and Ethernet frames.

The basis of MPLS operation is the classification and identification of IP packets at the ingress node with a short, fixed-length, and locally significant identifier called a label, and forwarding the packets to a switch or router that is modified to operate with such labels. The modified routers and switches use only these labels to switch or forward the packets through the network and do not use the network layer addresses [1, 2].

The basic components of MPLS are categorized as described below.

Control and Forwarding Planes

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

The original hop-by-hop forwarding architecture has remained unchanged since the invention of Internet architecture; the different forwarding architectures used by connection-oriented link layer technologies does not offer the possibility of a true end-to-end change in the overall forwarding architecture. For this reason, the most important change that MPLS makes to the Internet architecture is the forwarding architecture. It should be noted that MPLS is not a routing protocol but is a fast forwarding mechanism that is designed to work with existing Internet routing protocols, such as Open Shortest Path First (OSPF), Intermediate System-to-Intermediate System (IS-IS), or the Border Gateway Protocol (BGP). The control plane uses the previous routing protocols above to exchange information with other routers to build and maintain a forwarding table.

The control plane consists of network layer routing protocols to distribute routing information between routers, and label binding procedures for converting this routing information into the forwarding table needed for label switching. Some of the functions accomplished by the control plane are to disseminate decision making information, establish paths and maintain established paths through the MPLS network. The component parts of the control plane and forwarding plane [1] are illustrated in Figure 2.1.

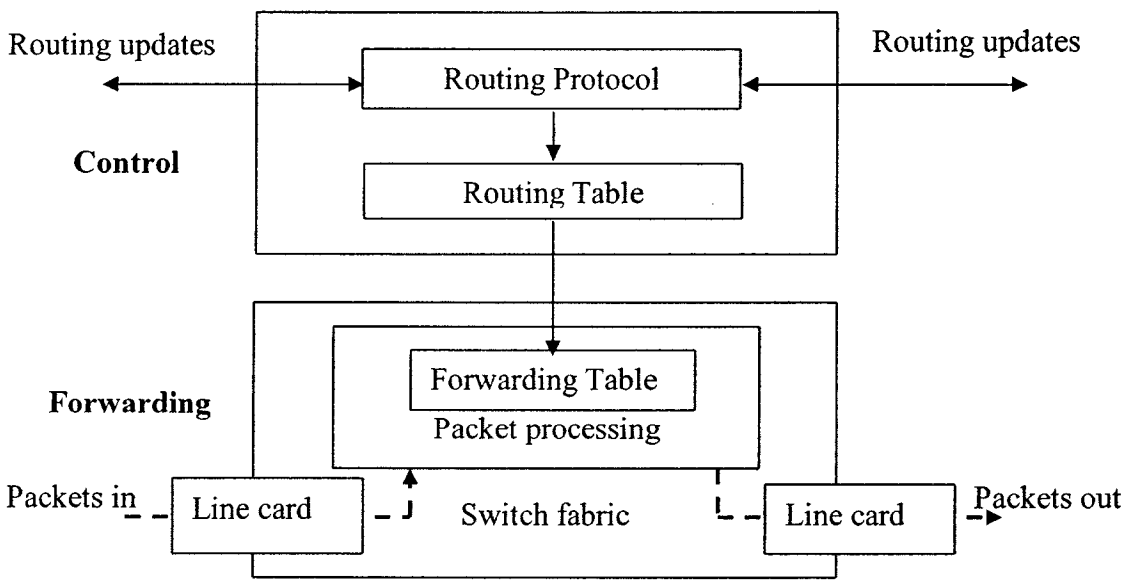


Figure 2.1 Control and forwarding planes [1]

When a packet arrives, the forwarding plane (based on a label swapping forwarding algorithm) searches the forwarding table maintained by the control plane to make a routing decision for each packet.

The data plane (forwarding plane) is responsible for relaying data packets between routers (LSRs) using label swapping. In other words, a tunnel is created below

the IP layer carrying client data. The concept of a tunnel (LSP tunnel) is a key because it means the forwarding process is not IP based but label based. Moreover, classification at the ingress, or entry point to the MPLS network, is not based solely on the IP header information, but applies flexible criteria to classify the incoming packets such as class of service required.

Forwarding Equivalent Class (FEC)

Forwarding Equivalent Class (FEC) is a set of packets that are treated identically by an LSR. Thus, a FEC is a group of IP packets that are forwarded over the same LSP and treated in the same manner and can be mapped to a single label by an LSR even if the packets differ in their network layer header information. Figure 2.2 shows this behavior. The label minimizes essential information about the packet. This might include destination, precedence, QoS information, and even the entire route for the packet as chosen by the ingress LSR based on administration policy. A key result of this arrangement is that forwarding decisions based on some or all of these different sources of information can be achieved by means of a single table lookup from a fixed-length label [2].

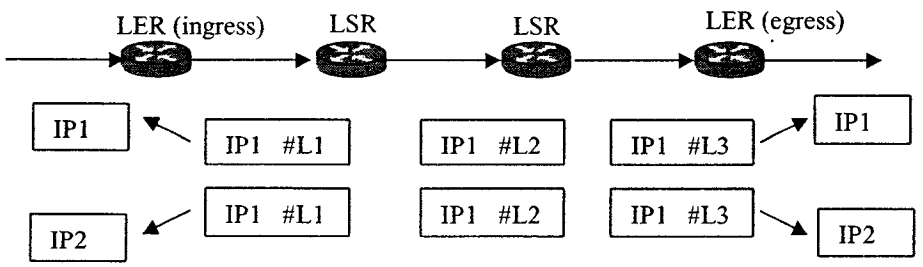
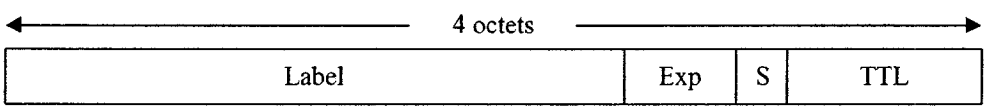


Figure 2.2 Forward Equivalent Class (FEC) [2]

This flexibility is one of the key elements that make MPLS so useful. Moreover, assigning a single label to different flows with the same FEC has advantages derived from "flow aggregation". For example, a set of distinct address prefixes (FECs) might all have the same egress node, and label swapping might be used only to get the traffic to the egress node. In this case, within the MPLS domain, the union of those FECs is itself a FEC. Flow aggregation reduces the number of labels which are needed to handle a particular set of packets, and also reduces the amount of label distribution control traffic needed. This improves scalability and reduces the need for CPU resources [2].

Labels

A label called a "shim label", or an MPLS "shim" header is a short, fixed-length, locally significant FEC identifier. Although the information on the network layer header is consulted for label assignment, the label does not directly encode any information from the network layer header like source or destination address. The labels are locally significant only, meaning that the labels are only useful and relevant on a single link, between adjacent LSRs. Figure 2.3 presents the fields of an MPLS "shim" header.



Label: Label value, 20 bits
Exp: Experimental bits, 3 bits. Can be used to support CoS (Class of Service).
S: Stack, 1 bit
TTL: Time to Live, 8 bits

Figure 2.3 MPLS "shim" header format

In MPLS the assignment of a particular packet to a particular flow is done just once, as the packet enters the network. The flow Forward Equivalence Class (FEC) which the packet is assigned to is encoded with a label. When the packet is forwarded to the next hop, this label is sent along with it, that is, the packets are "labeled". At subsequent hops there is no further analysis of the packet's network layer header. The label itself is used as hop index. This assignment eliminates the need to perform the longest prefix-match computation for each packet at each hop. The experimental bits in the MPLS header can be used to support the required class of service for packets as described in [91].

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

Label Switch Router (LSR)

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

the label on top of the IP packet and forwards the packet to the next hop. In this phase as the incoming packet is not labeled, the FEC-to-NHLFE (FTN) map module is used. Each intermediate/transit LSR examines only the label in the received MPLS packet, replaces it with the outgoing label present in the label information based forwarding table (LIB) and forwards the packet through the specified port. This phase uses the incoming label map (ILM) and next hop label forwarding entry (NHLFE) modules in the MPLS architecture. When the packet reaches the egress LSR, the label is popped and the packet is delivered using the traditional network layer routing module.

Label Switched Path (LSP)

A label Switched Path (LSP) is an ingress-to-egress switched path built by MPLS capable nodes which an IP packet follows through the network and which is defined by the label (Figure 2.4). The labels may also be stacked, allowing a tunneling and nesting of LSPs. An LSP is similar to ATM and FR circuit switched paths, except that it is not dependent on a particular Layer 2 technology.

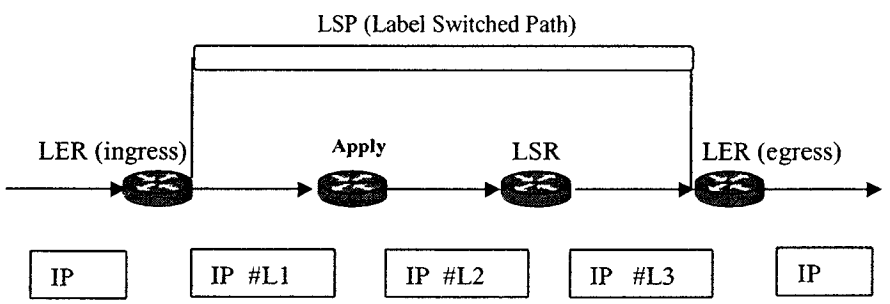


Figure 2.4 Label Switched Path (LSP) [2]

Label switching relies on the set up of switched paths through the network. The path that data follows through a network is defined by the transition of the label values

using a label swapping procedure at each LSR along the LSP. Establishing an LSP involves configuring each intermediate LSR to map a particular input label and interface to the corresponding output label and interface. This mapping is stored in the Label Information Base forwarding table (LIB).

There are two kinds of LSPs setup methods in MPLS networks. First, hop-by-hop routed LSPs where the label distribution protocol (LDP) is used. To illustrate more, in MPLS two adjacent label switching routers (LSRs) must agree on the meaning of labels used to forward traffic between them and through them. The label distribution protocol (LDP) is a protocol defined by IETF MPLS Working Group for distributing labels in MPLS networks. Therefore, LDP is a set of procedures and messages by which LSRs establish Label Switched Paths (LSPs) through a network by mapping network layer routing information directly to data link layer switched paths.

Figure 2.5 shows a simple process for requesting and assigning labels in a MPLS network. For example, an IP packet with IP prefix value 47.1 entering the MPLS network is assigned labels L_7 and L_4 to setup an LSP used to forward the packet from the ingress router towards the egress router. Therefore, each LSR that receives a MPLS packet with *Label IN* value, after checking its FEC value, forwards the packet to the next router with *Label Out* value.

The second kind for LSPs setup is explicitly routed if the path should take into account certain constraints like available bandwidth, QoS guarantees, and administrative polices. Explicit routing uses an advanced version of the LDP signaling protocol for this purpose because the basic LDP signaling protocol is incapable to do explicit routing. The

advanced version of the LDP is the constraint-routed label distribution protocol (CR-LDP) [3].

Resource Reservation Protocol with traffic engineering extensions (RSVP-TE) [4] is another signaling protocol used to establish LSPs. For example, in Figure 2.5 the RSVP *path* and *reservation* messages are used for requesting and mapping the labels.

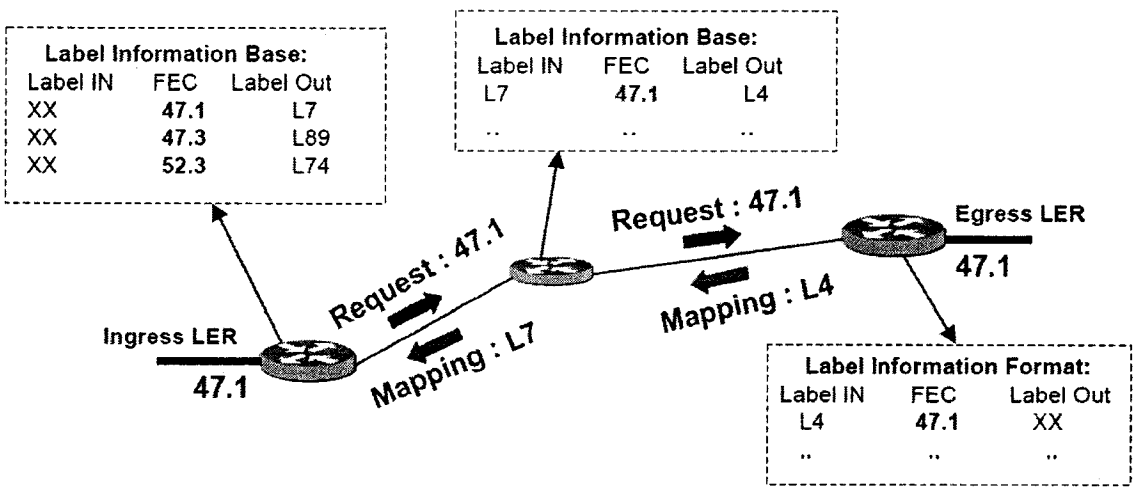


Figure 2.5 MPLS single path label distribution

The following section introduces the fault tolerance in MPLS and discusses the main techniques used for failure recovery. In addition, related work in MPLS fault tolerance is presented and discussed.

2.2 Failure Recovery in MPLS

The explosive growth of Internet and the real-time applications it supports, such as voice and video, has resulted in new requirements for service providers. The customers

expect the highest quality of service, including sustained continuity of service without any disruption, during the time they pay for the service. Service disruption due to a network failure may cause the customers significant loss of revenue during the network down time, which may lead to bad publicity for the service provider [5]. The major network failures are essentially of three types:

a. Node Failure due to equipment breakdown or equipment damage resulting from an event such as an accidental fire, flood, or earthquake; as a result, all or some of the communication links terminating on the affected node may fail.
b. Link failure due to inadvertent wire or fiber cables cuts; the cables carrying traffic from one telecommunication office to another is buried approximately 3 feet underground in a conduit, but due to ubiquitous construction activities, accidental cable cuts occur frequently, despite increased network care and maintenance efforts.
c. Software failure that can impact a large portion of the given network. If the software application in a router for example is corrupted or is not functioning correctly, then this will affect other network components that are connected to it [5]. This type of failure is not considered in this thesis.

In order to avoid these problems, it requires routers and other network system elements to be resilient towards node or link failures in the network. Therefore, a system that is able to continue operating properly in the event of failure of some of its parts is called a fault tolerant system.

The recovery time that can be tolerated for a highly reliable service is in the order of seconds down to 10s of milliseconds [6, 7]. The current routing algorithms used in IP networks are robust and survivable. However, the amount of time that is taken to recover a failure can be significant and unaccepted for time-critical communication such as real-time applications. MPLS network is very vulnerable to failures because of its connection oriented architecture. Path restoration is provided mainly by rerouting the traffic around a node/link failure in a LSP, which introduces considerable recovery delays and may incur packet loss.

2.2.1 Recovery Techniques in MPLS

Recovery schemes can be classified according to the following two criteria: dynamic protection vs. fast protection, local repair vs. path protection. Figure 2.6 provides a general classification of protection techniques for MPLS networks. The path supporting the transmission in normal conditions is called the primary path or working path. Backup paths or restoration paths are secondary paths to be used in case the primary path fails.

2.2.1.1 Recovery by Rerouting (Dynamic protection)

Recovery by rerouting or dynamic restoration is defined as establishing new paths or path segments on demand for restoring traffic after the occurrence of a fault. The new paths may be based upon fault information, network routing policies, pre-defined configurations and network topology information. Thus, upon detecting a fault, paths or path segments to bypass the fault are established using signaling. For this purpose, an

alternative or backup path apart from the primary path used by the current traffic is needed. It is important for the primary and backup paths to be disjoint or if not possible to be maximally disjoint.

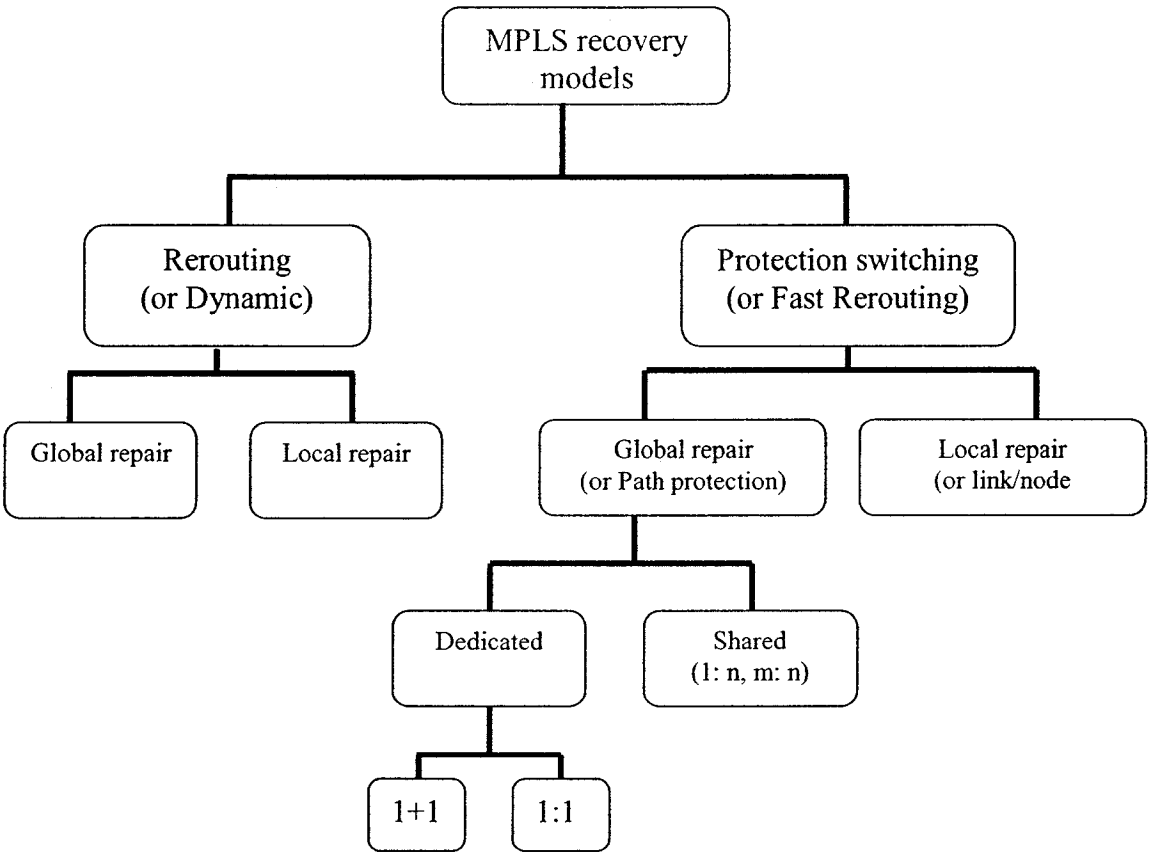


Figure 2.6 A General classification of protection techniques for MPLS networks [9, 10]

A complete rerouting technique is described in the framework presented in [7] and consists of several steps. The main steps that the rerouting method must accomplish are fault detection, fault notification, alternative path computing, and rerouting of traffic from the primary path to the alternative path.

2.2.1.2 Fast Protection

Fast protection recovery mechanisms pre-establish a recovery path or path segment, based on network routing policies, the restoration requirements of the traffic on the working path, and administrative considerations. The recovery path may or may not be disjoint with the working path. However, if the recovery path shares sources of failure with the working path, the overall reliability of the construct is degraded. When a fault is detected, the protected traffic is switched over to the recovery path(s) and restored. In fast restoration model the backup LSP is established and configured in advance, therefore bandwidth reservation has to be made.

The two common protection switching recovery mechanisms are local repair (or link/node protection) and global or end to end repair (path protection) according to the initialization locations of the rerouting process [11]. Indeed, both techniques local or global repair can be applied to dynamic restoration or fast rerouting, however, local repair is more common in dynamic restoration protection.

The global and local repairs in the scope of the fast rerouting technique can be explained as follows:

- **Global repair**: In global repair, the protection is activated on end-to-end basis, irrespective of where the failure occurs, as shown in Figure 2.7. That is, an alternative or a backup LSP is pre-established from ingress to egress routers of the path to be protected. It is noticed that in global repair a failure signal is propagated to the source (ingress router) before switching the traffic to the backup path, which

wastes valuable time because the failure notification has to traverse the entire network (MPLS domain).

- **Local repair**: Local repair aims to fix the failure at the point of failure or within a very short distance from the failure, thereby minimizing delay and total packet loss. If local repair is attempted to protect an entire LSP, each intermediate LSR must have the capability to initiate alternative, pre-established LSPs. This is because it is impossible to predict where failure may occur within an LSP. A very high cost has to be paid in terms of complex computations and extensive signaling is required to establish alternative LSPs from each intermediate LSR to the egress LSR.

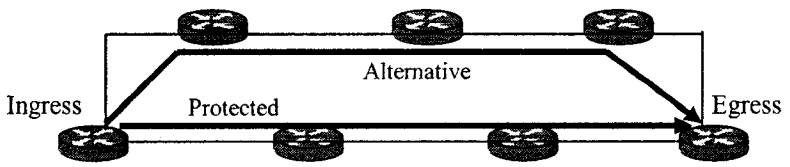


Figure 2.7 Global repair [2]

2.2.2 Recovery Factors in MPLS

Several criteria comparisons between different MPLS–based recovery schemes are defined in [12, 13]. The most important criteria are: packet loss, recovery time, re-ordering, full restoration time, vulnerability, and quality of protection. In the following we summarize the factors of our interest:

Packet Loss: Recovery schemes may introduce packet loss during switchover to/from a recovery or backup LSP. It is a critical parameter for a restoration mechanism. Throughput rates achieved for the service are seriously affected by packet losses. In real-time applications (i.e., VoIP, Multimedia, etc.) losses may interrupt the connection.

Recovery schemes should guarantee minimal or if possible no packet losses during the restoration period.

Packet Re-ordering: Recovery schemes may require packet re-ordering during switchover from/to the working LSP. Packet re-ordering is mainly required at the egress side.

Resource utilization: In order to provide fault tolerance in a system, especially pre-planned protection (e.g., fast protection), redundant bandwidth has to be reserved or dedicated. Therefore, the amount of redundant bandwidth required has to be effectively utilized.

Recovery time: It is the time from the start of failure detection, until the time when the packets start flowing through the alternative or backup LSP. The message that is used to inform the router responsible of switching traffic to the backup path is called Failure Indication Signal (FIS).

Table 2.1 summarizes the main aspects of different combination of restoration and repairing methods used to protect traffic from network failure.

Restoration and repair method	**Resource Requirements**	**Speed of Recovery**	**Packet Loss**	**Protection Path (length)**
Dynamic/Local Repair	No	Slow	Minimum	Might not be the SP available
Dynamic/Global Repair	No	As above + FIS	High (compared To above)	Path is shortest available
Fast protection/ Local repair	Yes	Fast	Minimum	May not be optimal
Fast protection/ Global repair	Yes	Fast, depends on FIS	High	Better than the above

Table 2.1 Comparison table for repair techniques, SP: shortest path, [12, 14].

The number of dropped packets in dynamic protection could be larger than fast protection. However, resource utilization is more efficient in dynamic protection than fast protection because reservation is only made after failure occurs. This scheme also provides more flexibility in the establishment of a new backup LSP. The main advantage of using a dynamic protection is that an optimal backup LSP may be established and the bandwidth requirements are minimal. However, there is no guarantee to find available bandwidth for the backup LSP as there is no reservation made in advance.

2.2.3 Related work in MPLS fault tolerance

The dynamic protection model may not be suitable for time sensitive applications because of its large recovery time [7, 14]. For this reason, in this thesis we consider the fast protection scheme because our main concern is to reduce the recovery delay and packet loss.

The Simple–Dynamic [15] scheme proposed a simple dynamic protection repair scheme, that is, the backup path is established after detecting a node failure in the working path. The backup path is established up to the path merge LSR (PML) (i.e., LSR1 in Figure 2.8) router along a shortest transmission path that does not utilize any working path. Figure 2.8 describes the protection scheme in [15]. When a failure in the link between LSR4 and LSR9 occurs, the traffic can be rerouted along the following backup LSP: LSR 4-3-7-8-9. Other rerouting or dynamic approaches can be found in [16, 17].

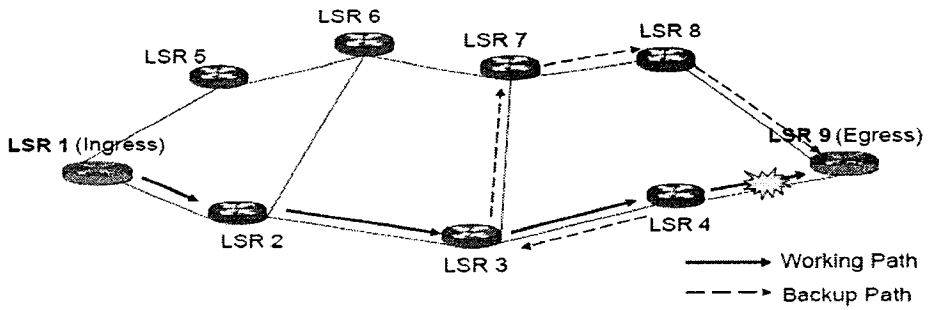


Figure 2.8 Simple dynamic restoration example in MPLS network

The 1+1 "one plus one" protection scheme discussed in [7, 8, 15] can provide path recovery without packet loss or recovery time. The resources (bandwidth, buffers, and processing capacity) on the recovery path are fully reserved, and carry the same traffic as the working path, requiring substantial amount of dedicated backup resources. Selection between the traffic on the working and recovery paths is made at the path merge LSR (PML) or the egress LER. In this scheme, the resources dedicated for the recovery of the working traffic may not be used for anything else.

Generally speaking, protection schemes use 1:1 path protection where every link can carry either regular traffic or backup traffic and thus does not require dedicated backup links. This can be extendible to 1:N, and M:N shared protection where N working paths share M backup paths for efficient bandwidth utilization. To support differentiated services, normally if the network is safe, the capacity of the backup paths is utilized to carry packets belonging to lower priority class types (e.g., best effort). In case of failure, the lower class traffic is blocked to support backup for high priority traffic [18].

The paper by Makam *et al.* [7] proposes a PSL (Path Switched LSR) oriented path protection mechanism that consists of three components: minimize the delay experienced by notification message traveling from the fault detection node to the protection

switching node (i.e., PSL) by building a fast and efficient reverse notification tree structure, a hello protocol to detect faults, and a lightweight notification transport protocol to achieve scalability. Basically, when a failure occurs a notification message is sent to the ingress router and the traffic is switched to the pre-established backup LSP.

The paper by Haskin *et al.* [19] is also based on pre-established alternative path. The backup path is comprised of two segments. The first segment is established between the last hop working switch and the ingress LSR in the reverse direction of the working path. The second segment is built between the ingress LSR and the egress LSR along an LSP that does not utilize any working path. In Figure 2.9, if the link between LSR4 and LSR9 fails, all the traffic in working path is rerouted along the backup path, LSR 4-3-2-1-5-6-7-8-9. Optionally, as soon as LSR1 detects the reverse traffic flow, it may stop sending traffic downstream of the primary path and start sending data traffic directly along the second segment.

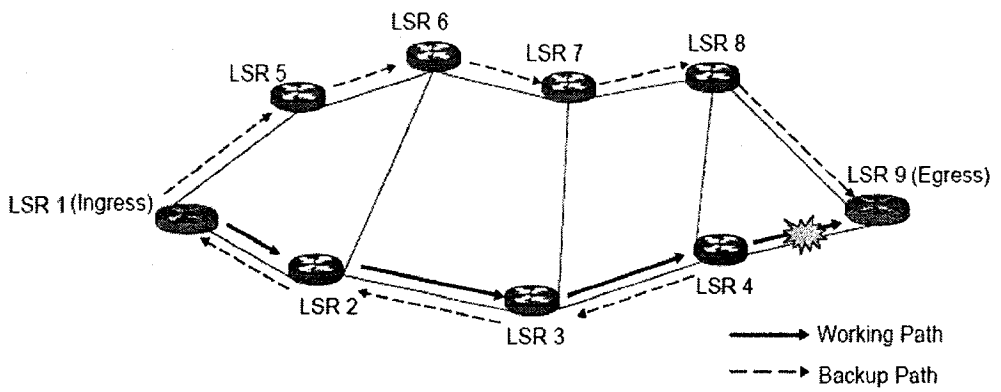


Figure 2.9 Path restoration examples [19]

In 1:1 protection method the bandwidth requirements are higher (because each working LSP should reserve equal bandwidth for the backup LSP) than in shared path protection (1: N or M: N) method. Figure 2.10 shows the basic principle for shared

backup path protection in MPLS networks [9] where the recovery path may be shared if its corresponding working paths are disjoint. In addition, the resources allocated for the recovery path can be fully available to pre-emptible low- priority traffic.

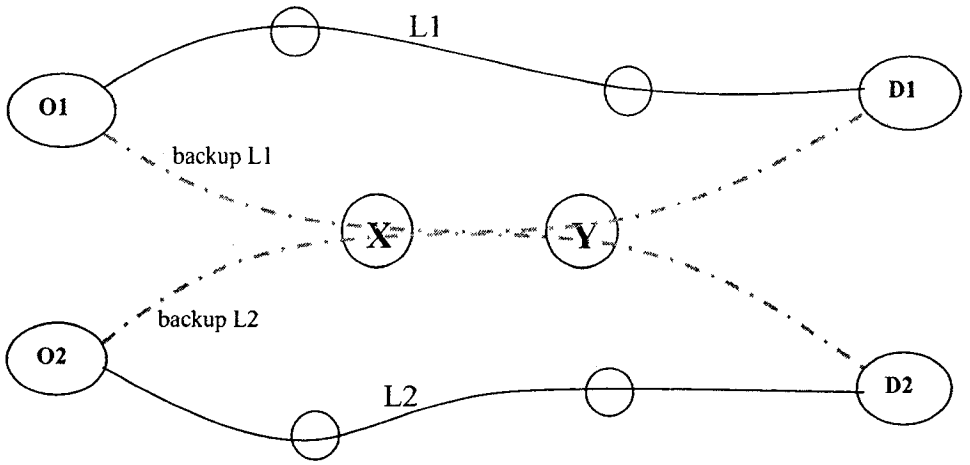


Figure 2.10 Working paths L1 and L2 may share backup capacity on XY only if L1 and L2 are disjoint [9]

The authors of [11] present a solution to further reduce the Spare Capacity Allocation (SCA) to near optimality by proposing a *successive survivable routing* (SSR) algorithm for mesh based communication networks. First, per-flow spare capacity sharing is captured by a spare provision matrix (SPM). The SPM matrix has a dimension of the number of failure scenarios by the number of links. It is used by each demand to route the backup path and share spare capacity with other backup paths. Next, based on a special link matrix calculated from SPM, SSR iteratively routes/updates backup paths in order to minimize the cost of total spare capacity. The network redundancy (which is the ratio of the total spare capacity over the total working capacity) for shared path restoration cases

measured in this paper ranges from 35% to 70%. The paper, however, does not consider recovery delay, packet loss and re-ordering.

To improve bandwidth utilization, [8] proposes path merging technique to allow bandwidth sharing on common links among a service LSP and its backup LSPs. The scheme extends the sharing principle and allows distributed sharing by including broader bandwidth sharing among any backup LSPs from different service LSPs.

Seok *et al.* proposed in [20] a fault tolerant multi-path traffic engineering scheme for MPLS networks with the objective to effectively control the network resource utilization. The proposed scheme consists of the maximally disjoint multi-path configuration and the traffic rerouting mechanism for fault recovery. The authors use the linear programming formulation to configure disjoint multi-path and the traffic rerouting solution. So when the statistical traffic demand is known between a source LSR and a destination LSR, then the traffic engineering can be applied with the following objectives: set all LSPs configuration in order to find disjoint paths for each node pair, subject to minimization of the maximum of link utilization. When some link failures are detected, the proposed mechanism routes the traffic flowing on the failed LSPs into available LSPs. Only bandwidth utilization is considered in this approach; where other fault tolerance issues such as recovery time and packet loss have not been discussed. However, this approach applies the same mechanism used in any other path protection approach where recovery time and packet loss occurs.

The paper by Buddhikot *et al.* [21] addresses guaranteed uninterrupted connectivity in case of link/node failure in primary path by finding a set of backup LSPs that protect the links along the primary LSP. The authors introduce the concept of "backtracking"

where the backup path may originate at the failed link (local restoration) or in the worst case the backup paths may originate at the ingress node. In other words, it provides algorithms that offer a way to tradeoff bandwidth to meet a range of restoration latency requirements.

The paper by Virk *et al.* [22] presents an economical global protection framework that is designed to provide minimal involvement of intermediate LSRs, reduction in the number of Path Switch LSRs responsible to switch the traffic from failed working path to the backup path, fast and cost effective fault notification. The proposed scheme uses a directory service that is logically centralized and physically distributed database to provide a fast lookup of information. In the paper, the performance evaluation only considers packet loss.

A distributed LSP scheme to reduce spare bandwidth demand in MPLS networks proposed in [24] is also based on pre-established protection scheme. Here the ingress LER groups the incoming LSP, which is carrying the incoming IP packet flows, into several sub-groups each of which is assigned to a distinct sub-LSP from a number of K sub-LSPs. A single failure is only assumed; hence only one backup sub-LSP of the amount 1/K is required. The scheme proposed in [24] does not however consider reducing recovery delay and/or packet loss. A path failure results in switching over to the backup path and therefore incurs considerable recovery delay.

The concept of spare capacity reallocation is also presented by authors in [25]. The proposed algorithm called Short Leap Shared protection with spare capacity reallocation (SLSP-R) that is used to initiate an on-line reconfiguration of the spare capacity, where both the network dynamics and computation efficiency are jointly considered. The author

in [26] presented a methodology on how to implement the path restoration strategy to achieve effective control, fast restoration, and economic network capacity placement in MPLS and ATM networks.

Wang *et al.* proposed in [27] a new scheme to utilize the network resources. The goal is to pre-identify backup paths but not to reserve them. The generalized model is to assign priorities for the primary and backup paths, based on a model for current networks where flows are classified by their importance. In this model, higher priority flows are allowed to preempt lower priority flows whenever there is any shortage of the bandwidth. The backup paths in this model are considered to be of lower priority than primary paths, which is differing from the reservation model where the backup paths are priced similarly as the primary paths. This approach has a tradeoff between the availability of backup paths and its cost. In particular, there is no guarantee that a backup path will be available to a flow in case of a primary path failure since it may be in use by higher application. To enhance the availability of backup paths, the scheme tries to achieve this by pre-identifying multiple backup paths protecting each link on the primary path.

Ruan *et a.l* [28] proposed a restoration scheme called UNIFR, which tries to provide fast restoration as local restoration repair scheme while achieving better bandwidth efficiency than the local restoration does. The key idea is to let the upstream node that is adjacent to the failure switch the traffic to a backup path from itself (i.e., the upstream node) to the destination node immediately after it detects a failure. Like local restoration repair scheme, UNIFR achieves fast restoration by eliminating the failure notification delay incurred in the global restoration scheme. Unlike local restoration repair scheme, this approach uses semi-global backup paths from the failure detecting node directly to

the destination node. The semi-global backup paths offer better opportunity for spare capacity sharing among themselves than the local backup paths used in local repair restoration scheme.

In [29], the authors proposed a scheduling mechanism for failure restoration in MPLS networks. The proposed scheme presents a scheduling mechanism for sending notification messages of connections with different rerouting priorities, which ensures that higher priority connections have preferential access to better routes and available network resources, so that QoS performance of higher priority connections could be guaranteed after restoration. In addition, recovery speed can be increased for higher priority LSPs by preventing the quick setup of low priority LSPs.

A mechanism to share and provide better bandwidth utilization [30] proposes path merging technique to allow limited bandwidth sharing on common links among a service LSP and its backup LSPs. The scheme extends the sharing principle and allows distributed sharing by including broader bandwidth sharing among any backup LSPs from different service LSPs.

The paper by Avallone *et al.* [41] proposes a splitting infrastructure for load balancing in an MPLS network. The paper proposes some operations to be performed at edge routers to handle multi-path routing issues such as packet reordering and processing resulted from the use of splitting technique. The mechanism proposed is based on the idea of splitting flows on a per-packet basis. The authors proposed a use of extra MPLS header in accordance of MPLS stacking to provide packet sequence numbering. However, this approach might face practical implementation issues as they use the experimental and TTL fields in MPLS header to identify splitting procedure.

It is seen from the previous related work in MPLS fault tolerance that recovery time, packet loss and bandwidth utilization are the main service parameters for real-time traffic. However, most of the approaches in literature focus on reducing working and recovery bandwidth utilization while considering the recovery delay. There is no scheme that can provide path protection with no packet loss and no recovery delay except the 1+1 protection at the cost of 100% redundant bandwidth reservation, or disperse routing [31] that can handle single failures with lower redundant bandwidth but requires to know the location of the failure.

The following section introduces the second objective of this thesis, which is MPLS security. Related work on MPLS security is discussed.

2.3 MPLS Network Security

With the increasing deployment of MPLS networks, security concerns have been raised. The basic architecture of MPLS networks does not support security aspects such as data confidentiality, data integrity, and availability. MPLS technology has emerged mainly to provide high speed packet delivery. As a result security considerations have not been discussed thoroughly until recent demands for security have emerged by most providers and researchers.

