A Dynamic Traffic Re-assignment Algorithm Targeted for MPLS Networks

By

Phillip Brade

A thesis submitted in partial fulfillment of the
requirements for the degree of

Master of Science

in

Information Systems Science

Department of Systems and Computer Engineering

Carleton University

Ottawa, Ontario, K1S 5B6, Canada

October 11th, 2002

© 2002 Phillip Brade

National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

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

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

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

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

0-612-79808-9

Canada

The undersigned recommend the Faculty of Graduate studies and Research
the acceptance of the thesis

**A Dynamic Traffic Re-assignment Algorithm Targeted for
MPLS Networks**

Submitted by

Phillip Brade, BSc.

In partial fulfillment of the requirements for the degree of

Master of Science

Professor ChangCheng Huang

Thesis supervisor

A handwritten signature in black ink, consisting of a series of loops and a long horizontal stroke, positioned above a solid horizontal line.

Professor Rafik A. Goubran

Chair, Department of Systems and Computer Engineering

A handwritten signature in black ink, featuring a large circular loop followed by a long horizontal stroke, positioned above a solid horizontal line.

ABSTRACT

The label switched paths of Multi-Protocol Label Switching (MPLS) makes it an Internet technology that can facilitate Traffic Engineering. This research evaluates the performance of a dynamic Traffic Re-assignment algorithm targeted for Multi-Protocol Label Switching networks. It extends the concepts of Multi-path Adaptive Traffic Engineering (MATE) and Fast Acting Traffic Engineering (FATE) mechanisms. We present a stochastic framework for partitioning an aggregate flow of Expedited Forwarding (EF) traffic and Assured Forwarding (AF) traffic unto alternate label switched paths when a path is experiencing congestion. This algorithm aims to reduce delay and packet loss in the MPLS domain by temporarily shifting individual traffic to alternate under-utilized paths.

The performance of the proposed algorithm was compared to the MATE algorithm. The performance measures of response time, delay and loss were used. The simulation results demonstrate the advantages of using the Traffic Re-assignment algorithm over the MATE algorithm.

To my dear mother

ACKNOWLEDGEMENTS

I would like to express my sincere thanks to my thesis supervisor Dr. ChangCheng Huang for his guidance, encouragement, patience and support during this research. Without his support I would not be able to generate some of the new ideas to contribute to the knowledge of MPLS networks. Sincere thanks also to Professors Yiqiang Zhao and Chung-Horn Lung for their comments and technical observations. I would like to acknowledge the financial funding provided by the International Council for Canadian Studies for making my studies possible.

It is impossible for me to name every person who contributed to this work, but I would like to thank my fellow researchers for their encouragement and support and everyone else who have participated in making my time at Carleton University enjoyable.

TABLE OF CONTENTS

ABSTRACT	III
ACKNOWLEDGEMENTS.....	V
LIST OF TABLES	VIII
LIST OF FIGURES	IX
LIST OF ACRONYMS	XI
LIST OF MATHEMATICAL FIGURES	XIV
DEFINITION OF VARIABLES.....	XVI
CHAPTER 1 INTRODUCTION.....	1
1.1. BACKGROUND.....	1
1.2 THESIS OBJECTIVE.....	3
1.3 THESIS CONTRIBUTIONS	4
1.4 THESIS ORGANIZATION.....	5
CHAPTER 2 MULTI-PROTOCOL LABEL SWITCHING (MPLS).....	7
2.1 INTRODUCTION.....	7
2.2 MPLS ARCHITECTURE	9
2.3 FLOW AGGREGATION AND ROUTE SELECTION	11
2.4 LABEL DISTRIBUTION PROTOCOL (LDP).....	13
2.4.1 LSR Neighbour Discovery.....	14
CHAPTER 3 TRAFFIC ENGINEERING AND MPLS	15
3.1 IP AND TRAFFIC ENGINEERING.....	15
3.2 MPLS AND TRAFFIC ENGINEERING.....	17
3.3 IMPORTANCE OF TRAFFIC ENGINEERING	17
CHAPTER 4 MODELS FOR DYNAMIC TRAFFIC SHIFTING MODELS	21
4.1 FATE.....	21
4.2 FATE +.....	26
4.2.1 Reroute via Alternative Downstream LSR.....	27
4.2.2 Reroute via Alternative Upstream LSR.....	28
4.3 MATE.....	29
4.3.1 Filtering and Distribution	31
4.3.2 Traffic Measurement and Analysis.....	32
4.4 DYNAMIC TRAFFIC PARTITIONING.....	34
CHAPTER 5 PROPOSED TRAFFIC RE-ASSIGNMENT ALGORITHM.....	44

5.1 PROBLEM.....	44
5.2 DETAILS OF THE PROPOSED ARCHITECTURE.....	44
5.2.1 <i>Computation at the Intermediate LSRs</i>	50
5.5.2 <i>Traffic Shift at the Ingress Router</i>	54
CHAPTER 6 SIMULATION MODEL AND RESULTS	56
6.1 INTRODUCTION.....	56
6.2 MODEL DESCRIPTION	57
6.2.1 <i>The Network Model</i>	57
6.2.2 <i>The Node Model</i>	58
6.2.3 <i>The Process Model</i>	60
6.3 COMPUTATION AND SIMULATION RESULTS.....	72
6.3.1 <i>The Single-Node Model</i>	73
6.3.2 <i>The Two-Node Model</i>	82
6.3.3 <i>Comparison of the Proposed Traffic Re-assignment Algorithm with MATE</i>	92
6.3.4 <i>Output Analysis of the Simulation Results</i>	104
CHAPTER 7 DISCUSSION.....	120
CHAPTER 8 CONCLUSION AND FURTHER WORK.....	126
8.1 CONCLUSION.....	126
8.2 FURTHER WORK.....	128
REFERENCES.....	131

List of Tables

TABLE 5.1 SENSITIVITY VALUES FOR DIFFERENT SIMULATION RUNS.....	52
TABLE 6.1 BACKGROUND THROUGHPUT VARIATION THROUGH LSR 2	74
TABLE 6.2 BACKGROUND THROUGHPUT VARIATION THROUGH LSR 2	74
TABLE 6.3 INDIVIDUAL BACKGROUND THROUGHPUT VARIATION	83
TABLE 6.4 AGGREGATE BACKGROUND THROUGHPUT VARIATION	83
TABLE 6.5 INDIVIDUAL BACKGROUND THROUGHPUT VARIATION	84
TABLE 6.6 AGGREGATE BACKGROUND THROUGHPUT VARIATION	85
TABLE 6.7 COMPARISON OF MEAN RESPONSE TIMES FOR LARGE AF VARIATION	100
TABLE 6.8 COMPARISON OF MEAN RESPONSE TIMES FOR LARGE EF VARIATION.....	101
TABLE 6.9 RESPONSE TIMES FOR THE MATE ALGORITHM.....	108
TABLE 6.10 RESPONSE TIMES FOR THE TRAFFIC RE-ASSIGNMENT ALGORITHM	109
TABLE 6.11 AVERAGE PACKET DELAY FOR THE MATE ALGORITHM.....	112
TABLE 6.12 AVERAGE PACKET DELAY FOR THE TRAFFIC RE-ASSIGNMENT ALGORITHM	113
TABLE 6.13 LOSS RATIO FOR THE MATE ALGORITHM	116
TABLE 6.14 LOSS RATIO FOR THE TRAFFIC RE-ASSIGNMENT ALGORITHM.....	117

List of Figures

FIGURE 2.1 MPLS SHIM HEADER FORMAT	8
FIGURE 3.1 CONGESTION AVOIDANCE USING LABEL SWITCHED PATHS	19
FIGURE 4.1 TRAFFIC FLOW MODEL	21
FIGURE 4.2 BUFFER CONFIGURATION.....	22
FIGURE 4.3 RENEGOTIATION PROCESS	25
FIGURE 4.4 RE-ROUTE VIA ALTERNATIVE DOWNSTREAM LSR	28
FIGURE 4.5 RE-ROUTE VIA DIFFERENT LSR	29
FIGURE 4.6 TRANSIT NETWORK RUNNING MATE	30
FIGURE 4.7 MATE FUNCTIONS IN AN INGRESS NODE	31
FIGURE 4.8 DELAY MEASUREMENT IN MATE NETWORK.....	33
FIGURE 4.9 LOSS MEASUREMENT IN MATE NETWORK.....	34
FIGURE 4.10 M/M/1 QUEUING NETWORK	35
FIGURE 4.11 M/M/1/K QUEUING NETWORK	35
FIGURE 4.12 CHANGE IN QUEUE ARRIVAL RATES ON A SINGLE LSP	39
FIGURE 4.13 CHANGE IN THE SOURCE AND QUEUE ARRIVAL RATES FOR N LSPs.....	41
FIGURE 5.1 STOCHASTIC MODEL FOR THE TRAFFIC RE-ASSIGNMENT ALGORITHM	45
FIGURE 5.2 CONGESTION AND NOTIFICATION PROCESS IN THE TRAFFIC RE-ASSIGNMENT NETWORK	47
FIGURE 5.3 QUEUE CONFIGURATION.....	49
FIGURE 6.1 NETWORK MODEL.....	58
FIGURE 6.2 THE INGRESS NODE MODEL.....	59
FIGURE 6.3 THE LSR NODE MODEL	60
FIGURE 6.4 THE INGRESS PROCESS MODEL	63
FIGURE 6.5 FLOW CHART FOR TRANSMITTER STATE.....	64
FIGURE 6.6 THE LSR PROCESS MODEL.....	65
FIGURE 6.7 FLOW CHART FOR ARRIVAL STATE	68
FIGURE 6.8 FLOW CHART FOR SERVICE-COMPLETE STATE.....	70
FIGURE 6.9 SOURCE PACKET FORMAT	71
FIGURE 6.10 NOTIFICATION PACKET FORMAT.....	72
FIGURE 6.11 THE SINGLE-NODE MODEL	73
FIGURE 6.12 VARIATION OF BACKGROUND THROUGHPUT AT LSR 2.....	76
FIGURE 6.13 VARIATION OF LOSS RATIO AT LSR 2	76
FIGURE 6.14 VARIATION OF PACKET DELAY AT LSR 2	77
FIGURE 6.15 INGRESS VARIATION	77
FIGURE 6.16 VARIATION OF BACKGROUND THROUGHPUT AT LSR 2.....	79
FIGURE 6.17 VARIATION OF LOSS RATIO AT LSR 2.....	79
FIGURE 6.18 VARIATION OF AVERAGE PACKET DELAY AT LSR 2	80