The reason why MPLS does not provide encryption mechanisms is related to the purpose it was built for. In the conventional IP networks, every router in the network has a role in analyzing IP packet headers, to classify, and to process every packet passing through it. This of course will add more overhead and delay in the network [32, 33]. In MPLS networks, only two routers (the ingress and egress routers) are responsible for this

task. Core or Label Switch Routers (LSRs) in MPLS network will only forward packets based on labels transmitted through a pre-established Label Switch Path (LSP). The use of security protocols such as IPSec requires the core MPLS routers to analyze and process packets' header, which will result in reducing the performance of MPLS network [34]. More discussion on issues regarding the application of IPSec over MPLS will be provided in Chapter 6.

Generally, network security covers issues such as:

- **Confidentiality:** The property that information is not made available or disclosed to unauthorized individuals, entities.
- **Integrity:** The property that information has not been modified or destroyed in an unauthorized manner.
- **Availability:** The property of a system or a system resource being accessible or usable upon demand by an authorized system entity.
- **IP spoofing:** refers to the creation of IP packets with a forged (spoofed) source IP address with the purpose of concealing the identity of the sender or impersonating another computing system.
- **Nonrepudiation:** A security service that provides protection against false denial of involvement in a communication.
- **Access control:** Protection of system resources against unauthorized access.
- **Authentication:** The process of verifying an identity claimed by a system entity.

In this thesis, we consider the types of attacks we intend to protect the MPLS network from. We are providing packet *data confidentiality* (including the header and payload) so that an attacker cannot collect and analyze traffic data or understand routing configuration. In this thesis we also propose a method to provide data integrity, availability, and protection against IP spoofing.

2.4 Related Work in MPLS Security

Every layer of communication has its own unique security challenges. Network security should be addressed at multiple layers to protect the network for different vulnerabilities. For example, IPSec is used to enable secure transfer of packets at the IP layer. A MPLS network requires the corporation of two network layers, IP and data link layers.

Most of the existing work on MPLS security concentrates on MPLS-VPN architecture. Behringer *et al.* [35] discussed MPLS VPN security. The authors present a practical guide to hardening MPLS networks. They assumed "zones of trust" for MPLS VPN environment. In other words, they assumed the MPLS domain to be secure or trusted. As a result, the focus was on implementing security measures on MPLS edges (i.e., Provider Edge routers). This assumption led to some security concerns such as VPN data confidentiality and integrity. Therefore, there is no guarantee to VPN users that packets do not get read or sniffed when they are in transit over the MPLS domain. The authors left the issue of securing MPLS core routers (if they are not trusted) as an open issue for more discussion. However, the authors proposed the application of IPSec over

MPLS VPN network in order to support security features such as confidentiality and integrity when the MPLS domain router are not trusted.

The implementation of IPSec in MPLS VPN networks may generally be applied between MPLS network borders, i.e., edge routers. Edge routers are also named PE (Provider Edge) routers while core LSRs are called P routers [35]. The P routers are only responsible for applying forwarding packets based VPN MPLS labels, while the PE routers have more complex tasks such as applying access control rules and analyses. IPSec can be applied at various points, i.e., Customer Edge to Customer Edge (CE-CE IPSec), or PE-PE IPSec, or Remote access as shown in Figure 2.11. In this work we only focus on PE-PE IPSec implementation since this part applies to MPLS core network.

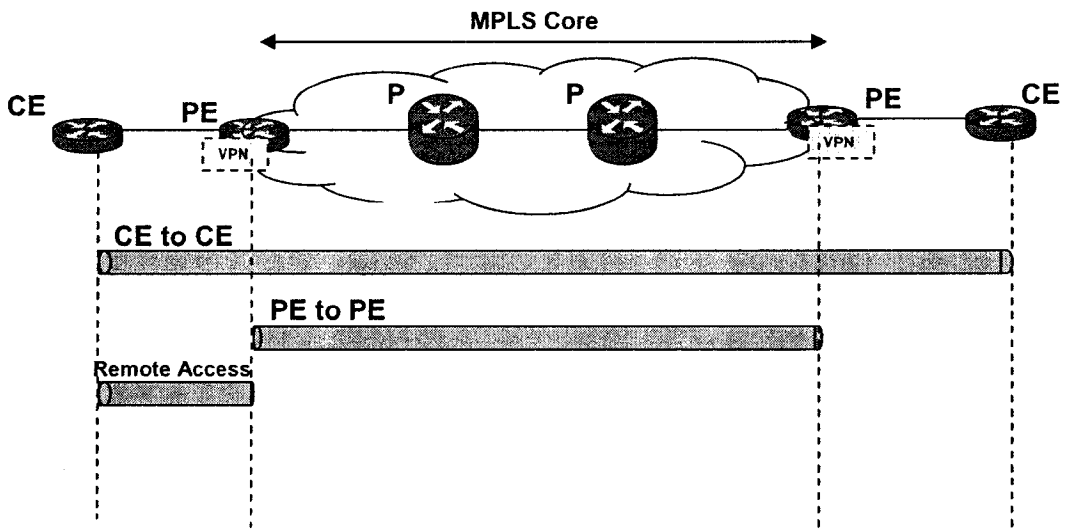


Figure 2.11 IPSec Termination Points in an MPLS VPN Environment [70]

IPSec PE-PE provides suitable protection for the following threats:

- Eavesdropping on lines between the PEs or P routers (The LSR router is called 'P' router), specifically, if the MPLS core is routed over untrusted areas.

- Misbehavior of P routers, which can lead to packets being changed or routed to the wrong egress PE.

Figure 2.12 shows a possible implementation of PE-PE IPSec in MPLS/VPN [70, 71]. In this mode, the LSPs in the MPLS core are replaced with IPSec tunnels. Therefore, it is also an alternative for running MPLS layer 3 VPN networks. The VPN label is pre-pended to the VPN packet as in normal MPLS; however, IPSec can only secure IP packets, not labeled packets. Therefore,

a) the labeled packet is first encapsulated in Generic Routing Encapsulation (GRE),

b) the GRE packet can then be secured with IPSec. Because the endpoints of the GRE tunnel are the same as for the IPSec tunnel, transport mode can be used, and this reuses the GRE header.

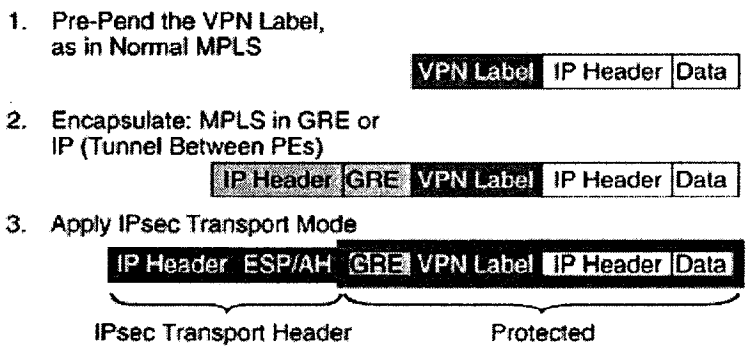


Figure 2.12 IPSec Encapsulation for PE-PE Security [70]

The paper by Lee *et.al.* [36] discusses some issues and procedures that should be considered when providing MPLS VPN services such as security, and scalability, QoS. The authors raised the concern that might occur by the use of security procedures in MPLS performance such as slowing down network connections.

Another study by T. Saad *et al.* [32] discusses the effect of MPLS-based tunnels on end-to-end virtual connection service and security. The study shows that applying IPSec in MPLS-based tunnels reduces overall throughput of TCP flow and adds more overhead.

A cryptographic protocol to protect MPLS labels was proposed by Barlow *et al.* [37]. The design applies simple encryption technique on labels to prevent header modification. The protocol does not provide data confidentiality. The authors proposed the Blowfish algorithm as an encryption algorithm. However, the approach proposes to use bits from the experimental field bits and TTL field (two bits) of MPLS header to increase the chance of detecting bad or invalid MPLS header resulted from label encryption. However, this practice is not practical as experimental bits are used in MPLS to support differentiated services.

Chung *et al.* [38] proposed a method for RSA [92] algorithm suitable for multi-path topology. It was mentioned that the algorithm can be applied to MPLS networks however the details are not provided. They proposed a modified RSA algorithm that can support multi-path routing.

The authors in [39, 40] propose a framework for MPLS-based network survivability against security threats. It requires digital signature of all the signaling messages for the MPLS control plane protocols. This type of protection is necessary to protect the routing information, e.g. the multi-path routing information.

In reference [43] the authors discuss possible solutions to prevent or reduce availability problems in IP/MPLS networks such as denial of service attack. The authors discuss some useful practices that can help to achieve the availability.

Another work that addresses the need to secure signaling protocols in MPLS is studied by Zhi *et al.* [42]. The work analyzes an authentication mechanism for securing the RSVP and RSVP-TE control messages, and studies their performance. The time for authenticating the signaling messages depends on the algorithm used, and increases slightly in the order of MD5, SHA-1, RIPEMD160 and SHA-256 [93]. The performance of the RSVP-TE with multiple sessions was measured.

The deployment of IP-MPLS networks between different ISP have some challenges that should also be taken into consideration. The paper by Fang *et al.* [44] addresses requirements, implementation option, and challenges for the deployment of IP-MPLS between different service providers. General requirements for the interconnections end users' and service providers' perspectives include security, scalability, and manageability, QoS, and end-to-end SLAs. From an implementation and challenges perspectives this includes methods for guaranteeing consistent QoS and security services across providers' boundaries.

The paper by Daugherty *et al.* [45] discusses the blind insertion attacks on MPLS VPNs. If MPLS infrastructure is not trusted, then MPLS VPNS that use IP tunnels expose customer networks to the possibility of blind-insertion attacks, in which an attacker circumvents a provider's defense perimeter of packet filters on edge routers and injects spoofed packets toward a PE router servicing a customer VPN.

From the previous discussion on related work of MPLS security we notice that most of the research proposals concentrate on MPLS-VPN point of view. In other words, the domain of a MPLS network is assumed to be trusted. The security control measurements are mainly applied to MPLS-VPN edge routers. It is also seen that the application of IPSec to provide confidentiality and integrity of data inside MPLS domain is accompanied by significant overhead as it will be discussed in more detail in Section 5.5. It is important to take into account not to reduce the performance of the MPLS network such as high speed networking when applying security protocols. The security of the MPLS domain may not always be assumed to be trusted. Therefore, in this thesis we tackle this case where we assume that the MPLS domain is not trusted. It is worth to note that the MPLS security subject is still a work in progress in IETF MPLS Working Group.

2.5 Summary

In this chapter we have introduced some background information about MPLS networks. Both fault tolerance and security literature for MPLS have been investigated. The literature review on MPLS fault tolerance shows that many approaches have been proposed to protect the network against failure. There are two main techniques used for recovery in MPLS networks, dynamic and pre-configured protection. Each of these techniques has pros and cons in terms of recovery time, packet loss, packet re-ordering, and bandwidth utilization.

The technique that we are interested to compare with is the pre-configured (fast protection scheme) as it performs better than the dynamic technique for time sensitive applications.

In this chapter we also explored MPLS security issues and introduced some of security approaches proposed until now. The security issue in MPLS network is an evolving issue and has not been until now standardized. It is still until now a work in progress in the IETF MPLS working group.

Chapter 3

Background on the Threshold Sharing Scheme

In this chapter we discuss the basic idea of the Threshold Sharing Scheme. A numerical example which illustrates the message distribution and reconstruction is presented. Moreover, some applications where the threshold sharing scheme can be used is discussed.

3.1 Threshold Sharing Scheme (TSS)

In cryptography, secret sharing refers to any method for distributing a secret amongst a group of participants, each of which is allocated a share of the secret. The secret can only be reconstructed when the shares are combined together; individual shares are of no use on their own. More formally, the idea behind the threshold sharing scheme (TSS) is to divide a message into n pieces, called shadows or shares, such that any k of them can be used to reconstruct the original message. Using any number of shares less than k will not help to reconstruct the original message.

Adi Shamir polynomial approach known as Shamir's (k, n) threshold scheme [47] will be used to show this concept. Shamir's (k, n) threshold scheme is one of the most well-known examples of secret sharing schemes, which provides a very simple and efficient way to share a secret among any k of the n participants.

Let p be a prime number. A polynomial in the intermediate x over the finite field Z_p (Appendix A) is an expression of the form

$$f(x) = a_{k-1}x^{k-1} + \ldots + a_2x^2 + a_1x + a_0 \qquad \text{mod } p \qquad (3.1)$$

where each $a_i \in Z_p$ and $n \geq k > 0$. The element a_i is called the coefficient of x_i in $f(x)$. The largest integer $k-1$ for which $a_{k-1} \neq 0$ is called the degree of $f(x)$, a_{k-1} is called the leading coefficient of $f(x)$. If $f(x) = a_0$ (a constant polynomial) and $a_0 \neq 0$, then $f(x)$ has degree 0. If all the coefficients of $f(x)$ are 0, then $f(x)$ is called the *zero* polynomial.

The following is a description of the distribution process of Shamir's (k, n) threshold scheme:

Let $K \in Z_p$, where $p \geq n+1$ is a prime. Hence, the key K will be an element of Z_p.

Initialization Phase:

The dealer or distributor, D, chooses n distinct, non-zero elements of Z_p, denoted x_i, $0 \leq i \leq n-1$. For $0 \leq i \leq n-1$, D gives the value x_i to P_i, (P_i is a participant).

Share Distribution:

- Suppose D wants to share a key $K \in Z_p$. D secretly chooses (independently at random) $k-1$ elements of Z_p to be coefficients $a_1, \ldots, a_{k-1}$.
- For $0 \leq i \leq n-1$, D computes $y_i = a(x_i)$, where

$$a(x) = K + \sum_{j=1}^{k-1} a_j x^j \bmod p \tag{3.2}$$

- For $0 \leq i \leq n-1$, D gives the share y_i to P_i.

In this scheme, the dealer D constructs a random polynomial $a(x)$ of degree at most $k-1$ in which the constant term is the secret key, K. Every participant obtains a point (x_i, y_i) on this polynomial.

Reconstruction:

The following is a description of the reconstruction process of (k, n) Shamir's threshold scheme.

The Shamir's (k, n) threshold scheme is based on the Lagrange interpolation for polynomials. The following Lagrange interpolation formula is used to reconstruct the message in an explicit formula for the unique polynomial $a(x)$ of degree at most $k-1$ which contains the pairs (x_i, y_i), $0 \leq j \leq k-1$.

$$a(x) = \sum_{i=0}^{k-1} y_i \prod_{0 \leq j \leq k-1, j \neq i} \frac{x - x_j}{x_i - x_j} \tag{3.3}$$

We will interpret Shamir's (k, n) threshold scheme from the point of view of solving systems of linear equations. An invariant polynomial $y = f(x)$ of degree at most $k-1$ is uniquely defined by k points (x_i, y_i) with distinct x_i, since they define k linearly independent equations in k unknowns.

Example: let the secret K be 11, then in a (3, 5) threshold sharing scheme, any three of five shares can reconstruct K. First a quadratic equation should be generated, where the coefficients a_1 and a_2 are chosen randomly, and the p value should be greater than any coefficient value, in this case 13. The x-coordinates are $x_i = i$, $1 \leq i \leq 5$, and

$$a(x) = (11 + 8x + 7x^2) \bmod 13$$

Where $a_1 = 8$, and $a_2 = 7$.

The following shares are calculated:

$a(1) \equiv 0 \mod 13$

$a(2) \equiv 3 \mod 13$

$a(3) \equiv 7 \mod 13$

$a(4) \equiv 12 \mod 13$

$a(5) \equiv 5 \mod 13$

a_0 is reconstructed from any three of the above shares (let's say $a(2)$, $a(3)$ and $a(5)$) using the following equation:

$$a(x) = [a_0\, a_1 \ldots a_{k-1}] \begin{bmatrix} 1 & 1 & 1 & \ldots & 1 \\ x_1 & x_2 & x_3 & & x_n \\ x_1^2 & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ x_1^{k-1} & x_2^{k-1} & \cdot & & x_n^{k-1} \end{bmatrix} \bmod p \qquad (3.4)$$

$(a_0 + a_1(2) + a_2(2)^2) \equiv 3 \mod 13$

$(a_0 + a_1(3) + a_2(3)^2) \equiv 7 \mod 13$

$(a_0 + a_1 (5) + a_2 (5)^2) \equiv 5 \bmod 13$

The following solution is obtained after applying equation (3.3):

$$a(x) = \left(3. \frac{(x-3)(x-5)}{(2-3)(2-5)} + 7. \frac{(x-2)(x-5)}{(3-2)(3-5)} + 5. \frac{(x-2)(x-3)}{(5-2)(5-3)} \right) \bmod 13$$

$$= \left(3. \frac{(x^2-8x+15)}{3} + 7. \frac{(x^2-7x+10)}{-2} + 5. \frac{(x^2-5x+6)}{6} \right) \bmod 13$$

$$= (3. (3)^{-1} (x^2-8x+15) + 7. (-2)^{-1} (x^2-7x+10) + 5. (6)^{-1} (x^2-5x+6)) \bmod 13$$

The inverse values in the above equation are calculated using the Extended Euclidean algorithm as provided in appendix A.

$$= (3. (9) (x^2-8x+15) + 7. (6) (x^2-7x+10) + 5. (11) (x^2-5x+6)) \bmod 13$$

$$= (124 x^2 - 785x + 1155) \bmod 13$$

$$= (7 x^2 + 8x + 11) \bmod 13$$

This system has a unique solution in Z_{13}: $a_0 = 11$, $a_1 = 8$, $a_2 = 7$. The key is therefore K= a_0 =11.

3.2 Applications of Threshold Sharing Scheme

In this section we present some of the areas which use threshold sharing to provide security and fault tolerance.

The use of threshold sharing from a networking perspective has been studied in many proposals. Zhou *et al.* [48] proposed to combine secret sharing and multi-path to improve the availability and security of certificate authority (CA) in ad hoc networks (MANET). The threshold signature scheme used in [48] is shown in Figure 3.1. Given a service consisting of three servers, let *K/k* be the public/private key pair of service. Using a (2, 3) threshold secret sharing scheme, each server *i* gets a share s_i of the private key *k*. For a message *m*, server *i* can generate a partial signature *PS* (*m*, s_i) using its share s_i. Servers 1 and 3 both generate partial signatures and forward the signatures to a combiner *C*. Even though server 2 fails to submit a partial signature, *C* is able to generate the signature $(m)_k$ of *m* signed by server private key *k*. Another distributed certificate issuing scheme has been suggested by Kang *et al.* [49].

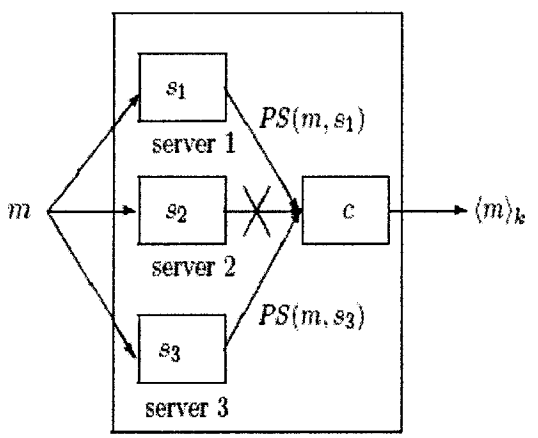


Figure 3.1 Threshold signature K/k [48]

The paper by Castelluccia *et al.* [50] presents two self-keying mechanisms for pair wise key establishment in mobile ad hoc networks which do not require any centralized support. The mechanisms are built using the threshold secret sharing, and are robust and secure against a collusion of up to a certain number of nodes.

The paper by Lou *et al.* [59] proposed to use the threshold secret sharing to provide confidentiality of data in ad hoc traffic with the consideration of saving bandwidth. The focus was to find the disjoint paths within the ad hoc environment. Other security parameters such as data integrity have not been addressed. In this thesis, we adopted the bandwidth saving mechanism used in this work.

The threshold secret sharing algorithm can protect a secret with high reliability and flexibility. These advantages can be achieved only when all the participants are honest, i.e., all the participants willing to pool their shares shall always present the true ones [47]. Cheating detection is an important issue in the secret sharing scheme. However, cheater identification may also be required. If some dishonest participants exist, the other honest participants will obtain a false secret, while the cheaters may individually obtain the true one [11].

The use of threshold secret sharing to provide security for stored data has been also proposed. The paper by Greenan *et al.* [51] propose a secure, distributed, very long-term archival storage system that eliminates the use of keyed encryption through the use of unconditionally secure secret sharing. Another proposal for the use of threshold secret sharing as a decentralized recovery for survivable storage systems is presented by Wong [23]. The use of threshold secret sharing to provide fault tolerance for *data networking* has not been considered to the best of our knowledge.

3.3 Another Coding Technique

There are other encoding models used to provide fault tolerant systems such as the disparity routing model proposed by Maxemchuk *et al.* [31]. The model requires

networks with multiple disjoint paths to spread the data from a source over several paths. To provide redundancy, the message is divided into fewer sub-messages than there are paths. Additional sub-messages are constructed as a linear combination of the bits in the original sub-messages (e.g., modulo-2, parity check codes), such that the original message may be reconstructed without receiving all of the sub-messages. For a (*N, K*) system, *N* is the number of paths used and the original message is subdivided into *K* parts, where $N > K$. For a modulo-2 system where $N = K+1$, and the system can only tolerate single path failure. The missing sub-message creates holes in the codeword whose position should be known and whose value can be reconstructed from the received sub-messages. The scheme is incapable of providing multiple failures. Therefore, in order to handle multiple failures their approach uses "Flooding" strategy (e.g., a *(3, 1)* system,) where the entire message is transmitted on all three paths until at least one path is received, thus requiring more redundant bandwidth.

Another work by Bheemarjuna *et. al* [52] has tackled the security and reliability issues in ad hoc wireless networks. The scheme addresses the reliability in the context of both erasure and corruption channels and quantifies the security factor in terms of the number of eavesdropping nodes. Indeed, the authors did not address other security issues like data integrity and IP spoofing. The coding scheme used is based on the Reed-Solomon.

3.4 Summary

In this chapter, a background on threshold sharing is introduced. The original scheme of Shamir's secret sharing has been studied to help us understand its

implementation which will be discussed in the next chapter. The scheme is used basically to provide security for secret key sharing. The threshold secret sharing scheme can also be used to provide resiliency for storage data and software. We also presented some literature work on the use of the threshold secret sharing.

In the next chapter, a modified threshold secret sharing scheme is proposed to provide data security and fault tolerance in MPLS networks. The proposed scheme provides fault tolerance in MPLS network in a non-conventional concept which is different from available recovery models for MPLS. The security of MPLS networks can also be improved using the proposed approach scheme.

Chapter 4

Modified Threshold Sharing for MPLS Networks

In the previous chapters, we introduced the fault tolerance and security issues (i.e., data confidentiality and integrity, availability, and IP spoofing throughout the thesis) in MPLS networks in addition to the threshold sharing scheme.

In this chapter, we present our proposed work and demonstrate how to integrate the original threshold sharing scheme to support fault tolerance and security in MPLS networks.

4.1 Modified Threshold Sharing Scheme (TSS)

Our proposed algorithm can be used to provide security and fault tolerance at the same time. We use the threshold sharing scheme described in previous chapter with a modified version to suit MPLS networking requirements such as bandwidth utilization. The original TSS algorithm is not suitable for network resource utilization as it will be explained later in this section.

From a network point of view, if the whole message follows the same path to the destination, the chance of risk that an attacker could intercept all information in the message is high. However using a multi-path routing protocol combined with (k, n) threshold sharing scheme makes it hard for the attacker to intercept all the information. This procedure requires the attacker to compromise at least k different node(s)/link(s) on

k disjoint paths (here two paths are considered independent if no shared node(s)/link(s) exist between a source and destination) to be able to reconstruct the original IP packet.

The same idea of threshold sharing scheme described in [59] can be applied to provide fault tolerance in MPLS considering other requirements that could be different from those needed in security. Recovery delay time and packet loss are the main parameters in the event of MPLS network failure for applications that are sensitive to delay and data loss. In order to overcome these problems, we introduce a novel approach that guarantees to continue the MPLS network operation with no packet loss or recovery delay and with reasonable network resource utilization in the event of single or multiple LSPs failure. Our approach requires the existing of at least three disjoint LSPs between an ingress and an egress routers to provide single LSP failure protection (i.e., $k = 2$, $n = 3$). If it is not possible to have three disjoint LSPs, the algorithm should look for maximally disjoint LSPs. Selecting paths which are maximally disjoint will affect the fault tolerance performance for parameters mentioned above. The definition of maximally disjoint paths was introduced in Section 1.2.

Our approach uses a modified version of the (k, n) threshold sharing algorithm with multi-path routing wherein k out of n LSPs are required to reconstruct the original message. For example, if we are using a (2, 3) threshold sharing algorithm, then it is only enough for the egress router to receive MPLS packets or shares coming from two LSPs to be able to reconstruct the original message which was divided and encoded at the ingress router.

As we mentioned earlier, our concern is to provide fault tolerance mechanism with reasonable bandwidth requirements in the network. We are using a modified version of

the threshold sharing algorithm to fulfill the bandwidth requirement because its original version is not suitable for the purpose of reducing redundant bandwidth. The idea to use the threshold sharing for fault tolerance in networks has never been studied and especially for MPLS networks to the best of our knowledge.

Figure 4.1 shows an architectural view of the system and shows how the original IP packet is distributed and reconstructed in a MPLS network.

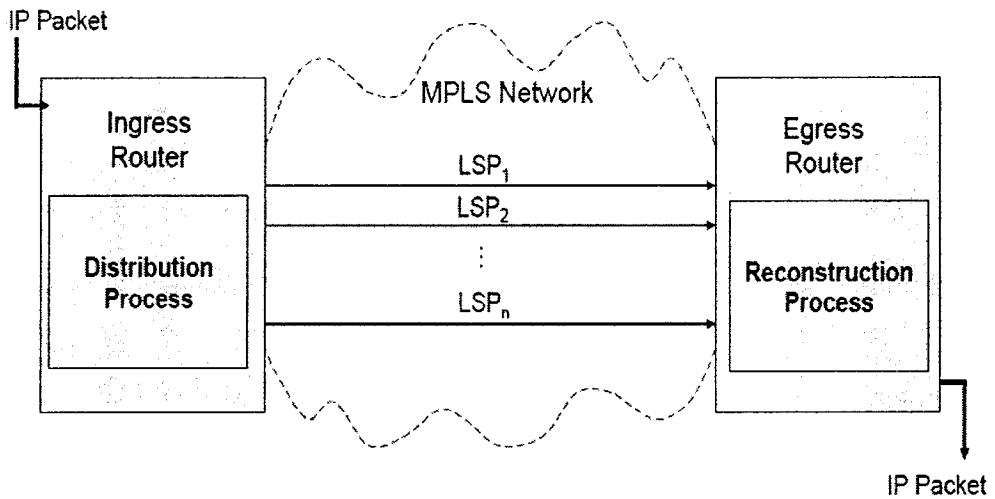


Figure 4.1 Distribution and Reconstruction processes

When an IP packet enters a MPLS ingress router, a distribution process at the ingress router is used to divide, encode and generate the n share messages that will become the payloads for n MPLS packets allocated over n disjoint or maximally disjoint LSPs obtained using multi-path routing [5, 53, 54, 59] or Equal-Cost Multi-path (ECMP) routing [55]. When an ECMP routing scheme is used, where all multiple LSPs found have the same delay cost, at least k MPLS packets will be received at the same time by the egress router. In case of other multi-path routing protocols, at least k MPLS packets should be received within the time frame of receiving a MPLS packet from the slowest

LSP. Thereafter the reconstruction process at the egress router generates the original IP packet from the first k MPLS packets received.

Our approach provides no involvement of intermediate LSRs in the core of the MPLS network. Only edge routers are responsible for the protection framework. The following sections describe distribution and reconstruction processes at ingress and egress routers.

4.1.1 Distribution Process

Figure 4.2 shows how the Distribution Process applies a (3, 4) Threshold Secret Sharing scheme into an IP packet. The IP packet is first divided into m blocks S_1, S_2, ... , S_m where each block size L is a multiple of k bytes. The effect of block size L on packet processing is shown later in Figure 4.4. From Equation (3.1), it can be easily seen that there are three coefficients, a_0, a_1 and a_2, for a (3, 4) scheme where $k = 3$. Each block is therefore divided in k equal parts (same as the number of polynomial coefficients), and these coefficients are assigned values from the block (unlike original TSS scheme where a_1 and a_2 are assigned random values [59]). For example $a_0 = 06$, $a_1 = 28$ (1C in hex), and $a_2 = 08$ for the block S_1. Next m quadratic equations $f(S_j, x)$, where $1 \leq j \leq m$, are generated using the three coefficients from each of the m blocks, that is, every block generates a quadratic equation. Each quadratic equation is solved n times using the n different x_i, $1 \leq i \leq n$, values as agreed between a sender (ingress) and a receiver (egress). Each MPLS packet payload therefore consists of m encoded values obtained from the m quadratic equations using the same x_i value, as shown in Figure 4.2. Each LSP can be

said to correspond to a x_i value. It can be easily seen that the size of each MPLS packet payload is $m \bullet L / k$ or in other words is equal to the size of an IP packet divided by k.

For example, for block S_1, the equation generated is

$$f(S_1, x) = (8x^2 + 28x + 6) \bmod 257$$

Figure 4.2 shows the equation generated for S_2.

The above equation is solved for different n values as follows:

$f(S_{j=1}, x = 1) = 42$ $\quad$ $f(S_{j=2}, x = 1) = 116$

$f(S_{j=1}, x = 2) = 94$ $\quad$ $f(S_{j=2}, x = 2) = 112$

$f(S_{j=1}, x = 3) = 162$ $\quad$ $f(S_{j=2}, x = 3) = 256$

$f(S_{j=1}, x = 4) = 246$ $\quad$ $f(S_{j=2}, x = 4) = 34$

In Figure 4.2, each of the four columns represents an MPLS packet share (i.e., MPLS packet 1, MPLS packet 2, MPLS packet 3, and MPLS packet 4). As seen from the figure, each MPLS packet share size is equal to 1/3 of the total original IP packet size. For a (3, 4) modified TSS, any three of these MPLS packet shares can be used to reconstruct the original IP packet. Therefore, since we have four MPLS packet shares, the data redundancy due to coding in this example is equal to 1/3 the size of original IP packet. The details of redundancy in general for modified (k, n) TSS scheme is discussed in Section 4.3.

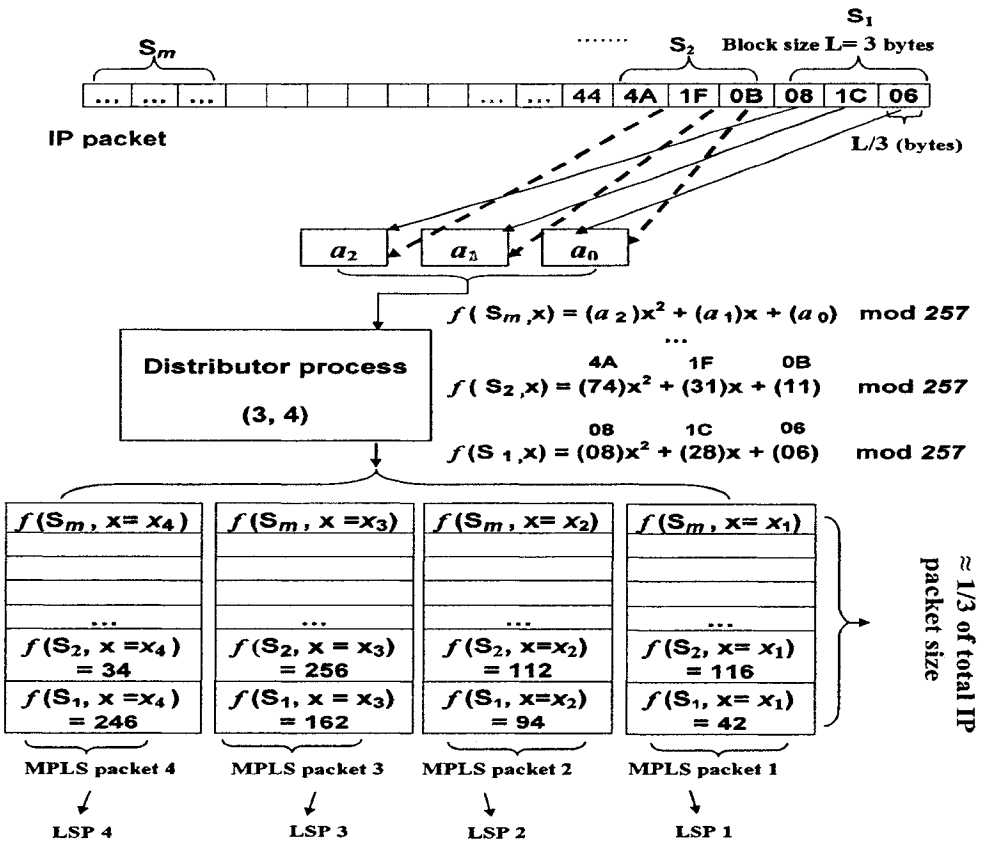


Figure 4.2 Distribution Process in the Ingress Router applying a (3, 4) modified TSS

Each block size used in Figure 4.2 consists of three bytes and therefore each coefficient size is equal to one byte or 8 bits. However, the value obtained for the share $f(S_{j=2}, x = 3)$ is equal to 256. This means that the number should be represented with more than 8 bits. Therefore, it is worth to note that in our implementation the operations are simplified for finite fields of order 2^w, also called GF (2^w) Galois fields [90] (where w is an integer value) or binary fields, making these fields appropriate for the proposed TSS

application. In other words, this will result in having share values that can be represented with an equal size of coefficient, i.e., 8 bits.

One way to construct GF (2^w) is to use a *polynomial basis representation* [90]. Here, the elements of GF (2^w) are the binary polynomials (polynomials whose coefficients are in the field GF (2) = {0, 1}) of degree at most $w-1$.

$$\text{GF}(2^w) = a_{w-1}\, z^{w-1} + a_{w-2}\, z^{w-2} + \ldots + a_2 z^2 + a_1 z + a_0 : a_i \in \{0, 1\}$$

An irreducible binary polynomial $f(z)$ of degree w is chosen (such a polynomial exists for any w and can be efficiently found). Irreducibility of $f(z)$ means that $f(z)$ cannot be factored as a product of binary polynomials each of degree less than w. Addition of field elements is the usual addition of polynomials, with coefficient arithmetic performed modulo 2. Multiplication of field elements is performed modulo the *reduction polynomial* $f(z)$. For any binary polynomial $a(z)$, $a(z)$ mod $f(z)$ shall denote the unique remainder polynomial $r(z)$ of degree less than w obtained upon long division of $a(z)$ by $f(z)$; this operation is called *reduction modulo* $f(z)$ [90].

The following are some examples of arithmetic operations in GF (2^8) with reduction polynomial $f(z) = x^8 + x^4 + x^3 + x + 1$. Here we represent each *byte* of the IP packet by a characteristic 2 finite field with 8 terms. In other words, this 8 terms characteristic 2 finite field represents the same concept of the prime number.

Addition: {0x CA in hex} + {0x 53 in hex} = $(x^7 + x^6 + x^3 + x) + (x^6 + x^4 + x + 1) = x^7 + x^4 + x^3 + 1$ = {0x 99 in hex} = {10011001} (in binary).

Subtraction : {0x CA in hex} - {0x 53 in hex} = $(x^7 + x^6 + x^3 + x) - (x^6 + x^4 + x + 1)$ = {0x 99 in hex}. (As explained before, since the operations are done in characteristic 2 finite field, the addition and subtraction give the same value).

Multiplication: {0x 53in hex} • {0x CA in hex} = $(x^6 + x^4 + x + 1)(x^7 + x^6 + x^3 + x)$

$$= x^{13} + x^{12} + x^9 + \mathbf{x^7} + x^{11} + x^{10} + \mathbf{x^7} + x^5$$

$$+ x^8 + \mathbf{x^7} + x^4 + x^2 + \mathbf{x^7} + x^6 + x^3 + x$$

$$= x^{13} + x^{12} + x^9 + x^{11} + x^{10} + x^5 + x^8 + x^4$$

$$+ x^2 + x^6 + x^3 + x$$

$$= x^{13} + x^{12} + x^{11} + x^{10} + x^9 + x^8 + x^6 + x^5$$

$$+ x^4 + x^3 + x^2 + x$$

Therefore,

$x^{13} + x^{12} + x^{11} + x^{10} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + x^2 + x$ modulo $x^8 + x^4 + x^3 + x + 1$ = (11111101111110 mod 100011011) = {01} where the operation can be done through long division, noticing that the XOR is applied in the example and not arithmetic subtraction.

The complexity of the distribution process is deduced from the explanation above; it is expressed in terms of the original packet size, the size of the blocks to be used, and the number of LSPs over which the resulting MPLS packets are sent. More precisely, if *a* is

the size of the original IP packet coming into the ingress router, b is the size of the blocks resulting from the division of the IP packet, and c is the number of LSPs used between the ingress and egress routers then the complexity of the distribution process is $O(\frac{a}{b} \times c)$.

4.1.2 Reconstruction Process

Figure 4.3 shows the reconstruction process of the IP packet from any k of the n MPLS packets received from the network of Figure 4.2. In the figure, MPLS packets received from the LSP2, LSP3 and LSP4 are considered. Since both ingress and egress routers use the same polynomial function, the order of coefficients a_0, a_1, ..., a_{k-1} are already preserved and does not depend on the location of shares in a group of LSPs, as can be seen in the following reconstruction process.

The following three equations are used to obtain the function $f(S_1, x)$ using Langrage Interpolation:

$$(a_0 + a_1(2) + a_2(2)^2) \equiv 94 \pmod{257} \text{ ; from LSP2}$$

$$(a_0 + a_1(3) + a_2(3)^2) \equiv 162 \pmod{257} \text{ ; from LSP3}$$

$$(a_0 + a_1(4) + a_2(4)^2) \equiv 246 \pmod{257} \text{ ; from LSP4}$$

Using Equation (3.3), the following is obtained:

$$f(S_1, x) = 8x^2 + 28x + 6 \mod 257$$

where the original values of the coefficients for block S_1 obtained are $a_2 = 8$ (0x08 in hex), $a_1 = 28$ (0x1C in hex), and $a_0 = 6$ (0x06 in hex)

Similarly, the following three equations are used to obtain the function $f(S_2, x)$ using Langrage Interpolation:

$$(a_0 + a_1(2) + a_2(2)^2) \equiv 112 \pmod{257} \text{ ; from LSP2}$$

$$(a_0 + a_1(3) + a_2(3)^2) \equiv 256 \pmod{257} \text{ ; from LSP3}$$

$$(a_0 + a_1(1) + a_2(1)^2) \equiv 34 \pmod{257} \text{ ; from LSP4}$$

Using Equation (3.3), the following is obtained:

$$f(S_2, x) = 74x^2 + 31x + 11 \bmod 257$$

where the original values of the coefficients for block S_2 obtained are $a_2 = 74$ (0x4A in hex), $a_1 = 31$ (0x1F in hex), and $a_0 = 11$ (0x0B in hex).

For the same reason mention before in the distribution process, the operations are simplified for finite fields of order 2^w, also called GF (2^w) Galois fields.

Similar to the distribution process, the complexity of the reconstruction process can be easily derived from the previous explanation. It is expressed in terms of the number of MPLS packets required to reconstruct the original IP packet, the number of blocks used, and the complexity of the Lagrange linear interpolation. More precisely, if a is the number of MPLS packets required and b is the number of blocks used then the complexity of the reconstruction process is $O(b \times a^3)$, where the complexity of the Lagrange linear interpolation is $O(a^3)$ according to [89].

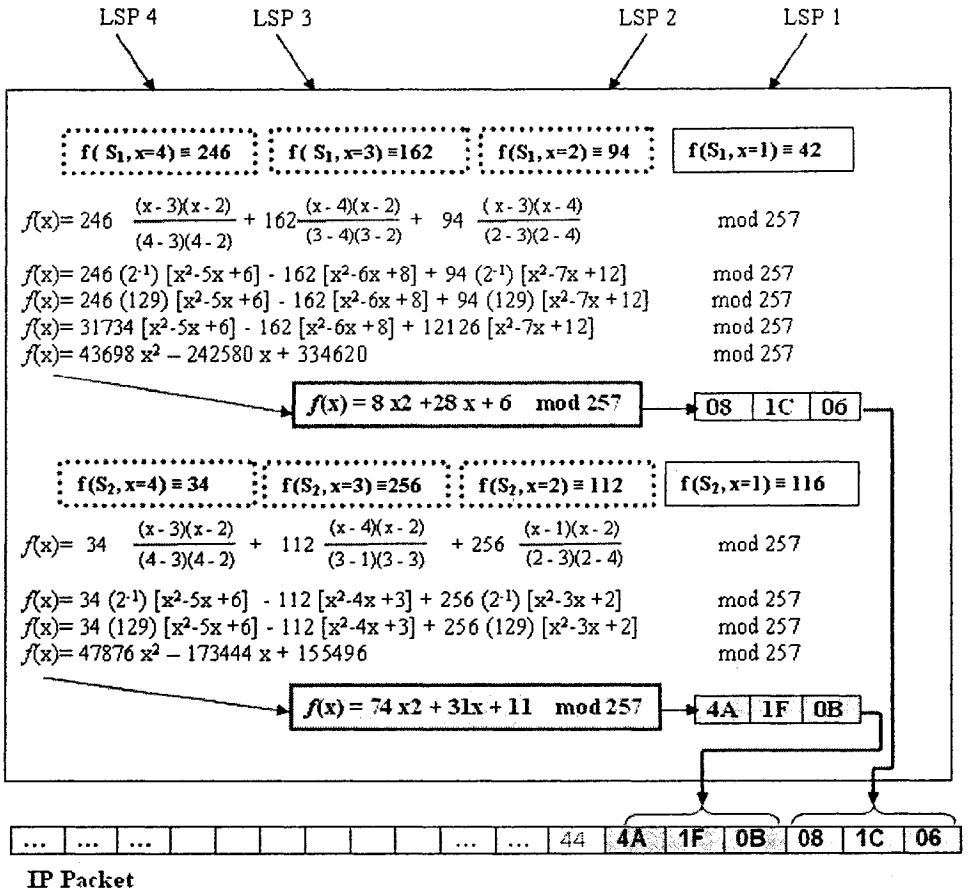


Figure 4.3 Reconstruction Process in the Egress Router applying a (3, 4) modified TSS

The processing time that is needed for the application of modified threshold sharing scheme on packet transmission is discussed in the next section.

4.2 Processing Time Measurements

This section illustrates the time required by the modified Threshold Sharing scheme (TSS) when processing packets at ingress and egress routers. We want to show that applying TSS application does not significantly affect the total time needed to send and receive a packet between two routers (ingress and egress).

The measurements were performed on 3 GHz Intel Pentium 4 CPU processor with 1 GB memory running under Linux operating system to measure the time taken by our modified TSS scheme to distribute the original IP packet to n MPLS packets and reconstruct it at the ingress and egress routers respectively.

The implementation considers variable packet sizes. The packet is basically a text file that is fed to the TSS application program which is written in C program, where we used the MPZ functions under GNU Multiple Precision (GMP) Arithmetic Library. The CPU time is measured for each packet that is processed by the modified (k, n) TSS application.

In our implementation the Galois binary field GF (2^w) is used. Indeed, the field arithmetic GF (2^w) is implemented on the base of irreducible polynomials of different degrees w. Therefore, the degree value of each irreducible polynomial depends on the block size selected.

The average packet processing time (in μsec) for different IP packet sizes obtained for the distribution process of a (k =3, n = 3) TSS at the ingress router is shown in Figure 4.4. Different block sizes L for the same IP packet size affects the average packet processing time in the distribution process. For larger block sizes, packet processing takes more time but is still of the order of μsec. It was observed during simulation that the packet processing time for reconstruction is similar to that required by the distribution process. It is also clear from the figure that the packet processing time (i.e., in microseconds) is much lower than the total packet transmission time (i.e., in milliseconds) and can thus be neglected.

The Maximum Transmission Unit (MTU) on a path across an MPLS network is equal to 1500 bytes [56]. For that reason, we give examples of up to 8 Kbytes IP packets for simulation purpose only, but in reality IP packets entering MPLS network should not exceed MTU size, otherwise it has to be fragmented.

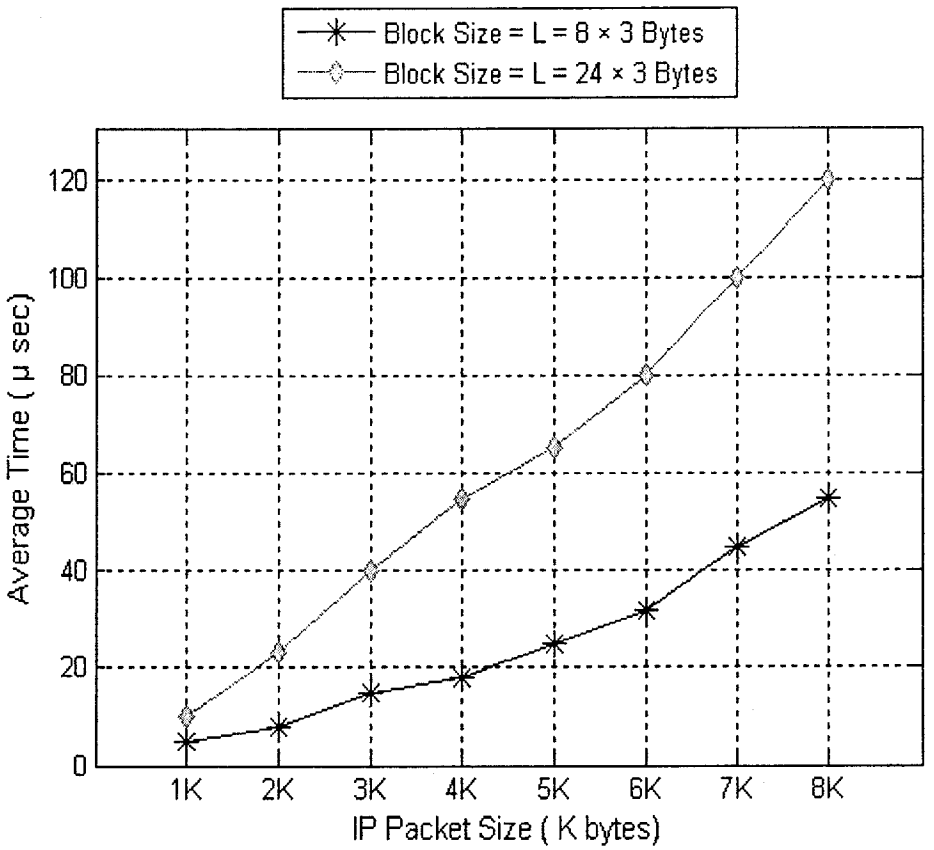


Figure 4.4 Average packet processing time for Distribution process using a modified (3, 3) TSS

The effect of variable (k, k) level, i.e., $n = k$, on IP packet processing time is shown in Figure 4.5 for a IP packet size of 2Kbytes and a block size of (24 x k)bytes. It is worth to note that when (k, k) modified TSS is used, this means that the value of n is equal to the value of k, there are no redundant shares required and as a result there is no redundant

bandwidth required. We conclude from the results that variable (k, k) level has no significant effect on the IP packet processing time. The processing time results obtained are within 10s of microseconds. Therefore, comparing to the total packet transmission time which is in order of 10s milliseconds, our method does not impose a significant impact.

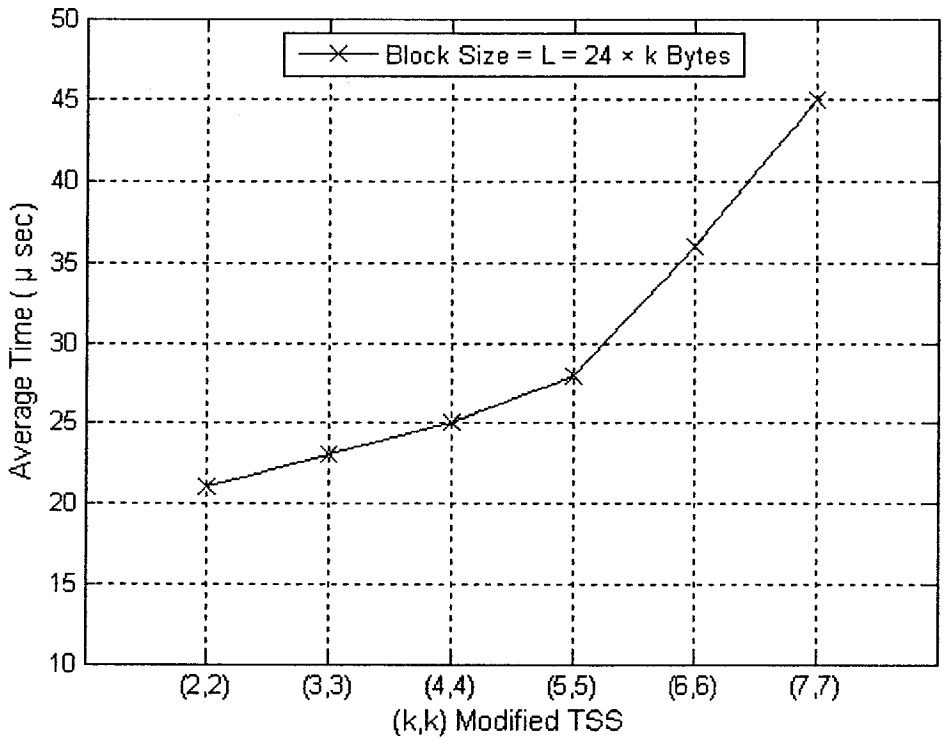


Figure 4.5 The effect of variable (k, k) modified TSS level on packet processing time

We also compared our method with the VSR model [57] that is used for survivable storage systems and it uses the threshold sharing scheme for its implementation. The author used MIRACL package for arithmetic integer precision. The scheme targets to distribute data storage in multiple servers to preserve it for long time. The VSR model is designed for data storage and was not intended for networking aspects. We compared the

processing time of our modified TSS scheme with VSR model as shown in Figure 4.6. Our implementation improves processing time significantly because we performed our modified TSS in smaller block sizes as it was indicated in by Figure 4.4, where in VSR model the whole packet is considered as one block. Our implementation was able to perform the job with much smaller block size to be able to reduce the packet processing time. Our scheme is targeting networking application which is sensitive to the time issue.

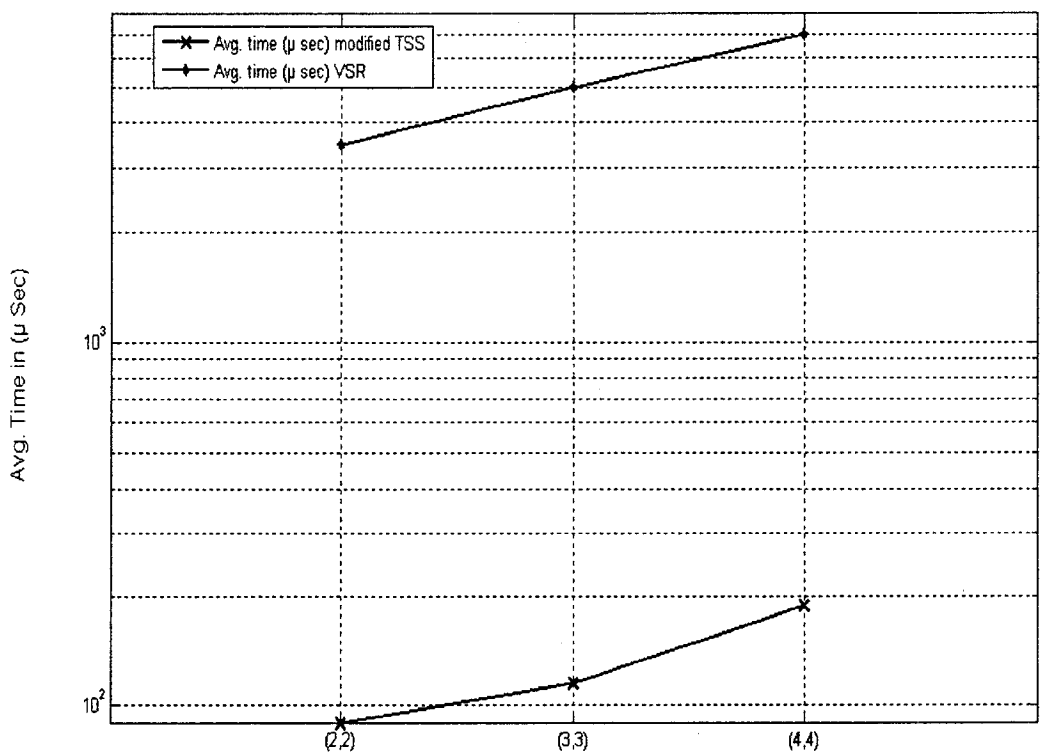


Figure 4.6 Comparison between Modified TSS and VSR model for packet size equal to 8K bytes

We also compared our work as shown in Figure 4.7 with the encryption technique used in SSL (Secure Socket Layer) application [58]. The authors discussed the impact of applying encryption on packet processing time. The purpose of this comparison is to

show that when it comes to networking environment, any security method should not add a significant processing time to the total transmission time. Moreover, from the results obtained by our implementation for the threshold sharing scheme it is noticed that the processing time is acceptable compared to available encryption techniques.

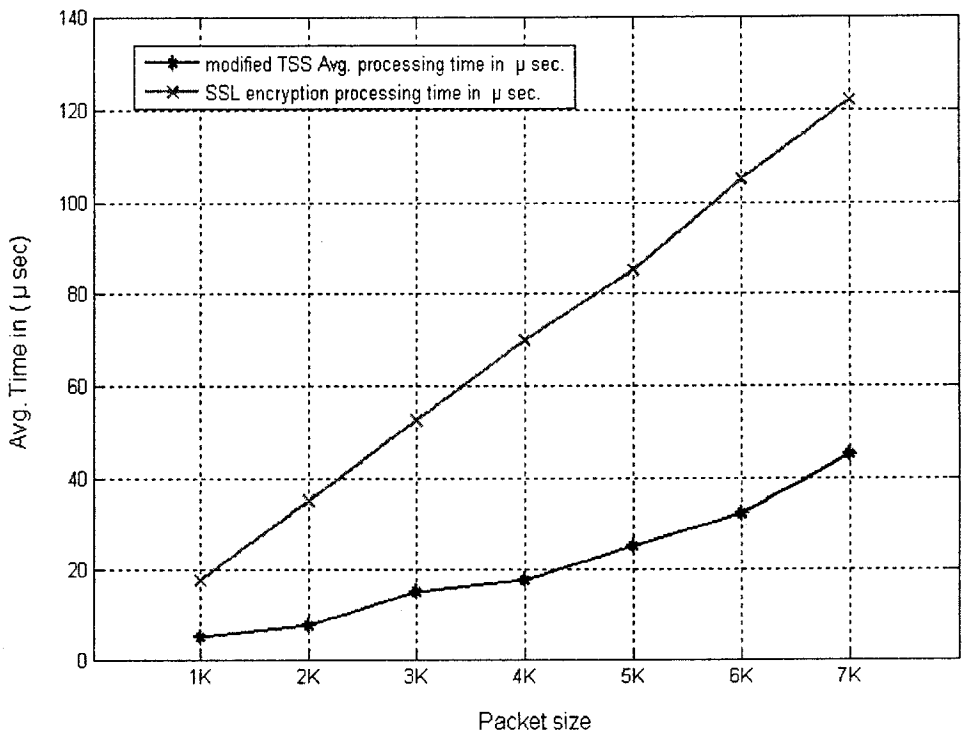


Figure 4.7 A comparison between modified (k, k) TSS where (k = 3) and the encryption application used in SSL

4.3 Network Resource Utilization

In this discussion, we compare between our modified TSS and the original TSS scheme in terms of network resource utilization. MPLS header size (equals 4 bytes) is assumed to be neglected compared to the total MPLS packet size.

In the original TSS scheme, the secret message is the only value supplied to the polynomial coefficients (to coefficient a_0) where other coefficients values are selected randomly. Assuming an IP packet of size P_{size}= Q bytes, and using the original (k, k) Threshold Secret Sharing scheme. The total size of k MPLS packets is:

$$\text{Total } k \text{ packets size } T_{originalTSS(k)} = k \times \text{Q bytes} \tag{4.1}$$

The original (k, k) TSS generates k shares of size Q bytes each. This is because the coefficients a_1 and a_2 were selected to have random values. Figure 4.8 (a) demonstrates that each generated share size, using for example the original (k = 3, n = 3) TSS, is equal to Q bytes which is the same size as the original IP packet size. Using a modified (k, k) TSS, since all coefficients' values of the polynomial are obtained from the original IP packet, it results in having the total packets size of k shares/MPLS packets equal to:

$$\text{Total } k \text{ packets size } T_{modifiedTSS(k)} = P_{\text{size}} = Q \text{ bytes} \tag{4.2}$$

Figure 4.8 (b) illustrates the result obtained by equation (4.2). The original IP packet is coded and divided using a modified (3, 3) TSS into three shares. Each share value in this example is equal to Q/3 bytes. In other words, each MPLS packet share size is equal to (Q / k) bytes.

Now, for a (k, n) modified TSS level, the total size for n MPLS packets for a Q bytes IP packet in the modified TSS is:

$$\text{Total } n \text{ packets size } T_{modifiedTSS(n)} \approx \frac{n}{k} \times Q \quad bytes \tag{4.3}$$

and,

$$\text{Redundant bandwidth required} \approx \frac{(n-k)}{k} \times Q \quad bytes \tag{4.4}$$

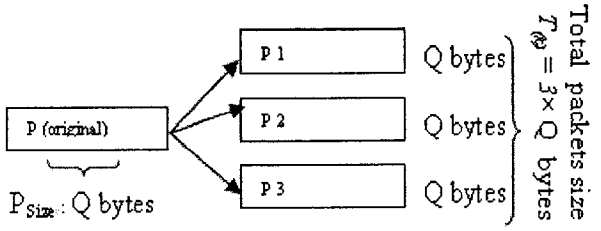


Figure 4.8 (a) Original TSS of (3, 3) security level

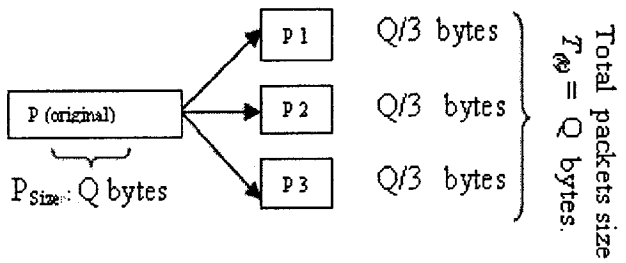


Figure 4.8 (b) Modified TSS of (3, 3) security level

From a fault tolerance point of view, we can conclude from equation (4.4) that the redundant bandwidth required to protect one single failure is only equal $\left(\frac{1}{k}\right)^{th}$ of the original IP packet size. This requires a (k, n) modified TSS of $n = k + 1$.

Assuming a (3, 4) modified Threshold Sharing scheme used and an IP packet size of 3Kbytes entering the MPLS network. Referring to Figure 4.8, the distribution process generates four MPLS packets, each of size ≈ 1 Kbytes (i.e., 1024 bytes payload + 4 bytes MPLS header + 4 bytes for MPLS extensions). The total size of four MPLS packets

created is therefore ≈ 4Kbytes. The redundant bandwidth for one extra share for a (3, 4) TSS is 1/3 of the total original bandwidth. On the other hand, using the original TSS method, the total size of four MPLS packets will be ≈ 4 × 3 = 12Kbytes

Now, from security point of view, data confidentiality can be the only security level required that can be achieved using a (k, k) modified TSS. This means that no additional bandwidth is required (except for the additional header overhead for k shares) as can be concluded from Figure 4.8 (b). Detailed discussion on bandwidth utilization in modified TSS is discussed while covering security and fault tolerance deployment in the next chapter.

In the following sections, we discuss some issues related to the application of multi-path routing. In other words, the application of multi-path approach in MPLS security and fault tolerance requires some additional considerations. These considerations include packets sequencing to identify packets disordering due to transmission errors, and considering multiple classes of quality of service.

4.4 MPLS Packets Sequencing

As explained earlier, each IP packet creates n MPLS packets; each MPLS packet is sent over an LSP. MPLS packets generated are in the same order as the IP packet received by the ingress router, and therefore the MPLS packets at each LSP will also be received in the same order if there is no MPLS packet lost, for example packets not received in sequence due to transmission errors, long queuing delays. Figure 4.9 illustrates an example for a (3, 3) modified TSS algorithm. It is clearly noticed that if MPLS share 3 that belongs to IP packet 2 is assumed to be lost (e. g., due to transmission

error), then group 2 in this case is going to include MPLS share 3 of IP packet 3. That means if this share is to be included in the reconstruction process, then the obtained original IP packet 2 is not correct. Therefore, we need a mechanism to preserve the ordering of the shares received at the egress router. To identify packets lost due to transmission errors, a sequence number can be used.

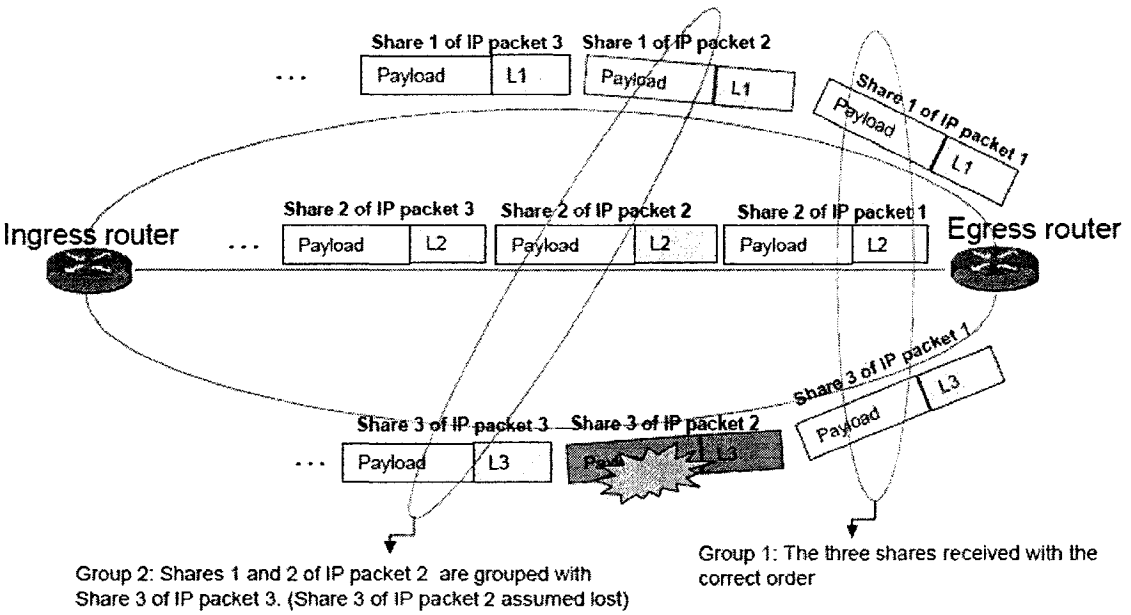


Figure 4.9 Packet loss example for a (3, 3) TSS

We adopted the work of [64] to use sequence numbering for supporting packets ordering for MPLS packet shares received from multiple LSPs at the egress node. However, there are some challenges and issues that should be considered. In MPLS, the shim header length is only four bytes long. The header format does not provide space for the packet sequence number. Therefore, a Control Word (CW) can be added to the MPLS

packet share payload. This requires the CW to be carried as the first four bytes of the MPLS share payload. The format of the control code word is shown in Figure 4.10.

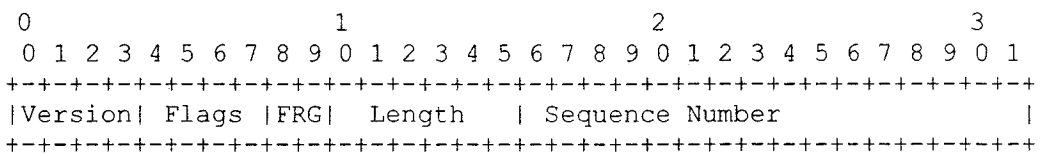


Figure 4.10 Control Word (CW) format [64]

Flags (bits 4 to 7): These bits may be used for per-payload signaling.

FRG (bits 8 and 9): These bits are used when fragmenting MPLS packet share payload.

Length (bits 10 to 15): When the path between ingress and egress nodes includes an Ethernet segment, the MPLS packet shares arriving network edges may include padding appended by the Ethernet Data Link Layer. The length field is used to determine the size of the padding added by the MPLS network, and hence extract the payload required for reconstruction.

Sequence number (Bit 16 to 31): used for MPLS packet shares ordering.

The use of the version field in the CW is summarized as follows: All IP packets start with a version number that is checked by LSRs performing MPLS payload inspection [64]. Therefore, to prevent the incorrect processing of packets, MPLS packet payload must not start with the value 4 (IPv4) or the value 6 (IPv6) in the first nibble, as those are assumed to carry normal IP payloads. Indeed, in our proposed scheme, the payload of a MPLS packet share is not an IP packet. In other words, it is an encoded payload produced

by the distribution process at the ingress node. For that reason, to avoid having the previous values in the beginning of MPLS packet share payload, the version field has to be given a different value.

On the other hand, the egress node is responsible for checking the content of the CW in the MPLS payload, and accordingly uses the sequence number value to synchronize MPLS packet shares. The location of CW in MPLS packet is shown in Figure 4.11.

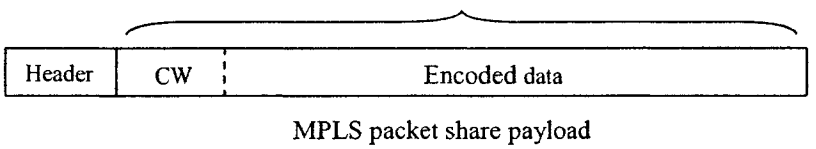


Figure 4.11 CW as part of MPLS packet payload

The following procedure must be used for setting the sequence number:

- The initial MPLS packet shares allocated to multiple LSPs must be sent with the sequence number one.
- Subsequent MPLS packet shares must increment the sequence by one.
- The sequence number that follows 65535 (Maximum unsigned 16-bit number) is set to one.

Packets that are received out of order may be buffered until shares with same sequence number are received on other LSPs. Buffering requirements at the egress node will be discussed in Chapter 7.

Supporting packet ordering requires extra bytes to be added. The R_{CW} represents the effect of adding the CW bytes on the size of MPLS packet share which is given by the following ratio:

$$R_{CW} = \frac{S_{CW}}{\frac{P}{k} + H + S_{CW}} \tag{4.5}$$

where:

S_{CW} : indicates the size of control word, 4 bytes

P: Original IP packet size (in bytes)

H: MPLS header size, 4 bytes

Indeed, the control word is considered an extra overhead in addition to the MPLS header. The effect of this header overhead is more noticeable on small IP packets.

4.5 Considering multiple classes of Quality of Service

Our approach requires the use of the reserved bandwidth of all n LSPs that are part of the (k, n) modified TSS. This means that even if there is no failure, the redundant bandwidth can not be used to serve other traffic. Therefore, this is a drawback in our approach. In order to make our approach more practical, we can use a nice feature in MPLS networks where different treatments can be applied for different traffic by the use of multiple classes of services. To clarify this point, consider the example shown in Figure 4.12. There are two types of traffic traversing through the network. The first type is a high priority traffic demonstrated by C1 and C2 which are allocated into three

disjoint LSPs following a (2, 3) TSS scheme. In our approach, high priority traffic does not tolerate recovery delay or packet loss, and therefore can not be pre-empted. The second type is a low priority traffic demonstrated by C3. This type has no stringent traffic requirements such as recovery delay or packet loss, and accordingly can be pre-empted. In other words, the amount of traffic dedicated for C3 on LSP 3 can be pre-empted to serve another high priority, i.e., traffic C4, which can be a part of another network traffic connection. In this case, C3 is reconstructed at the egress node from only two LSPs. As a result, there will be no protection provided for traffic C3 if link/node failure occurs on the two LSPs. Therefore, to recover from failure of traffic C3, fast protection techniques can be used. In summary, our approach may only apply (k, k) modified TSS, where $k = n$, for traffics that can tolerate recovery delay and packet loss.

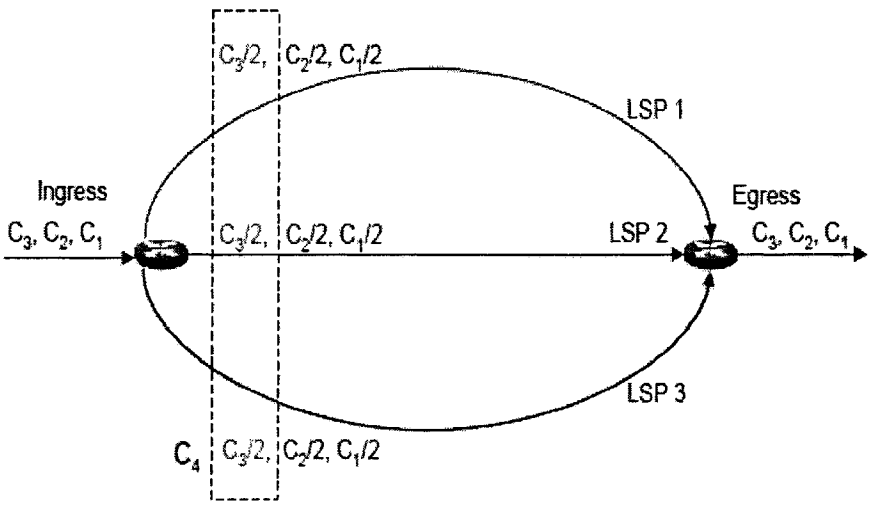


Figure 4.12 Example on traffic protection with/without pre-emption

4.6 RSVP-TE Signaling Protocol for Multi-path Routing

In order for LSPs to be used, the forwarding tables at each LSR must be populated with the mappings from {incoming interface, label value} to {outgoing interface, label value}. This process is called LSP setup, or labels distribution. The RSVP-TE and LDP-CR are main signaling protocols used for label distribution in MPLS networks. These protocols only consider point-to-point signaling between edge routers.

Currently, extension for RSVP-TE and LDP to support Point-to-Multipoint (P2MP) is still work in progress [67]. An important application for P2MP is multicasting. However, there is no work made so far for RSVP-TE or LDP signaling to support multi-path routing in MPLS. There is a slight difference between P2MP and multi-path routing. In multi-path routing there is one sender and one receiver wherein P2MP there is one sender and multiple receivers. Moreover, non-edge routers in P2MP signaling may be required to duplicate packets. In this part of the thesis, we discuss extensions for RSVP-TE signaling protocol for path and reservation messages to support multi-path routing in MPLS networks, which satisfy the set of requirements described in [4, 68].

For a Multi-Path LSP (MP-LSP) signaling, each path or tunnel is now identified by a MP-LSP Session object. This object contains the identifier of the MP-LSP Session, which includes the MP-LSP Identifier (MP-LSP ID), a tunnel Identifier (Tunnel ID), and an extended tunnel identifier (Extended Tunnel ID). The MP-LSP ID is a four octet number and is unique within the scope of the ingress router.

The difference between the specifications proposed for MP-LSP and P2MP is that in MP-LSP signaling we only focus in a multiple LSP connection which consists of one ingress router and one egress router.

According to [68], the ingress router may initiate one path message or multiple path messages to support P2MP signaling. The use of multiple path messages maybe required if the path message may not be large enough to contain all the multiple LSPs. The use of one path message maybe feasible for P2MP signaling, however, for a MP-LSP approach we believe multi-path messages are needed. Explanations to this requirement for MP-LSP signaling can be summarized as follows:

- In MP-LSP routing, the modified TSS scheme requires that multiple LSPs per connection to be disjoint. As a result, the multiple disjoint LSPs are initiated from the ingress router toward the egress router.
- It is more convenient to use multiple path messages in MP-LSP signaling as these path messages contains the upstream hops traversed by the path message. This information is located in Record Route Object (RRO) which is part of the sender descriptor object in the path message.

- In MP-LSP, the LSRs are not required to process additional tasks such as label merging. However, label merging maybe considered for maximally disjoint LSPs as nodes are shared.

The path messages are sent to the egress node where each message contains a different Explicit Route object. The path message format should be modified to include MP-LSP signaling.

The MP-LSP is identified by the combination of the MP-LSP destination address, MP-LSP ID, and Extended Tunnel ID that are part of the MP-LSP Session object, and the tunnel sender address and LSP ID of the MP-LSP Sender Template object. The Session

object has a similar structure as the existing point-to-point RSVP-TE Session object [4]. The destination address is set to the MP-LSP ID instead of the unicast tunnel endpoint address. All LSPs that are part of the same MP-LSP share the same Session object. This Session object identifies the MP-LSP tunnel ID.

The combination of the Session object, the Sender Template object identifies each path of the MP-LSP. This follows the existing P2P RSVP-TE notion of using the Session object for identifying a P2P Tunnel, which in turn contain multiple LSPs, each distinguished by a unique Sender Template object.

The new MP-LSP Session and Sender Template objects are defined in Figures 4.13 and 4.14. The specifications are related to IPv4 protocol.

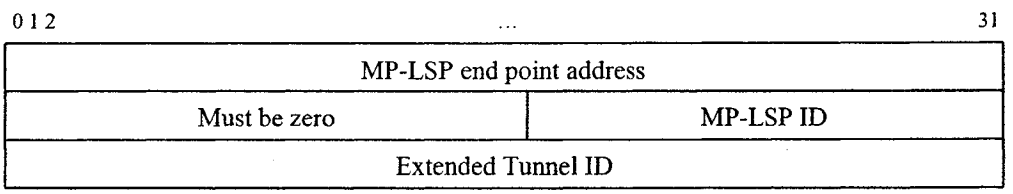


Figure 4.13 Extended MP-LSP Session object Format

MP-LSP end point address: IPv4 address of the egress node for the MP-LSP tunnel.

MP-LSP ID: A 16-bit identifier used in the Session object that remains constant over the life of the MP-LSP tunnel.