FIGURE 6.19 INGRESS VARIATION	80
FIGURE 6.20 THE TWO-NODE MODEL.....	82
FIGURE 6.21 INDIVIDUAL BACKGROUND THROUGHPUT AT LSR 2	86
FIGURE 6.22 INDIVIDUAL BACKGROUND THROUGHPUT AT LSR 3	86
FIGURE 6.23 AGGREGATE BACKGROUND THROUGHPUT AT LSR 3	87
FIGURE 6.24 VARIATION OF LSP LOSS RATIO	87
FIGURE 6.25 VARIATION OF INGRESS SOURCE THROUGHPUT.....	88
FIGURE 6.26 INDIVIDUAL BACKGROUND THROUGHPUT AT LSR 2	89
FIGURE 6.27 INDIVIDUAL BACKGROUND THROUGHPUT AT LSR 3	90
FIGURE 6.28 AGGREGATE BACKGROUND THROUGHPUT AT LSR 3	90
FIGURE 6.29 VARIATION OF PACKET DELAY ALONG LSP	91
FIGURE 6.30 VARIATION OF INGRESS SOURCE THROUGHPUT.....	91
FIGURE 6.31 AVERAGE PACKET DELAY VARIATION FOR LARGE EF VARIATION.....	93
FIGURE 6.32 LOSS VARIATION FOR LARGE EF VARIATION	94
FIGURE 6.33 AVERAGE PACKET DELAY VARIATION FOR LARGE AF VARIATION.....	95
FIGURE 6.34 LOSS VARIATION FOR LARGE AF VARIATION.....	96
FIGURE 6.35 CONGESTION DETECTION FOR TRAFFIC RE-ASSIGNMENT ALGORITHM.....	97
FIGURE 6.36 CONGESTION DETECTION FOR MATE ALGORITHM	98
FIGURE 6.37 MEAN RESPONSE TIMES WITH STRICTLY EF TRAFFIC IN NETWORK.....	102
FIGURE 6.38 MEAN RESPONSE TIMES WITH STRICTLY AF TRAFFIC IN NETWORK.....	103
FIGURE 6.39 MEAN RESPONSE TIMES WHEN BOTH AF AND EF TRAFFIC ARE USED IN THE NETWORK	104

List of Acronyms

AF	Assured Forwarding
ATM	Asynchronous Transfer Mode
AS	Autonomous System
BGP	Border Gateway Protocol
CR-LDP	Constraint-based Routed Label Distribution Protocol
CR-LSP	Constraint-based Routed Label Switch Path
Diffserv	Differentiated Services Model
ECMP	Equal Cost Multi-path
EF	Expedited Forwarding
EGP	Exterior Gateway Protocol
FATE	Fast Acting Traffic Engineering
FEC	Forward Equivalence Class
FIFO	First-In First-Out
IETF	Internet Engineering Task Force
IGP	Interior Gateway Protocol
IP	Internet Protocol
IS-IS	Intermediate System-Intermediate System
ISP	Internet Service Provider

LIB	Label Information Base
LAN	Local Area Network
LDP	Label Distribution Protocol
LER	Label Edge Router
LSP	Label Switched Path
LSR	Label Switching Router
MATE	Multi-path Adaptive Traffic Engineering
MPLS	Multi-Protocol Label Switching
NHLFE	Next Hop Label Forwarding Entry
OSPF	Open Shortest Path First
OPNET	Optimized Network Engineering Tools
QoS	Quality of Service
RED	Random Early Detection
RSVP	ReSerVation Protocol
RSVP-TE	ReSerVation Protocol and Traffic Engineering
SLA	Service Level Agreement
SONET	Synchronous Optical Network
TA	Traffic Re-Assignment algorithm
TCP	Transmission Control Protocol
TLV	Type Length Value

TOS	Type of Service
TTL	Time-to-Live
UDP	User Datagram Protocol
WAN	Wide Area Network

List of Mathematical Figures

λ	Source rate
$\lambda_{increase}$	Increase in source rate
λ_i	Ingress source rate along LSP i
$\gamma_{i,j}$	Background source rate at queue j of LSP i
P	Traffic mapping vector
R_i	The number queues traversed by the ingress flow along LSP i
τ_{avg}	Average packet delay in the network
$\rho_{i,j}$	Utilization of queue j on LSP i
S_Γ	Loss sensitivity due to background traffic
S_π	Loss sensitivity due to ingress traffic
S_γ	Delay sensitivity due to background traffic
S_λ	Delay sensitivity due to ingress traffic
K	Queue capacity
L	Loss ratio
$\Delta\gamma_{EF_j}$	Increase in the background EF traffic at LSR j
$\Delta\gamma_{AF_j}$	Increase in the background AF traffic at LSR j

$Y_{k,i}$	k^{th} sample of system i
θ_j	Mean output from system j
D_p	Differences in output for replication number p
$\overline{D}$	The point estimate
$\overline{Y}_{p,i}$	Sample mean performance measure for system i
S_D^2	Sample variance of the differences
$s.e(\overline{D})$	Standard error

Definition of Variables

λ (λ):	Source rate
$\Delta\gamma_{EF}$ ($\Delta\gamma_{EF}$):	Variation of the EF background traffic
$\Delta\gamma_{AF}$ ($\Delta\gamma_{AF}$):	Variation of the AF background traffic
$\Delta\lambda_{EF}$ ($\Delta\lambda_{EF}$):	Amount of EF ingress traffic shift
$\Delta\lambda_{AF}$ ($\Delta\lambda_{AF}$):	Amount of AF ingress traffic shift
$\Delta\lambda$ ($\Delta\lambda$):	Amount of aggregate flow shifted at the ingress
num_trans :	Number of packets transmitter at the queue
$num_dropped$:	Number of packets dropped at the queue

Chapter 1 Introduction

1.1. Background

The evolution of the networking environment has led to the transmission of increasingly more mission critical traffic over networks that use the Internet Protocol. Although IP networks offer flexibility and scalability, the existing IP networks need to be enhanced in the areas of availability, dependability and Quality of Service (QoS), in order to provide a mission-critical networking environment [1]. In these traditional networks, the software-based forwarding is too slow to handle the large traffic loads in the Internet. Although techniques such as fast-table lookup were implemented in these systems, the load on the router is often more than the router can handle. This sometimes leads to loss of traffic, lost of connections and overall poor performance. Also, the IP-based networks offer more delays and jitters to mission-critical traffic such as speech. As the packet traverses through the nodes of these networks, it encounters both delay and variable delay depending on how long it takes the table lookup and the number of packets that must be processed in a given period. The overall effect is the accumulation of the variable delays that are encountered at each node between the sender and receiver. The increasing number of new users signing on to the Internet has also increased the task of the routers and the complexity of the address tables in these systems.

The features of Multi-protocol Label Switching (MPLS) have addressed many of the limitations of the traditional IP networks. MPLS can be described as a technology that promises the speed and flexibility of future networks [2]. MPLS is the convergence of connection-oriented forwarding techniques and the Internet routing protocols. The most prominent pre-standard incarnations of MPLS, leveraged the high-performance cell switching capabilities of Asynchronous Transfer Mode (ATM) switch hardware [3], and melded them together into a network using existing IP routing protocols [4]. Currently, IP traffic is routed and forwarded over the Internet using standard IP routing protocols.

Interior Gateway protocols (IGPs) such as Open Shortest Path First (OSPF) are used within Autonomous Systems (ASs), while Exterior Gateway protocols (EGPs) are used to interconnect ASs. IGPs and EGPs offer little or no readily available traffic control capabilities, such as reporting and incorporating in routing, the information on the network resources availability or utilization. The routing algorithms such as OSPF used by routing protocols tend to converge traffic on the same network links or interfaces, which contributes to congestion and unbalanced network resource utilization.

Features that can easily load balance, support dynamic fall over to backup paths or re-optimize the traffic routes for better network resource utilization and transport performance are not provided by the existing connectionless router networks. This makes it difficult for network administrators to implement effective policies to address network performance

problems [1]. Inadequate Traffic Engineering capability results in poor QoS performance, inefficiency and difficulty in network resources management.

Several researchers have proposed to add Traffic Engineering capabilities in traditional datagram networks using shortest path algorithms. Although such schemes have shown to improve the efficiency of the network, they suffer from several limitations including:

1. Load sharing cannot be accomplished among paths of different costs,
2. Traffic/policy constraints are not taken into account,
3. Modifications of link metrics to re-adjust traffic mapping tend to have network-wide effects, causing undesirable and unanticipated traffic shifts, and
4. Traffic demands must be predictable and known a priori.

MPLS performs Traffic Engineering for IP networks by providing for Constraint-based Routing Label Switched Paths (CR-LDPs) with associated Traffic Engineering constraints. In MPLS networks, the Traffic Engineering building block is a Label Switched Path (LSP) that can be manipulated, and managed by network administrators to direct the traffic. It is based on the limitations of IP networks and the features of MPLS, that we are proposing this MPLS-based Traffic Re-assignment scheme.

1.2 Thesis Objective

This work presents a method for dynamically shifting traffic away from a congested LSP to an alternate under-utilized path and is targeted for the MPLS domain. It extends the

methodology proposed in [5] to cover the simultaneous flow of different classes of traffic along the LSPs in the MPLS domain. Given the user's desired QoS and the network resources availability, our proposed procedure is able to compute the amount of each class of traffic that should be shifted away from a congested LSP and redistribute it unto the alternate LSP. The solution is based on a linear approximation and we use a first order approximation for the system dynamics to model the effect of changes.

1.3 Thesis Contributions

The first contribution is the design of a monitoring and measurement scheme at the intermediate LSRs along the label switched paths. The intermediate LSRs are able to monitor the packet loss ratios and the average packet delays of the AF and EF traffic respectively. In comparison to the MATE algorithm, which does the measurement of the loss and delay at the egress router, this measurement scheme provides the advantage of a faster response to congestion at the LSRs. The intermediate LSRs are also capable of measuring the variation of the individual background traffic that are transmitted through them and compute the amount of traffic that should be shifted away from them if they are experiencing congestion. This feature is not present in the MATE algorithm.

A second contribution of this thesis is a method for shifting the individual EF and AF traffic at the ingress router to reduce the average packet delay and loss respectively. The ingress router compares the amount of EF and AF traffic that should be shifted after

receiving these values from the congested downstream LSRs. If the amount of EF traffic that needs to be shifted is greater than or equal to the amount of AF traffic that needs to be shifted, the ingress router only shifts the equivalent amount of EF traffic. If however, the amount of EF traffic is less than the amount of AF traffic, the ingress router shifts the equivalent amount of EF traffic and the difference in the amounts of the AF and EF traffic for the AF traffic. This proposed method conserves the available bandwidth along the congested path and does not lead to the over-utilization of the available bandwidth on the alternate label switched path. The MATE algorithm assumes the shift of the aggregate traffic at the ingress router.

The third contribution of this thesis is the design of a stochastic model of the proposed architecture and the fourth is the development of the simulation model using the network simulator, OPNET. The process models are programmed with C/C++ language.

1.4 Thesis Organization

The remaining chapters of the thesis are organized as follows:

Chapter 2: Introduces the concepts of MPLS.

Chapter 3: Discusses Traffic Engineering in MPLS networks.

Chapter 4: Reviews the Dynamic Traffic Shifting models used in the research.

Chapter 5: Describes the Proposed Traffic Re-assignment scheme.

Chapter 6: Gives the simulation model and the results.

Chapter 7: Gives the discussion of the related work.

Chapter 8: Gives the conclusion and the recommendations for future work.

Chapter 2 Multi-Protocol Label Switching (MPLS)

2.1 Introduction

Multi-protocol Label Switching (MPLS) is an emerging technology that aims to address many of the existing issues associated with packet forwarding in today's internetworking environment. The MPLS architecture describes the mechanisms to perform label switching, which combines the benefits of packet forwarding based on Layer 2 switching with the benefits of Layer 3 routing, [6]. Similar to Layer 3 networks (for example, Frame relay or ATM), MPLS assigns labels to packets for transport across packet- or cell-based networks. The forwarding mechanism throughout the network is label swapping, in which units of data (for example a packet or cell) carry a short fixed-length label that tells switching nodes along the path how to process the forwarding data.