Extended Tunnel ID: A 32-bit identifier used in the Session object that remains constant over the life of the MP-LSP tunnel. Ingress LSRs that wish to have a globally unique identifier for the MP-LSP tunnel should place their tunnel sender address here. A

combination of this address, MP-LSP ID, and Tunnel ID provides a globally unique identifier for the MP-LSP tunnel.

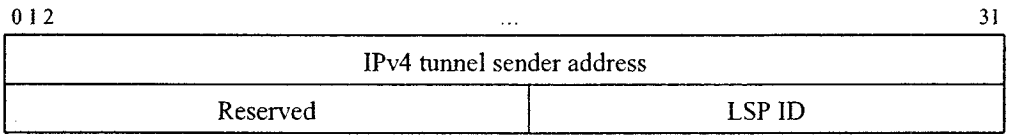


Figure 4.14 Extended MP-LSP Sender Template object Format

IPv4 tunnel sender address: IPv4 address for sender node.

LSP ID: A 16-bit identifier used in the Sender Template and the Filter_Spec that can be changed to allow a sender to share resources with itself. Thus, multiple instances of the MP-LSP tunnel can be created, each with a different LSP ID. The instances can share resources with each other. The S2L sub-LSPs corresponding to a particular instance use the same LSP ID.

The reservation style for the MP-LSP is similar to the one in [4]. However, MP-LSP extensions described above has to be considered. All MP-LSP paths that belong to the same MP-LSP tunnel must be signaled with the same reservation style.

4.7 Summary

In this chapter we discussed our proposed approach to provide MPLS fault tolerance and security. The modification of the original TSS scheme is illustrated with examples. To verify that our approach does not require long processing time, we conducted simulations which show that modified TSS processing time does not affect significantly

the packet transmission time. We also show how our scheme can better utilize the network resources.

It should be mentioned that identifying packets with transmission error is considered in our approach, however, packet re-ordering may not be required if it is caused by failure. We also discussed issues that may be raised when supporting multi-path routing signaling for RSVP-TE.

Chapter 5

Deploying Modified TSS to Provide Fault Tolerance and Security

In this chapter, we explain how the modified TSS can be applied in MPLS networks to provide fault tolerance, and security (i.e., data confidentiality and integrity, availability, and IP spoofing). Moreover, this chapter discusses the case of maximally disjoint LSPs where the n paths in a (k, n) modified TSS application are not disjoint, i.e., neither node-disjoint nor link-disjoint. Moreover, a comparison between the modified TSS and IPSec security protocol is presented. In addition, the buffering requirement limitation in our approach is discussed.

5.1 Deploying modified TSS to Provide MPLS Fault Tolerance

The basic idea in our approach uses the threshold sharing scheme combined with multi-path LSP routing in order to provide fault tolerance in MPLS network. Our proposal for fault tolerance can easily handle single or multiple path failures.

In order to do so, an IP packet entering MPLS network is partitioned into n MPLS packets, which are assigned to disjoint or maximally disjoint LSPs across the MPLS

network. Receiving MPLS packets from k out of n LSPs are sufficient to reconstruct the original IP packet.

In this thesis, we focus on the following fault tolerance factors which have been introduced earlier in Section 2.2.2:

- Network resource utilization
- Failure recovery time
- Packet loss
- Packets out-of-order

The above factors are discussed and elaborated throughout Theorems 5.1 – 5.4.

Property 5.1:

By the application of the modified (k, n) TSS algorithm, we provide fault tolerance for single/multiple LSPs failures with an approximate redundant bandwidth which is equal to $\approx \frac{(n-k)}{k} \times Q$ *bytes, where Q is the size of IP packet in bytes. Single failure protection requires n = k+1 paths, and multiple LSP failure protection requires n > k +1 paths to handle n – k failures. This property is valid under the assumption of being able to find n node-disjoint LSPs between a source and a destination.*

Explanation of property 5.1 can be obtained from equation (4.4) in the discussion of bandwidth utilization of the modified TSS scheme in Section 4.3. The property 5.1 can be explained as follows:

- The ideal bandwidth "I" is equal to Q bytes.
- The bandwidth used for each share is Q/k, therefore the total bandwidth "T" used for the n shares is equal to $n \times Q/k$.
- The redundant bandwidth is simply the difference between T and I.

Now, to show the advantages and limitations of bandwidth utilization in our approach for fault tolerance, the modified TSS is compared with 1: N and 1+1 models as shown in Section 5.1.1 and 5.1.2.

5.1.1 Compared to (1: N) Shared Protection Scheme

Figures 5.1 and 5.2 show an example of bandwidth utilization in (1:3) shared protection scheme and our method, respectively. Consider Figure 5.1 where the ingress router in the MPLS network receives three different IP traffic Φ_1, Φ_2, and Φ_3. For each working path W_1, W_2, W_3 the bandwidth required for Φ_1, Φ_2, and Φ_3 is reserved respectively. The backup path, which protects the three disjoint working paths and handles single failure at any time of any of the three working paths, is also disjoint and reserves bandwidth required for the largest traffic value as given by equation 5.1.

$$\textit{Extra Bandwidth for } (1\text{: N} = 3) \textit{ scheme} = \text{Max } (\Phi_1, \Phi_2, \Phi_3) \qquad (5.1)$$

On the other hand, our method presented in Figure 5.2 handles the same amount of traffic received differently. The ingress router encodes each of the three traffic into four shares allocated to four disjoint LSPs W_1, W_2, W_3, and W_e. Each LSP therefore needs a bandwidth of Average (Φ_1, Φ_2, Φ_3). The extra bandwidth needed in our method is represented by the W_e LSPs which amounts to the following:

Extra Bandwidth for (k = 3, n = 4) TSS scheme ≈ Average (Φ_1, $\Phi 2$, Φ_3) (5.2)

which is less than Equation (5.1) for a *(1: N = 3)* scheme.

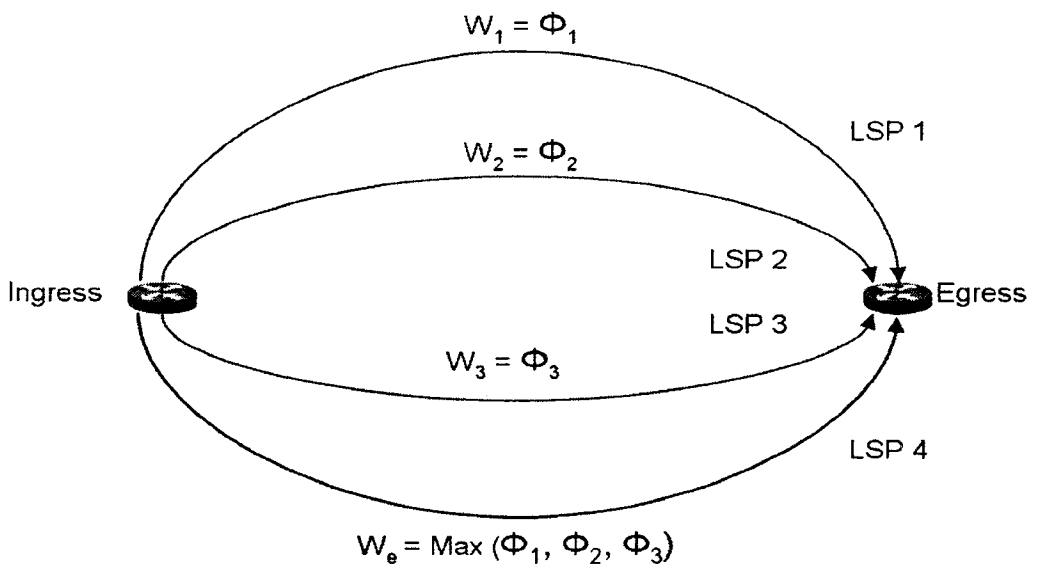


Figure 5.1 Shared Path Protection scheme: three working traffic sharing one backup path

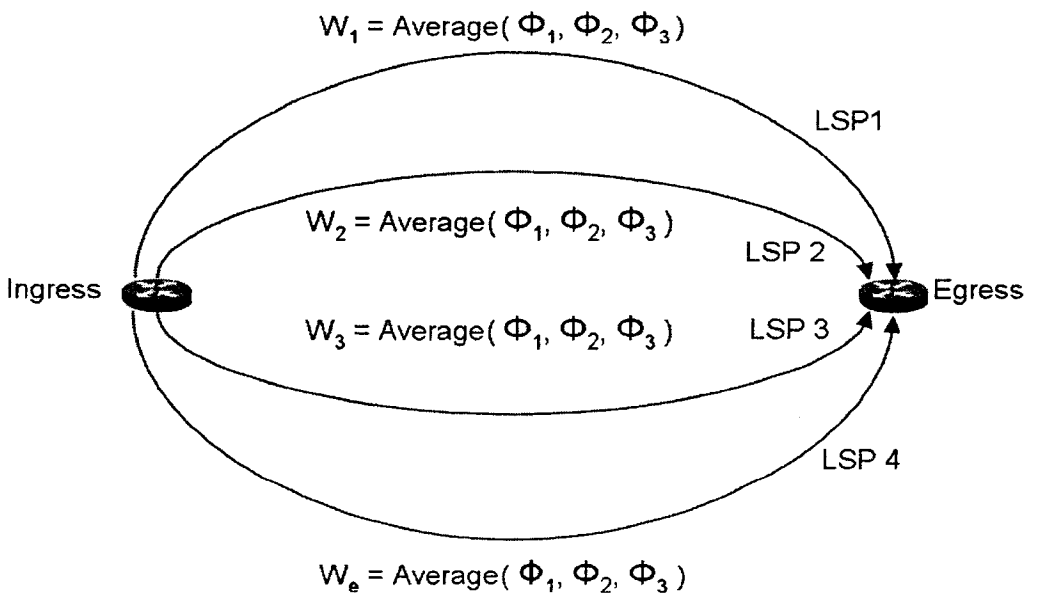


Figure 5.2 Proposed (*k, n*) TSS: three working traffics encoded into four equal shares

5.1.2 Compared to (1+1) Protection Scheme

In the 1+1 protection, the backup path carries the same traffic as the working path, requiring 100% of dedicated redundant bandwidth reservation for each working traffic.

$$\textit{Extra Bandwidth for (1+1) scheme} = \text{Sum}\ (\Phi_1, \Phi_2, \Phi_3) \tag{5.3}$$

Table 5.1 compares redundant bandwidth required for different traffic sizes for (1: N = 3), (1+1), and our modified *(k, n) TSS* approach. It is clear that our approach requires ≈ 33.3 % (≈1/3) of redundant bandwidth for a (3, 4) scheme, and is not affected by the variation of different traffic values. It also performs better than the other approaches where redundant bandwidth depends on the different traffic values.

(Φ_1, Φ_2, Φ_3) ***Traffic sizes***	***Redundant Bandwidth for***		
	(1: 3)	**(1+1)**	**(3,4) TSS**
6, 6, 6	6 or 33.3 %	18 or 100 %	6 or 33.3 %
6, 6, 9	9 or 43 %	21 or 100 %	7 or 33.3 %
6, 3, 9	9 or 50 %	18 or 100 %	6 or 33.3 %
3, 3, 12	12 or 66.6 %	18 or 100 %	6 or 33.3 %

Table 5.1 Redundant bandwidth required for (1: 3), (1+1) and our (3, 4) TSS approach

On the other hand, there are some limitations for the use of our approach compared to 1+1 and 1: N protection techniques. These limitations are described below.

1) Compared to 1: N shared protection:

   The (k, n) modified TSS approach uses all the bandwidth reserved for all n LSPs. Therefore, redundant bandwidth is used even if there is no failure. However, in the 1: N shared protection, the bandwidth reserved for the backup LSP can be used to serve low priority traffic.

2) Compared to 1+1 protection:

   In our approach we need at least three LSPs to be able to support fault tolerance (i.e., $k = 2$, $n = 3$). On the other hand, 1+1 protection only needs two disjoint LSPs. Therefore, the ability to find three disjoint LSPs is more difficult than finding only two. Moreover, the ability to find more than two disjoint LSPs that maintain acceptable variations between different paths lengths becomes more difficult in terms of buffering size required at the destination side and more dependable on the network topology that is used.

Figure 5.3 shows the effect of (k, n) on the redundant bandwidth when handling single failures, where $n = k + 1$. It is seen that the redundant bandwidth needed goes down as higher (k, n) TSS level is used. However, the number of disjoint LSPs is usually limited by the network topology. The choice of the values of k and n used therefore depends on the number of disjoint paths available.

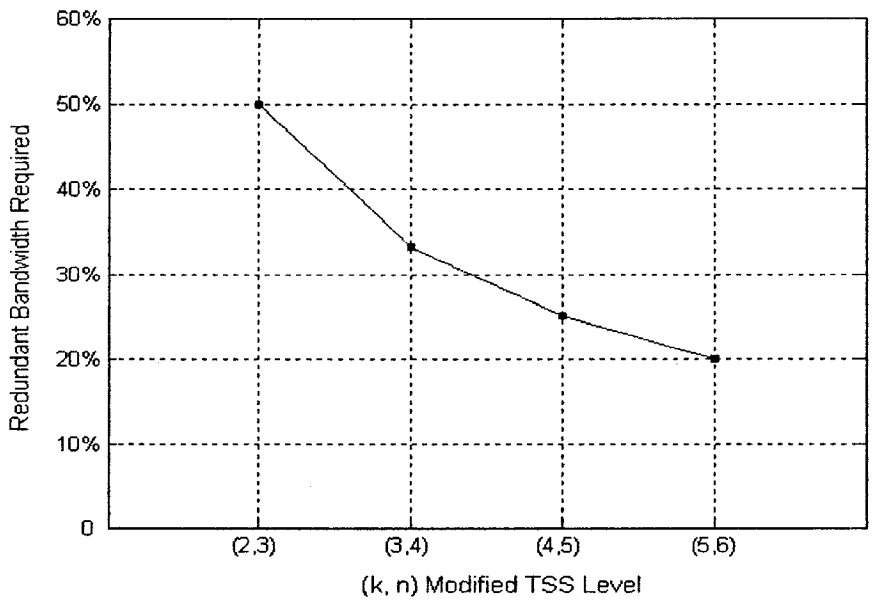


Figure 5.3 The effect of k multiple paths on redundant bandwidth where $n = k+1$

Property 5.2:

By the application of the modified (k, n) TSS algorithm, we provide fault tolerance for single/multiple LSPs failures with a recovery time that is equal to zero. Single failure protection requires $n = k+1$ paths, and multiple LSP failure protection requires $n > k+1$ paths to handle $n - k$ failures. This property is valid under the assumption of being able to find n node- disjoint LSPs between a source and a destination.

The explanation of the property is seen from the basic concept of the (k, n) TSS model. In our approach, the egress node has to receive packet shares from at least k LSPs in order to be able to reconstruct the original IP packets. So, if there are n disjoint LSPs available, then failure of LSR(s)/link(s) in node-disjoint paths or link(s) failure in link-disjoint paths, in any number of (n - k) LSPs will not affect the reconstruction process at the egress node. The concept of recovery time in terms of detecting a failure, sending

fault indication signals, and switching from the working LSP to the backup LSP does not apply in our approach and as a result, in our approach the network is able to continue operating properly without introducing any recovery delay. To explain more, consider the network topology shown in Figure 5.4. The path between the ingress and egress nodes is recognized by the three disjoint LSP1, LSP2 and LSP3. Therefore, for example using a (2, 3) TSS, a failure of any node(s)/link(s) in LSP_i will not disrupt the operation of egress node as it can reconstruct the original IP packets from two other LSPs. Note that all LSPs in this example are node-disjoint.

Property 5.3:

By the application of the modified (k, n) TSS algorithm, we provide fault tolerance for single/multiple LSPs failures with zero packet loss. Single failure protection requires n = k+1 paths, and multiple LSP failure protection requires n > k +1 paths to handle n – k failures. This property is valid under the assumption of being able to find n node-disjoint LSPs between a source and a destination.

The same argument used to explain property 5.2 can be used to prove this property. To illustrate more, consider the example of Figure 5.4. If node(s)/link(s) in LSP1 fail, then packets lost or dropped from the failed link(s) or node(s) will not affect the operation of the reconstruction process in the egress node. The egress node can still be able to reconstruct the original IP packets from packet shares coming from LSP2 and LSP3. Therefore, packets shares lost in LSP1 have no impact on the operation of the egress node, and subsequently there will be no packets missing in the egress outgoing IP packet traffic. Packet loss due to reasons other than node(s)/link(s) failure does not apply for this

property. For example, packet loss due to transmission errors or long queuing delays is discussed in Section 4.4.

Property 5.4:

By the application of the modified (k, n) TSS algorithm, we provide fault tolerance for single/multiple LSPs failures with zero packets that is out-of-order. Single failure protection requires n = k+1 paths, and multiple LSP failure protection requires n > k +1 paths to handle n – k failures. This property is valid under the assumption of being able to find n node-disjoint LSPs between a source and a destination.

The same argument used to explain properties 5.2 and 5.3 can be used to explain this property. To illustrate more, when switching the traffic from the working LSP to/from the backup LSP, packets received by the egress node can be out-of-order because for example when switching from backup LSP after fixing the node(s)/link(s) failure in the working LSP, packets arriving from the backup path may arrive at the same time with packets arriving from the working path. This concept does not apply in our proposed approach since the concept of switching between working and backup LSPs does not apply. Therefore, our approach does not introduce packets that are out-of-order. Moreover, the MPLS packet shares in each LSP are assumed to be arrived in order at the egress node. However, if there are some MPLS packets shares lost or dropped due to transmission error, in this case packet sequencing is required so that the egress node will be able to detect any packets lost due to transmission and will be able to preserve the order of shares, and consequently be able to reconstruct IP packets.

In the following section we present our simulation results that confirm the above theorems with regard to the recovery time, packet loss, and out of order packets.

5.1.3 Simulation Results

All existing recovery approaches aim to minimize the recovery time when switching and rerouting the traffic to a backup LSP. Unlike other approaches, the recovery delay time and packet loss has nothing to do with our approach due to the way the threshold sharing scheme works. The egress router's reconstruction process just needs to receive k MPLS packets from any of the n LSPs. Therefore, if failure occurs in any of the $(n - k)$ LSPs, it does not affect the reconstruction process at the egress router, and consequently no delay and packet loss occurs to reconstruct the original IP packet. The concept of sending a fault notification message to the router that is responsible for rerouting the traffic to another backup path is not required in our approach. The restoration time that includes the failure detection time, notification time, recovery operation time, and traffic restoration time [12] are therefore all absent in our approach.

Figures 5.5, 5.6 and 5.7 show our simulation results compared with other path protection mechanisms (Makam [7] and Haskin [19]). The simulation was done on the network shown in Figure 5.4 with the following parameters: link delay of 10ms, link bandwidth 1 Mbps, IP packet size 200 bytes, Constant Bit Rate (CBR) traffic at 400Kbps and each node uses drop tail queue. Single failure was injected at different link locations of LSP1. Results from Figure 5.5 confirms that the recovery time is zero in our approach if link/node failure occurred at any location in LSP1. In the simulation, the egress router was able to reconstruct the original IP packets received from the other two LSPs (LSP 2

and LSP 3) using a (2, 3) TSS. However, for the other two approaches (Makam and Haskin), the recovery time was dependent on the location of the link failure. It increases as the distance of the failure location increases from the ingress router, which is the switching node for the backup path in the example shown.

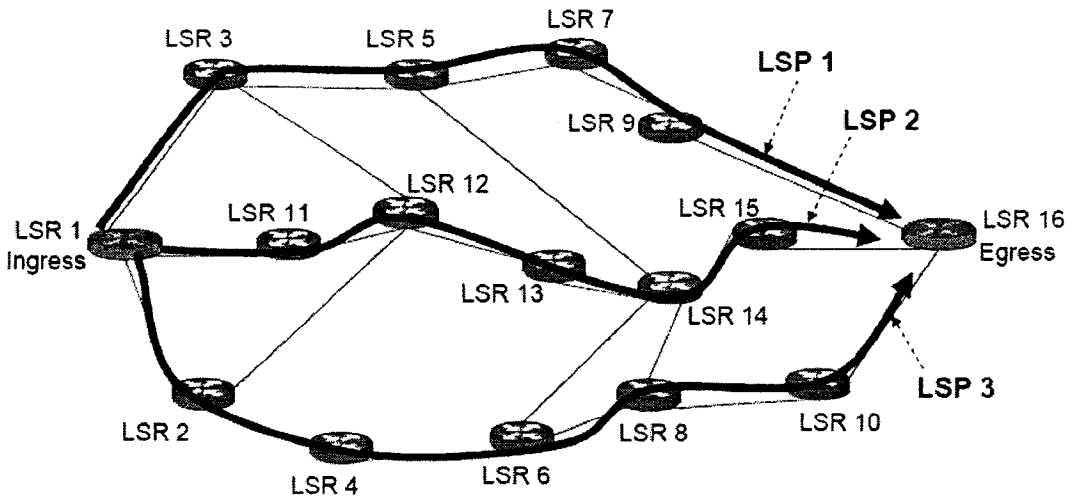


Figure 5.4 Multi-path routing topology for a (2, 3) TSS

Figures 5.6 and 5.7 confirm that there is no packet loss in our approach as well as all packets received are in order. For the other two approaches (Makam and Haskin), the number of packets lost were dependent on the location of the link failure. It increases as the distance of the failure location increases from the ingress router. Makam's scheme produces no packet re-ordering since it does not reroute packets from the failed link or node before the switch over to backup path takes place as Haskin approach does, however, more packets are lost in Makam's approach.

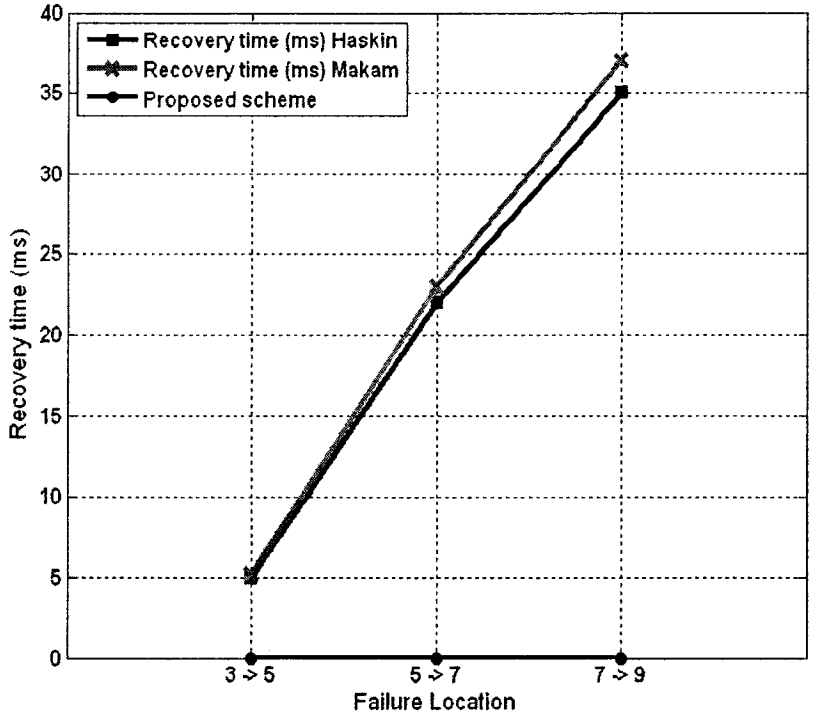


Figure 5.5 Recovery time for different link failure locations in LSP1

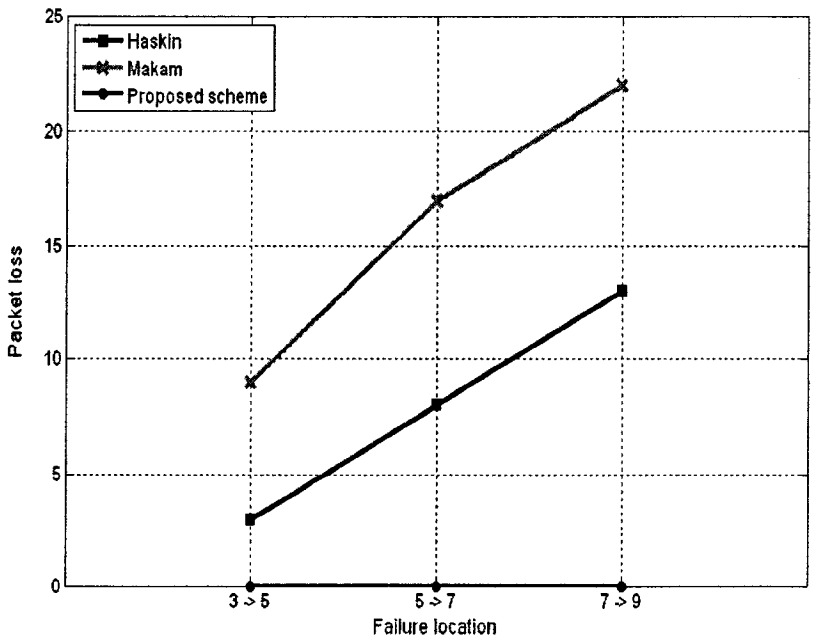


Figure 5.6 (a) Packet loss for different link failure locations in LSP1

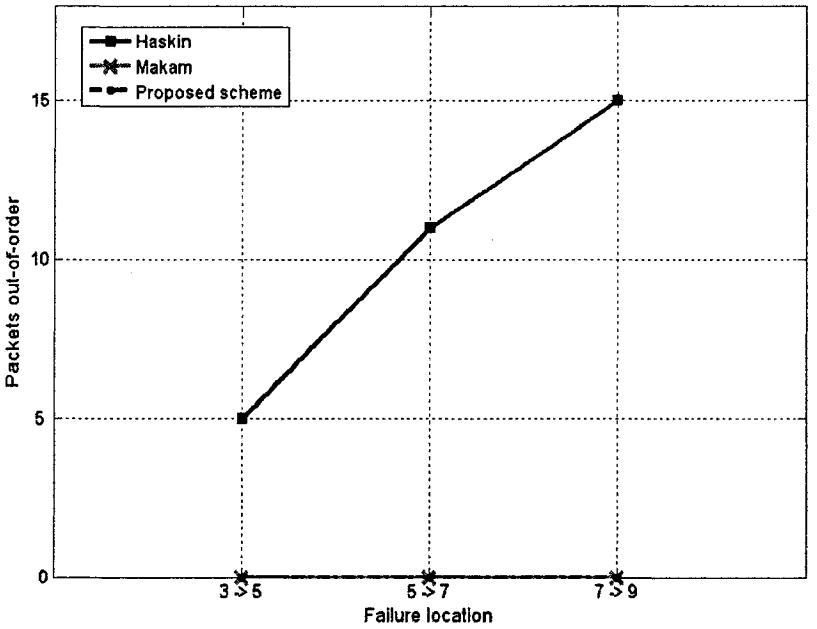


Figure 5.6 (b) Out-of-Order Packets received for different link failure locations in LSP1

It is worth to note that delay, packet loss, and reordering may occur when the traffic is to be switched back from the backup LSP again to the working LSP, this is called *reverting*. However, in our approach this case does not exist because reverting from failure has nothing to do with the reconstruction process.

5.1.4 Handling multiple LSPs failures

Our approach can be easily extended to provide multiple path failures by increasing the number of n disjoint paths. The following equation provides the level of protection based on the values of n and k.

$$Level\ of protection = \begin{cases} n - k = 0, \text{ path protection can not be provided} \\ n - k = 1, \textit{single path} \text{ failure protection} \\ n - k > 1, \textit{multiple paths} \text{ failures protection} \end{cases}$$

Our approach can handle multiple nodes or links failures within the same failed LSP. In our approach, the number of nodes or links failed within the same LSP does not affect the reconstruction process at the egress router. To illustrate more, consider the example shown in Figure 5.7 that illustrates this point. Assume a (2, 3) modified TSS is applied. It can be noticed that if one or more (node/link) failures occur within LSP1, these failures will not affect the reconstruction of traffic at the egress side.

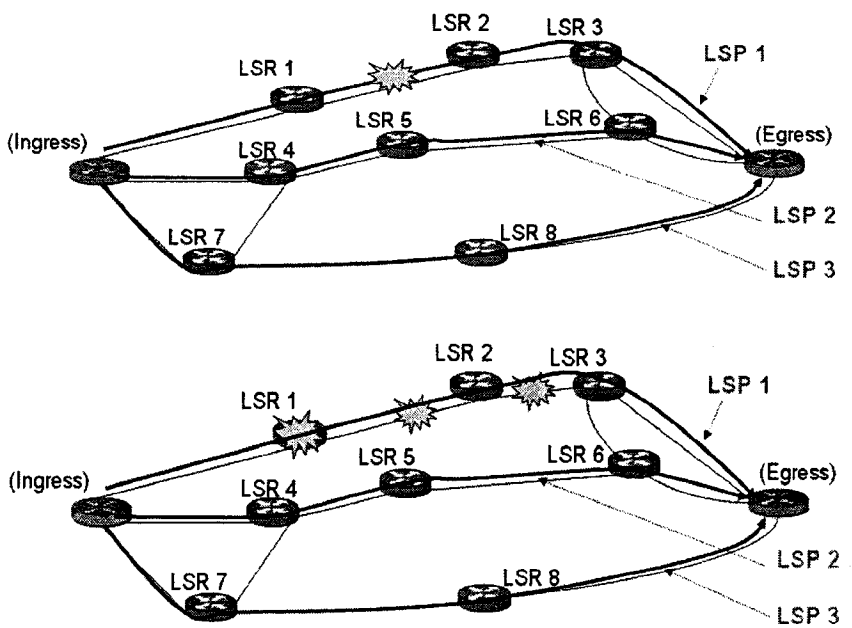


Figure 5.7 Example of multiple node(s)/link(s) failure within LSP1

It is clear that multiple disjoint failures protection is resource consuming. In other words, more path protection requires more LSPs to be set up between the ingress and egress routers, which means more extra bandwidth should be dedicated to the traffic being carried. In general, the required bandwidth for multiple path failures protection is given by:

$$\text{Extra bandwidth for (k, n) TSS scheme} \approx \text{Average } (\Phi_1, \Phi 2, \ldots, \Phi_k) \times (n - k) \qquad (5.4)$$

5.2 Deploying modified TSS to Provide MPLS Security

In this section, a security analysis is made to show how our approach can support data confidentiality, data integrity, availability, and IP spoofing in MPLS network.

5.2.1 Confidentiality

The data confidentiality is basically the main goal of the original threshold secret sharing algorithm. The shares generated by the distributor process carry an encoded form of the original IP packet. The confidentiality of data is therefore actually inherited from the original TSS algorithm as shown in section 4.1.

It is worth to note that when our approach is applied to provide confidentiality only, then there is no redundant bandwidth required. In other words, consider equation 4.2 and Figure 4.8 – (b) which illustrate the redundant bandwidth required for any (k, k) modified TSS. It will be noticed that our approach can provide confidentiality without introducing extra overhead. However, the only overhead introduced results from the need of MPLS header for each packet share. In other words, the (k, k) modified TSS produces k MPLS packet shares. The total size of the k shares is approximately equal to the original IP

packet size (equation 4.2). Indeed, this approximation is due the MPLS header overhead for each share generated in addition to the control word bytes that can be used to provide packet ordering as it was mention in section 4.4.

5.2.2 Integrity

This section discusses the applicability of our approach in providing *data integrity*. An IP packet entering an MPLS network may be attacked and modified along its LSP path at one or multiple nodes toward egress node. Data integrity is a term used to ensure that the data transferred between the source and destination nodes has not been modified.

We want to make sure that the egress or destination node is able to detect and identify the LSP path(s) under attack. In this section, we present a method to enforce the security of the modified TSS scheme with the ability to detect and identify the modified shares which belong to the hijacked LSP.

5.2.2.1 Detection of data modification

To support data integrity we need to apply a (k, n) modified TSS scheme where $n > k$. This means extra or redundant shares are needed. The impact of adding more shares on network resource utilization was discussed in Section 4.3.

The detection of modified share(s) can simply be obtained by comparing values reconstructed from different groups of shares as shown in Figure 5.8. The original IP packet is divided into three shares e_1, e_2, e_3 and allocated into LSP1, LSP2, and LSP3 respectively using a (2, 3) modified TSS algorithm. As previously discussed in Section 4.1 each share represents an MPLS packet. At the egress node, the reconstruction process

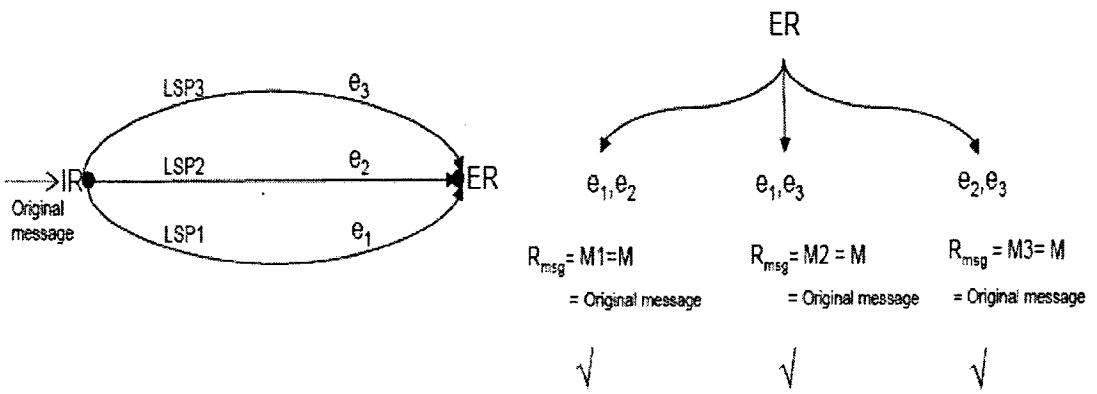
requires a group of at least two shares or MPLS packets to be able to reconstruct the original message. The number of groups which contains different combinations of shares is given by:

$$G_{(k,n)TSS} = \binom{n}{k} \quad \text{where } n > k \tag{5.5}$$

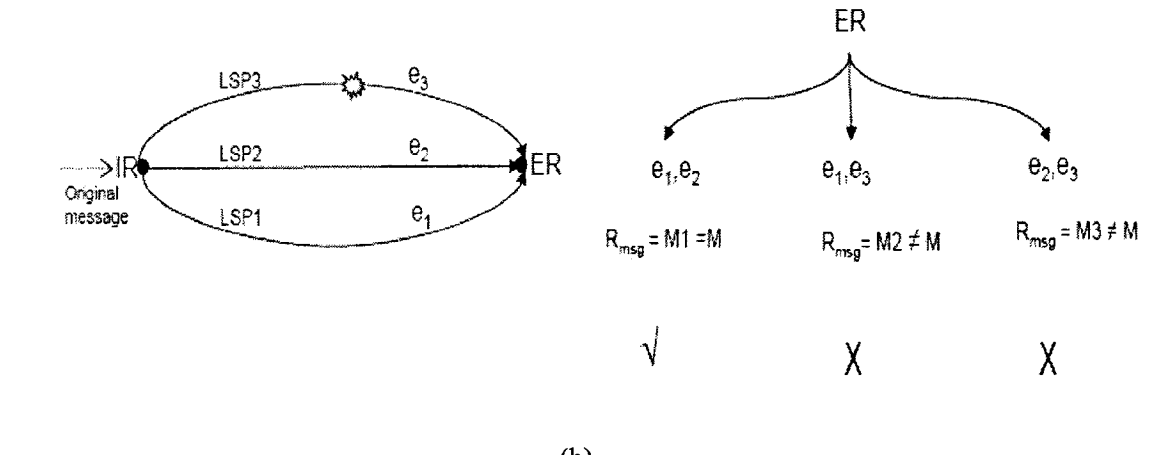
For $n = k+1$, the number of groups $G_{(k,\,k+1)TSS}$ is equal to:

$$G_{(k,k+1)TSS} = \binom{k+1}{k} = k+1 \tag{5.6}$$

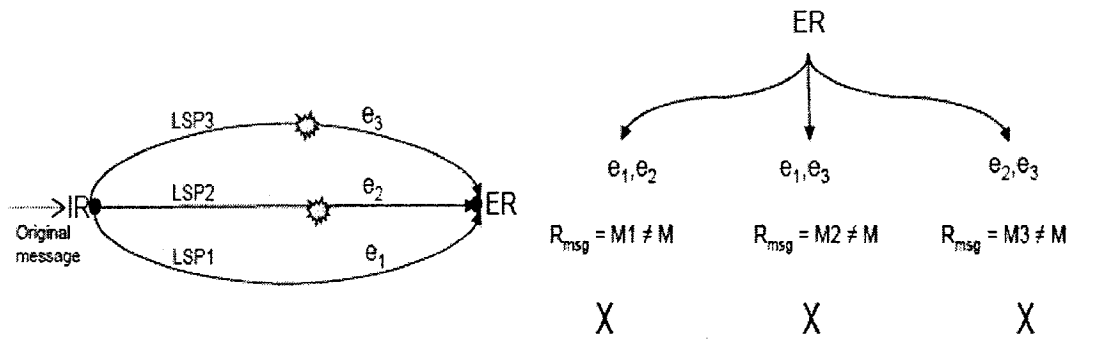
In Figure 5.8, for a (2, 3) modified TSS scheme the total number of groups obtained are 3. In Figure 5.8-(a), all groups reconstruct the same message which represents the true value of the original IP packet. However, if there is data modification the groups may contain not-attacked, or all attacked, or partially attacked shares resulting in different reconstructed messages, as shown in Figures 5.8-(b, c, d). The (k, $k+1$) modified TSS therefore results in providing detection of data modification. In other words, if the attack occurred at any LSP path, our method can detect if the data integrity is at risk.