The significant differences between MPLS and traditional WAN technologies are the way the labels are assigned and the capability to carry a stack of labels attached to the packet. The concept of label stack enables new applications, such as Traffic Engineering, Virtual Private Networks and fast re-routing around link and node failures, [4], [7] and [8]. Packet forwarding in MPLS is in stark contrast to today's connectionless network environment, where each packet is analyzed on a hop-by-hop basis, its Layer 3 header is checked, and an independent forwarding decision is made based on the information extracted from the network layer routing algorithm.

The architecture is split into two separate components: the Forwarding component (also called the data plane) and the Control component (also called the control plane), [6]. The Forwarding component uses a label-forwarding database maintained by a label switch to perform the forwarding of data packets based on labels carried by packets. The Control component is responsible for creating and maintaining label-forwarding information (referred to as bindings) among a group of interconnected label switches.

Figure 2.1 MPLS shim header format

The label field of the MPLS shim header consists of 20 bits and carries the label. The 3-bit Experimental field is not fully defined but can be used to implement different service class using the Diffserv model, [9] or to signal congestion, [10]. The one bit Stack field is used to identify the end of stack when multiple labels are used and the Time-to-Live (TTL) field provides conventional IP time-to-live functionality.

The MPLS nodes run one or more IP routing protocols (or rely on static routing) to exchange routing information with other MPLS nodes in the network. In this sense, every

MPLS node is an IP router on the control plane. In an MPLS node, the IP routing table is used to determine the label binding exchange, where adjacent MPLS nodes exchange labels for individual subnets that are contained within the IP routing table. The MPLS IP routing control process uses labels exchanged with adjacent MPLS nodes to build the Label Forwarding table, which is the forwarding plane database that is used to forward packets through the MPLS network.

2.2 MPLS Architecture

Several new terms are introduced to describe the devices that make up the architecture and the services offered by this technology. Some of these new terms describe the functionality of each device and their roles within the MPLS domain.

Label Switching Router (LSR): any router or switch that implements label distribution procedures and can forward packets based on labels. The basic function of label distribution procedures is to allow an LSR to distribute its label bindings to other LSRs within the MPLS network, [6].

Edge-LSR: can receive an IP packet, perform Layer 3 lookups, and impose a label stack before forwarding the packets into the MPLS domain. It can also receive a labelled packet, remove labels, perform Layer 3 lookups, and forward the IP packet towards its next-hop.

Forwarding Equivalence Class (FEC): a group of IP packets that are forwarded in the same manner (e.g., over the same path, with the same forwarding treatment).

Label Switched Path (LSP): the path through one or more LSRs at one level of the hierarchy followed by packets in a particular FEC.

MPLS Edge node: an MPLS node that connects an MPLS domain with a node, which is outside of the domain, either because it does not run MPLS, and/or because it is in a different domain.

MPLS label: a label which is carried in a packet header, and which represents the packet's FEC.

Some FECs correspond to address prefixes that are distributed via a dynamic routing protocol. The set up of LSPs for these FECs can be done in two ways: Independent LSP Control or Ordered LSP Control.

In Independent LSP Control, each LSR upon noting that it recognizes a particular FEC, makes an independent decision to bind a label to that FEC and to distribute that label to its label distribution peers. In Ordered LSP Control, a LSR only binds a label to a particular FEC if it is the egress LSR for that FEC, or if it has already a label binding for that FEC from its next hop for that FEC.

Ordered LSP Control is used if one wants to ensure that traffic in a particular FEC follows a path with the same specified set of properties (e.g. that the traffic does not traverse any node twice, that a specified amount of resources is available to the traffic, that the traffic follows an explicitly specified path, (etc). In Independent LSP control, some

LSRs may begin label switching the traffic in the LSP before the LSP is completely set up, and thus some traffic in the FEC may follow a path that does not have the specified set of properties. Ordered LSP Control is also used if the recognition of the FEC is a consequence of the setting up of the corresponding LSP. Ordered LSP set up may be initiated by either the ingress or the egress node. Ordered LSP Control and Independent LSP Control are fully inter-operable. However, unless all LSRs in an LSP are using Ordered LSP Control, the overall effect on network behaviour is largely that of independent control, since one cannot be sure that an LSP is not being used until it is fully set up, [6].

2.3 Flow Aggregation and Route Selection

Classifying traffic into different FECs is an important aspect of MPLS technology. One way traffic is partitioned into FECs is by creating a separate FEC for each address prefix that appears in the routing table. Within a particular MPLS domain, this may result in a set of FECs in which all traffic in all those FECs follow that same route. In this case, the union of those FECs is itself an FEC. This creates a choice; should a distinct label be bound to each component FEC, or should a single label be bound to the union, and that label applied to all traffic in the union.

According to the MPLS Architecture draft, [6] the procedure for binding a single label to a union of FECs which is itself an FEC, and of applying that label to all traffic in the

union, is known as aggregation. Aggregation may reduce the number of labels needed to handle a particular set of packets and may also reduce the amount of label distribution control traffic needed. Given a set of FECs that are “aggregatable”, it is possible to aggregate them into a single FEC, aggregate them into a set of FECs or not aggregate them at all. Thus we can speak of the granularity of aggregation.

In [6], two options for route selection are proposed. The first proposed route selection is Hop-by-Hop routing that allows each node to independently choose the next hop for each FEC. This is the usual mode today in existing IP networks, [11], [12], [13] and [14]. A “hop-by-hop routed LSP” is an LSP whose route is selected using Hop-by-Hop routing. The other route selection option is Explicit routing which does not allow each LSR to choose the next hop but rather a single LSR, generally the LSP ingress or egress, which specifies some or all of the LSRs in the LSP.

Our emphasis in this work will be on Explicit routing since it facilitates Traffic Engineering in the MPLS domain. In MPLS, the explicit route needs to be specified at the time that the labels are assigned, but the explicit route does not have to be specified with each IP packet, [15]. This makes MPLS Explicit routing much more efficient than the alternative of IP source routing.

2.4 Label Distribution Protocol (LDP)

Label distribution protocol is used between nodes in an MPLS network to establish and maintain the label bindings. In order for MPLS to operate correctly, label information needs to be transmitted reliably, and the label distribution protocol messages pertaining to a particular FEC need to be transmitted in sequence. Flow control is also desirable, as is the capability to carry multiple label messages in a single datagram. E. Rosen et. al., [6] propose the use of TCP as the underlying transport to meet these goals.

The LDP specification is the work of a design team formed after the second MPLS working group meeting. LDP is based on control driven protocols, [16]. LDP has the following characteristics:

1. It provides an LSR “discovery” mechanism to enable LSR peers to find each other and establish communication.
2. It defines four classes or messages:
 - DISCOVERY messages
 - ADJACENCY messages, which deal with initialization, keepalive, and shut down of sessions between LSRs
 - LABEL ADVERTISEMENT messages, which deal with label binding advertisements, requests, withdrawal, and release

- NOTIFICATION messages, used to provide advisory information and to signal error information
3. It runs over TCP to provide reliable delivery of messages (with the exception of DISCOVERY messages).
 4. It is designed to be easily extensible, to using messages specified as collections of TLV (Type, Length, Value) encoded objects.

The TLV encoding means that each object contains a Type field to say what sort of object it is (e.g., a label binding), a Length field to say how long the object is, and a Value field, the meaning of which depends on the type, [17].

2.4.1 LSR Neighbour Discovery

The LDP's discovery protocol runs over UDP. An LSR periodically multicasts a HELLO message to a well-known UDP port on the "all routers on this subnet" multicast group. All LSRs listen on this UDP port for the hello messages. Thus, at some point a LSR will learn about all the other LSRs to which it has direct connections. When the LSR learns the address of another LSR by this mechanism, it establishes a TCP connection to that LSR. At this point a LDP session can be established between two LSRs. A LDP session is bi-directional, that is, the LSR at each end of the connection can advertise and request bindings to and from the peer at the other end of the connection.

Chapter 3 Traffic Engineering and MPLS

3.1 IP and Traffic Engineering

Three complimentary technical instruments that Internet Service Providers (ISPs) have employed to respond to the challenges of the Internet growth are Network Architecture, Capacity Expansion and Traffic Engineering. Network Architecture deals with the abstract structure of networks, the components of object classes of the network, their functions, and the relationships between them. Good, scalable network architectures, premised on sound architectural principles, are imperative in the quickly evolving Internet environment.

The second instrument employed by large ISPs to respond to traffic growth is Capacity Expansion. In 1996, most large ISPs in the United States operated backbones with DS3 (44.736 Mb/s) links. In 1997 and 1998, OC-12c (622 Mb/s) links became pervasive. In 1999 a number of major ISPs upgraded to OC-48c (2,488.32 Mbps) links, [1].

The third instrument employed by service providers to address the Internet growth challenge is Traffic Engineering. This subject has attracted significant attention in recent times [1], [5], [8], [11], [18] and [19]. A motivation for doing traffic engineering is the realization that architectural paradigms and simple capacity expansion are necessary but not sufficient, to deliver high quality Internet service under all circumstances. Internet Traffic Engineering is the aspect of Internet Network Engineering that addresses the issue of performance optimization in Operational networks. It encompasses the application of

technology and scientific principles to the measurement, modeling, characterization, and control of Internet traffic, [18]. It also includes the application of knowledge and techniques to achieve specific performance objectives (including reliable and expeditious movement of traffic through the network), efficient utilization of network resources, and planning of network capacity. Ultimately, good Traffic Engineering increases the value of a network to both the service provider and the Internet user community.

Historically, Traffic Engineering has been difficult to achieve in public IP networks. The reason for this is the limited functional capabilities of conventional IP technologies. One particular shortcoming of conventional IP systems is the inadequacy of measurement functions. The limitations on intra-domain routing control functions are another issue of conventional IP systems. IGPs such as Intermediate System-Intermediate System (IS-IS) and Open Shortest Path First (OSPF), commonly used to route traffic within ASs in the Internet, are topology-driven and employ per-packet progressive connection control. Each router makes independent routing decisions using local instantiation of a synchronized routing area link-state database. Route selection is based on shortest path computations using simple additive link metrics, [11]. This approach is highly distributed and scalable, but flawed. The flaw is that these protocols do not consider the characteristics of offered traffic and network capacity constraints when making routing decisions. This results in subsets of network resources becoming congested, while other resources along the alternate

paths remain underutilized, [8], [18] and [19]. This kind of congestion is a symptom of poor resource allocation, and is an issue that Traffic Engineering specifically attempts to redress.

3.2 MPLS and Traffic Engineering

Recent developments in MPLS, [5], [8], [11] and [18] have opened new possibilities to address some of the limitations of IP systems concerning Traffic Engineering. A Framework for MPLS is provided in [20] and [21] and the Architecture for it is described in [6]. The requirements for Traffic Engineering over MPLS were articulated in [18]. Although MPLS is a relatively simple technology (based on classical label swapping), it enables the introduction of sophisticated control capabilities that advance the Traffic Engineering function in IP networks, [18] and [20]. A particular interesting aspect of MPLS is that it efficiently supports origination connection control through Explicit LSPs. When MPLS is combined with Differentiated Services and Constrained Based routing, they become powerful and complimentary abstractions for quality of service (QoS) provisioning in IP networks. In this work the Traffic Engineering function decides when and how to shift traffic among the LSPs. There are several uses for MPLS Traffic Engineering, [12].

3.3 Importance of Traffic Engineering

Traffic Engineering is an iterative process of Network Planning and Network Optimization. Network Planning improves the topology and link capacity of a network in a systematic

way so that the network is robust, adaptive and easy to operate. Generally, Network Optimization improves the mapping performance and distribution of traffic over the existing network infrastructure. This leads to the avoidance or reduction of congestion levels in the network and to satisfactory service delivery.

Network Optimization is necessary, because it does not matter how well a network is designed, random incidents such as fibre cuts or changes in traffic demand will occur. When they occur, they cause congestion and other problems to manifest in an operational network. This is aggravated by the fact that current IGPs always use the shortest paths to forward traffic.

Traffic Engineering is difficult to do with IGP in large networks because among the Equal-Cost Multi-Paths (ECMPs) from the source, every path will have an equal share of load. This equal ratio cannot be changed. Therefore, one of the paths may end up carrying significantly more traffic than other paths because it also carries traffic from other sources. Load sharing cannot be done among multiple paths of different cost.

Combining MPLS, Constraint-Based routing and an enhanced state IGP, Traffic Engineering can be done effectively, [22] and [23]. For example by setting reserved bandwidth of each link, and setting the bandwidth requirement for each LSP, Constraint-Based routing will automatically avoid too many LSPs on any link, [24]. This solves the

problem of congestion on the links. In figure 3.1, Constraint-Based routing will automatically place LSP LG on a longer path to avoid congestion in link MG.

Figure 3.1 Congestion avoidance using label switched paths

Traffic Engineering in IP networks is a challenging problem. Singularities and discontinuities characterize Internet growth. Very rapid growth occurs over relatively short periods of time. The rapid growth is then followed by modest growth over relatively longer intervals of time. Accurate forecasting is therefore quite difficult. Furthermore, Internet traffic exhibits very dynamic behaviour with characteristics that are not yet understood. Traffic also tends to be highly asymmetric.

The operational environment is also in a continual state of flux. New resources are added constantly. Resources also fail regularly. New Internet applications with bandwidth

requirements (which may have significant global impact) are introduced all the time. Additional complications are introduced by inter-domain traffic traversing autonomous systems' boundaries. These environmental factors result in the network topology not usually correlating with the traffic matrix. Addressing these issues require continual monitoring and performance optimization of public IP networks. The following chapter discusses some of the novel methods proposed for addressing the problem of congestion in the MPLS domain, due to the dynamic changes of traffic along the LSPs.

Chapter 4 Models for Dynamic Traffic Shifting Models

4.1 FATE

FATE (Fast Acting Traffic Engineering) is a set of mechanisms that can be employed to either pre-empt congestion or respond to its occurrence in a LSR along which a CR-LSP has already been established, [25]. Each LSR and also the ingress and egress router have two buffers, one catering for High Priority traffic, i.e., loss stringent applications and one for Best Effort traffic.

Figure 4.1 Traffic flow model

Figure 4.1 depicts the traffic flow between the sources and the destinations passing through the two different buffers at each LSR. The LSPs passing through each buffer

could represent an aggregate of LSPs from a number of customer sites. FATE assumes that explicit routing is used in the route selection and this allows the LER to specify all the LSRs along the LSP. Additional to the path selection process, FATE allows the ingress router to identify the class of LSP and so enable it to be distributed through the appropriate buffer at each LSR. The buffer configuration is shown in figure 4.2 below.

Figure 4.2 Buffer Configuration

Each LSR periodically calculates the packet loss in the individual LSPs passing through each input buffer stream. If the value exceeds a given threshold, i.e., the loss probability

assigned to that particular buffer, it is taken as a sign of possible congestion in that buffer stream within the LSR.

The number of packets belonging to the Best Effort LSP and the number of packets destroyed due to the failed insertion are recorded at the input buffers. These values are used to calculate the packet loss in the LSP just before the packet is forwarded to the output buffer.

Once the packet loss in the Best Effort LSP has arisen above the predetermined threshold value, for an extended time period, the LSR creates a notification message and sends it to the ingress router. When the ingress router receives this notification message it can perform one of the three decisions:

1. Decide that the packet loss it is currently experiencing remains sufficiently low and it continue to meet its SLA requirement, permitting no further action to be taken.
2. Re-negotiate for new quality requirements along the existing LSR
3. Negotiate for new quality requirement along an alternative LSP

The information carried in this notification message is the identity of the LSP that is experiencing congestion, the current loss in the buffers, the LSR this loss is occurring in and the current loss the LSP is experiencing. FATE allows each intermediate LSR upon receipt of a notification message to append relevant information to it concerning their status

if they are experiencing congestion. It also uses a timer to control the number of notification messages sent by the LSR.

If the ingress router decides to re-negotiate along an existing path for a higher service class, it carries out the following procedures:

The ingress router formulates a label request message to indicate that this is an existing CR-LSP along which the traffic parameters are being modified. The label request message contains the newly requested modified traffic parameters along with the service class it requires.

When each LSR receives this label request message, it checks its buffer table to determine which buffer stream that is being used by the particular LSP. The LSR then chose a higher buffer stream to the one the flow is currently traversing. It then determines whether it can allocate the bandwidth and the minimum loss probability requested to a higher buffer stream. If it can, it assigns the amount in the new buffer stream. It alters the available bandwidth within the buffer request table and forwards the request message to the next hop. If all the LSRs along the CR-LSP are able to meet the requirement, on receipt, the egress router will create a notification message, indicating the resources have been reserved and send it to the upstream LSR towards the ingress router of the CR-LSP. On receipt of this message the ingress router updates its FEC/label binding to reflect the higher buffer stream through which the flow will now be routed.

If however, an LSR cannot allocate the additional resources, it will send a RenegFailure message to the ingress router and not propagate the Label Request message. On receipt of this RenegFailure message, the ingress router tries negotiating for new quality requirements along an alternative LSP, figure 4.3.

Figure 4.3 Renegotiation Process

- (1). Initial flow of packets from ingress to egress.
- (2). Transmission of notification message to ingress router.
- (3). Transmission of packets through a higher buffer.
- (4). Transmission of packets through an alternate LSP.

4.2 FATE +

In the FATE scheme previously discussed, when a LSR determines that it is suffering congestion along a CR-LSP, it informs the ingress router by sending a notification message. Upon receipt, the ingress router decides the appropriate action. FATE + is an extension of FATE and it permits congested LSRs to negotiate alternative QoS path arrangements in the event of congestion occurring without involving the ingress router, [25]. This is done by redistributing traffic flows from the point of congestion within the network.

FATE+ therefore enhances the basic scheme by removing the responsibility of deciding what action should be taken in the event of congestion occurring from the ingress router, and places it in the congested LSR. The scheme assumes the CR-LSPs have been “loosely” configured within the region of interest.

In the FATE + scheme a congested LSR can take any of the decisions stated below:

1. Transfer the flows from CR-LSP on to a higher buffer
2. Re-route the flows from the CR-LSP via an alternative downstream LSR
3. Re-route the flows from the CR-LSP via an alternative upstream LSR

4.2.1 Reroute via Alternative Downstream LSR

The first decision is the same used in the FATE scheme. If the congested LSR decides to re-create the traffic flows via an alternative downstream LSR it carries out the following procedure:

The congested LSR assumes the role of an ingress router and determines what the alternative next hop is, eliminating the one currently being used. It then formulates a Label request message and sends it to the next hop for the destination, that is LSR 5 illustrated in figure 4.4. On receiving the label mapping from downstream LSR 5, LSR 3 is able to re-map the traffic flows along the new CR-LSP through LSR 5.

When LSR 3 detects congestion has receded, it can go back to using the label it received initially (from LSR 4) whilst maintaining the newly received label in case of further congestive situations.

Figure 4.4 Re-route via alternative downstream LSR

4.2.2 Reroute via Alternative Upstream LSR

In a situation where a congested LSR has detected congestion and is unable to accommodate CR-LSPs through itself, it has to consider re-routing the traffic flows via an alternative upstream LSR. In this situation, the congested LSR carries out the following procedure: it has to inform the upstream LSR that it is no longer able to meet the quality requirements of the CR-LSP it initially received a Label Mapping from. To do this, it formulates a No Resources notification message and sends it to the upstream LSR.

Figure 4.5 Re-route via different LSR

On receiving the No Resources notification message, the upstream LSR will instigate proceedings to re-map the flows within the CR-LSP into a CR-LSP via a different downstream LSR capable of accommodating the path, LSR 5 as shown in figure 4.5 above.

4.3 MATE

The main goal of the MATE (Multi-path Adaptive Traffic Engineering) algorithm is to avoid network congestion by adaptively balancing the load among multiple paths based on measurement and analysis of path congestion, [5]. In this algorithm, the intermediate LSRs are not required to do any traffic engineering or measurement but simply to switch packets.

The algorithm is targeted for switched networks such as Multi-Protocol Label Switching (MPLS) networks.

Figure 4.6 Transit network running MATE

The MATE algorithm assumes that several Explicit LSPs (typically ranging from two to five) have been established between the ingress and egress routers in the MPLS domain using a standard protocol. The three main functions of the MATE algorithm are implemented at the ingress router. The algorithm is intended for traffic that does not require bandwidth reservation. These functions are:

1. Filtering and Distribution
2. Measurement and Analysis
3. Traffic Engineering

Figure 4.7 MATE Functions in an ingress node

4.3.1 Filtering and Distribution

The Filtering and Distribution function first distributes the traffic to be balanced for a given ingress-egress pair equally among M bins where the number of bins determines the amount of traffic that can be shifted. For example, if the incoming ingress traffic flow is R bps, then each bin would receive $r = R/M$ bps. The MATE algorithm uses the traffic from the M bins to map unto the LSPs.

The distribution and filtering of the traffic into the M bins can be done in different ways. The MATE algorithm assumes a per-packet basis using a round-robin fashion. Packets can also be filtered and distributed on a per-flow basis or a per-destination basis.

After the traffic is distributed into the M bins, a second function maps each bin to the corresponding LSP. The rule for mapping this traffic is very simple. If one LSP is to receive twice the traffic as another, then this LSP should receive twice as many bins.

4.3.2 Traffic Measurement and Analysis

The MATE algorithm does not require each LSR to perform traffic measurement. The measurements are only done at the ingress and egress routers. The packet delay on a LSP is obtained by transmitting a probe packet from the ingress router to the egress router. The probe packet is time-stamped at the ingress router at time T_1 and recorded at the egress router at time T_2 . A time difference of T_d is used to account for any difference in the speeds of the two clocks. The total packet delay is $T_2 - T_1 + T_d$. The estimate of the mean packet delay is obtained by sending a number of probe packets to the egress router. This mean value is then stamped to a probe packet at the egress and sent back to the ingress router, figure 4.8 below.

Figure 4.8 Delay measurement in MATE network

The MATE algorithm is also assumed to be able to measure the packet loss probability. A group of probe packets are sent from the ingress router. Each probe packet is stamped with a sequence number N . This sequence number is to notify the egress router how many probe packets have been transmitted by the ingress router and another field is used in the probe packet to indicate how many probe packets N_R have been received by the egress router. When a probe packet returns, the ingress router is able to estimate the one-way packet loss probability based on the number of probe packets that have been transmitted and the number that has been received, figure 4.9.

Figure 4.9 Loss measurement in MATE network

4.4 Dynamic Traffic Partitioning

E. Dinan et. al., [26] proposed a framework for dynamically partitioning EF and Best Effort traffic in the MPLS network. The analysis is based on minimizing the average end-to-end delay of EF traffic and the packet loss ratio of Best Effort traffic.