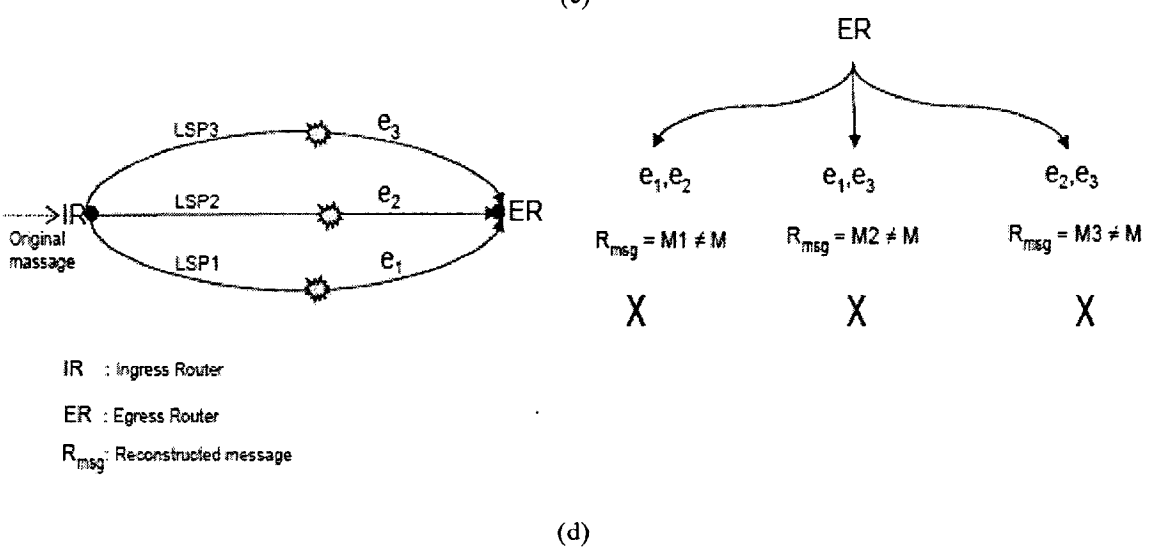
(a)



(b)



(c)



(d)

Figure 5.8 An example of a (2, 3) modified TSS used to detect data modification. (a) no attack , (b) single path attack, (c, d) multiple path attacks detection

From the previous discussion we conclude the following theorems:

Property 5.5:

For a (k, k +1) modified TSS algorithm, the reconstruction process at the egress node is able to detect data modification that may occur at any LSP of the multi-path connection.

Explanation: The use of (k, k +1) modified TSS requires only k shares or MPLS packets to reconstruct the original message. Therefore, according to equation 5.6 there will be k +1 groups available and each group consists of k shares. The modification of one or multiple shares will result in having different calculated values of all groups according to the threshold secret sharing scheme [47]. By referring to the example in Figure 4.16, the set of different group combinations consists of the following shares: (e_1, e_2), (e_1, e_3), (e_2, e_3) and the reconstruction process at the egress router produces the following different reconstructed message values respectively R_{msg} = M1, R_{msg} = M2, and R_{msg} = M3. Therefore, the presence of different calculated values indicates that the values of some shares have been modified.

Property 5.6:

For a (k, k +1) modified TSS algorithm; if an attack on the integrity of data occurs at only one LSP, then there will be only one group that is able to reconstruct the true original message. If the attack occurs on more than one LSP, then no group will be able to reconstruct the true original message.

Explanation: From equation 4.9, the total number of groups is equal to k +1. Only one LSP attack is considered to occur at a time. Therefore by removing the shares coming

from this path, the total number of groups which produce the true original message is equal to $\binom{(k+1)-1}{k} = \binom{k}{k} = 1$.

However, with $(k, k+1)$ algorithm it is not possible to know which reconstructed message is the true one, because all reconstructed values shown in Figure 5.8 have different values and therefore there is no way to know which one is the correct one. Identification of the true reconstructed message is discussed in the next section.

5.2.2.2 Identification of modified shares

To support identification of modified shares, the following requirement should be available, which is $n > k + 1$. Figure 5.9 illustrates this with an example of (2, 4) modified TSS.

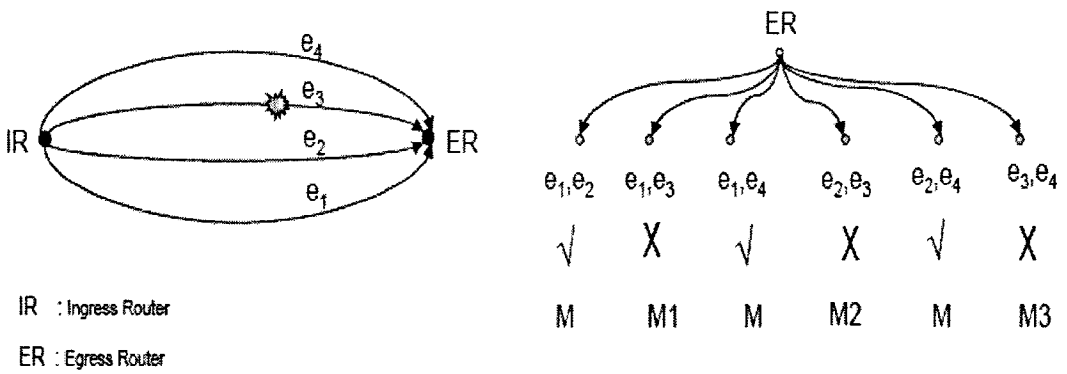


Figure 5.9 Identification of modified shares with (2, 4) modified TSS

The following observations can be obtained:

- The groups that are able to reconstruct the same true original message are 3 and they are (e_1,e_2), (e_1,e_4) and (e_2,e_4). We deduce that the reconstructed value is correct *iff* it is reconstructed more than once. Note that LSP $\mathbf{e_3}$ is not part of any group that reconstructs the true original message. The remaining groups (e_1,$\mathbf{e_3}$), (e_2,$\mathbf{e_3}$) and ($\mathbf{e_3}$,e_4) reconstruct different values. Notice again that $\mathbf{e_3}$ is the common element among these different values and not in the other groups. We can conclude that the shares coming from $\mathbf{e_3}$ are the modified shares.
- For any (*k, n*) modified TSS algorithm and for any *m* = number of multiple LSPs attacks, the following result can be obtained.

The total number of groups producing true original message G_T is given by:

$$G_T = \binom{n-m}{k} = \frac{(n-m)!}{k!(n-m-k)!} \quad \text{where } m \leq (n\text{-}k) \tag{5.7}$$

For a (*k, n*) modified TSS algorithm, if attacks on the integrity of data occurs at m LSPs, then the probability P_{G_T} of being able to reconstruct the true original message is given by:

$$P_{G_T} = \frac{\binom{n-m}{k}}{\binom{n}{k}} = \frac{(n-m)!(n-k)!}{n!(n-m-k)!} \tag{5.8}$$

For n = 4, k = 2, and m = 1, P_{G_T} = ½, which is shown in Figure 5.9, that is 3 out of 6 groups are able to reconstruct the true original message.

5.2.3 Availability

We focus on *Denial of Services (DoS)* as an example of attacks on network services availability. A denial of service attack is an incident in which a user or organization is deprived of the services of a resource they would normally expect to have. Although a DoS attack does not usually result in the theft of information or other security loss, it can cost the target person or company a great deal of time and money.

For a (k, n) modified TSS where $n > k$, the service continues if at most $(n-k)$ LSPs are under DoS attacks, since k LSPs are enough to reconstruct the original message. This is again inherited from the basic TSS model.

5.2.4 IP Spoofing Protection

IP spoofing is a technique used to gain unauthorized access to computers, whereby the intruder sends messages to a computer with an IP address indicating that the message is coming from a trusted host. To engage in IP spoofing, a hacker must first use a variety of techniques to find an IP address of a trusted host and then modify the packet headers so that it appears that the packets are coming from that host.

Our method can provide protection against IP spoofing because the IP packet entering MPLS network is divided, encoded, and then allocated to multiple LSPs, where each divided part is considered an MPLS payload. Therefore, in our approach the IP header is itself part of the original IP packet division process. If an attack occurred at any of the LSRs in the MPLS network, the hacker can not gain any information of the original IP header and therefore has no knowledge of the IP address. Figure 5.10 shows an example of how our approach can provide protection against IP spoofing.

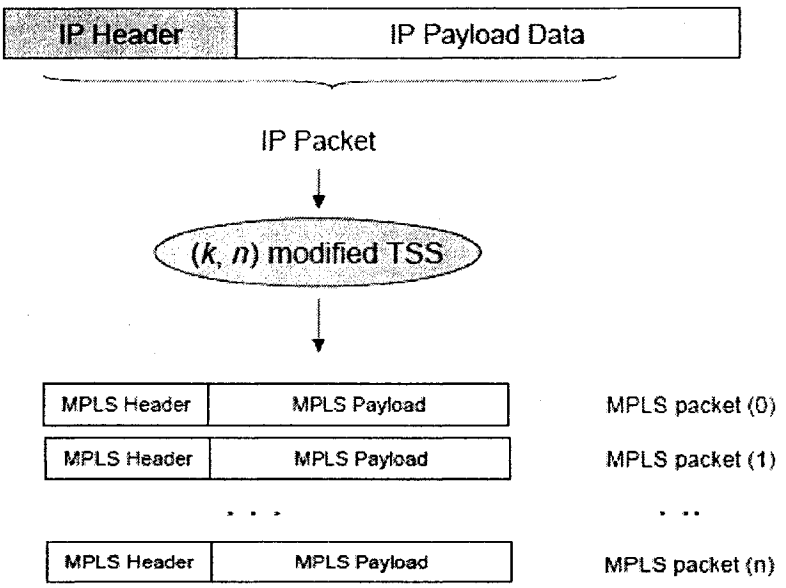


Figure 5.10 IP header as part of the whole IP packet division and encoded using a (k, n) modified TSS

5.3 Disjoint and Maximally Disjoint LSPs

In this thesis, generally we assume that n disjoint paths are available. We use the expression *disjoint paths* to denote either node or link disjoint paths. When two paths are said to be node disjoint, this also indicates that there are no links in common, however the opposite is not true (i.e., when two paths are said to be link disjoint, it does not necessarily indicate that they are node disjoint). In this thesis, we can use the disjoint paths expression to denote node or link disjoint paths *iff* we assume that the node(s) in link-disjoint case are secure and do not fail. More details can be referred to the discussion made in Chapter 1. Figure 5.11 (a) illustrates the scenario case for a link-disjoint multi-path connection. A number of research proposals have focused on finding $n > 2$ disjoint paths to provide network survivability. Bhandari [5] has proposed an algorithm to find $n > 2$ disjoint paths by extending the modified Dijkstra approach that finds $n = 2$ disjoint

paths in the same manner to obtain $n > 2$ disjoint paths. The $n > 2$ disjoint paths are obtained from n iterations of the modified Dijkstra in a graph modified at the end of each iteration. There are other approaches proposed to find multiple disjoint paths, some of these approaches are found in [53, 54] [60-63].

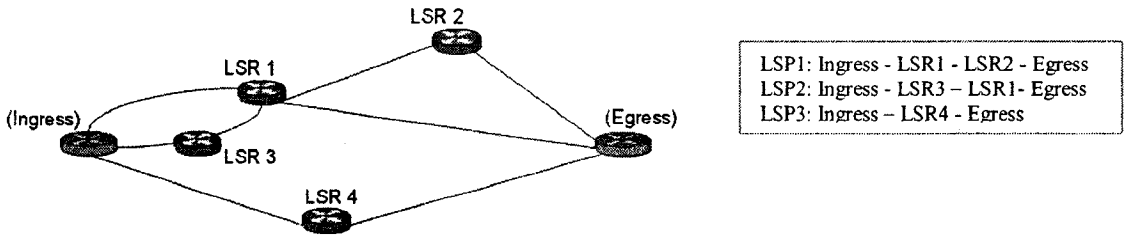


Figure 5.11-(a) The connection is link-disjoint, however, it is not node-disjoint as LSR1 is shared between LSP1 and LSP2.

In case those n disjoint LSPs are not available, maximally disjoint paths should be found between the ingress and the egress pair, where one or more than one link(s) are shared between two or more LSPs. Therefore, a pair of source and destination is said to be maximally disjoint if the number of links shared by at least two LSPs is minimum. However, network failure in the shared links will not be protected unless packets are received from at least k paths.

In this thesis, referring to a *maximally disjoint* multi-path connection between a source and a destination indicates that the multi-path connection has *shared* link(s) which inherently means has shared nodes. This type of multi-path connection has more impact on fault tolerance and security because the shared node(s) and link(s) may result in preventing the egress router from receiving the required shares to be able to reconstruct the original IP packets. Moreover, from security point of view, e.g., the confidentiality of data, the maximally disjoint LSPs may result in giving the attacker enough number of

shares to reconstruct the original IP packets and as a result of that being able to reveal their contents. In Figure 5.11(b) the failure in link between LSR1-LSR2, or the failure of any of these two nodes will have a serious impact on the reconstruction process at the egress router. In other words, any shared link failure will prevent the egress node from reconstructing the original traffic. The same applies for the failure in the link between LSR3-LSR4 or the failure in the nodes them selves. In Figure 5.11 (c), the link between LSR1-LSR2 will be overlapped by *all* LSPs between the ingress and egress routers. This case represents the worst case where the modified TSS will not work.

In our approach, the search for maximally disjoint paths is needed when there are not enough n disjoint paths between the ingress and egress routers for the required (k, n) modified TSS scheme that is applied. There are research approaches which can compute maximally disjoint paths between a pair of nodes. The work by Lee *et al.* [99] proposes an algorithm for finding K-best paths between a pair of nodes. These K paths are found with the least number of node(s)/link(s) in common. The complexity of the algorithm is $O(c(n,m,l))$, where $c(n,m,l)$ is the time complexity for minimum cost network flow algorithm for a graph with n nodes, m links, and l units of flow. The algorithm output the k paths with the lowest total cost and minimum number of common node(s)\link(s).

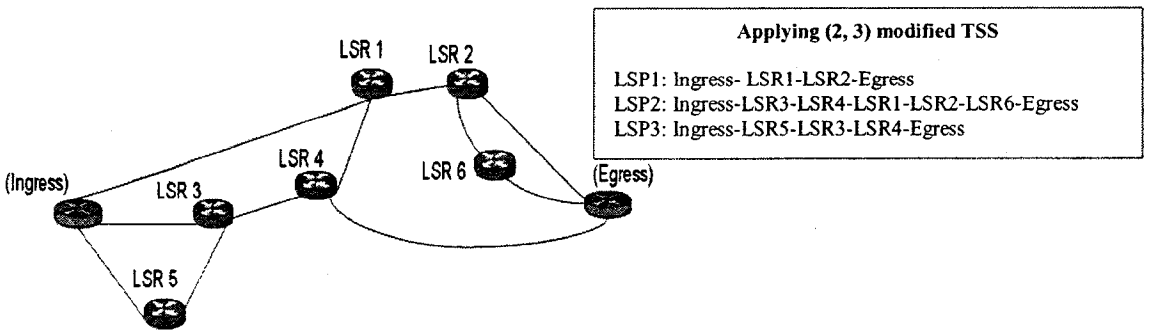


Figure 5.11 (b) A node(s)/Link(s) shared multi-path connection

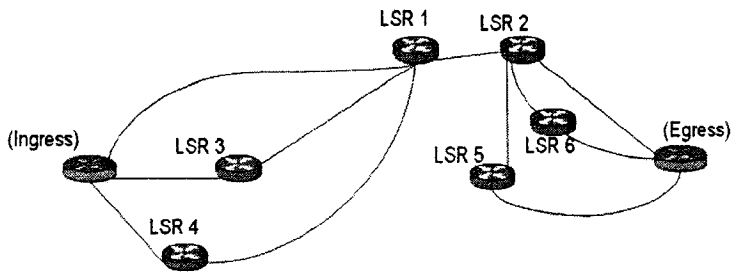


Figure 5.11 (c) The worst case where TSS will not work

The following provides an algorithm to find the best maximally disjoint multi-path group that is comprised of n LSPs for the required (k, n) modified TSS application. The algorithm used has the following limitations:

1. It assumes that the ingress router is able to compute more than one path toward the egress router where these computed LSPs are saved in a matrix. This algorithm does not compute these LSPs. It takes as input the K best LSPs obtained from other algorithms found in literature such as in [99] mentioned above

2. It only takes into account the path length constraint parameter as its cost. Other constraints can be considered for better evaluation of the best group.

Algorithm 5.1: Selecting the best maximally disjoint multi-path routes.

The purpose of this algorithm is to find the best maximally disjoint multi-path connection between ingress and egress routers. The algorithm assumes that the best set of K LSPs between the ingress and the egress routers are given (e.g., using the algorithm in

From the previous discussion we conclude the following theorems:

Property 5.5:

For a (k, k +1) modified TSS algorithm, the reconstruction process at the egress node is able to detect data modification that may occur at any LSP of the multi-path connection.

Explanation: The use of (k, k +1) modified TSS requires only k shares or MPLS packets to reconstruct the original message. Therefore, according to equation 5.6 there will be k +1 groups available and each group consists of k shares. The modification of one or multiple shares will result in having different calculated values of all groups according to the threshold secret sharing scheme [47]. By referring to the example in Figure 4.16, the set of different group combinations consists of the following shares: (e_1, e_2), (e_1, e_3), (e_2, e_3) and the reconstruction process at the egress router produces the following different reconstructed message values respectively R_{msg} = M1, R_{msg} = M2, and R_{msg} = M3. Therefore, the presence of different calculated values indicates that the values of some shares have been modified.

Property 5.6:

For a (k, k +1) modified TSS algorithm; if an attack on the integrity of data occurs at only one LSP, then there will be only one group that is able to reconstruct the true original message. If the attack occurs on more than one LSP, then no group will be able to reconstruct the true original message.

Explanation: From equation 4.9, the total number of groups is equal to k +1. Only one LSP attack is considered to occur at a time. Therefore by removing the shares coming

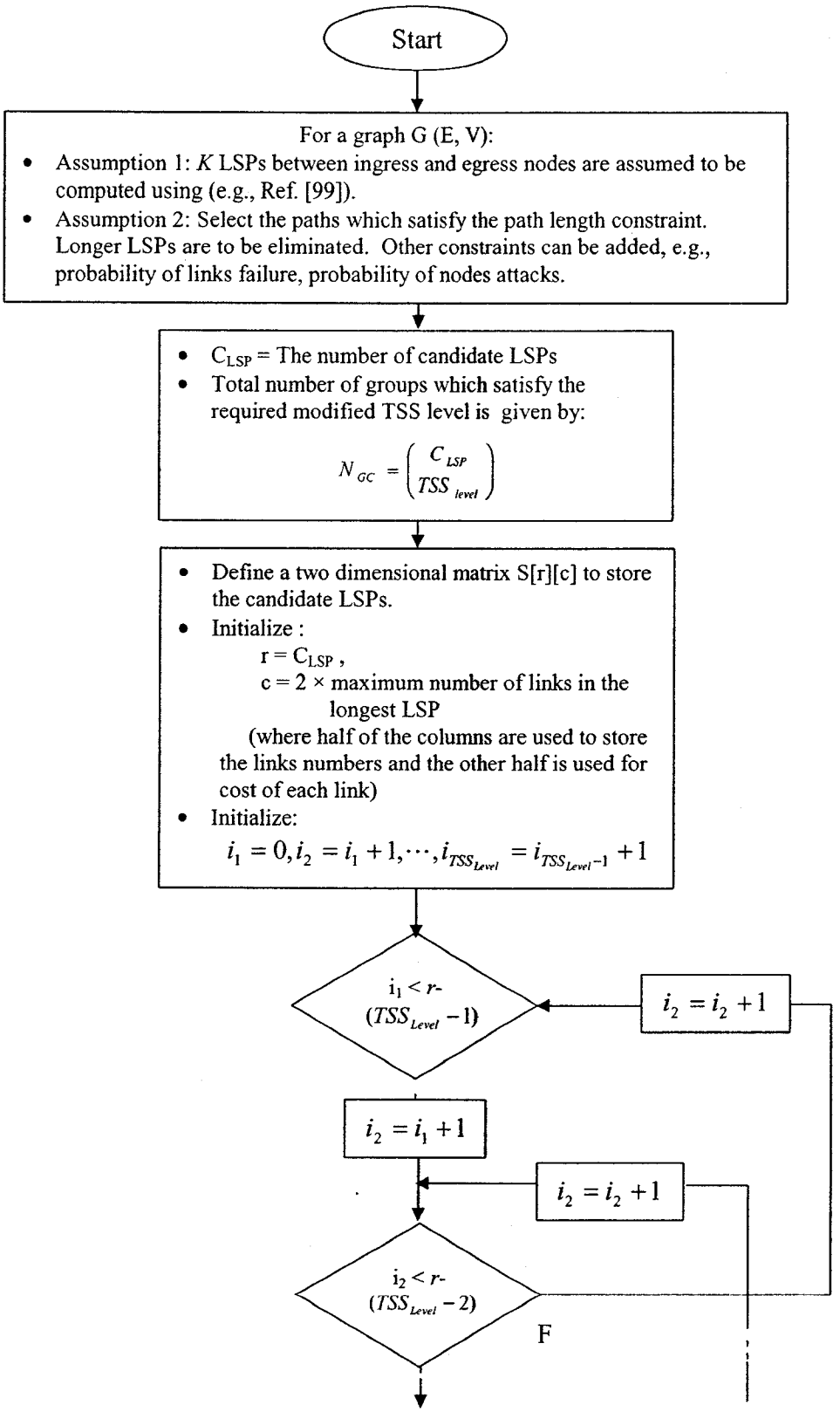


Figure 5.12 Selecting the best maximally disjoint multi-path routes

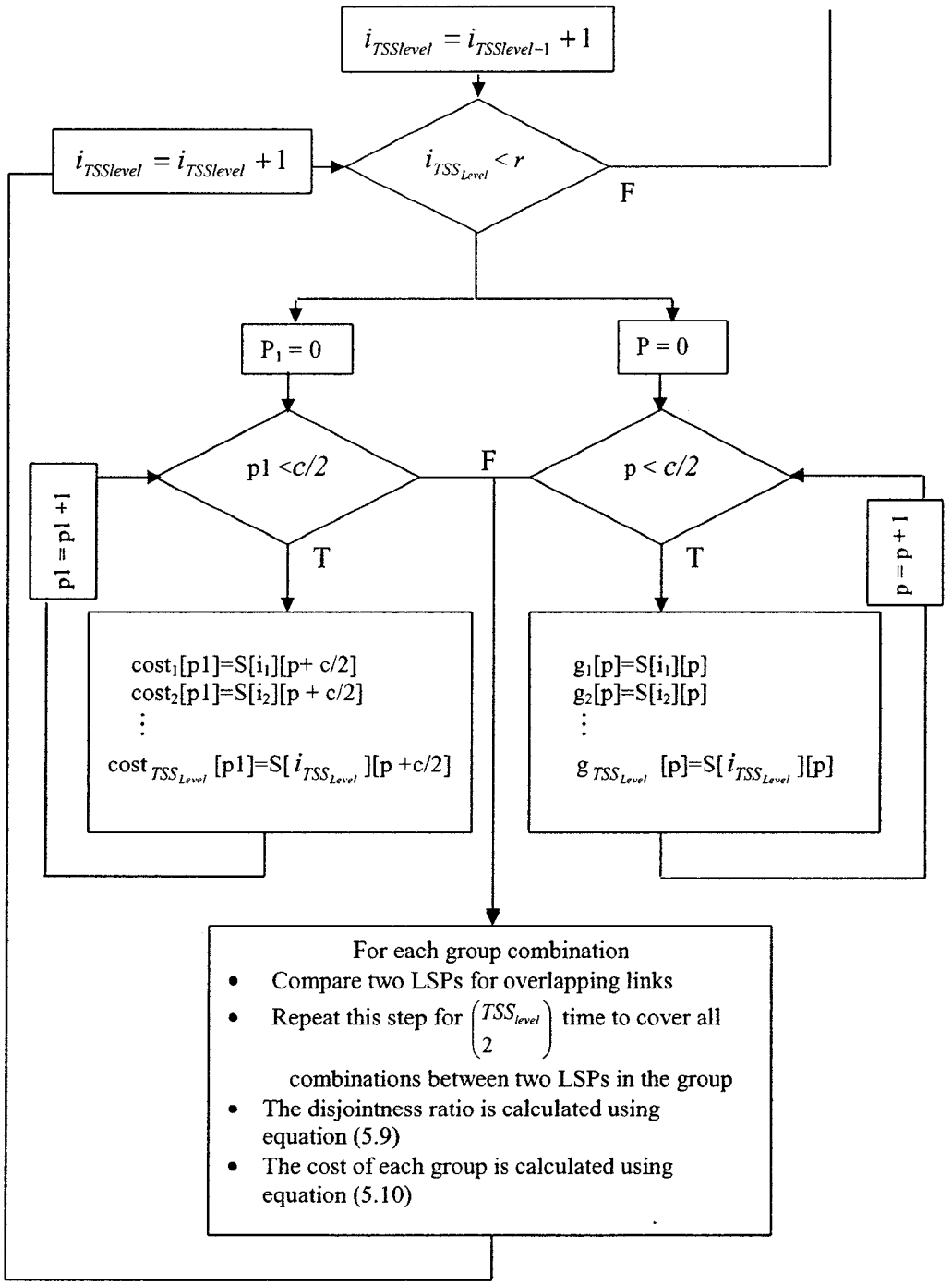


Figure 5.12 Selecting the best maximally disjoint multi-path routes

The complexity of the algorithm is $O(N_{GC} \times G_I)$ where N_{GC} is the number of different groups which consists of n LSPs, and G_I is the number of iterations to find the overlapping links between two LSPs in a group of n LSPs.

To measure the disjointness factor for each group of LSPs, the following formula can be used:

$$Disjointness = 1 - \frac{T_{OI} \mid T_{OL} \leq n\text{-}k}{T_{LPC}} \tag{5.9}$$

where:

T_{OI}: Total number of overlapping links value
T_{OL}: The number of overlapped LSPs in each link
T_{LPC}: Total number of links in the connection

From equation (5.9) it can be noticed that a group of LSPs is completely disjoint if there is no overlapping. However, the disjointness factor alone may not be enough as a selection method and especially if there exist groups that have the same disjointness ratio value. Therefore, other factors can be used for group selection such as cost of each group, where the cost function may be defined as the following:

$$\text{Cost}(g) = \sum_{\forall L_{i,j} \in g} Cost(L_{i,j}) \tag{5.10}$$

There may be overlapped links which can cause a serious threat to the reconstruction process at the egress router. To illustrate more, consider the example shown in the Figure 5.13-(a). It can be found that the total number of overlapping links value is equal to 4, i.e., {(overlapping value of LSR1->LSR2 = 1) + (overlapping value of LSR2->LSR3 = 1) + (overlapping value of LSR1->LSR8 = 1) + (overlapping value of LSR8->LSR9 = 1) =

4}, and the modified TSS level used has $k = 2$ and $n = 4$. The scheme can provide protection if a failure or an attack occurs in the overlapping links because the number of overlapping LSPs in each link is not exceeding the value $(n-k) = 2$, i.e. each link may overlap with a maximum of two LSPs.

Now, Part (b) of Figure 5.13 has the same value for total number of overlapping links (i.e., equal to 4), however, it is not a suitable choice to provide protection because the link between LSR1 and LSR2 overlaps with three LSPs, which is greater than the value $(n-k)$, and therefore a failure or an attack in this link make it impossible to protect the connection between the ingress and egress routers. In this case the egress will receive MPLS shares from only one LSP (i.e., LSR1 –> LSR8 –> LSR 9 –> LSR10 –> LSR11), and therefore one LSP is not enough to reconstruct the IP packets using a (2, 4) TSS scheme.

The overlapping links values in Figure 5.13 (a, b) are calculated as the following:

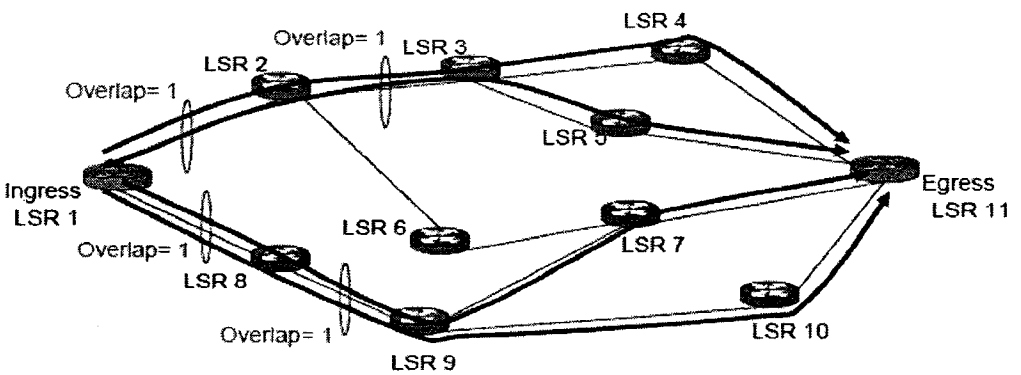
- A combination of two LSPs passing through a link is considered at each time.
- The total number of possible combinations representing the link overlapping value for two LSPs passing through a link is equal to $\binom{2 \quad LSPs\ passing \quad a \quad link}{2} = 1$, (e.g., see all links overlapping values in Figure 5.13 (a)). Moreover, for three LSPs passing through a link the overlapping value is equal to $\binom{3 \quad LSPs\ passing \quad a \quad link}{2} = 3$, (e.g., refer to the overlapping value for the link LSR1-LSR2 in Figure 5.13(b)).

The overlapping values in Figure 5.13 (c, d) are calculated by a different procedure:

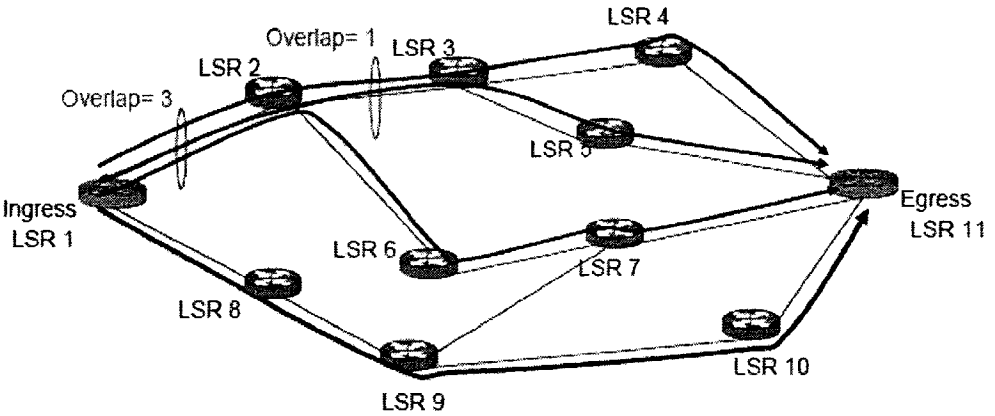
- The overlapping value for a link is calculated by counting the number of LSPs passing through this link. For example, if two LSPs are passing through a link, then the overlapping value is equal to two.

Therefore, in Figure 5.13(c, d), the total number of overlapping links is equal to 6 as shown in Figure 5.13-(c), and equal to 5 as in shown in Figure 5.13-(d). Indeed, part (d) will remain not suitable to be chosen for the threshold sharing connection for the same reason mentioned before although it has lower number of total overlapping links value. As a result of that, we can conclude the following:

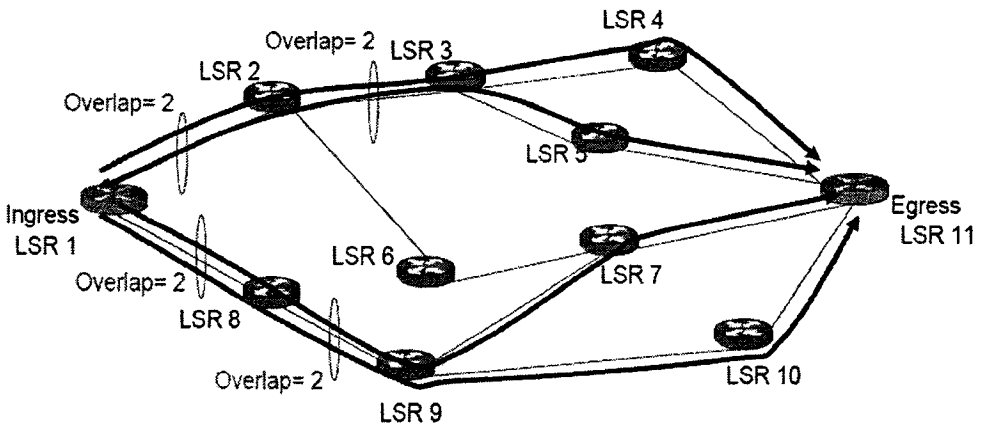
It is not preferable to consider a multi-path connection for a (k, n) threshold sharing scheme when it consists of a link that has an overlapping value larger than n-k although its total overlapping links value is lower compared to other connections.



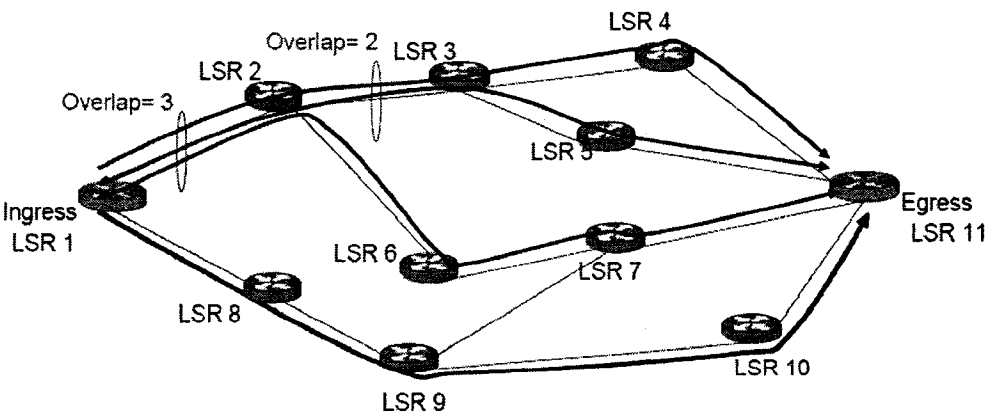
(a)



(b)



(c)



(d)

Figure 5.13 Different group combinations with variable overlapping values.

Case study:

For the network graph shown in Figure 5.14, a connection is established between the ingress node N_5 and the egress node N_{13}. Assume the following LSPs are selected to be the candidate LSPs along with their links' cost as shown in Table 5.2.

	Link #							
LSP1	10	11	16	12	17			
Cost	0.1	0.11	0.16	0.12	0.17			
LSP2	9	14	13	19				
Cost	0.09	0.14	0.13	0.19				
LSP3	10	11	15	7	13	19		
Cost	0.1	0.11	0.15	0.07	0.13	0.19		
LSP4	9	20	8	17				
Cost	0.09	0.2	0.08	0.17				
LSP5	9	20	8	12	18	19		
Cost	0.09	0.2	0.08	0.12	0.18	0.19		
LSP6	9	20	1	2	11	16	18	19
Cost	0.09	0.2	0.01	0.02	0.11	0.16	0.18	0.19
LSP7	9	14	7	15	16	12	17	
Cost	0.09	0.14	0.07	0.15	0.16	0.12	0.17	
LSP8	9	14	7	15	16	18	19	
Cost	0.09	0.14	0.07	0.15	0.16	0.18	0.19	
LSP9	10	11	16	18	19			
Cost	0.1	0.11	0.16	0.18	0.19			

Table 5.2 Selected candidate LSPs and their corresponding link cost

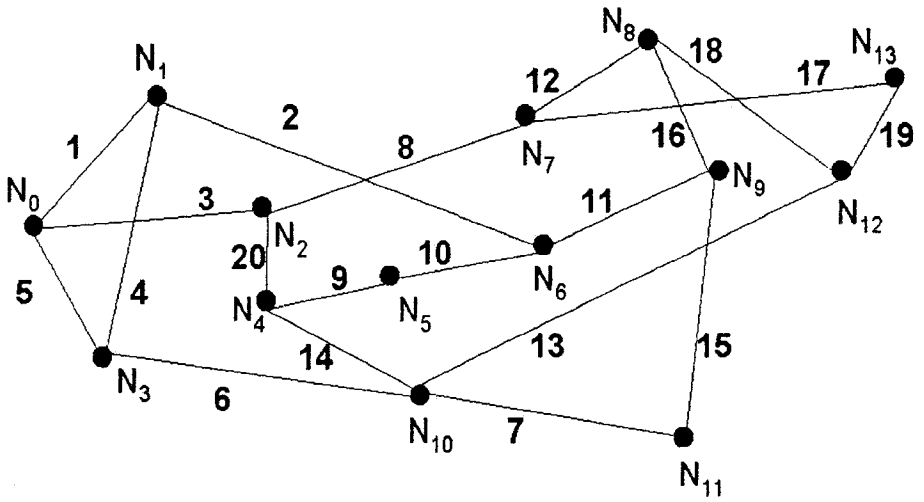


Figure 5.14 NSF network topology

The results in Figure 5.15 show the total number of maximally disjoint groups which can be created for two modified TSS levels (i.e., $TSS_{level} = 3$, $TSS_{level} = 4$). The TSS_{level} refers to the total number of LSPs in each group. For example, a (2, 3) modified TSS indicates that the modified TSS application requires three LSPs in each group to be able to provide fault tolerance or data integrity, and accordingly the modified TSS level is equal to 3. We used the link overlapping criteria of equation 5.9 to measure the disjointness for each group of LSPs. For a TSS level = 3 the number of overlapping links for each group starts from having a total overlapping value = 1 per connection, and reaches for other connections to a total overlapping value = 10. For a $TSS_{level} = 4$, the total number of overlapping links starts from having 6 per connection, and reaches for other connections to 19. It can be noticed that for a higher TSS level the number of overlapping links increase and therefore it becomes difficult to find groups with high maximally disjoint ratio.