Considering the case where the traffic belongs only to the EF class, the system performance of EF traffic is measured by the average packet transmission delay. The focus is on minimizing the average delay and neglecting the impact of delay variation. It is assumed that the blocking probability is relatively small and each LSP is approximated by a series of M/M/1 queues. A single ingress-egress pair is considered in this case. Figure 4.10 shows the queuing model used in the analysis.

Figure 4.10 M/M/1 queuing network

Figure 4.11 M/M/1/k queuing network

The vector to the input (ingress) traffic rates onto the LSPs at time t is given as:

$$\lambda(t) = [\lambda_1, \lambda_2, \dots, \lambda_n]^T$$

The queue arrival rate at time t is given as:

$$\lambda_{i,j}(t) = \lambda_i(t) + \gamma_{i,j}(t)$$

That is, the queue arrival rate consists of the ingress arrival rate λ_i and the background traffic arrival rate γ_{ij} .

As an example, from figure 4.10, the queue arrival rate at time t of queue 1 on LSP 1 is given as:

$$\lambda_{1,1}(t) = \lambda_1(t) + \gamma_{1,1}(t) \quad (4.1)$$

The objective of the analysis is to find the traffic-mapping vector $P(t) = [p_1(t), p_2(t), \dots, p_n(t)]^T$ that optimizes the system performance. From their model, $[\lambda_1(t), \dots, \lambda_n(t)]^T = [p_1, \dots, p_n]^T \lambda(t)$ and $\sum_{i=1}^n p_i(t) = 1$. This represents the proportion of the ingress arrival rate that should be transmitted to each LSP in the network to obtain an optimal performance.

Finding the column vector $P(t)$ that minimizes the average delay, τ_{avg} , is one method of solving this optimal problem. Considering a single LSP i in the model, the average packet delay along the LSP is given by

$$\tau_i = \sum_{j=1}^{R_i} \frac{\rho_{ij}(t)}{(1 - \rho_{ij}(t))\lambda_{ij}(t)} p_i(t) \quad (4.2)$$

Here R_i is the total number of queues utilized by the ingress traffic on LSP i . The average delay, τ_{avg} in the network can therefore be written as follows:

$$\tau_{avg}(t) = \sum_{i=1}^n \left(\sum_{j=1}^{R_i} \frac{\rho_{ij}(t)}{(1 - \rho_{ij}(t))\lambda_{ij}(t)} p_i(t) \right) \quad (4.3)$$

Here, it is assumed that there are n LSPs in the network.

The set $\{\lambda_i(t), 1 \leq i \leq n\}$ that minimizes τ_{avg} can be found by iteration or Lagrange Multiplier methods, [27] and [28]. The computation involved in finding the set $\{\lambda_i(t), 1 \leq i \leq n\}$ that minimizes τ_{avg} is considered to be significant since the queue arrival rate vectors and the ingress node arrival rate varies. E. Dinan et. al., [26] developed a simple method to calculate the optimal vector $P(t+1)$ when the value of $P(t)$ is known. The minimum value of τ_{avg} is obtained once and $P(t)$ is updated periodically with a simple algorithm in each time step t_d .

Two functions used in [26] in the development of the solution to this linear optimization problem are the sensitivity to the average delay to the background load traffic rate $S_{\gamma_{i,j}}(t)$, and the sensitivity of the average delay to the path arrival rate $S_{\lambda_i}(t)$.

$$S_{\gamma_{i,j}}(t) = \frac{\partial \tau_{avg}}{\partial \gamma_{i,j}(t)} \quad (4.4)$$

$$S_{\lambda_i}(t) = \frac{\partial \tau_{avg}}{\partial \lambda_i(t)} \quad (4.5)$$

The value of $S_{\gamma_{i,j}}(t)$ is dependent on the queue and the LSP and the value of $S_{\lambda_i}(t)$ is only dependent on the LSP in question. These elements are positive since the delay increases with increasing arrival rate.

An increase in the average delay along a LSP can be caused by the increase in the source arrival rate ($\lambda_{increase}$) at the ingress router or the increase in the background traffic rate ($\Delta\gamma_{i,j}$) at an intermediate LSR or both. To minimize this delay along the LSP, we must consider what caused the increase delay. In [26], two algorithms are proposed each depending on the factor that caused an increase in the delay. The objective is to find how much ingress traffic should be removed from the LSP that is experiencing increasing congestion levels. E.Dinan et al. using computation find this amount.

In the simplest case, suppose that during the time interval t_d only the arrival rates of queues in a single LSP changes and this change is caused specifically by changes in the background traffic arrival rate $\Delta\gamma_{i,j}$, figure 4.12.

Figure 4.12 Change in queue arrival rates on a single LSP

To compensate for the effect of variation of the arrival rates, the ingress arrival rate of LSP i must be changed by $\Delta\lambda_i$ in a way that the average delay at some time t is the same as at time $t+1$. Equation 4.6 below gives the linear approximation used to find the change needed for the ingress arrival rate.

$$\tau_{avg}(t+1) = \tau_{avg}(t) + \sum_{j=1}^{R_i} S_{\gamma_{i,j}}(t) \Delta\gamma_{i,j}(t) + S_{\lambda_i}(t) \Delta\lambda_i = \tau_{avg}(t) \quad (4.6)$$

When this is the case, the amount of ingress traffic that must be shifted to/from this LSP is given as:

$$\Delta\lambda_i(t) = \frac{-\sum_{j=1}^{R_i} S_{\gamma_{i,j}}(t) \Delta\gamma_{i,j}(t)}{S_{\lambda_i}(t)} \quad (4.7)$$

A traffic assignment algorithm was used in [26] to distribute this extra traffic to the LSPs. The traffic assignment algorithm is as follows:

if $(-\Delta\lambda_i) > 0$

Shift the traffic $(-\Delta\lambda_i)$ from LSP i to the path with lowest value of $S_{\lambda}(t)$

Else (... .. $(-\Delta\lambda_i) < 0$)

Shift the traffic $(-\Delta\lambda_i)$ from the path with the highest value of $S_{\lambda}(t)$ to LSP i

end.

The main idea is that if there is an increase in traffic on an LSP, the delta traffic is added to the path whose transmission delay is least sensitive to λ to minimize the increase of τ_{avg} . If there is a decrease in traffic on a LSP path, delta traffic is removed from the path that is most sensitive to λ to this path, [26].

In a real network, increases in congestion levels can be caused by both increases in the source traffic rate ($\lambda_{increase}$) and background traffic rate $\Delta\gamma_{i,j}$, figure 4.13.

Figure 4.13 Change in the source and queue arrival rates for n LSPs

In [26], both increases were considered in some time interval t_d . Assuming the traffic variations ($\lambda_{increase}$ and $\Delta\lambda_i$) are small due to small t_d of smooth changes in the network behaviour, the following algorithm was recommended by E. Dinan et al.

$$\text{Calculate } \Delta\lambda_i = \frac{\sum_{j=1}^{R_i} S_{\gamma_{i,j}}(t) \Delta\lambda_{i,j}(t)}{S_{\lambda_i}(t)}, 1 \leq i \leq n$$

$$\text{Calculate } \Delta\lambda = \lambda_{increase} + \sum_{i=1}^n \Delta\lambda_i$$

Execute the traffic assignment algorithm for the delta traffic $\Delta\lambda$.

The partitioning vector obtained by the above algorithms and the optimum solution depends on the error of the linear approximation. Smaller changes during time interval t_d should lead to more exact results.

With only traffic of the Best Effort group present in the network, an optimal mapping vector can be determined in order to minimize loss rate in the network. To represent the bottleneck of the system each LSP is modeled by one M/M/1/k queue, figure 4.11.

The objective is therefore to find the column vector $P(t)$ such that the loss rate $L(t) = \sum_{i=1}^n P_{b_i}(t) \lambda_i(t) = P_b(t)^T \Delta\lambda(t)$ is minimized. To reduce the computational complexity of the computation involved, the same approach that was used for the EF traffic can be used. That is, to calculate the mapping vector $P(t+1)$ in terms of the system parameter variations during the time interval t_d , given $P(t)$. In this model, the vector $S_{\gamma,j}(t)$ is reduced to a scalar value $S_{\gamma_i}(t)$ since each LSP is now modeled by one queue and there is a close form equation for the sensitivity functions for a single LSP.

$$S_{\gamma} = S_{\lambda} = \frac{\partial L}{\partial \lambda} = \rho^K \times \frac{(1+K)(1-\rho^{K+2}) - (2+K)(\rho - \rho^{K+2})}{(1-\rho^{K+1})^2} \quad (4.8)$$

That is, for a single node LSP, the loss sensitivity due to the background traffic and the loss sensitivity due the ingress traffic are the same and they can be computed by using the close form equation. In equation 4.8 above, K represents the queue size, ρ represents the utilization of the server and L is the loss rate. The Traffic Assignment algorithm used previously for the EF can be used with the above sensitivity functions.

In this chapter we gave the details of the four main algorithms used in the development of the proposed Traffic Re-assignment algorithm. The next chapter gives the problem statement and discusses the proposed architecture.

Chapter 5 Proposed Traffic Re-assignment Algorithm

5.1 Problem

It is assumed that a user wants to transmit traffic that requires two different service guarantees across an MPLS domain. One traffic type is Expedited Forwarding (EF) traffic and the other is the Assured Forwarding (AF) traffic, [29]. The user demands a guarantee on the EF traffic, which is to have low packet delay, and for the AF traffic, little packet loss. The network is able to provide these guarantees if the user agrees for the traffic to be dynamically shifted to alternate LSPs at the ingress router of the network. It is assumed that a set of parallel LSPs has been pre-established between the ingress and egress routers using CR-LDP, RSVP-TE, [30] and [31] or manually configured. Since the LSPs are pre-established, there is no requirement for the traffic demand.

Given these assumptions, the objective of our work is to develop a dynamic Traffic Re-assignment algorithm that can be used to reduce the loss of the AF traffic and the delay of the EF traffic. The algorithm should be implemented in the MPLS domain. We are assuming that both service classes of traffic are transmitted simultaneously along the LSPs.

5.2 Details of the Proposed Architecture

In the network, each LSR along a LSP is modeled as a Single-Server queuing system. The queuing system itself consists of two queues one for each class of traffic, [32]. We used a Priority scheduling discipline for servicing the different classes of traffic. Our convention

is that priority increases with increasing priority numbers. That is, the EF class has priority 1 (highest priority) and the AF class has a priority of 0 (lower priority). We are assuming that the traffic from each service class arrive at the queues according to a Poisson process, [33].

Figure 5.1 Stochastic model for the Traffic Re-assignment algorithm

In [26], the background traffic introduced at an intermediate LSR exits the network at the same intermediate LSR. That is, the background traffic introduced at one intermediate LSR has no effect on the traffic arrival rates of the downstream LSRs along this LSP. In

our model, figure 5.1, the background traffic introduced at the intermediate LSRs exit at the egress router. Therefore, the traffic arrival rate at a downstream LSR is influenced by the background traffic arrival rate of an upstream LSR and the external background source. The background traffic transmitted from an upstream LSR and the background traffic transmitted from the external background source, to an intermediate LSR represent the aggregate background traffic variation of that LSR. This background traffic is used to increase the congestion levels at the queues of the LSRs and along the LSPs.

Wolf, [34] Jaswal, [35] and Fraser and Morgan, [36] have summarized research on Priority Queuing systems. Wolf, [34] provides a basic analytical approach, which may be used to study message delays under a Round-Robin service discipline for the case of only one message class. Fraser and Morgan [36] have analyzed the Priority First-in-First-out followed by Round-Robin discipline that services two classes of traffic. The analytical details of message switching, with prioritized packet transmission, have been covered in [37] and [38]. A similar approach is used in our model for the transmission of traffic from the two service classes. The model presented here is simpler and more flexible than those presented in [38] and [39].

Since we are using the LSPs from the MPLS domain, the analysis of one of the queuing systems along the path can be used to generalize the performance of this path. That is, the LSP can be modeled as a series of Single-Server queuing systems. The

queuing system consists of the M/D/1 queue that models the EF class and the M/D/1/K queue that models the AF class. The M/D/1 queue was considered since it represents an infinite buffer queue in which packets can only experience delays but no loss. The M/D/1/K queue has finite buffer space and as a result traffic destined for this queue is liable to experience packet loss if the buffer is full.

Figure 5.2 Congestion and Notification process in the Traffic Re-assignment network

Figure 5.2 above shows the Congestion and Notification processes when both LSRs along the LSP experience increase congestion levels. Here $\Delta\gamma_{EF_j}$ and $\Delta\gamma_{AF_j}$ represent the increase in the individual background EF and AF classes respectively at LSR j . Each LSR along the LSP monitors the traffic variation through it. The packet loss ratio and the packet delay are measured at the LSRs. Also the Aggregate background traffic variation of the AF and EF classes is measured at these Intermediate LSRs. These values are stored in the buffer table of each intermediate LSR. Figure 5.3 below shows the configuration of the queues at each LSR. The EF queue has an infinite capacity and therefore models the M/D/1 queuing system and the AF queue capacity is finite. Two thresholds are established at these LSRs, the loss threshold for the AF queue and the packet delay threshold for the EF queue.

Figure 5.3 Queue Configuration

When the loss or the delay at a LSR is greater than the pre-determined upper threshold, the LSR creates a notification packet and forwards it upstream to either the ingress router or another intermediate LSR. This notification packet contains the values of the amount of each class of traffic that must be shifted, the packet loss ratio and the packet delay. Before this LSR forwards this notification packet upstream it compares its computed amount of traffic that should be shifted with the values received from the downstream LSRs. The greatest of these values is stamped to the notification packet that is forwarded upstream.

When the ingress router receives the notification packet, it obtains the computed amounts of the traffic shift values, the loss ratio and the packet delay from the packet. The

ingress router compares the loss and delay values with the thresholds set at the ingress and the values from other LSPs. If one or both of these values are greater than the threshold, the ingress router shifts the individual classes to the alternate LSP.

If however, the loss or delay at a LSR or along the LSP is below the lower threshold, the LSR sends a notification packet with only the value of the delay or the loss upstream to the ingress router. This packet is used to inform the ingress router that traffic can now be shifted to the original LSP.

5.2.1 Computation at the Intermediate LSRs

The approach used to compute the amount of each class of traffic that must be shifted away from the congested queue is similar to approach used in [26]. Two functions used in the computation of the amount of the AF class that must be shifted are the Loss Sensitivity due to the background traffic and the Loss Sensitivity due to the ingress traffic.

The Loss Sensitivity due to the background traffic compares the variation of the loss ratio with the variation of the background traffic rate and the Loss Sensitivity due to the ingress arrival rate compares the variation of the loss ratio with the variation of the ingress arrival rate, for a single LSR.

$$S_{\Gamma} = \frac{\partial L}{\partial \gamma} \quad \text{Loss Sensitivity due to background traffic}$$

$$S_{\pi} = \frac{\partial L}{\partial \lambda} \quad \text{Loss Sensitivity due to ingress traffic}$$

Since the loss increases with increasing arrival rate, the elements of S_γ and S_π are always positive. In the case of the EF class traffic, these functions are the Delay Sensitivity due to the background traffic and the Delay Sensitivity due to the ingress traffic. These represent the variation of the delay with the background source rate and the ingress arrival rate respectively.

$$S_\gamma = \frac{\partial T}{\partial \gamma} \quad \text{Delay Sensitivity due to background traffic}$$

$$S_\lambda = \frac{\partial T}{\partial \lambda} \quad \text{Delay Sensitivity due to ingress traffic}$$

From these functions the Loss Sensitivity ratio and the Delay Sensitivity ratio are obtained.

$$\text{Loss Sensitivity ratio} = S_\gamma / S_\pi$$

$$\text{Delay Sensitivity ratio} = S_\gamma / S_\lambda$$

These values are estimated empirically by collecting values of the loss, delay and background and source throughputs within small interval times of simulations. The background and source traffic were injected independently into the LSR. To obtain these values, the differences in the delay and loss as well as the changes in the background throughput and ingress source were measured at consecutive simulation time intervals. A

specific example of how S_π is computed for every 10 seconds of the simulation time is shown below.

$$S_\pi = \frac{L(k+10) - L(k)}{\lambda(k+10) - \lambda(k)}$$

Here L represents the loss at time index k and $k+10$, and λ is the actual arrival rate. Only the non-negative values of S_π were considered.

Varying mean arrival times were introduced in the background source generator since it is expected that the background traffic throughput will be changing. To increase the background rate, the inter-arrival time of the background source generator was decreased by 0.001 seconds every ten seconds. Each simulation was run for 5000 seconds. The table below shows the results obtained for 10 consecutive runs.

	1	2	3	4	5	6	7	8	9	10
S_γ	1.8632	1.8744	1.8731	1.8732	1.8659	1.8666	1.8723	1.8756	1.8754	1.8743
S_λ	0.9821	0.9801	0.9923	0.9903	0.9812	0.9856	0.9876	0.9844	0.9876	0.9832
S_Γ	1.7345	1.7656	1.8021	1.8036	1.759	1.7643	1.7325	1.7432	1.8001	1.7089
S_π	0.7123	0.7567	0.7723	0.7045	0.6989	0.7078	0.7056	0.7379	0.7461	0.7913

Table 5.1 Sensitivity values for different simulation runs

The table shows that the variation of the loss and delay with respect to the background traffic is larger than with respect to the ingress source traffic. The variation of the ingress source traffic is much smaller and also causes smaller fluctuations in the loss values.

$$\text{Loss Sensitivity ratio} = \frac{\text{mean}(S_{\gamma})}{\text{mean}(S_{\lambda})} = 1.88$$

$$\text{Delay Sensitivity ratio} = \frac{\text{mean}(S_{\Gamma})}{\text{mean}(S_{\pi})} = 2.29$$

The amount of each class of traffic shift computed at an intermediate node r , of a single LSP is therefore:

$$\text{delta_lambda_EF} = \text{delay sensitivity ratio} * \sum_{j=1}^r \Delta\gamma_{EF_j} \quad (5.1)$$

$$\text{delta_lambda_AF} = \text{loss sensitivity ratio} * \sum_{j=1}^r \Delta\gamma_{AF_j} \quad (5.2)$$

The MATE algorithm discussed earlier, measure the delay and loss of the packets at the egress and ingress router using probe packets. The advantages of using our proposed method for measuring the delay and loss is that each LSR is able to advertise to the upstream LSRs or the ingress router its loss or delay and receive a faster reaction from the ingress router. The FATE and FATE+ schemes also measure the loss at the intermediate LSRs, but they do not measure the variation of the background traffic that is causing the

congestion at the LSR. Also the MATE, FATE and FATE+ schemes discussed previously do not compute the amount of traffic that should be shifted away from the congested LSR.

In the proposed Traffic Re-assignment algorithm, a notification packet is sent to the ingress router if the loss or delay has arisen to the pre-determined upper thresholds or if it has dropped below the pre-determined lower thresholds. In the FATE scheme, the notification message is sent to the ingress router only if the loss has arisen above the pre-determined threshold value for an extended time period. The advantage therefore of our proposed scheme is that fewer packets will be lost since the notification message is created and sent immediately following an increase in the loss or delay to the threshold value. That is, each LSR does not monitor how long the loss or delay remains at or above the threshold values before sending the notification message, but sends a message the first time this happens.

5.5.2 Traffic Shift at the Ingress Router

Let $\Delta\lambda_{EF}$ and $\Delta\lambda_{AF}$ represent the computed amount of EF and AF traffic classes that must be shifted respectively at the LSR. Before each class can be shifted, the ingress router compares the values for *delta_lambda_EF* and *delta_lambda_AF* obtained from the notification packet. The shifting process is discussed below.

When $\Delta\lambda_{EF} \geq \Delta\lambda_{AF}$, $\Delta\lambda_{EF}$ of the ingress EF class is shifted to the alternate LSP. If however $\Delta\lambda_{EF} < \Delta\lambda_{AF}$, we shift $\Delta\lambda_{EF}$ of the ingress EF class and $(\Delta\lambda_{AF} - \Delta\lambda_{EF})$ of the

ingress AF class, to the alternate LSP. That is, if more of the EF class needs to be shifted at the ingress router, than the AF class, then only the EF class is shifted. If however, more of the AF class needs to be shifted, then the equivalent amount of the EF class is shifted and the difference in the computed amounts for the two classes is shifted for the AF class. Shifting the different classes of traffic using this method conserves the available bandwidth on the alternate path by $\Delta\lambda_{AF}$ in the first case and $\Delta\lambda_{EF}$ in the latter, as compared to shifting the computed amount of both classes.

The MATE, FATE and FATE+ schemes consider shifting the aggregate traffic at the ingress router. Also in the FATE and FATE+ schemes, all the traffic is taken away from the congested LSP and shifted to the alternate LSP. This method could lead to the under-utilization of the congested LSP after the shift and the over-utilization of the alternate path.

Our proposed method prevents this from happening by shifting a proportion of the traffic to the alternate path. This proportion is the equivalent amount of traffic that will reduce and stabilize the delay or loss along the congested path. The proposed method also allows the ingress router to redistribute the traffic back to the original LSP if the loss or delay along the path drops below the pre-determined lower thresholds.

Chapter 6 Simulation Model and Results

6.1 Introduction

In the previous section, a description of the proposed architecture was given. It is necessary to investigate the effect of different physical implementations and traffic variations of the functional architecture on the performance of the Traffic Re-assignment algorithm.

The commercial simulator used was OPNET, a general purpose telecommunications network simulator tool. OPNET uses a graphical interface where simulation models are defined at five levels. The first level is the Project level. At this level, high-level components of real-world networks are identified, models are created and edited, simulations are run and results are analyzed. The Network level is the second level of the OPNET simulator. At this level the interconnections of nodes are defined. The third level of the simulator is the Node level and here the elements that make up the nodes and their interconnections are defined; these elements include queues, processes, sources and receivers, and transmitters. The functionality of each queue or process element is defined in terms of a finite state diagram and the transitions between the states. This functionality makes up the fourth level. The final level is where the processing in each of these states in the finite state machine is defined in C code.

6.2 Model Description

6.2.1 The Network Model

The Network model used is shown in figure 6.1 and consists of generators, ingress and egress routers, intermediate routers and point-to-point duplex links. The link connections from the ingress router to the egress router represent the LSP. Packets generated from the source generator are switched to the appropriate LSP based on the outgoing stream number placed in the Destination field of the packet. Packets with a destination address of 0 are transmitted through LSP 0 and packets with a destination address of 1 are transmitted through LSP 1. The background source generator is used to create temporary statistical variation in the traffic along the LSP. The variation in the background traffic leads to temporary congestion at the LSRs. The packets generated at the background source have the same destination address as the ingress packets flowing along the path. All packets generated from the source generator or the background source generators are stamped with priority 1 or priority 0. Only a Poisson traffic model was used in the network. The link speed used was 155Mb/sec and the packets had a constant size of 400 bytes.