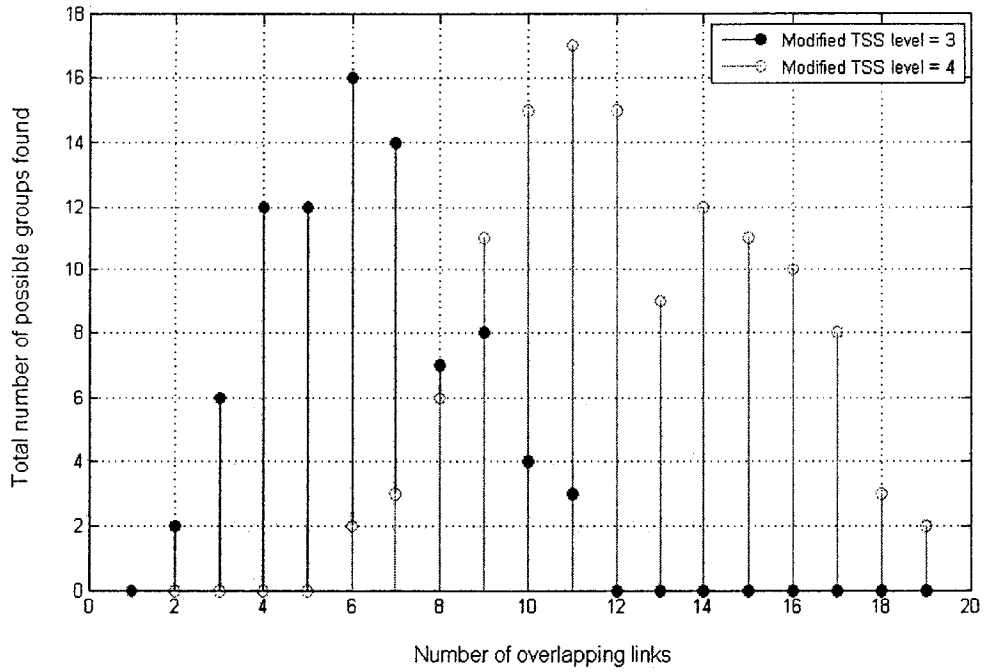


Figure 5.15 The total number of possible groups for each number of overlapping links, where two TSS levels are used.

The results obtained from Figure 5.16 – 5.17 can help to select the best connection group which contains the required *n* LSPs for the applied (*k, n*) TSS application.

Figure 5.16 shows the disjointness ratio and the cost of each group. The minimum total links overlapping value obtained are for the groups (1, 2, 4) and (2, 4, 9) for a TSS level equals to three. There are two overlapping links in each group. It is noticed that both groups have the same disjointness ratio because they have the same number of overlapping links. The costs of overlapping links in each group are nearly the same.

On the other hand, Figure 5.17-(a) shows another example where the number of overlapping links is equal to 6. The TSS level applied is also equal to 3. However, the disjointness ratio varies for each group. The cost for each group is also presented in the

same figure. Therefore, the group with the highest disjointness ratio and lowest cost can be selected.

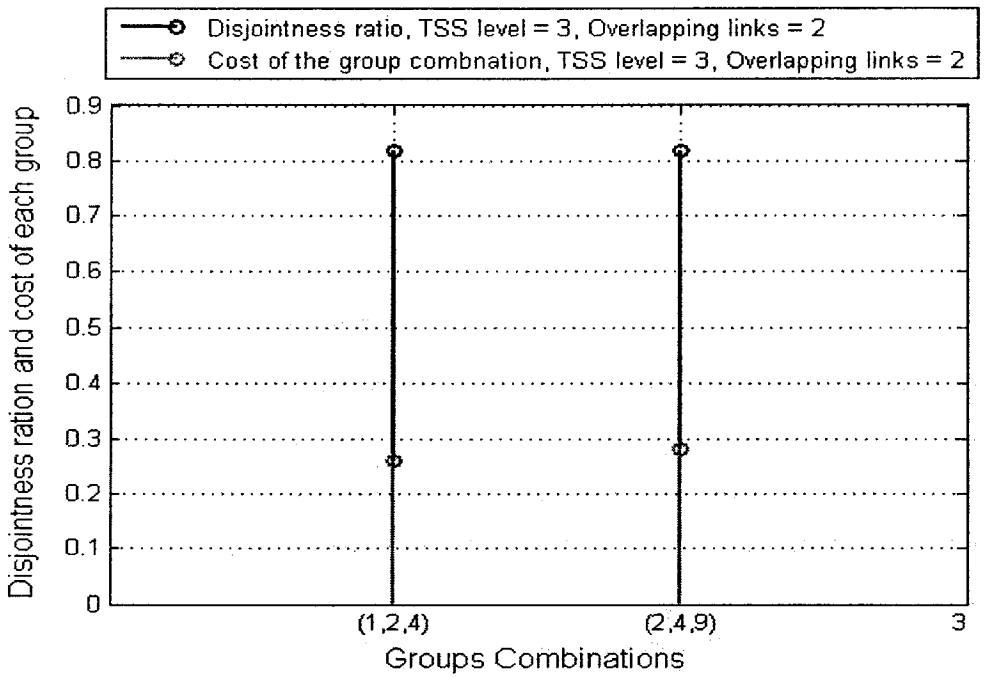


Figure 5.16 Disjointness ratio and cost, overlapping links = 2, TSS level = 3

The disjointness ratio and cost of each group for a TSS Level equal to 4 is shown in Figure 5.17-(b). The cost for a group is larger than a group cost obtained in Figure 5.17 (a) since the group combination for a TSS level = 4 consists of a larger number of LSPs than a group combination of a TSS level = 3.

Moreover, in the previous case study example, both the ingress (N_5) and the egress (N_{13}) nodes are of degree 2 (i.e., the number of incoming/outgoing links is equal to 2). The TSS_{level} used was equal to 3. By considering another scenario where the ingress node is N_5 (of degree = 2) and the egress node is N_8 (of degree = 3), the number of overlapped links can be reduced, and therefore the disjointness ratio is increased. The number of overlapping links in a group combination of, (LSP1: {10, 11, 16}, LSP2: {9, 20, 8, 12},

LSP3: {9, 14, 13, 18}), is only one and therefore, the disjointness ratio is improved and equal to 90%.

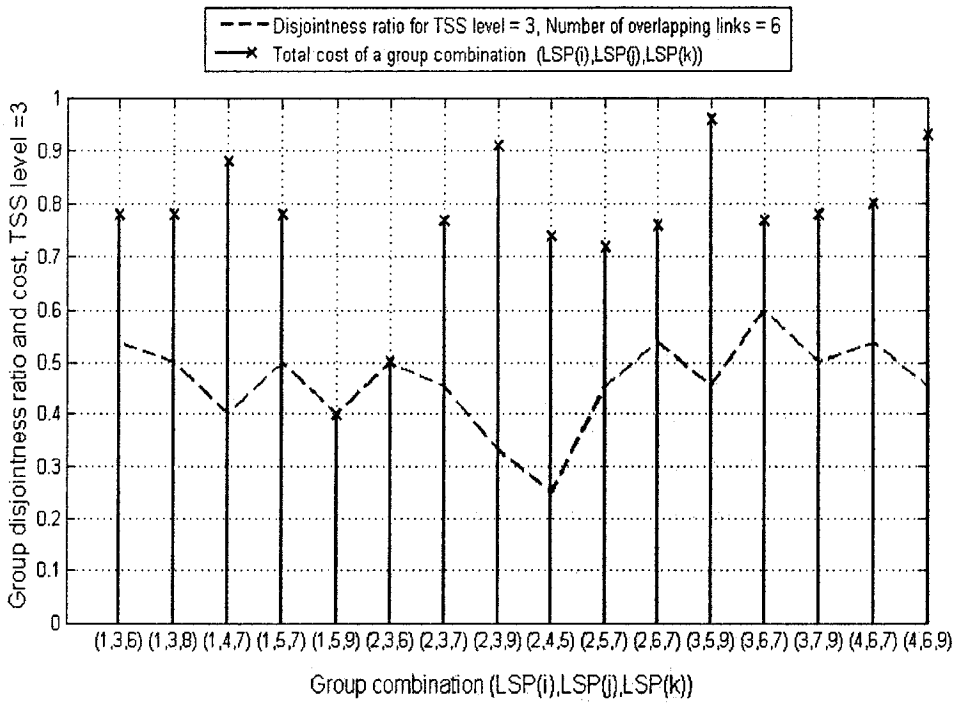


Figure 5.17-(a) Disjointness ratio and cost of groups combinations, overlapping links = 6

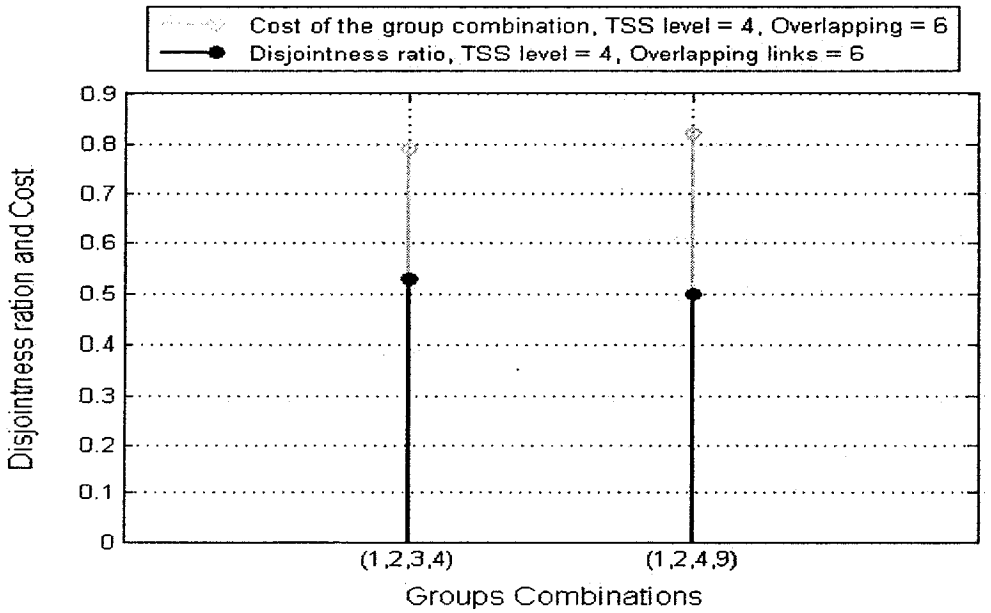


Figure 5.17-(b) Disjointness ratio and cost, overlapping links = 6, TSS level = 4

In the next section we investigate the impact of using threshold sharing scheme combined with multi-path routing on MPLS security and fault tolerance. Afterward, a comparison between the application of threshold sharing scheme and the IPSec security protocol in MPLS is also covered. Moreover, we investigate the advantages of using the modified TSS in MPLS technology domain instead of using it in the IP domain.

5.4 The Impact of Using Multiple and Single LSP Routing on MPLS Fault Tolerance and Security

This section analyzes the impact of multiple LSP traffic allocation on MPLS network fault tolerance and security. The analysis consists of two parts. The first part analyzes the reliability performance obtained when a virtual connection takes multiple disjoint LSPs to the egress node. The second part shows the contribution of our proposed algorithm in providing a high failure protection level. Afterward, the impact on MPLS security is investigated and primarily on finding the information leakage probability.

5.4.1 The Impact of Using Multiple and Single LSP Routing on MPLS Fault Tolerance

We define that an LSP fails when at least one LSR on this LSP fails. The *connection failure probability* is defined as the probability that all of the traffic sent by the ingress node fails to reach the egress node. In the single path or LSP routing, it is only possible that either all the traffic fails or successes to reach the destination.

When a connection disperses traffic on multiple LSPs, the connection failure probability equals the probability that all LSPs of the connection fail. However, in a (k, n)

TSS multi-path traffic scheme, there also exists a possibility that only part of the traffic fails which corresponds to the event that only part of the LSPs that a connection takes may fail.

The following discussion requires LSPs that take part in the multi-path connection to be disjoint. The discussion does not cover the maximally disjoint LSP connection case. For simplicity, but without loss of generality, we further assume that all the LSRs on the same LSP have the same failure probability while other LSRs in other LSPs might have different failure probability. Table 5.3 lists notations that will be used in the following discussion.

To be able to measure the traffic failure probability, we have to calculate the whole connection failure probability.

n	the number of LSPs taken by a TSS (k, n) connection.
$P(n)$	the probability that n LSPs are in failure.
P_i	the probability that a LSR on the LSP i fails.
l_i	the number of LSRs on LSP i

Table 5.3 Notations (Fault tolerance)

Theorem 5.1:

If there are n node-disjoint LSPs allocated to a connection, the probability that the whole connection is in failure, in other words, each LSR/link in each of the n LSPs fails; P(n) is given by:

$$P(n) = \prod_{i=1}^{n} [1 - (1 - P_i)^{l_i}] \tag{5.11}$$

Equation 5.11 can be used to determine how many disjoint paths, n, are required to provide the desired connection failure probability. Equation (5.11) holds because the n LSPs are node-disjoint and under the assumption that the paths' failures are independent. Equation (5.11) can also be applied to link-disjoint LSPs given that node failure is assumed not to happen, otherwise the equation does not hold. The equation does not hold for maximally disjoint paths cases. With regard to the availability of disjoint LSPs between ingress node to egress nodes, and suppose we have a predetermined threshold ζ required that a connection failure probability, $P(n)$, is not larger than ζ, i.e.

$$P(n) \leq \zeta \tag{5.12}$$

As a result, the lower bound of the modified (k, n) TSS algorithm, k, can be determined. On the other hand, the connection failure probability $P(n)$ has the property that it is a monotonic decreasing function of n and can be proven as follows.

$$P(n) = \mathrm{P}(LSP_1 failure \wedge LSP_2 failure \wedge . \quad . \quad . \wedge LSP_n\ failure)$$

$$= \mathrm{P}(LSP_1 failure) \times \mathrm{P}(LSP_2 failure). \quad . \quad . \times \mathrm{P}(LSP_n failure) \tag{5.13}$$

Note: LSPs belong to different SRLG [94]

Based on the fact that for any LSP the probability of failure is in the range of $0 \leq \mathrm{P}(\mathrm{LSP}_{\mathrm{failure}}) \leq 1$, then $P(n)$ keeps monotonically decreasing as n increases as shown in Figure 5.18. Therefore, the following result is obtained.

$$P(n) < P(Single_{LSPfailure}) \tag{5.14}$$

Where $P(Single_{LSPfailure})$ denotes the probability failure for a connection that is comprised of one or single path between a source and a destination.

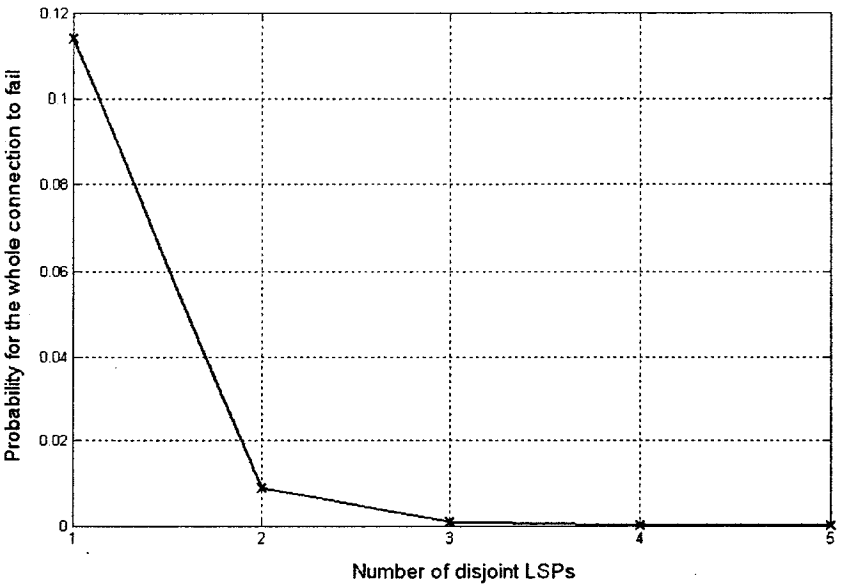


Figure 5.18 The effect of multiple LSP allocation in reducing the probability that the whole connection fails, for P_i =0.02

5.4.2 The Impact of Using Multiple and Single LSP Routing on MPLS Security

To complete this discussion, we briefly summarize the impact of multiple LSP routing on MPLS security. Table 5.4 shows notations related to security and also extended to serve for the presentation of the *information leakage probability*. We defined security terms the same way as for fault tolerance. Therefore, we define that "an LSP is attacked" if at least one LSR on this LSP is attacked. The *connection intrusion probability* is defined as the probability that all of the traffic sent by the ingress node to the egress node is attacked.

n	the number of LSPs taken by a TSS (k, n) connection.
$P(n)$	the probability that n LSPs are attacked.
P_i	the probability that a LSR on the LSP i is attacked
l_i	the number of LSRs on LSP i
N	the number of LSRs being attacked in the whole connection.
k_i	the number of LSRs being attacked at LSP i.
Y	a subset of the n disjoint LSPs which contain Y LSPs.

Table 5.4 Notations (Security)

Theorem 5.2:

If there are n disjoint LSPs allocated to a connection, the probability that the whole connection is attacked, in other words, each LSR in each of the n LSPs is attacked; P (n) is given by:

$$P(n) = \prod_{i=1}^{n} [1 - (1 - P_i)^{l_i}] \tag{5.15}$$

Equation (5.15) holds because the n LSPs are node-disjoint . Equation (5.15) can also be applied to link-disjoint LSPs given that node attack is assumed not to occur, otherwise the equation does not hold. The equation does not hold for maximally disjoint paths cases.

Moreover, the following equations are obtained:

Suppose we have a predetermined threshold ζ that requires that the connection intrusion probability, *P(n)*, is not larger than ζ, i.e.

$$P(n) \leq \zeta \tag{5.16}$$

and,

$$P(n) = \mathrm{P}(LSP_1 attack \wedge LSP_2 attack \wedge . \quad . \quad . \wedge LSP_n \ attack)$$

$$= \mathrm{P}(LSP_1 attack) \times \mathrm{P}(LSP_2 attack). \quad . \quad . \times \mathrm{P}(LSP_n attack) \tag{5.17}$$

$$P(n) < P(Single_{\ LSPattack}) \tag{5.18}$$

where $P(Single_{LSPattack})$ denotes the intrusion probability for a connection that is comprised of one or single path between a source and a destination.

From the previous equations, we notice that the connection intrusion probability for our proposed method is smaller compared to connections which consist of only one path between a source and a destination.

Information Leakage Probability:

This part focuses on the information leakage probability. It is noticed that when N_R routers are attacked, these attacks may concentrate only on a subset of the *n* LSPs taken by a connection (i.e., Y), which will be part of the information to be leaked out. In our (*k, n*) modified TSS algorithm, if the number of these attacked LSRs includes *k* or more of LSPs, then our (*k, n*) modified TSS algorithm fails to protect the *confidentiality* of data.

Lemma 1 (mainly extended from [69]): Among *n* LSPs of a connection, the probability that N_R of the LSRs routers are attacked, P [N = N_R], is given by:

$$P[N = N_R] = \sum_{k_1=0}^{\min(l_1, N_R)} \cdots \sum_{k_{n-1}=0}^{\min(l_{n-1}, N_R - \sum_{i=1}^{n-2} k_i)} \prod_{j=1}^{n} \binom{l_j}{k_j} p_j^{k_j} (1-p_j)^{(l_j - k_j)} u(l_n - k_n) \quad (5.19)$$

where $k_n = N_R - \sum_{i=1}^{n-1} k_i$ and $u(.)$ is defined as:

$$u(x) = \begin{cases} 1 & if \quad x \geq 0 \\ 0 & otherwise \end{cases} \quad (5.20)$$

Proof: The probability that at LSP j there is k_j ($k_j \leq l_j$) LSR routers being attacked, p_{l_j} is given by *Binomial* probability distribution:

$$p_{l_j} = \binom{l_j}{k_j} P_j^{k_j} (1 - P_j)^{l_j - k_j} \quad (5.21)$$

where $k_j > l_j$, $p_{l_j} = 0$. Now, the probability that at LSP_1 there are k_1 routers being attacked, at LSP_2 there are k_2 LSR routers being attacked, ... , at LSP_n there are k_n routers being attacked, while totally there are N_R routers being attacked is given by:

$$P[k_1, k_2, ..., k_n, N = N_R] = \prod_{j=1}^{n} \binom{l_j}{k_j} p_j^{k_j} (1-p_j)^{l_j - k_j} u(l_j - k_j) \quad (5.22)$$

By summarizing all the possible values of k_j (j =1, 2, ..., n -1), the marginal probability can be achieved. Figure 5.19 shows the impact of Multiple LSP allocation in reducing the probability that N_R of the LSRs are attacked.

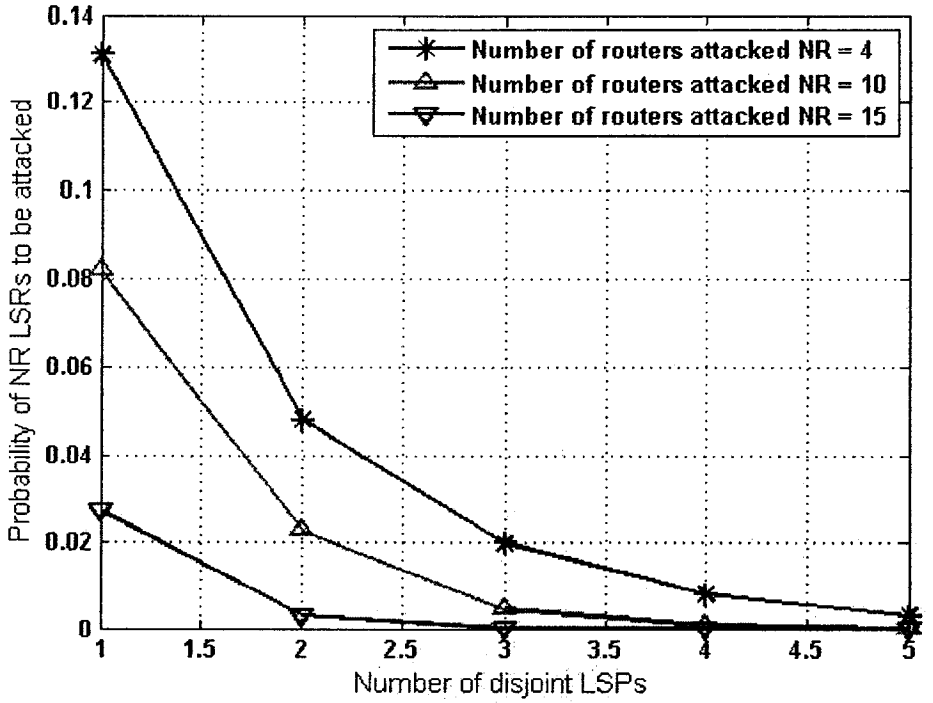


Figure 5.19 The effect of multiple LSP allocation on reducing the probability that N_R of the LSRs routers are attacked, P_i used = 0.2

Lemma 2 (mainly extended from [69]): The probability that a subset of the n disjoint LSPs, Y, which contains LSP i_1, i_2, ..., i_Y , is attacked while N_R routers have been attacked, P[Y, N=N_R], is given by:

$$P[Y, N = N_R] = \sum_{k_{s_1}=0}^{\min(l_{s_1}, N_R)} \cdots \sum_{k_{iY-1}=0}^{\min(l_{s_{Y-1}}, N_R - \sum_{s=s_1}^{s_{Y-2}} k_s)} \prod_{j=s_1}^{i_Y} \binom{l_j}{k_j} p_j^{k_j} (1-p_j)^{(l_j - k_j)} u(l_{s_Y} - k_{s_Y}) \prod_{s=1}^{n} (1-p_s)^{l_s} \quad (5.23)$$

where $k_{s_Y} = N_R - \sum_{i=i_1}^{s_Y - 1} k_s$, where $s \neq i_1 \ldots i_Y$.

Proof: from the result obtained in lemma 1, the probability that in Y LSPs there are N_R routers being attacked is given by:

$$P(Y) = \sum_{k_{i_1}=0}^{\min(l_{i_1}, N_R)} \cdots \sum_{k_{i_{Y-1}}=0}^{\min(l_{i_{Y-1}}, N_R - \sum_{i=i_1}^{i_Y-2} k_i)} \prod_{j=i_1}^{i_Y} \binom{l_j}{k_j} p_j^{k_j} (1 - p_j)^{(l_j - k_j)} u(l_{i_Y} - k_{i_Y}) \quad (5.24)$$

The probability that no other router is being attacked is given by:

$$\prod_{s=1}^{n} (1 - p_s)^{l_s}, \text{ where } s \neq i_1 \ldots i_Y.$$

Therefore, $P[Y, N = N_R] = P(Y) \prod_{s=1}^{n} (1 - p_s)^{l_s}$ (5.25)

From the definition of the conditional probability, we can obtain the following Theorem.

The probability that a subset Y of the n disjoint LSPs in the a (k, n) modified TSS algorithm are attacked given the condition that N_R routers have been attacked, $P[Y \mid N = N_R]$, is given by:

$$P[Y \mid N = N_R] = \frac{P[Y, N = N_R]}{P[N = N_R]} \quad (5.26)$$

Therefore, based on equation (5.26) we can measure the probability of attack for any (k, n) TSS level by substituting the value of k instead of the Y.

In the single path case, when an attack is successfully made, all the information along that LSP is leaked out. Otherwise, there is no information leaked out. In the multiple LSP allocation case, each LSP hold an encoded part of the whole message of IP packet. In our proposed method, an attack on a subset of LSPs less than k will not help in revealing the content of the original IP packet. We can define an information leakage

ratio L used to guarantee that the leakage information ratio is smaller than the threshold ratio which is given by:

$$\textit{Modified TSS threshold ratio, } X = \frac{k}{n} \tag{5.27}$$

We can also conclude the following result obtained from Theorem 5.9:

For a connection that uses a (k, n) modified TSS algorithm with multiple LSPs allocation, the leakage information ratio for N_R LSRs that are attacked among the n LSPs compared to the single path allocation case is given by:

$$P_{multiple-LSp}[L < X \mid N = N_R] \geq P_{SINGLE}[L < X \mid N = N_R]$$

$$= P_{SINGLE}[L = 0 \mid N = N_R] \tag{5.28}$$

Indeed, in our (k, n) modified TSS approach we should have a predefined information leakage threshold value X to maintain the security performance. That is, when information leakage value $L \geq X$, it will cause severe problems. Therefore, we want to minimize the probability that such an event occurs. This is equivalent to find a (k, n) TSS algorithm so that $P_{multiple-LSP}[L < X \mid N = N_R]$ is maximized.

5.5 Modified TSS vs. IPSec

In computer networks, IPSec is considered the most widespread protocol in use to secure the network layer (IP) [32, 88]. Other internet security protocols that are widely used include SSL, TSL and SSH, operate from transport layer and up (OSI layers 4-7).

IPSec can be implemented and deployed in the end hosts or in the gateways/routers or in both. In other words, IPSec deployment depends on the security requirements of the users. The use of IPSec security protocols is combined with extra overhead. There are two factors that can affect the amount of overhead. One factor is the cryptographic algorithm overhead related to padding. The second factor is IPSec packet formatting overhead related to the various IPSec modes.

Cryptographic algorithm overhead is created by padding that must be added to packets for encryption and authentication algorithms before processing. Each of the common encryption/decryption (DES, 3DES, AES) and hashing (SHA-1, MD5) algorithms used for IPSec is a block-based algorithm that operates on specific size blocks of data [88]. Figure 5.20 shows block sizes for common IPSec algorithms.

When data including minimum padding are not divisible by these block sizes, padding must be added to reach the desired block size prior to algorithmic processing. For example, SHA-1 and MD5 require 512-bit blocks (64 bytes). When considering the implied 64-bit length field, the real limit is 448 bits. If a packet came in at 456 bits (57 bytes), 504 bits (63 bytes) of padding would be added to "right size" the data to the appropriate block size. For randomly sized packets, padding as a percent of throughput increases as packet size decreases [88]. In extreme situations padding can nearly double packet size and therefore decrease performance by nearly 50 percent. The effect of worst case padding overhead by packet size is shown by Figure 5.21.

Algorithm	Block size (in bits)
DES, 3DES	64
AES	128
SHA-1	512
MD5	512

Figure 5.20 Block sizes for common IPSec algorithms

In addition to cryptographic algorithm overhead, IPSec incurs significant overhead caused by the addition of headers and message authentication bytes [88]. The IPSec protocol requires that IPSec headers be added on top of the IP header. The overhead varies across the four common IPSec modes.

AH provides robust authentication of IP packets without confidentiality (encryption). ESP provides both confidentiality and authentication. Transport mode is designed for host-to-host communication; tunnel mode can operate a) host to host, b) gateway to gateway, c) or gateway to host. Tunnel mode adds a new 20-byte IP header in front of the transported IP packet. ESP adds an additional 8-byte ESP header, 0-16 byte Initialization Vector (IV), and a 16-byte ESP Trailer. AH adds a 24-byte AH header. The result is significant overhead related to the various modes. The padding is up to 7 bytes in DES/3 DES, up to 15 bytes in AES, and up to 63 bytes in SHA1/MD5.

The conclusion from above discussions is that the use of IPSec may add considerable overhead. The overhead is resulted from cryptographic algorithm overhead and ESP/AH header overhead. Indeed, both overhead factors may add more than 100 bytes overhead. The effect of this overhead is more significant on a small packet size.

Algorithm	Block size (bits)	Packet (57 bytes)	Packet (113 bytes)	Packet (169 bytes)	Packet (393 bytes)	Packet (1537 bytes)
MD5	512	49%	32%	24%	14%	4%
Algorithm	**Block size (bits)**	**Packet (49 bytes)**	**Packet (113 bytes)**	**Packet (161 bytes)**	**Packet (401 bytes)**	**Packet (1537 bytes)**
AES	128	23%	11%	9%	4%	1%

Figure 5.21 The effect of worst case padding overhead by packet size

To compare between our modified TSS approach and the IPSec approach we need to consider the following factors:

- The effect of overhead introduced is determined by the security level (Confidentiality only ESP, Authentication Only AH, ESP and AH).
- The effect of overhead introduced is determined by the security mode applied, i.e., transport or tunneled mode.
- The modified (k, n) TSS level applied.

The use of IPSec protocol security can provide data confidentiality, authentication, confidentiality and authentication. Generally, additional security requirements are needed for the application of IPSec over MPLS. For example, considering a GRE encapsulation is also needed, the total overhead ratio can be calculated by equation 5.29.

(IPSec-MPLS) Overhead ratio =

$$\frac{New\_IP\_header + IPSec\_header + GRE\_header + VPN\_label}{IP\_header + IP\_payload\_size} \quad (5.29)$$

Where:

IPSec_header: can be ESP header, or AH only, or ESP and AH

New_IP_header: the new IP header added by the tunnel mode

To summarize the previous discussion, it is noticed that the use of IPSec in MPLS adds considerable overhead. The effect of this overhead on small IP packets is very large. Therefore, applying IPSec is notably affecting the network utilization. On the other hand, our approach may have better performance over IPSec in this case because the overhead incurred from the use of modified TSS will be less. For example, for an IP packet of size 90 bytes, and a (3, 4) modified TSS level, the redundant overhead is 90* (1/3) bytes + MPLS header (4 bytes) + (4 bytes control word) = 38 bytes.

5.6 Modified TSS over IP networks

The application of modified TSS over other networks such as IP or ad hoc networks may face some problems that could lower its efficiency. The major challenges are summarized as follows:

- The size of MPLS header is only 4 bytes while IP header size is at least 20 bytes. Therefore, IP shares' headers add more overhead than MPLS shares do, especially for small shares.
- Our approach depends on multi-path routing. Multiple Paths should follow disjoint routes from a sender to a destination. To specify a specific route in IP forwarding, other mechanisms have to be used such as source routing. As a result, the size of IP packet header will get larger than 20 bytes.
- The addition of, at least 20 bytes, IP header to each share is considered large compared to MPLS header.

It is clearly noticed from the points above that the performance of the modified TSS is reduced when it is applied on networks such as IP or ad hoc networks such as in [59] for the reasons mentioned above. The architecture of MPLS networks can provide explicit routing using labeled switched paths, and this makes the application of this approach within MPLS environment more suitable especially for voice over IP applications. In addition, the routing information in ad hoc networks keeps changing, as a result this makes multiple paths routing more difficult and hard to be maintained.

5.7 Considering Variable Path Length (Buffer allocation)

One of the issues related to multi-path routing is the case of uneven length of paths. The transfer delay may be different for the LSPs used to route the MPLS packets because one LSP may be longer than the other. So, for the egress router to be able to reconstruct successfully the original IP packet, it should on one hand possess enough buffers for

storing the arriving MPLS packets and on the other hand use a timer to wait for the latest MPLS packet to arrive. The value of the timer is dictated by the slowest LSP used [84]. To formalize this, let us consider an original traffic flow f with n subflows $f_1, f_2 \ldots, f_n$. The end-to-end delay for a subflow f_t, t = 1 ... n, towards the egress node is denoted by $d^{f_t,egress}$ and is calculated as follows:

$$d^{f_t,egress} = \sum_{(i,j)\in LSP_t} d_{ij} \tag{5.30}$$

where d_{ij} is the delay of each link (i, j) in LSP_t. Hence, the delay for the slowest f_t belonging to f is then:

$$d^{f}_{slowest} = \max_{t=1\ldots n}(\{d^{f_t,egress}\}) \tag{5.31}$$

Therefore, the timer used by the egress router to reconstruct the original IP packet should be at least equal to $d^{f}_{slowest}$.

Moreover, the buffer size required at the egress should be large enough to store all the *k-1* subflows received while waiting for the slowest one, as well as any other traffic originated from other IP packets and received before the slowest subflow of f. In other words, the buffer size required for each subflow f_t is:

$$B^{f_t,egress} = (d^{f_t,egress}_{slowest} - d^{f_t,egress}) \cdot b_{f_t} \tag{5.32}$$

and the buffer size needed by the reconstruction process is:

$$B^{f,egress} = \sum_{\forall LSP_t} B^{f_t,egress} \tag{5.33}$$

where b_{f_t} is the bit rate arrival for the egress node from flow f_t.

Let us consider an example to illustrate more the calculation of the required buffer size and the value of the timer at the egress router. Figure 5.22 shows a network topology with a modified (2, 3) TSS model. The egress node receives shares from three disjoint LSPs (LSP1, LSP_2 and LSP_3). In this case:

$$d^{f_{1,egress}} = \sum_{(i,j)\in LSP_1} d_{ij} = d_{ingress,3} + d_{3,5} + d_{5,7} + d_{7,9} + d_{9,egress} = 4+5+3+5+3 = 20$$

$$d^{f_{2,egress}} = \sum_{(i,j)\in LSP_2} d_{ij} = d_{ingress,11} + d_{11,13} + d_{13,14} + d_{14,15} + d_{15,egress} = 4+5+4+5+3=21$$

$$d^{f_{3,egress}} = \sum_{(i,j)\in LSP3} d_{ij} = d_{ingress,2} + d_{2,4} + d_{4,6} + d_{6,8} + d_{8,10} + d_{10,egress} = 4+5+5+4+3+4=25$$

For this example, the value of the timer to be used by the egress router should be at least equal to 25 ms and therefore the total buffer size equals to 1152 bits, when the bit rate for all links is assumed to be equal to 128Kbps, as shown in Table 5.5.