Figure 6.1 Network Model

6.2.2 The Node Model

The Node model is responsible for the direct connection of the transmitter to the receiver of the same node. Each port on the node therefore supports the bi-directional transmission of packets along the links.

Figure 6.2 below shows the Node model for the ingress router. It consists of the Source module, the Process module and three transmitters and receivers. The packets generated at the Source module are forwarded to the Process module and after the processing of these packets they are forwarded to the transmitters. The packets received at

the receivers are forwarded to the Process module. All the receivers and the transmitters are configured to accept packet formats used in the simulation and the link data rates.

Figure 6.2 The Ingress Node Model

The Node model for the LSRs and the egress router is shown in figure 6.3 below. It consists of four receivers connected to four transmitters via the Process module. Packets received at the receivers are forwarded to the Process module. After these packets are processed they are forwarded to the transmitters. The Process module represents the queue of the node and is configured with the queue attributes.

Figure 6.3 The LSR Node Model

6.2.3 The Process Model

- The Ingress Process Model

The Process model of the ingress router is composed of five states, the Initial, Idle, Transmitter (xmt), the Receiver 1 (rcv1) and the Receiver 2 (rcv2) states. Figure 6.4 shows the Ingress Process model.

- The Initial state

The variables are initialized from this state. These variables include the packet loss and packet delay values of the downstream LSRs. The throughput variation along the LSPs and the throughput of the ingress traffic, are also initialized at this state. These variables are initialized to zero.

- Receiver 1 and Receiver 2 states

At the Receiver states the parameters carried by the packets received at the receiver of the ingress router are read. These parameters include the packet loss and delay at the downstream LSRs and the proportion of traffic to be shifted (*delta_lambda*). After obtaining these values the packets are destroyed. There is a transition to these states when a packet from the downstream LSRs arrives at the receiver stream of the Ingress router.

- The Transmitter state

The process transitions to the Transmitter (xmt) state when there is a packet arrival from the Source module. The Traffic Re-assignment algorithm is implemented in this state. When a packet is received from the Source module the process places a destination address in the Destination field of this packet and forwards this packet unto the stream. Initially (for the first 3 seconds), all the packets are forwarded unto the same LSP. At the end of this simulation time, the process checks the values of the parameters of interest. That is,

the loss ratio, packet delay, and $\delta_{\lambda_{EF}}$ and $\delta_{\lambda_{AF}}$. If no packets were received at the ingress router from the downstream LSR, then these values are zero. The process therefore continues to transmit packets along the same path.

If however, packets are received from the downstream LSRs and the values of these parameters are changed, the process compares the values obtained for the loss ratio and the packet delay from the specific path with its upper and lower threshold values set. It also compares these values with the values obtained from the other LSP. If any of these values or both of these values are greater than the upper threshold and the parameters of the other path, the process performs the traffic shift. That is, the computed amount of traffic is switched to the alternate path. This is achieved by decrementing the count of each packet switched to the paths. The initial count for the packets switched to the alternate path is the computed amount of traffic that must be shifted. It is assumed that the alternate path is lightly utilized and the packet loss and delay values are minimal.

When the packet loss and delay values are below the threshold, the traffic is switched back to the original path. This ensures that the alternate path does not get congested. The increase in the background traffic is temporary. Figure 6.5 below gives the flow chart for the Transmitter state.

Figure 6.4 The Ingress Process Model

Figure 6.5 Flow chart for Transmitter state

- The LSR Process Model

The queuing discipline discussed in the previous chapter is defined in the LSR Process model, figure 6.6. The LSR Process model consists of five states. These are the Initial, the Arrival, Service and Service-Complete, and Idle states.

Figure 6.6 The LSR Process Model

- The Arrival state

A transition is made to the Arrival state from the Initial or Idle states when a packet arrives at the receivers of the LSRs. These packets could be from either the downstream or

upstream LSRs or the ingress router. The process first acquires the packet and checks its origination. If the packet is from the background source generators, the process increments a counter to measure the throughput from this stream. The packet is then queued for service. A counter is used to keep track of the queue size. At consecutive intervals of simulation time the change in throughput from the background source generators and the upstream LSRs are computed. The state variable *delta_gamma* is used to represent this change.

If the packet is from the ingress source, a counter is also used for measuring the throughput. When the packet comes from either the ingress source or the background source stream, the priority of this packet is checked. If the priority of the packet is 0 the queue size is checked. If the queue size is greater than the queue capacity the packet is destroyed (lost). The state variable *num_dropped* is used to record the number of packets dropped. If however the queue size is less than the queue capacity, the packet is queued for service and the queue count is incremented. The state variable *num_trans* is used to count the number of packets transmitted. The packet loss ratio is therefore computed as the ratio of the number of packets dropped to the number of packets transmitted. When the priority of the packet is 1, the simulation time is placed in the Time field of the packet and it is queued for service.

If the packet is from a downstream LSR, the packet contents are collected. These are the packet loss and delay values and *delta_lambda* for the priority1 and priority 0 packets. The packet is then destroyed. Figure 6.7 gives the flow chart for the Arrival state of the LSR.

Figure 6.7 Flow chart for Arrival state

- The Service state

The process transitions to the Service state when the server is not busy and the insertion of a packet was successful. At this state, the process gets the packet at the head of the queue and determines its packet service time. This time is determined by obtaining the packet length and service rate of the server.

- The Service-Complete state

The packet at the head of the queue, which had just been serviced, is obtained at this state. The priority of the packet is checked. If the priority is 1, the time of arrival of this packet is acquired from the time field. The difference between the current simulation time and the time of arrival is computed. This time represents the delay of this packet at the queue. For all packets, the destination addressed is checked and forwarded to the appropriate stream.

At this state, the packet loss and delay parameters are also monitored. If both or any of these values are greater than the upper thresholds set at this node, *delta_lambda* is computed for both types of traffic.

A notification packet is then created and the packet loss and delay values and *delta_lambda_EF* and *delta_lambda_AF* are recorded into the appropriate fields. The packet is then sent to the upstream LSR. Also if the loss and/or delay are less than the lower thresholds a notification packet is sent to the ingress router. Figure 6.8 below gives the flow chart diagram for the Service-Complete state.

Figure 6.8 Flow chart for Service-Complete state

- The Egress Process Model

The Router state of the egress router simply collects the packets from the upstream LSR and sends them to the destination.

- The Packet Format

Two different packet formats were used in the simulation. Those transmitted from the source generators and those created at the queues. Figures 6.9 and 6.10 show the Packet formats. The Source packet fields include the Destination field, which carries the address of the packets, the Priority field, which is stamped at the queue if the priority of the packet is one, and the Time field. The notification packet has the Loss and Delay fields and also fields for recording the increase in the background throughput.

Figure 6.9 Source Packet format

Figure 6.10 Notification Packet format

6.3 Computation and Simulation Results

Simulation experiments were used to evaluate the effectiveness of the proposed Traffic Re-assignment algorithm. Two different networks were used in the analysis. The first is the single-node model and the second is the two-node model. The first subsection compares the computation results of the traffic shift with the simulations done and the second section highlights the performance of the algorithm over the MATE algorithm.

Figure 6.11 The Single-Node Model

6.3.1 The Single-Node Model

The variations of the background EF and AF class traffic were considered at LSR 2 of the single-node model, figure 6.11. The path **ingress** --> **LSR 2** --> **egress** is considered the highly utilized path in this network. The background and ingress source traffic rates are set to be minimum along path **ingress** --> **LSR 1** --> **egress**. Two different scenarios were considered in the analysis. Scenario 1 considers a high loss ratio at LSR 2 due to a larger increase in the background AF traffic and is referred to as the loss sensitive case in this

research and scenario 2 considers a large delay due to a large variation of the background EF traffic and is referred to as the delay sensitive case.

- Computation
- Scenario 1: Loss sensitive case

<i>delta_gamma_EF</i> (packets/sec)	1000
<i>delta_gamma_AF</i> (packets/sec)	3900

Table 6.1 Background throughput variation through LSR 2

The amount of traffic that should be shifted for each class of traffic is computed as follows:

$\text{delta_lambda_EF} = \text{delay sensitivity ratio} * \text{delta_gamma_EF} = 2.29 * 1000 = 2290$
packets/sec;

$\text{delta_lambda_AF} = \text{loss sensitivity ratio} * \text{delta_gamma_AF} = 1.88 * 3900 = 7332$
packets/sec;

The actual amount of traffic that is shifted at the ingress router for the EF and AF classes are 2,290 packets/sec and 5,042 packets/sec respectively.

- Scenario 2: Delay sensitive case

<i>delta_gamma_EF</i> (packets/sec)	3000
<i>delta_gamma_AF</i> (packets/sec)	1500

Table 6.2 Background throughput variation through LSR 2

The amount of traffic that should be shifted for each class of traffic is computed as follows:

$$\text{delta_lambda_EF} = \text{delay sensitivity ratio} * \text{delta_gamma_EF} = 2.29 * 3000 = 6870 \text{ packets/sec;}$$

$$\text{delta_lambda_AF} = \text{loss sensitivity ratio} * \text{delta_gamma_AF} = 1.88 * 1500 = 2820 \text{ packets/sec;}$$

The amount of traffic that should be shifted at the ingress router is 6,870 packets/sec. This shift is needed only for the EF ingress class.

- Simulation Results

Simulation 1 gives the results obtained when the values of the background traffic from scenario 1 is used and simulation 2 uses the background variation values from table 6.2 of scenario 2.

Simulation 1: Loss sensitive case

Figure 6.12 Variation of background throughput at LSR 2

Figure 6.13 Variation of Loss ratio at LSR 2

Figure 6.14 Variation of packet delay at LSR 2

Figure 6.15 Ingress variation

Figures 6.12 to 6.15 show the results obtained for the single-node model. Figure 6.12 gives the variation of the background traffic for the EF and AF class at LSR 2. The traffic rates increase temporarily from a mean of 4,000 packets/sec to a mean of 8,000 packets/sec for the AF class and from a mean of 4,000 packets/sec to a mean of 5,000 packets/sec for the EF class. This increase occurred at 4 seconds of simulation time. This variation in the traffic rates accounts for a larger loss ratio and packet delay at LSR 2. Figure 6.13 shows the increase in the loss ratio at LSR 2. Also figure 6.14 gives the packet delay variation.

Figure 6.15 gives the ingress traffic shift from the ingress to LSR 2. The figure shows the individual shift for each class of traffic. The mean values recorded from this plot are a shift of 2,100 packets/sec for the EF class and 5,200 packets/sec for the AF class.

- Simulation 2: Delay sensitive case

Figure 6.16 Variation of background throughput at LSR 2

Figure 6.17 Variation of Loss ratio at LSR 2

Figure 6.18 Variation of Average Packet delay at LSR 2

Figure 6.19 Ingress variation

Figures 6.16 to 6.19 are the results obtained for the delay sensitive scenario. For this scenario, the variation of the EF traffic at LSR 2 is made larger than the variation of the AF traffic increase, figure 6.16. The background EF and AF traffic rates increase from a mean of 4,000 packets/sec to a mean of 7,000 packets/sec and 5,500 packets/sec respectively. Figure 6.17 shows the variation of the loss ratio at LSR 2. A smaller increase in the loss ratio was recorded in comparison to the previous scenario (loss sensitive case). The variation of the average packet delay is shown in figure 6.18. When the average delay value reaches the threshold set at the ingress router, this triggers the Traffic Re-assignment algorithm and causes the traffic shift, figure 6.19 above. The shift only occurs for the ingress EF traffic. The recorded mean value for this shift is 6,400 packets/sec. The computed value for this scenario is a shift of 6,870 packets/sec. The difference in these values is due to the fact that the queues are measuring the actual background throughput variation but the computation uses the mean background throughput variation.