	$d^{f_{1,egress}}$ ms	max	Partial Buffer (bits)	Total buffer size (bits)
$f_{1,egress}$	20		(25-20).128 = 640	1152
$f_{2,egress}$	21		(25-21).128 = 512	
$f_{3,egress}$	25	25	0	

Table 5.5 Buffer allocation required at the egress node

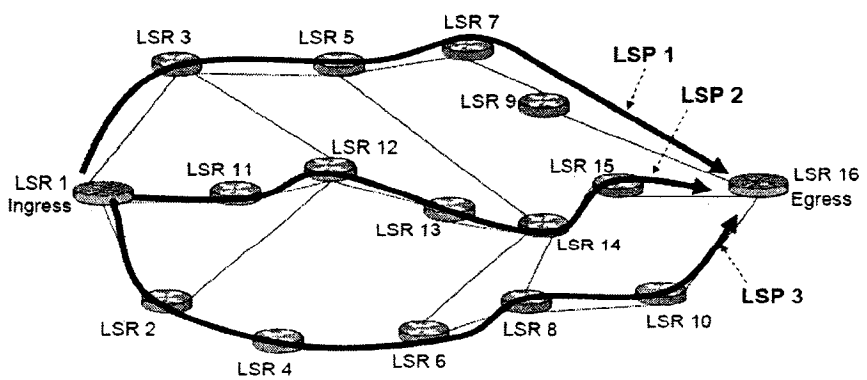


Figure 5.22 A (2, 3) modified TSS example

The calculation shown above is only used to show how to calculate the buffering at the egress router for the (2, 3) modified TSS multi-path connection. However, in real networks, the bit rate value can be very large (i.e., in Mega or Giga bits per second). In this case our approach may have a real challenge as the buffering size can be very large, especially when there are high length variations between the LSPs.

5.7 Summary

In this chapter we introduced our proposed scheme that is used to provide fault tolerance and improve security in MPLS network. From the security point of view, the modified TSS can provide data confidentiality, integrity, availability and IP spoofing. Providing data confidentiality does not require additional resources (not including header overhead of shares) while the other security factors do. Fault tolerance in MPLS can be supported with reasonable redundant bandwidth. The recovery from node/link failure can be done with no delay, packet loss or packet re-ordering.

The impact of multi-path routing on MPLS security and fault tolerance has been discussed in this chapter. The connection intrusion probability and connection failure probability have shown lower values when multi-path routing is used.

The application of IPSec security protocol in MPLS networks has also been investigated. The need to provide security within the MPLS networks is required if the network is not trusted. Therefore, IPSec can be introduced as a solution, however, some challenges may arise when it is implemented. These concerns and challenges are presented and compared with the modified TSS approach. Finally, the feasibility of applying modified TSS on IP networks has been investigated.

Chapter 6

Modified TSS in MPLS Multicast Networks

All the work done so far in the previous chapters was in the scope of unicast networks. In this chapter, we discuss the feasibility of applying the modified (*k, n*) TSS approach in MPLS multicast networks. There are some challenges that can add limitations to the application of our approach in real or practical MPLS multicast networks. This chapter starts by an overview on multicast networks and related work to MPLS in terms of fault tolerance and security. After that, we present our approach for application of the modified TSS in MPLS multicast.

6.1 An Overview on Multicast Networks

Several applications like web services, video/audio on demand services, and teleconferencing consume a large amount of network bandwidth. Multicasting is a useful operation for supporting such applications. Using multicasting services, data can be sent from one or more sources to several receivers at the same time. The data is distributed with the use of a tree structure which is called a multicast tree. Multicast trees fit into two categories: (1) source-specific trees where this category includes the multicast routing protocols: DVMRP [72], MOSPF [73] and PIM-DM [74]; (2) Group- shared trees that are built by the multicast protocols PIM-SM [75] and CBT [76]. The differences between

these protocols lie mainly in the type of multicast routing trees they build. DVMRP, MOSPF, and PIM-DM build multicast spanning trees that use shortest paths from every source to any destination. PIM-SM, CBT build spanning trees that are shortest path from a known central core, also called Rendezvous point (RP), where all sources in the session share the same spanning tree. PIM-SM is the most widely implemented protocol.

The establishment and maintenance of multicast trees make multicast routing more challenging than unicast routing. First, the creation of a tree requires the establishment of routing state. For network-level multicasting, this state has to be installed in the participating routes. As a result, the routing state is proportional to the number of active multicast groups. Second, reliability and fault tolerance become exponentially more challenging due to the multiplicity of receivers. For example, in contrast to unicast communications, any packet loss or link failure can lead to a large number of "complaining" nodes. Thus, a simple closed feedback protocol from unicast communications cannot be applied here [77].

6.2 Related Work

MPLS Multicast is an evolving area of discussion. The security and fault tolerance are the two main issues discussed in this thesis.

6.2.1 Related Work on MPLS Multicast Security

To the best of our knowledge, no work has been published on MPLS multicast security. Very few resources have been found on this topic. Multicast operation in MPLS VPN has been discussed in [78]. The confidentiality, integrity and other security issues

have not been discussed. In MPLS working group of IETF [67], work is still in progress to provide an MPLS security framework. The framework will address security mechanisms for MPLS VPN deployment. Moreover, multicast security applications are to be considered.

6.2.2 Related Work on MPLS Multicast Fault Tolerance

Several approaches have been proposed to deploy multicasting over MPLS networks. Moreover, many approaches have been proposed to employ MPLS based failure recovery methods in multicast trees. Most of these approaches satisfy some of the quality of service requirements (e.g., recovery delay, packet loss, bandwidth utilization), but there is no scheme that satisfies most of them. The existing schemes can be grouped in two categories: preplanned and on-demand. On-demand approaches do not need to compute backup routes beforehand, the computational and maintenance cost is therefore low. However, these schemes usually experience longer recovery delay, since the whole rerouting procedure is triggered on demand or in other words after the occurrence of the failure. In contrast, preplanned failure restoration predefines the backup routes, which introduces a large amount of computational and maintenance cost when there are large number of groups ongoing in the network.

However, the advantage of preplanned protection type has shorter recovery delay, which is desired by many real-time communications and some other time-critical applications (e.g., video conferencing or streaming). Recovery from failure can be local or global. Local recovery is used to protect segments of the working path and it is used to protect against a link or a node failure. The main advantage of local recovery is that it

minimizes the amount of time required for failure propagation, but it consumes more network resources. On the other hand, global recovery is used to protect traffic against any link or node failure on a path. Global recovery is called end-to-end path recovery. Usually, in global recovery the ingress router is considered responsible for switching the traffic between the working path and the recovery path. In global protection, there is a tradeoff between bandwidth utilization and recovery time.

The authors in reference [77] propose the use of aggregated multicast concept to achieve better network design goals such as scalability and fast recovery. The key idea is to force several multicast groups to share a single distribution multicast tree called aggregated tree. This procedure helps in reducing the number of trees that are needed to set up and maintain in the core of the network. Moreover, the paper reviewed four protection techniques that can be applied for MPLS multicast fault tolerance. The possible techniques are categorized as follows: link protection, path protection, Dual tree protection, redundant tree protection (e.g. 1+1 tree protection).

Within the scope of the preplanned fast protection, the paper by Deshmukh. *et. al* [80] proposes a method to set up the backup paths that are based on segmentation cluster formation, in which backup paths are provided connecting segmentation points rather than providing a backup between the receiver edge routers. Another paper by Banimelhem *et al.* [82] proposes a method to divide the multicast tree into several domains where each domain represents a sub-tree of the original one. In this approach, the backup paths are built between the root of the domain and each leaf router which is called a border router. The goal of this paper is to reduce the total capacity required for reserving the backup paths and to reduce the time of the failure notification signal.

The paper by Font *et al.* [85] discusses a solution to minimize the traffic concentration in the shared group multicast trees. Usually, the Rendezvous Points are selected in such a way to minimize the delay, however the authors propose an algorithm that can distribute and select the RPs with regard to avoid traffic concentration.

6.3 Modified TSS in MPLS Multicast Networks

In this section, we investigate the application of the modified TSS approach in MPLS multicast networks. We propose a novel approach for fault tolerance and security for MPLS network based on multiple trees. From fault tolerance point of view, our scheme falls in the category of preplanned protection, and is based on the modified Threshold Sharing Scheme (TSS) discussed in chapter 4. We show that our scheme can provide failure recovery without introducing delay or packet loss while minimizing the required bandwidth overhead. In a multiple trees approach, trees may have variable lengths or delays, which require the receivers to apply buffering. Therefore, this issue will also be discussed in our approach.

In addition, our scheme can also be used to improve the security to current MPLS multicast architecture. The same security issues discussed in the unicast application can be applied to MPLS multicast networks (i.e., data confidentiality and integrity, availability and IP spoofing). However, as it was mentioned earlier, there might be some limitations for the application of our scheme. The ability to find enough disjoint multicast trees for the modified (k, n) TSS may not be possible for all network topologies.

In multicast networks, the trees are established between a source and receivers that belong to the same tree. In a multicast graph, more than one sub-graph can be found

where any of these sub-graphs should contain all the receivers of the group. The term spanning tree is used to indicate these sub-graphs. However, these spanning trees may have different lengths. Therefore, when building a multicast spanning tree, it is preferable to find the minimum length spanning tree (MST).

There are several algorithms that are used to build minimum spanning trees. The main algorithms used to find the minimum spanning trees are:

- Prim's algorithm [95]: This is an old algorithm used to find the MST. It runs in $O\ (E \log V)$ time, where E represents the number of links (edges) and V the number of nodes (vertices) in the graph. The Prim's algorithm grows the MST tree T one edge at a time. Initially, T is an arbitrary vertex. In each step of the Prim's algorithm, T is augmented with the least-weight edge (x, y) such that $x \in T$ and $y \notin T$.
- Kruskal's algorithm [96]: is an algorithm in graph theory that finds a minimum spanning tree for a connected weighted graph. This means it finds a subset of the edges that forms a tree that includes every vertex, where the total weight of all the edges in the tree is minimized. The algorithm creates a set of trees where each vertex (node) in the graph is a separate tree. It also creates a set S containing all the edges (links) in the graph. The algorithm repeats the following steps a, b, and c until the set S is empty: (a) remove an edge with minimum weight from S, (b) if that edge connects two different trees, then add it to the tree, (c) otherwise discard that edge. At the termination of the algorithm, the tree has only one component and forms a minimum spanning tree of the graph.

In order to apply our approach, we need to find the required n trees. It is worth to note that in our approach, we assume that the spanning tree does not include all the receivers in the graph (like in the Prim's or Kruskal' and algorithm), because this will make the process of computing multiple disjoint trees not possible, and this is a limitation in our approach.

The modified (k, n) TSS requires the trees to be disjoint or if not possible, maximally disjoint. In our approach, multiple trees are said to be disjoint if there is no node or link in common between any of the trees. Specifically, node disjoint trees indicate that there are no links in common. A multiple tree connection is said to be maximally disjoint if the number of links in common among the trees are minimum. We can use the algorithm proposed by B. Mukherjee *et al.* [97] to establish multiple link-disjoint trees. The algorithm to compute link-disjoint trees works as follows:

Computation of the link-disjoint trees: because the minimum cost spanning tree problem is an NP-complete problem, a heuristic is used to compute the minimum-cost spanning tree. The algorithm in [97] employs two common heuristics: pruned Prim's heuristic (PPH) [95] and minimum-cost path heuristic (MPH) [98]. In the first heuristic, a minimum spanning tree (MST) is constructed first using Prim's MST algorithm and then pruned by eliminating unwanted links. In the second heuristic, the closest destination nodes are picked one by one and added to a partially built tree.

The steps to compute the link-disjoint trees are:

Step 1. Create the first tree using heuristic H.

Step 2. Remove the links along the tree established from step1.

Step 3. Create the second tree in the partial graph using heuristic H.

Step 4. Repeat steps 2 and 3 until all n trees are computed.

The time complexity for the PPH heuristic as mentioned before is *O(E log V)*, where it is $O(V^3)$ for the minimum-cost path heuristic (MPH). The algorithm in [97] does not provide a procedure to compute node-disjoint trees.

In order to find multiple trees which may share links (called in this thesis maximally disjoint trees), Mukherjee *et al.* propose an algorithm to compute *arc-disjoint* trees in a directed optical WDM mesh networks. The notion of "arc disjointness" indicates that two arc-disjoint trees may share a link in opposite directions only. However, the authors argue that a failure protection scheme which uses the link-disjoint and arc-disjoint trees is not always an efficient approach (i.e., it is not always possible to find two or more link or arc disjoint trees from a source to destination nodes); therefore they propose to protect each segment in the primary tree by finding a segment-disjoint path. Another approach proposed by *Eppstein* [100] can be used to generate a number of "good" trees, and then find the best K best spanning trees in time $O(E \log \beta(E,V) + K^2)$. In the literature, other approaches that are used to compute multiple best trees can be found in [101, 102].

It is worth to note that in the next discussions for fault tolerance and security, we assume that the disjoint or maximally disjoint trees are computed using available algorithms such as [97].

6.3.1 Fault Tolerance

Our approach uses the (k, n) threshold sharing scheme with multiple tree routing wherein k out of n trees are required to reconstruct the original message. For example, if

we are using a (2, 3) threshold sharing scheme, then it is only enough for the egress router to receive MPLS packets or shares coming from two trees to be able to reconstruct the original message which was divided at the ingress router.

In this section we investigate two possible scenarios for the application of modified TSS on MPLS fault tolerance. The first scenario is applied to the source-specific tree category while the second scenario is applied to the group-shared tree category.

6.3.1.1 Source-Specific Trees Scenario

One of the multicast tree routing protocols is the Source-Specific Tree protocol. In this protocol, there is only one sender for each group (G) which is represented by (S, G). We propose to apply the modified (k, n) TSS scheme to provide fault tolerance by the use of multiple disjoint or maximally disjoint trees. The fault tolerance factors (recovery delay, packet loss, packet re-ordering, and bandwidth utilization) can be provided as we have discussed before in Section 5.1.

The source specific tree is comprised of the following:

- One sender: the sender is the ingress node, therefore, each ingress node establishes a tree that is able to reach all the receivers.
- Receivers: the receivers are the egress nodes; therefore, each group (G) is comprised of the corresponding egress nodes that belong to this group.

In order to be able to apply the modified (k, n) TSS scheme, we need to find n disjoint or maximally disjoint trees which connect the ingress router with its corresponding receivers for the required multicast group.

In the next section we present our scheme with examples. The well-known NSF (National Science Foundation) [84] network shown in Figure 6.1 (Number of nodes |N| = 14) is chosen. Indeed, two scenario cases are presented here. The first scenario illustrates a multiple disjoint tree connection and the second scenario illustrates a maximally multiple disjoint tree connection.

In the first scenario, we consider a subset of egress routers (receivers) $E_1 = \{N_4, N_9\}$ which have the same source N_0. For this subset, we can build three disjoint trees T_1, T_2, and T_3 as shown in Figure 6.1. At the source router (ingress N_0), the original traffic f is split into three different shares based on a modified (2, 3) TSS model. Each share f_n will carry an encoded traffic of amount equal to the half of the original traffic f and allocated to any one of the three trees T_1, T_2, and T_3. Accordingly, each receiver in E_1 should receive at least two shares from any two trees to be able to reconstruct the original traffic f. A failure in one tree (node(s) or link(s)) will not affect the reconstruction of the original traffic as long as the other two trees have no failure.

The trees that are selected to apply the modified TSS may have some overlapping or shared links in between. The example shown in Figure 6.2 illustrates this case and is similar to the previous scenario example shown in Figure 6.1 except that the subset E_1 now contains more participating receivers in the group $E_1 = \{N_4, N_9, N_{12}\}$. Figure 6.2 shows a possible establishment of maximally disjoint trees that cover E_1. Note that T_1 and T_2 are sharing the link between N_8 and N_9. However, if a failure occurred on this link, neither N_9 nor N_{12} will be affected because N_9 and N_{12} are still able to reconstruct the original traffic from $\{T_2$ and $T_3\}$ and $\{T_1$ and $T_3\}$ respectively.

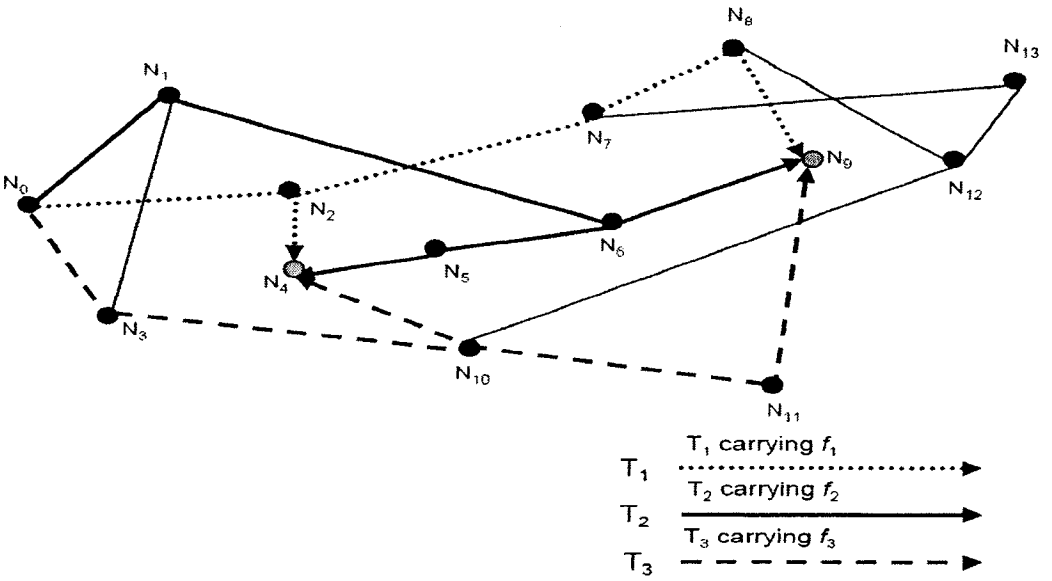


Figure 6.1 Disjoint trees coverage

However, there are limitations on the maximally disjoint trees case as it depends on the location of the overlapping between the trees. This issue has been studied previously in Section 5.3, however, links overlapping in multicasting trees may have different impact as will be discussed later in this chapter.

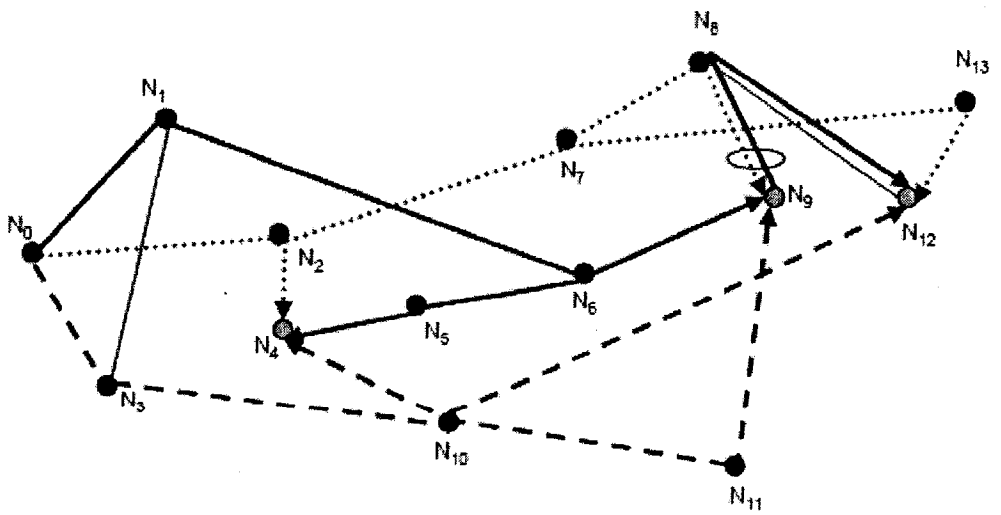


Figure 6.2 Maximally disjoint trees coverage

6.3.1.2 Group Shared Trees Scenario

In multicast networks the PIM-SM is the most widely implemented protocol. The basic idea of PIM-SM can be summarized as the following: When more than one sender belongs to the same multicast group, i.e., (*, G), they can share one central point called Rendezvous Point (RP). A RP is a router that acts as a meeting point between all the senders and receivers of the same group. Thus, one shared tree is built from the RP to all the receivers, allocating group state information only in the routers along the shortest path from the receivers to the RP. The same tree is valid for all the senders, because all the sources of the same group send the information to the same RP.

The PIM-SM approach uses a single RP point which inherits the drawbacks of centralized networking. The drawbacks for using a single RP approach can be summarized as follows [85]:

- Relying on one RP for a multicast operation can result in a single point of failure.
- Traffic concentration on RP results in longer delays and congestion.
- From a security point of view, the use of a single RP approach is considered a security problem since it forces network providers to rely on a third party when the RP is located in a network controlled by a different organization.

There are some papers that discuss the benefits of using multiple RPs over single RP in PIM-SM [85-87]. In this thesis, we propose the use of multiple RPs scheme to provide fault tolerance in MPLS multicast networks. Figure 6.3 shows an example of a multicast group, G, consisting of egress routers LER1, LER2, LER4 and LER5. The nodes RP_1,

RP_2 and RP_3 are selected to be the rendezvous points. It is apparently seen that LER2 can only receive data from two rendezvous points RP_1 and RP_2. The senders for the group G can be any of the following ingress nodes combinations {LER0}, {LER3}, {LER6}, {LER0, LER3}, {LER0, LER6}, {LER3, LER6}, {LER0, LER3, LER6}. For an example of (2, 3) modified TSS, three shared trees have to be established to provide fault tolerance.

Consider two different multicast group connections. The first scenario case demonstrates a maximally disjoint connection between the senders and the corresponding RPs. The second scenario case demonstrates another maximally disjoint connection, however, this time the connection between the RPs and the receivers contains shared links.

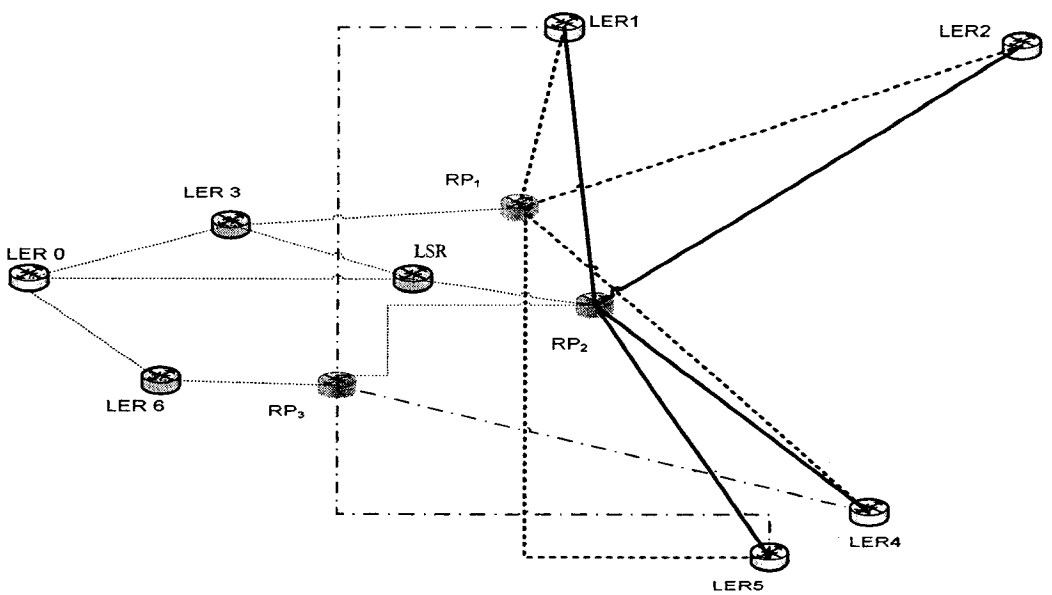


Figure 6.3 A multicast example scenario with three RPs: RP_1, RP_2 and RP_3

For the first scenario, Figure 6.4 represents a multiple multicast shared trees. The multicast session Session1 (*, G) is represented by the senders S = {LER1 and LER2}, and the receivers are represented by the group G = {LER3, LER4, LER5}. The graph below shows only the involved senders, receivers and core routers for Session1. Multiple rendezvous points RP1, RP2 and RP3 are chosen for this session. A (2, 3) modified TSS is applied to illustrate single failure case.

The following observations are obtained for the first scenario case:

- One LSP is established between each sender and RPs. For the sender LER1, the following LSPs are established, {LER1 -> LSR1 -> RP1}, {LER1, -> LSR2 -> RP2} and {LER1 -> LSR3 -> LSR5 -> RP3}. Similarly for the sender LER2, the following LSPs are established, {LER 2 -> LSR 4 -> RP1}, {LER 2, -> LSR2 -> RP2} and {LER 2 -> LSR5 -> RP3}.

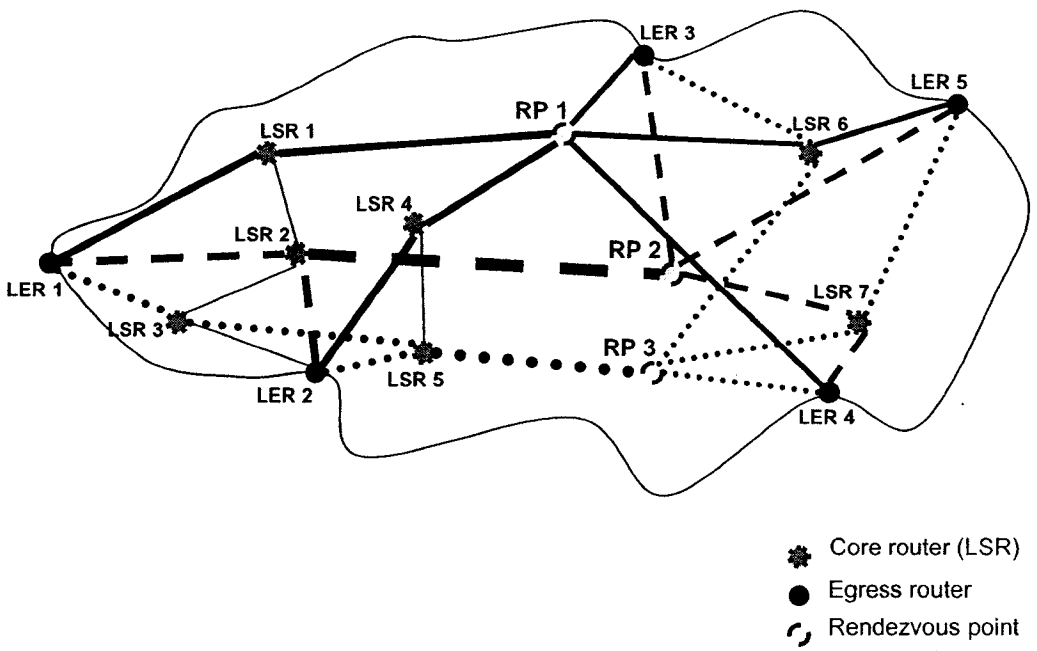


Figure 6.4 Multiple trees connection, disjoint from RPs side to receivers, and maximally disjoint from senders side to RPs.

- The links {LSR5 -> RP3} and {LSR2 -> RP2} are shared. However, this does not affect traffic reconstruction in the receivers in the event of single failure since at least two of the three RPs will receive MPLS packet shares. This feature is very important because we do not have to find disjoint paths for every sender. Therefore, we do not impose any additional requirements on multicast routing between the sender and RPs.

- From each RP point to receivers in the group G, a tree is established. This tree can be link disjoint from other trees established by other RPs or maximally disjoint as will be shown later.

- The trees established from RPs to receivers in Figure 6.4 are link disjoint but are not node disjoint. However, a single failure in nodes LSR6 or LSR7 does not affect the reconstruction of Session1 in any of the receivers in the group G.

- A failure in one RP point does not affect the reconstruction of the multicast session in any of the receivers in the group G1. Therefore, our approach can protect the network against single RP failure because the other RPs will be able to operate the network properly without introducing any recovery restoration delay or packet loss. Our approach overcomes the drawback of *centralized* RP architecture where if a single failure occurs in one RP point, the whole shared tree is disconnected which results in session disruption. Additionally, the centralized RP architecture suffers from the concentration of traffics in one point which leads also to traffic congestion.

Beside the above observations, it is important to measure the network utilization when applying multiple trees protection using modified threshold sharing scheme. Figure 6.5 provides a comparison analysis between the proposed scheme (using a (2, 3) TSS) and the (1:2) protection scheme, where there are two working trees and one backup tree, for the multicast topology of Figure 6.4. The analysis measures the bandwidth reservation required. Equation 6.1 calculates the total bandwidth reservation in each shared tree where each shared tree is identified by its corresponding rendezvous point.

Let us assume that there are three kinds of traffics that are to be sent from LER1 to the receivers in the group based on Figure 6.4 topology. Traffic A consists of two equal sub flows (14 units, 14 units) and they are allocated to Tree 1 and Tree 2 identified by RP1 and RP2 respectively. Now, the backup tree is Tree 3 identified by RP3. The other two traffics (B and C) are carrying two different sub flows. In other words, traffic B consists of sub flows (18 units, 14 units) and traffic C consists of sub flows (24 units, 14 units) and respectively allocated to Tree 1 and Tree 2. The backup Tree 3 should be allocated the highest sub flow value. From Figure 6.5 we notice that our approach requires less bandwidth reservation compared to a (1: 2) protection scheme when variable traffic amounts are used as in the case of B and C. However, both schemes perform equally when an equal traffic sub flows amounts are used as in the case of A.

Total bandwidth required for tree T_j for each sender i is given by:

$$BW_{T_j} = B_{ij} + \sum_{\forall e \in G} B_{je} \tag{6.1}$$

Where:

j : Represents the Rendezvous Point router number.

i: Represents sender (ingress router) number.

e: Represents receiver (egress router) number $\in$ G.

G: Represents a group of receivers.

B_{ij} : Reserved bandwidth for the path between an ingress router i and a Rendezvous Point j.

B_{je} : Reserved bandwidth for the path between Rendezvous Point j and egress router e.

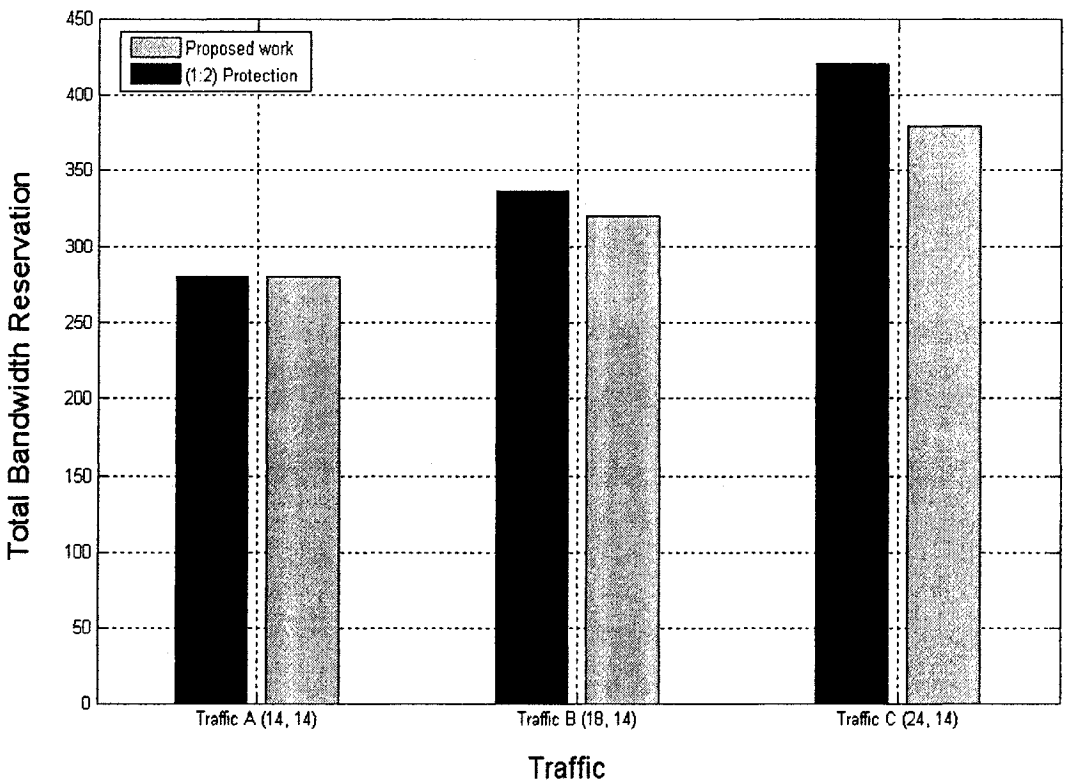


Figure 6.5 Bandwidth comparisons (in units) between (2, 3) modified TSS and (1:2) protection scheme

- The selection of RPs is very important issue. For example, the selection of LSR6 to be the rendezvous point instead of RP1 does not produce disjoint trees for the multicast group G.

- Final observation resulted from the use of multiple trees approach is the ability to support multiple failures in different trees without the need to use higher modified TSS level. For example, if the links {LSR6 -> LER3} and {LSR7 -> LER5}, {RP2 -> LSR7} all have failed simultaneously, then the receivers will be able to continue operating properly (i.e., able to reconstruct original IP Packets).

Let us now consider the second scenario that is shown in Figure 6.6. The figure represents another case where the trees established between the receivers and RPs are shared in some links {LSR6 -> LSR8 and LSR7 -> LSR10}. Also, an additional receiver LER6 joined the group G. Additional core routers {LSR8, LSR9 and LSR10} are involved this time wherein the previous example they were hidden. It is clearly seen that the overlapping between trees does not affect the reconstruction of traffic at the receivers' nodes when single failure occurs. However, simultaneous failures on the shared links may lead to disruption in service for some of the receivers (i.e., LER5 and LER 6).

It is noticed that if the trees are overlapping it may or may not affect the reconstruction of data at the receivers. We can obtain the following conclusions:

- The overlapping in the case of multicast may not have a direct or actual impact on the reconstruction of original data at the receivers compared to

the case of unicast. The topology and trees selection play an important factor.

- In multicast application, the multiple failures of links in more than one tree may not disrupt the operation of *all* receivers in the group. An example of this case was shown in Figure 6.6. Obviously, this behavior can not be achieved in unicast application unless $n > k+1$.

In the following section we explore the security aspect of the application of the modified TSS in MPLS multicast networks.

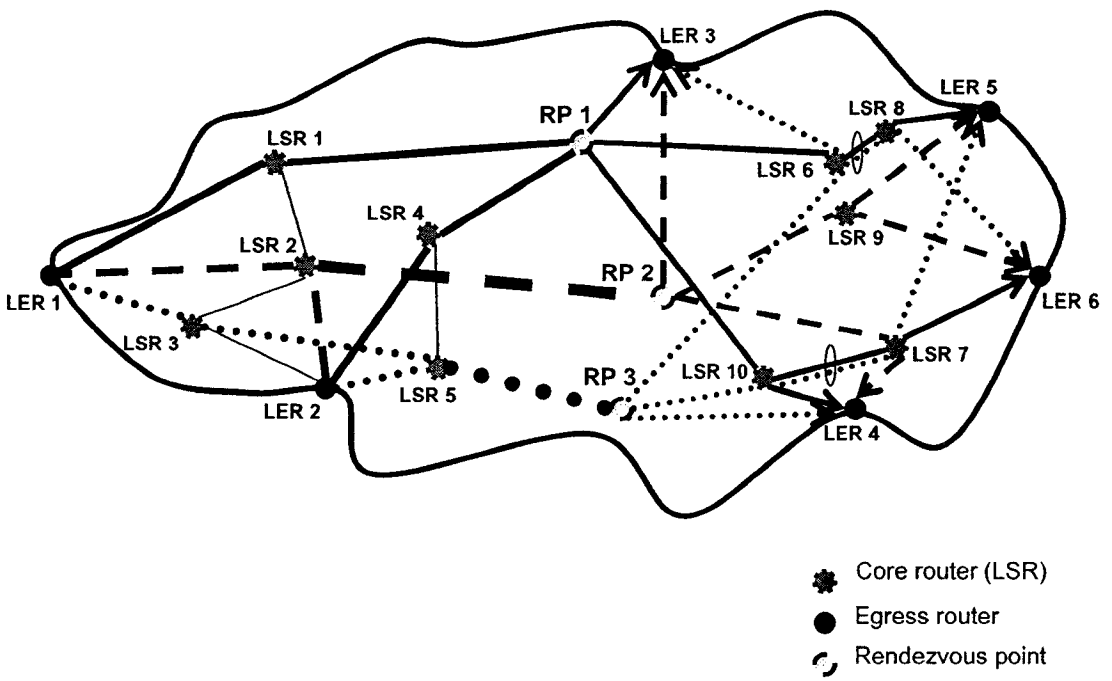


Figure 6.6 Multiple trees connection, maximally disjoint from RPs side to receivers, and maximally disjoint from senders to RPs.

6.3.2 Providing Security

MPLS multicast security can be provided by using modified threshold sharing scheme. The same procedures for MPLS multicast fault tolerance can be used for the application of the TSS on MPLS multicast security. The modified TSS level is determined by the security option required. The security options that can be provided are:

- Confidentiality of data: The modified TSS level required is (k, k) TSS.
- Data integrity: The modified TSS level required is (k, n) TSS where n > k.
- Relying on a single RP point may be considered a security threat if the RP is established in a non-trusted location or domain, and if the RP itself is not trusted.
- If a RP point is hijacked (e.g., DoS attack), the service availability or reliability could be at risk. Therefore, the distributed RPs or multiple RPs can protect the network from such possible attacks.

Finally, an important issue that may result from the use of multiple trees is the synchronization of packets, which may need buffer allocation in receivers. In next section we discuss how to measure the required buffer size.

6.4 Limitations of our approach in MPLS multicast networks

The application of our modified (k, n) TSS scheme in MPLS multicast networks can be faced by some challenges and limitations which are described below.

1. Our approach depends on the network topology. The application of the modified (k, n) TSS scheme requires to obtain n disjoint or maximally disjoint trees for the multiple trees connection. This requirement can be very hard to achieve in real networks especially for the case of n disjoint trees.

2. The number of receivers (egresses) of the multicast group plays an important role in the ability to compute the n disjoint trees. Therefore, if the number of receivers increases in a multicast group, then it will be more difficult to find the required number of disjoint trees. It is worth to note that in our approach, we assume that the spanning tree can not include all the nodes in the graph (like in the Prim's or Kruskal' and algorithm), because this will make the process of computing multiple disjoint trees impossible.

3. The variation in length between trees may require high buffering allocation in the receivers' side. This point is discussed in details in Section 6.5.

6.5 Buffer Allocation

The proposed scheme may require to buffer packets at the receiving nodes (egress routers) due to the possibility of having some trees faster than the others in the multiple trees. Therefore, buffer allocation is required [84] to synchronize packets received from shorter paths. Here we calculate the buffer size required.

Consider an original traffic flow f with n subflows or shares f_t (t = 1, 2, ..., n). The end-to-end delay for each share towards an egress node e is given by:

$$d^{f_t,e} = \sum_{(i,j)\in T_t} d_{ij} \tag{6.2}$$

where according to our model, d_{ij} is the delay of each link (i, j) in a tree T_t.

The delay for the slowest f_t belonging to the f to egress node e is:

$$d^{f_t,e}_{slowest} = \max(\{d^{f_t,e}\}) \quad \text{for all } T_t \in T \tag{6.3}$$

Therefore, the buffer size $B^{f_n,e}$ required for each f_n flow is:

$$B^{f_t,e} = (d^{f_t,e}_{slowest} - d^{f_t,e}) \ . \ b_{f_t} \tag{6.4}$$

where b_{f_t} is the bit arrival rate for node e from flow f_t. It is noticed that a buffer for the slowest path is not required. The total buffer size in an egress router e for an original traffic flow f is:

$$B^{fe} = \sum_{\forall T_t \in T} B^{f_t,e} \tag{6.5}$$

To illustrate more on how to calculate the required buffer size at the egress router (a receiver), we present this example. The chosen topology is the NSF network shown in Figure 6.7. The costs on the links represent the delays d_{ij}. The multicast group subset to be considered is E = {N4, N9}, see Figure 6.7. In the case of applying a modified (2, 3) TSS model, the original traffic flow is allocated into three disjoint trees.

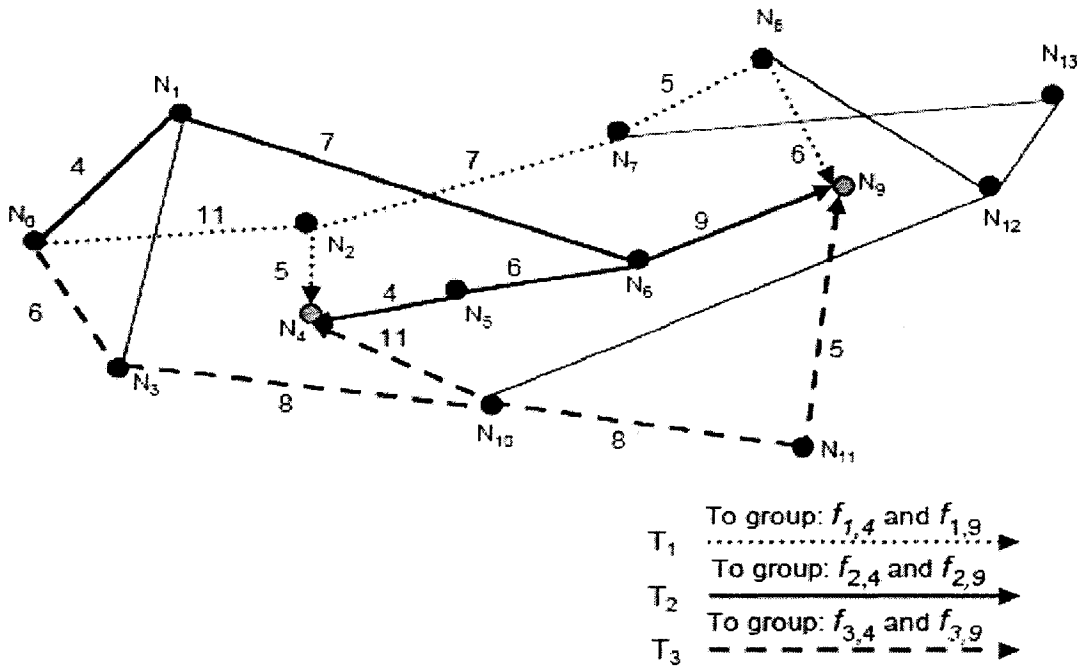


Figure 6.7 Representation of different link costs for T_1, T_2 and T_3

Each egress router (N_4 and N_9) receives shares from three disjoint trees T_1, T_2 and T_3. In this case for tree T_1, $d^{f_{1,4}} = \sum_{(i,j)\in T_1} d_{ij} = d_{0,2} + d_{2,4} = 11+5 = 16$. In the same way for the rest, we obtain the values of Table 1. The total buffer size required at N_4 and N_9 are 1664 and 1408 respectively when the bit rate for all links is 128Kbps.

	$d^{f_{i,e}}$ ms	max	Partial Buffer (bits)	Total buffer size (bits)
$f_{1,4}$	16		(25-16).128 = 1152	
$f_{2,4}$	21		(25-21).128 = 512	1664
$f_{3,4}$	25	25	0	
$f_{1,9}$	29	29	0	
$f_{2,9}$	20		(29-20).128 = 1152	1408
$f_{3,9}$	27		(29-27).128 = 256	

Table 6.1 Buffer allocation required at each egress router

The calculation shown above is only used to show how to calculate the buffering for each receiver in the multicast group. However, in real networks, the bit rate value can be very large (i.e., in Mega or Giga bits per second). In this case our approach may have a real challenge as the buffering size can be very high, especially when there are high length variations between the trees.

6.5 Summary

In this chapter we have investigated the feasibility of applying the modified threshold sharing scheme on MPLS multicast networks. We have presented two application scenarios. The first scenario is based on the source specific tree approach and the second is based on the group shared tree approach. The main discussion in this chapter is focusing on MPLS multicast fault tolerance.

It is worth to note that when applying the modified TSS for multicasting, if some links are shared among different trees, then the failure on these links may not prevent the receivers from being able to reconstruct the original data as shown in Figure 6.7.

Another advantage of using the modified threshold sharing scheme is the ability to prevent the single point of failure as has been shown for the group shared trees scenario. The proposed scheme is able to provide network fault tolerance in case of a failure of rendezvous point RP(s) which also depends on the threshold sharing level used.

Finally, the same mechanisms used in providing fault tolerance in MPLS multicasting can also be considered to provide multicast security.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

With the current demand on Internet Service Providers, which support MPLS technology, to provide Quality of Service (QoS) guarantees, fault tolerance and security become essential part of any network design. Fault tolerance is an important QoS factor that needs to be considered to maintain network survivability. It is the property of a system that continues to operate the network properly in the event of failure of some of its parts. Moreover, with increasing deployment of MPLS network, the security of traffic traversing through it has become a crucial concern.

However, providing fault tolerance and security in MPLS network is still accompanied by many problems and limitations. The solutions to these problems have to be proposed accordingly.

At the beginning of this thesis we have introduced some background information about MPLS networks. Both fault tolerance and security literature for MPLS have been investigated. The literature review on MPLS fault tolerance shows that many approaches have been proposed to protect the network from failure. In literature, there are two main

techniques used for providing recovery which are dynamic and pre-configured protection. Each of these techniques has pros and cons in terms of recovery time, packet loss, packet re-ordering, and bandwidth utilization. The technique that we are interested to compare with is the pre-configured (fast protection scheme) as it performs better than the dynamic technique for time sensitive applications in terms of previously mentioned factors. On the other hand, we also explored MPLS security approaches. The security issue in MPLS network is an evolving issue and has not been until now standardized.

Furthermore, a background on the threshold sharing is introduced. This scheme is used essentially to provide security for secret key sharing.

In this thesis, we have proposed a new method to provide fault tolerance and to enhance the security in MPLS networks. Our approach uses a modified *(k, n) threshold sharing scheme* combined with multi-path routing. An IP packet entering MPLS network is partitioned into n MPLS packets, which are assigned to node/link disjoint LSPs across the MPLS network. Receiving MPLS packets from k out of n LSPs are sufficient to reconstruct the original IP packet.

In order to provide fault tolerance, our scheme requires $n > k$. From security point of view, the modified TSS can provide data confidentiality, integrity, availability and IP spoofing. Providing data confidentiality does not require additional resources (i.e., $k = n$) while the other security aspects do.

Fault tolerance in MPLS can be supported using reasonable resources. The recovery from node/link failure can be achieved with no delay or packet loss. It should be mentioned that identifying packets in case of error transmission is considered in our

approach, however, packet re-ordering is not required if it is caused by failure. To verify that our approach does not require long processing time, we conducted simulations which show that modified TSS processing time does not affect significantly the packet transmission time. Extensions required supporting multi-path routing signaling in RSVP-TE and packet re-ordering requirements are also studied.

The impact of multi-path routing on MPLS security and fault tolerance has been investigated. The connection intrusion probability and connection failure probability have shown lower values when multi-path routing is used. The application of IPSec security protocol in MPLS networks has also been investigated and results showed that our proposed work performs better than IPSec in terms of bandwidth overhead, especially for small IP packet. As a conclusion, the need to provide security within the MPLS networks is required if the network domain is not trusted.

Finally, we have investigated the feasibility of applying the modified threshold sharing scheme on MPLS multicast networks. We have presented two application scenarios. The first scenario is based on the source specific tree approach and the second is based on the group shared tree approach. Our proposed work can provide fault tolerance with out producing recovery delay and packet loss, and it requires reasonable redundant bandwidth compared to other approaches. In addition, we also presented the use of multiple Rendezvous Points to support fault tolerance and security. We discussed the advantages of this approach in providing a protection against the single point of failure and availability.

In addition to security features such as confidentiality, integrity, and availability, our approach can avoid the case of relying on a single RP point which may be considered a

security threat if the RP is established in a non-trusted location or domain, or even if the RP itself is not trusted.

7.2 Future Work

As a future work, we intend to study additional extensions, enhancements and mechanisms required for the set up of multi-path routing and multiple multicast trees in MPLS.

Besides, more performance evaluation for other network topologies should be explored in order to strengthen this approach.

In terms of security aspect, we plan to investigate other security issues in MPLS such as user authentication and the security of signaling protocols used to convey multi-path and multicast routing information.

With regard to multicast application, the receivers in a group can join and leave a session anytime. Therefore, this issue has to be considered when applying our proposed scheme for multicast security.

We also plan to explore the security of Generalized Multiprotocol Label Switching (GMPLS) which is an open area of discussion. Moreover, we intend to study the security of MPLS when it belongs to heterogeneous networks. Some inter-provider regulations and standards may vary between each and another. Therefore, the managing between these networks is by itself a problem that needs to be clarified and hopefully standardized.

References

[1] E. Rosen et al, "Multiprotocol Label Switching Architecture", IETF, RFC 3031, 2001, http://www.ietf.org/rfc/rfc3031.txt (last time accessed: Feb. 2008).

[2] H. G. Lemma, "Enhanced Fast Rerouting Mechanisms for Protected Traffic in MPLS Networks", PhD Thesis, Technical University of Catalonia, Barcelona, Spain, 2003.

[3] B. Jamoussi, L. Anderson, R. Callon, R. Dantu, L. Wu, P. Doolan, "Constraint-Based LSP Setup using LDP", IETF, RFC 3212, January 2002.

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

[5] R. Bhandari, "Survivable Networks, Algorithms for Diverse Routing", Kluwer Academic Publishers, 1999.

[6] O. Bonaventure, C. Filsfils, P. Francois, "Achieving Sub-50 Milliseconds Recovery upon BGP Peering Link Failures", IEEE/ACM Transactions on Networking, Vol. 15, Issue 5, pp. 1123-1135, Oct. 2007.

[7] K. Makam, C. Huang, and V. Sharma, "Building Reliable MPLS Networks Using a Path Protection Mechanism," IEEE Communication Magazine, March 2002, pp. 156 – 162.

[8] D. Wang, G. Li, "Efficient Distributed Solution for MPLS Fast Reroute", Networking 2005, LNCS 3462, pp. 804-815.

[9] W. Grover, "Mesh-Based Survivable Networks, Options and Strategies for Optical, MPLS, SONET, and ATM Networking". Prentice Hall PTR, 2004.

[10] J. Vasseur, M. Pickavet, P. Demeester, "Network recovery: protection and restoration of optical, SONET-SDH, IP, and MPLS", Elsevier Inc 2004. Publisher: Morgan Kaufmann. ISBN: 012715051X.

[11] Y. Lui, D. Tipper, P. Siripongwutikorn, "Approximating Optimal Spare Capacity Allocation by Successive Survivable Routing", IEEE/ACM Transactions on Networking, Vol. 13, No. 1, pp.198-211, February 2005.

[12] A. Autenrieth, "Recovery Time Analysis of Differentiated Resilience in MPLS", Proceeding of the Fourth International Workshop on Design of Reliable Communication Networks, Alberta, Canada, pp. 333- 340, Oct. 2003.

[13] V. Sharma et al, "Framework for Multi-Protocol Label Switching (MPLS)-based Recovery", RFC3469, February 2003.

[14] L. Hundessa, J.D. Pascual, " Optimal and Guaranteed Alternative LSP for Multiple Failures", Proceedings ICCCN 2004, 13th International Conference on Computer Communications and Networks, Chicago, USA, October, 2004.

[15] G. Ahn, W. Chun, "Simulater for MPLS path restoration and performance evaluation", ATM (ICATM 2001) and High Speed Intelligent Internet Symposium, 2001, Joint 4th IEEE International Conference, pp. 32-36, April 2001.

[16] M. Kodiallam, T. Laksham, "Dynamic Routing of Bandwidth Guaranteed Tunnels with Restorations", Proceeding of IEEE, INFOCOM, pp. 902-911, March 2000.

[17] T. Oh, T. Chen, J. Kennington, "Fault Restoration and Spare Capacity Allocation with QoS Constraints for MPLS Networks", *Proceeding* of the IEEE GLOBECOM'00, pp. 1731-1735, Nov 2000.

[18] V. Rajeev, C.R. Muthukrishnan, "Reliable Backup Routing in Fault Tolerant Real-Time Networks," Proceeding of the Ninth IEEE International Conference on Networks, pp. 184-189, Oct. 2001.

[19] D. Haskin, "A method for setting an Alternative Label Switched Paths to Handle Fast Reroute," Internet Draft, July 2001.

[20] Y. Seok, Y. Lee, and N. Choi, "Fault –tolerant Multipath Traffic Engineering for MPLS Networks," IASTED International Conference on Communications, Internat, and Information Technology, USA, pp. 91-101, Nov. 2003.

[21] L. Li, M. Buddhikot, C. Chekuri, K. Guo, "Routing bandwidth guaranteed paths with local restoration in label switched networks," IEEE Journal on Selected Areas in Communications, Feb, 2005. pp. 437 – 449.

[22] A. Virk, and R. Boutaba, "Economical protection in MPLS networks," Computer Communications, Vol. 29, Issue 3, pp. 402-408, February 2006.

[23] A. Bremier-Brarr, Y. Afek, H. Kaplan, E. Cohen, M. Merritt, "Fast Recovery of MPLS Paths", *Proceedings* of the twentieth annual ACM symposium on Principles of distributed computing , Newport, Rhode Island, USA, August 2001.

[24] K-S Sohn, Y.N. Seung, D.K. Sung, "A distributed LSP scheme to reduce spare bandwidth demand in MPLS networks", IEEE Transactions on Communications, Vol. 54, Issue 7, pp. 1277 – 1288, July 2006.

[25] P. Han, H. Mouftah, "Reconfiguration of Spare Capacity for MPLS-based Recovery for Internet backbone Networks", IEEE/ACM Transaction on Networking, Vol.12, No.1, pp.73-85, February 2004.

[26] S.Wang, "Survivable Label Switching Networks", Ph.D Thesis, University of California, Los Angeles, 2002.

[27] D. Wang, F. Ergun, "Path protection with pre-identification for MPLS networks", proceedings of the 2nd Int'l Conf. on Quality of Service in Hetrogeneous Wired/Wireless Networks (QShine'05), August 2005.

[28] L. Ruan, Z Liu, "Upstream Node Initiated Fast Restoration in MPLS Networks", ICC2005, IEEE International Conference on Communications, Seol, May 2005.

[29] T. Shan, S. Rabie, "A Scheduling Mechanism for Service Aware Restoration of MPLS Networks", Proceeding of the Military Communications Conference, MILCOM October 2005.

[30] D. Wang, G. Guangzhi, "Efficient Distributed Solution for MPLS Fast Reroute", Proceeding of the 4th International IFIP-TC6 Networking Conference, Waterloo, Canada, May 2005.

[31] N. Maxemchuk, N. Murray, "Dispersity Routing on ATM Networks," INFOCOM'93, pp. 347-357, San Francisco, USA, 1993.

[32] T. Saad, B. Alawieh, H. Mouftah, "Tunneling Techniques for End-to-End VPNs: Generic Deployment in an Optical Testbed Environment", IEEE Communication Magazine, pp. 124-132, 2006.

[33] J. Chung, S. Panguluru, L. Dongfang, R. Garcia,"Multiple LSP Routing Network Security for MPLS Networking", IEEE-MWSCAS, pp. 605-608, 2002.

[34] "Security of the MPLS Architecture", Cisco Press, 2001.

[35] M. Behringer, M. J. Morrow, "MPLS VPN- Security", Cisco Press, 2005. ISBN: 1-58705-183-4.

[36] H. Lee; J. Hwang; B. Kang; K. Jun, "End-to-end QoS architecture for VPNs: MPLS VPN deployment in a backbone network", *Proceedings* the International Workshops on Parallel Processing, pp. 479 – 483, Canada, 2000.

[37] D. Barlow, V. Vassilio, H. Owen, "A cryptographic protocol to protect MPLS Labels", Proceeding of IEEE Workshop of Information Assurance, 2003.

[38] J. Chung, "Multiple LSP Routing Network Security for MPLS Networking", IEEE-MWSCAS, 2002.

[39] F. Palmieri , U. Fiore, "Enhanced Security Strategies for MPLS Signaling", Journal of Networks, Academy Publisher, Vol. 2, Issue 5, pp. 1-13, September 2007.

[40] F. Palmieri , U. Fiore, " Securing the MPLS Control Plane", HPCC 2005, LNCS 3726, Springer, pp. 511-523, 2005.

[41] S. Avallone, V. Manetti, M. Mariano, S. Romano, " A splitting infrastructure for load balancing and security in an MPLS network", 3rd International Conference on Testbeds and Research Infrastructure for the Development of Networks and communities, pp. 1-6, May 2007.

[42] J. Zhi, C. Lung, X. Xu, A. Srinivasan, Y. Lei, " Securing RSVP and RSVP-TE signaling protocols and their performance study", Proceeding of 3rd International Conference on Information Technology: Research and Education, pp. 90-94, 2005.

[43] B. Harman, L. Burness, G. Corliano, A. Murgu, F. El-moussa, L. He, "Securing Network Availability", BT Technology Journal, Springer, pp. 65-71, 2006.

[44] L. Fang, N. Bita, J. Roux, J. Miles, "Interprovider IP-MPLS services: requirements, implementations, and challenges", IEEE Communication Magazine, Vol. 43, Issue 6, pp. 119-128, June 2005.

[45] B. Daugherty, C. Metz, "Multiprotocol label switching and IP. Part I. MPLS VPNs over IP tunnels", IEEE Internet Computing, Vol. 9, Issue 3, pp. 68-72, June 2006.

[46] C. Phillips, J. Bigham, L. He and B. Littlefair, "Managing dynamic automated communities with MPLS-based VPNs", BT Technology Journal, , Springer,Vol. 24, Issue 2, pp. 79-84, 2006.

[47] A. Shamir, "How to share a secret", Communications of ACM, vol. 22, Issue 11, pp. 612-613, Nov. 1979.

[48] L. Zhou, Haas, Z, "Securing Ad Hoc Networks", IEEE Network, Volume 13, Issue 6, pp. 24-30, Nov.-Dec. 1999.

[49] J. Kang, D. Nyang, A. Mohaisen, Y. Choi, K. Kim, "Certificate Issuing Using Proxy and Threshold Signatures in Self-initialized Ad Hoc Network", Vol. 4707/2007 LNCS, Springer, pp. 886-899, 2007.

[50] C. Castelluccia , N. Saxena, J. Yi, "Robust self-keying mobile ad hoc networks", Computer Networks 51, Elsevier,1169–1182, 2007.

[51] K. Greenan, M. Storer, E.. Miller, C. Maltzahn, "Storing Data for the Long-term Without Encryption", Third IEEE International Security in Storage Workshop (SISW'05), pp. 12-20, 2005.

[52] T. Bheemarjuna, S. Sriramb, B. Manojc, C. Murthy, "MuSeQoR: Multi-path failure-tolerant security-aware QoS routing in Ad hoc wireless networks", Computer Networks, Vol. 50, Issue 9, pp. 1349-1383, June 2006.

[53] D. Sidhu, R. Nair, S. Abdallah, "Finding Disjoint Paths in Networks", Proceeding ACM-SIGCOMM'91 Symposium, pp. 43-51, 1991.

[54] M. Blesa, C. Blum, "Ant colony optimization for the- maximum edge-disjoint paths problem". In Raidl *et al.*, editor, 1st (EvoCOMNET'04), volume 3005 of Lecture Notes in Computer Science, pages 160-169, Coimbra, April 2004.

[55] G. Deyun, S. Yantai, L. Shuo , O. Yang, "Delay-based adaptive load balancing in MPLS networks", IEEE International Conference on Communications, ICC'02, pp. 1184-1188, USA, 2002.

[56] M. Lewis, "Troubleshooting Any Transport over MPLS Based VPNs", Cisco Press article, June 2005.

[57] T. Wong, "Decentralized Recovery for Survivable Storage Systems", Ph.D Thesis, Carnegie Mellon University, 2004.

[58] R. Kokku, T. Riche, A. Kunze, H. vin, "A case for run time adaptation in packet processing systems", ACM SIGCOMM Computer Communication Review, Vol. 34, Issue 1, pp. 107-112, January 2004.

[59] W. Lou, Y. Fang, "A multi-path routing approach for secure data delivery" IEEE MILCOM, Vol., pp. 1467-1473, 2001.

[60] Y. Shiloach, "A Polynomial Solution to the Undirected Two Paths Problem", Vol. 27, Issue 3, pp. 445 – 456, 1980.

[61] J. Park, H. Kim and H. Lim, "Many-to-Many Disjoint Path Covers in a Graph with Faulty Elements", book chapter, Lecture Notes in Computer Science, Springer Berlin / Heidelberg, Vol. 3341, pp. 742-753, 2004.

[62] R. Shenai and K. Sivalingam, "Hybrid Survivability Approaches for Optical WDM Mesh Networks," Journal of Lightwave Technol. Vol. 15, No. 10, 2005.

[63] M. Henning, "Graphs with large paired-domination number", Journal of Combinatorial Optimization, Springer Netherlands, Vol. 13, No., January, 2007.

[64] S. Bryant, G. Swallow, L. Martini, D. McPherson, "Pseudowire Emulation Edge-to-Edge (PWE3) Control Word for Use over an MPLS PSN", RFC4385, IETF, Feb. 2006.

[65] S. Deering, R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, IETF, 1998.

[66] G. Swallow, S. Bryant, L. Anderson, "Avoiding Equal Cost Multipath Treatment in MPLS Networks", RFC4928, IETF, June 2007.

[67] "MPLS Working Group", IETF, www.ietf.org/html.charters/mpls-charter.html.

[68] R. Aggarwal, D. Papadimitriou, S. Yasukawa, " Extensions to Resource Reservation Protocol – Traffic Engineering (RSVP-TE) for point to Multipoint TE Label Switch Paths (LSPs)", IETF, RFC 4875, May 2007.

[69] J. Yang, S. Papavassiliou, "Improving network security by multipath traffic dispersion", Proceeding of the Military Communication Conference (MILCOM), pp. 34-38, 2001.

[70] Microsoft TechNet: IPSec Architecture, http://technet.microsoft.com/en-us/library/bb726946.aspx, (last time accessed on April, 2008).

[71] E. Rosen, J. Clercq, Y. Joens, C. Sargor, " Architecture for the use of PE-PE IPSec Tunnels in BGP/MPLS VPNS", Internet Draft, IETF, 2005.

[72] D. Waitzman, C. Patridge, "Distance Vector Multicast Routing Protocol", RFC 1075, November 1988.

[73] J. Moy, "MOSPF: Analysis and Experience", RFC 1585, March 1994.

[74] A. Adams, J. Nicholas, W. Siadak, "Protocol Independent Multicast-Dense Mode (PIM-DM): Protocol Specification", RFC 3973, January 2005.

[75] B. Fenner, M. Handley, H. Holbrook, I. Kouvelas, "Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification (Revised)", RFC 4601, August 2006.

[76] A. Ballardie, "Core Based Trees (CBT) Multicast Routing Architecture", RFC 2201, September 1997.

[77] J. Cui, M. Faloutsos, M. Gerla, " An Architecture for Scalable, Efficient, and Fast Fault-Tolerant Multicast Provisioning", IEEE Networks, pp. 26-34, March/April 2004.

[78] I. PepeInjak, J. Guichard, J. Apcar, " MPLS and VPN Architectures", Cisco press, Vol. 2, 2003.

[79] Y. Pointer, "Link Failure Recovery for MPLS Networks with Multicasting", Master thesis, University of Virginia, August 2002.

[80] R. Deshmukh, A. Agarwal, "Failure Recovery in MPLS Multicast Network using Segmented Backup Approach", International Conference on Networking, pp. 344-349, Guadeloupe, French Carribbean, March 2004.

[81] M. Kodialam, T. Lakshman, "Dynamic routing of bandwidth guaranteed multicasts with failure backup", Proceeding 10th IEEE International Conference on Network Protocols, pp. 259-268, Nov. 2002.

[82] O. Banimelhem, A. Agarwal, J. Atwood, "A Tree Division Approach to Support Failure Recovery for Multicasting in MPLS Networks", Proceedings of the Systems Communications, ICW'05, pp. 249-254, 2005.

[83] W. Chao, W. Shong, C. Joan, "Centralized Control and Management Architecture Design for PIM-SM Based IP/MPLS Multicast Networks", IEEE Global Telecommunications Conference, GLOBECOM '07, pp. 443-447, Nov. 2007.

[84] X. Hesselbach, R. Fabregat, B. Baran, "Hashing based traffic partitioning in a multicast-multipath MPLS network model", LANC'05, October 10-12, 2005, Colombia, 2005.

[85] F. Font, D. Mlynek, "Choosing the set of rendezvous points in shared trees minimizing traffic concentration", IEEE International Conference on Communications, ICC '03, Vol. 3, pp. 1526-1530, 2003.

[86] A. Swan and L.A. Rowe, "The Case for a Multicast Session Layer", Open Mash Consortium, TR 2003-168, July 2003.

[87] N. Chefai, N. Georganas, G. Bochman, "Preemptive bandwidth allocation protocol for multicast, multi-streams environments", Proceedings of the ninth ACM international conference on Multimedia, pp. 528-530, 2001.

[88] R. Savarda, M. Karash, "Explaining the Gap between Specification and Actual Performance for IPSec VPN Systems", The Internet Security Conference Newsletter, Insight, Volume 3, Issue 9, 2001.

[89] M. Iwaki, K. Toraichi, R. Ishii, " Fast Polynomial Interpolation for Remez exchange Method", IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, pp. 411-414, 1993.

[90] Chapter 2: Finite Field Arithmetic, http://www.springer.com/?SGWID=2-102-45-110359-4, (last time accessed May, 2008).

[91] F. Faucheur *et al.* "Multiprotocol Label Switching (MPLS) Support of Differentiated Services", RFC 3270, IETF, 2002.

[92] R. Rivest, A. Shamir, L. Adleman. "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems". Communications of the ACM, Vol. 21 (2), pp.120 – 126. 1978.

[93] C. Madson, R. Glenn, " The use of HMAC-SHA-1-96 within ESP and AH", RFC 2404, IETF, 1998.

[94] L. Shen, X. Yang, B. Ramamurthy, "Shared Risk Link Group (SRLG)-Diverse Path Provisioning Under Hybrid Service Level Agreements in Wavelength-Routed Optical Mesh Networks", IEEE/ACM TRANSACTIONS ON NETWORKING, Vol. 13, No. 4, 2005, pp. 918-932.

[95] R. Prim, "Shortest connection networks and some generalisations. In: Bell System Technical Journal", 36, 1957, pp. 1389–1401.

[96] J. Kruskal, "On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem". In Proceedings of the American Mathematical Society, Vol 7, No. 1, Feb, 1956, pp. 48–50.

[97] B. Mukherjee, N. Singhal, L. Sahasrabuddhe, "Provisioning of survivable multicast sessions against single link failures in optical WDM mesh networks", Journal of Lightwave Technology, Vol. 21, Issue 11, 2003, pp. 2587-2594.

[98] H. Takahashi, A. Matsuyama, "An approximate solution for the steiner problem in graphs," Journal of Math. Japonica, pp. 573–577, 1980.

[99] S. Lee, C. Shong, "A K -Best Paths Algorithm for Highly Reliable Communication Networks", IEICE Trans. Commun., Vol. E82-B, No. 4., April 1999, pp. 586-590.

[100] D. Eppstein, "Finding the *k* smallest spanning trees", in the 2nd Scand. Worksh. Algorithm Theory, Lecture Notes in Computer science, 1990, pp. 38-47.

[101] N. Katoh, T. Ibaraki, H. Mine, "An Algorithm for Finding k Minimum Spanning Trees", SIAM J. Comput. 10, 1981, pp. 247-255.

[102] G. Frederickson, "Data Structures for On-Line Updating of Minimum Spanning Trees with Applications", SIAM J. Comput. 14 (4), 1985, 781-798.

Appendix (A)

The extended Euclidean algorithm for integers

Let a and b be integers, not both 0. The *greatest common divisor (gcd)* of a and b, denoted gcd(a,b), is the largest integer d that divides both a and b. Efficient algorithms for computing gcd(a,b) exploit the following simple result. [Chapter 2: Finite Field Arithmetic, http://www.springer.com/?SGWID=2-102-45-110359-4]

Theorem 1: Let a and b be positive integers. Then gcd(a,b) = gcd$(b- ca, a)$ for all integers c.

In the classical Euclidean algorithm for computing the gcd of positive integers a and b where $b \geq a$, b is divided by a to obtain a quotient q and a remainder r satisfying $b = qa+r$ and $0 \leq r < a$. By Theorem 1, gcd(a,b) = gcd(r,a). Thus, the problem of determining gcd(a,b) is reduced to that of computing gcd(r,a) where the arguments (r,a) are smaller than the original arguments (a,b). This process is repeated until one of the arguments is 0, and the result is then immediately obtained since gcd$(0,d)$ = d. The algorithm must terminate since the non-negative remainders are strictly decreasing. Moreover, it is efficient because the number of division steps can be shown to be at most $2k$ where k is the bit length of a.

The Euclidean algorithm can be extended to find integers x and y such that $ax+by = d$ where $d = \gcd(a,b)$. Algorithm 1 maintains the invariants $ax_1 + by_1 = u$, $ax_2 + by_2 = v$, $u \leq v$.

The algorithm terminates when $u = 0$, in which case $v = \gcd(a,b)$ and $x = x_2$, $y = y_2$ satisfy $ax + by = d$.

Algorithm 1 Extended Euclidean algorithm for integers

INPUT: Positive integers a and b with $a \leq b$.
OUTPUT: $d = \gcd(a,b)$ and integers x, y satisfying $ax + by = d$.

1. $u \leftarrow a$, $v \leftarrow b$.
2. $x_1 \leftarrow 1$, $y_1 \leftarrow 0$, $x_2 \leftarrow 0$, $y_2 \leftarrow 1$.
3. While $u \neq 0$ do
   3.1 $q \leftarrow \lfloor u/v \rfloor$, $r \leftarrow v - qu$, $x \leftarrow x_2 - qx_1$, $y \leftarrow y_2 - qy_1$.
   3.2 $v \leftarrow u$, $u \leftarrow r$, $x_2 \leftarrow x_1$, $x_1 \leftarrow x$, $y_2 \leftarrow y_1$, $y_1 \leftarrow y$.
4. $d \leftarrow v$, $x \leftarrow x_2$, $y \leftarrow y_2$.
5. Return(d, x, y)

The extended Euclidean algorithm for polynomials

Let a and b be binary polynomials, not both 0. The *greatest common divisor (gcd)* of a and b, denoted $\gcd(a,b)$, is the binary polynomial d of highest degree that divides both a and b. Efficient algorithms for computing $\gcd(a,b)$ exploit the following polynomial analogue of Theorem 2.

Theorem 2: Let a and b be binary polynomials. Then $\gcd(a,b) = \gcd(b - ca, a)$ for all binary polynomials c.

In the classical Euclidean algorithm for computing the gcd of binary polynomials a and b, where $\deg(b) \geq \deg(a)$, b is divided by a to obtain a quotient q and a remainder r satisfying $b = qa+r$ and $\deg(r) < \deg(a)$. By Theorem 2, $\gcd(a,b) = \gcd(r,a)$. Thus, the problem of determining $\gcd(a,b)$ is reduced to that of computing $\gcd(r,a)$ where the arguments (r,a) have lower degrees than the degrees of the original arguments (a,b). This process is repeated until one of the arguments is zero—the result is then immediately obtained since $\gcd(0,d)=d$. The algorithm must terminate since the degrees of the remainders are strictly decreasing. Moreover, it is efficient because the number of (long) divisions is at most k where $k = \deg(a)$. In a variant of the classical Euclidean algorithm, only one step of each long division is performed. That is, if $\deg(b) \geq \deg(a)$ and $j = \deg(b)-\deg(a)$, then one computes $r = b+z\,ja$. By Theorem 2, $\gcd(a,b) = \gcd(r,a)$. This process is repeated until a zero remainder is encountered. Since $\deg(r) < \deg(b)$, the number of (partial) division steps is at most $2k$ where $k = \max\{\deg(a),\deg(b)\}$. The Euclidean algorithm can be extended to find binary polynomials g and h satisfying $ag+bh = d$ where $d = \gcd(a,b)$. Algorithm 2 maintains the invariants

$$ag_1 + bh_1 = u$$
$$ag_2 + bh_2 = v.$$

The algorithm terminates when $u = 0$, in which case $v = \gcd(a,b)$ and $ag_2+bh_2 = d$.

Algorithm 2 Extended Euclidean algorithm for binary polynomials

---

INPUT: Nonzero binary polynomials a and b with $\deg(a) \leq \deg(b)$.
OUTPUT: $d = \gcd(a,b)$ and binary polynomials g, h satisfying $ag+bh = d$.

1. $u \leftarrow a$, $v \leftarrow b$.
2. $g_1 \leftarrow 1$, $g_2 \leftarrow 0$, $h_1 \leftarrow 0$, $h_2 \leftarrow 1$.
3. While $u \neq 0$ do
   3.1 $j \leftarrow \deg(u) - \deg(v)$.
   3.2 If $j < 0$ then: $u \leftrightarrow v$, $g_1 \leftrightarrow g_2$, $h_1 \leftrightarrow h_2$, $j \leftarrow -j$.
   3.3 $u \leftarrow u + z^j v$.
   3.4 $g_1 \leftarrow g_1 + z^j g_2$, $h_1 \leftarrow h_1 + z^j h_2$.
4. $d \leftarrow v$, $g \leftarrow g_2$, $h \leftarrow h_2$.
5. Return(d, g, h)

Suppose now that f is an irreducible binary polynomial of degree w and the nonzero polynomial a has degree at most $w-1$ (hence $\gcd(a, f) = 1$). If Algorithm 2 is executed with inputs a and f, the last nonzero u encountered in step 3.3 is $u = 1$. After this occurrence, the polynomials $g1$ and $h1$, as updated in step 3.4, satisfy $ag_1 + f h_1 = 1$. Hence $ag_1 \equiv 1 \pmod{f}$ and so $a^{-1} = g_1$. Note that h_1 and h_2 are not needed for the determination of g_1. These observations lead to Algorithm 3 for inversion in $GF(2^w)$.

Algorithm 3 Inversion in $GF(2^w)$ using the extended Euclidean algorithm

---

INPUT: A nonzero binary polynomial a of degree at most $w-1$.
OUTPUT: $a^{-1} \bmod f$.

1. $u \leftarrow a$, $v \leftarrow f$.
2. $g_1 \leftarrow 1$, $g_2 \leftarrow 0$.
3. While $u \neq 1$ do
   3.1 $j \leftarrow \deg(u) - \deg(v)$.
   3.2 If $j < 0$ then: $u \leftrightarrow v$, $g_1 \leftrightarrow g_2$, $j \leftarrow -j$.
   3.3 $u \leftarrow u + z^j v$.
   3.4 $g_1 \leftarrow g_1 + z^j g_2$.
4. Return(g_1).