Figure 6.20 The Two-Node Model

6.3.2 The Two-Node Model

The two-node model considered in this section is shown in figure 6.20 above. Initially the ingress source traffic is sent through LSP **ingress --> LSR 2 --> LSR 3 --> egress**. In figure 6.20 above, LSR 2 represents the first intermediate LSR along the LSP and LSR 3 is the second. The background traffic variations are injected at LSR 2 and LSR 3. LSP **ingress --> LSR 5 --> LSR 4 --> egress** is lightly utilized.

For this section, scenario 3 considers a high loss ratio at LSR 3 due to a larger increase in the background AF traffic and scenario 4 considers a large delay due to a large variation of the background EF traffic at LSR 2 and LSR 3. All the computations are done using equations 5.1 and 5.2 from chapter 5. Since LSR 2 is the first intermediate LSR on the LSP, the computation done for LSR 2 considers the individual background variation at LSR 2. For LSR 3 the aggregate background traffic variation is used since the congestion at LSR 3 is caused by the background traffic variations at LSRs 2 and 3.

- Computation
- Scenario 3: Loss sensitive case

Tables 6.3 and 6.4 below show the individual and aggregate variation of the background throughput considered on each of the LSRs respectively.

	LSR 2	LSR 3
<i>delta_gamma_EF</i> (packets/sec)	1000	1200
<i>delta_gamma_AF</i> (packets/sec)	1500	1800

Table 6.3 Individual background throughput variation

	LSR 2	LSR 3
<i>delta_gamma_EF</i> (packets/sec)	1000	2200
<i>delta_gamma_AF</i> (packets/sec)	1500	3300

Table 6.4 Aggregate background throughput variation

The amount of traffic that should be shifted from LSR 2 is therefore:

$$\delta_{\lambda_{EF}} = 2.29 * 1000 = 2290 \text{ packets/sec};$$

$$\delta_{\lambda_{AF}} = 1.88 * 1500 = 2820 \text{ packets/sec};$$

The amount of traffic that should be shifted from LSR 3 is:

$$\delta_{\lambda_{EF}} = 2.29 * 2200 = 5038 \text{ packets/sec};$$

$$\delta_{\lambda_{AF}} = 1.88 * 3300 = 6204 \text{ packets/sec};$$

The upstream node closest to the ingress router therefore selects 5,038 and 6,204 packets/sec as the amount of EF and AF traffic that should be shifted respectively and send these values via the notification packet to the ingress router. The actual amount of traffic that is shifted at the ingress router for the EF and AF traffic are 5,038 packets/sec and 1,166 packets/sec respectively.

- Scenario 4: Delay sensitive case

Tables 6.5 and 6.6 below show the individual and aggregate variation of the background throughput considered on each of the LSRs respectively.

	LSR 2	LSR 3
$\delta_{\gamma_{EF}}$ (packets/sec)	1000	1200
$\delta_{\gamma_{AF}}$ (packets/sec)	500	600

Table 6.5 Individual background throughput variation

	LSR 2	LSR 3
<i>delta_gamma_EF</i> (packets/sec)	1000	2200
<i>delta_gamma_AF</i> (packets/sec)	500	1100

Table 6.6 Aggregate background throughput variation

The amount of traffic that should be shifted from LSR 2 is therefore:

$$\text{delta_lambda_EF} = 2.29 * 1000 = 2290 \text{ packets/sec;}$$

$$\text{delta_lambda_AF} = 1.88 * 500 = 940 \text{ packets/sec;}$$

The amount of traffic that should be shifted from LSR 3 is:

$$\text{delta_lambda_EF} = 2.29 * 2200 = 5038 \text{ packets/sec;}$$

$$\text{delta_lambda_AF} = 1.88 * 1100 = 2068 \text{ packets/sec;}$$

The upstream node closest to the ingress router therefore selects 5,038 and 2,068 packets/sec as the amount of EF and AF traffic that should be shifted respectively. The actual amount of traffic that is shifted at the ingress router is 5,038 packets/sec. This shift is only needed for the EF ingress traffic.

- Simulation results

Simulation 3 gives the results obtained when the values of the aggregate background traffic from scenario 3 is used and simulation 4 uses the aggregate background variation values from table 6.6 of scenario 4.

- Simulation 3: Loss sensitive case

Figure 6.21 Individual background throughput at LSR 2

Figure 6.22 Individual background throughput at LSR 3

Figure 6.23 Aggregate background throughput at LSR 3

Figure 6.24 Variation of LSP loss ratio

Figure 6.25 Variation of ingress source throughput

The objective of this scenario was to establish a high packet loss ratio along the LSP **ingress ---> LSR 2 --> LSR 3 --> egress**. The LSP loss ratio represents the sum of the loss ratios at both of the LSRs. Figures 6.21 and 6.22 above show the individual background variation used at LSRs 2 and 3 for each class of traffic. Figure 6.23 gives the aggregate variation at LSR 3. As shown in figure 6.23, there is an aggregate increase of 3,300 packets/sec of the AF traffic at LSR 3. This increase causes a large packet loss at LSR 3 and therefore along the LSP. Figure 6.24 shows the variation of the loss ratio at this LSP before and after the Traffic Re-assignment algorithm is triggered.

The individual ingress traffic shifts are shown in figure 6.25 above. The mean values recorded for the ingress traffic shifts are 5,000 packets/sec and 1,000 packets/sec for the EF

and AF classes respectively. The slight differences obtained for the computation and the simulation results are due the variation of the loss sensitivity during the simulations. Also actual traffic variations are measured during the simulations compared to the mean values used for doing the computation.

- Simulation 4: Delay sensitive case

Figure 6.26 Individual background throughput at LSR 2

Figure 6.27 Individual background throughput at LSR 3

Figure 6.28 Aggregate background throughput at LSR 3

Figure 6.29 Variation of packet delay along LSP

Figure 6.30 Variation of ingress source throughput

The background throughput variations for each class of traffic at LSRs 2 and 3 for this scenario are shown in figures 6.26, 6.27 and 6.28 above. This scenario was used to establish a large packet delay along the LSP **ingress --> LSR 2 --> LSR 3 --> egress**. The increase of the aggregate EF class at LSR 3 was 2,200 packets/sec. The average LSP packet delay represents the sum of the average delays at both LSRs. Figure 6.29 shows the variation of the average LSP packet delay before and after the Traffic Re-assignment algorithm is triggered. For this scenario, only the EF ingress traffic needs to be shifted. This is shown in figure 6.30 above. A mean value of 5,000 packets/sec was recorded for the ingress traffic shift.

The proposed scheme has shown to be effective for both the loss sensitive and the delay sensitive cases. The next section compares the performance of the scheme with the MATE algorithm.

6.3.3 Comparison of the Proposed Traffic Re-assignment Algorithm with MATE

A comparison was made between the proposed Traffic Re-assignment algorithm and the MATE algorithm using the two-node model. For the MATE algorithm the delay and loss values are collected at the egress router and the notification packet is sent to the ingress router. The proposed model computes the delay and loss values at the intermediate LSRs and forwards the packet to the ingress router. Three performance parameters were used in

the comparison. The drop of the loss ratio, the average packet delay and the response times for the two algorithms were compared.

The comparison was done for two scenarios. The first scenario considers the case when there is a large packet delay along the congested LSP, due to large variations of the EF background throughput, and the second scenario considers the case of a high loss ratio due to a large AF variation. The figures below show the performance of both algorithms. These values were obtained by running the simulation a number of times using different simulation seeds for each run. The mean of these values were then used to produce the given plots. The values were collected at specific time periods during the simulation.

Figure 6.31 Average packet delay variation for large EF variation

Figure 6.32 Loss variation for large EF variation

Figures 6.31 and 6.32 above show the performance of the proposed Traffic Re-assignment algorithm and the MATE algorithm for the delay sensitive case. Figure 6.31 shows large delay values were recorded for both algorithms but the values recorded for the average delay at specific simulation times are less when the Traffic Re-assignment algorithm is used instead of the MATE algorithm. Although the loss ratio was not high, both algorithms were able to reduce the loss values within the simulation time, figure 6.32. The MATE algorithm recorded higher loss values at specific simulation times than the Traffic Re-assignment algorithm. The main reason for the improvement in the drop of the delay in the

proposed Traffic Re-assignment algorithm is because the individual ingress traffic classes are shifted, while the MATE algorithm shifts the aggregate traffic. That is, for the delay sensitive case only the EF ingress traffic needs to be shifted. The switching of the aggregate traffic in the MATE algorithm reduces the amount of ingress EF traffic that can be shifted and therefore larger values of the average delay is observed.

Figure 6.33 Average packet delay variation for large AF variation

Figure 6.34 Loss variation for large AF variation

The performance of the proposed Traffic Re-assignment algorithm and the MATE algorithm for the loss sensitive case is shown in figures 6.33 and 6.34 above. High loss values were recorded for both algorithms but the loss values recorded at the specific simulation times were less when the Traffic Re-assignment algorithm is used instead of the MATE algorithm, figure 6.34. Also the values of the average packet delay are less during the specific simulation times when the Traffic Re-assignment algorithm is used, figure 6.33. For the loss sensitive case, both classes of traffic are shifted at the ingress router

when the Traffic Re-assignment algorithm is used. This results in lower values of both the delay and the loss along the congested path using the Traffic Re-assignment algorithm.

The mean response time was also used as a performance measure for the two algorithms. The mean response time is defined as the difference of the first time that the LSR loss ratio or the average packet delay reaches the threshold value and the packets are shifted to the alternate path. It can also be defined as the first time an intermediate LSR or the egress router generates a notification packet and the first time a packet is switched to the alternate LSP. Figures 6.35 and 6.36 below show the approximate times involved for the two-node model for the two algorithms.

Figure 6.35 Congestion detection for Traffic Re-assignment algorithm

Figure 6.36 Congestion detection for MATE algorithm

The following tables below give a comparison of the mean response times for the algorithms when different ingress arrival rates are used. For all the arrival rates used in the simulations the Traffic Re-assignment algorithm responds faster to congestion than the MATE algorithm. The main reason for this difference is that in the MATE algorithm the loss and delay values collected at the egress router, are sent to the ingress router through the notification packets which experience transmission and propagation delays at the intermediate LSRs and the LSPs respectively. In the proposed Traffic Re-assignment algorithm the notification packet is sent to the ingress router by the downstream node closest to it.

Table 6.7 gives the mean response times when the loss ratio is high due to a large variation of the background AF traffic at the intermediate LSRs. The mean response time for the Traffic Re-assignment algorithm is 8.5283 milliseconds for a mean arrival rate of 16,000 packets/sec. The equivalent mean response time for the MATE algorithm is 8.5557 milliseconds. Table 6.8 shows the case for a large packet delay due to a large variation of the background EF traffic at the intermediate LSRs. For a mean arrival rate of 16,000 packets/sec the mean response time recorded for the Traffic Re-assignment algorithm is 8.5284 milliseconds and for the MATE algorithm it is 8.5557 milliseconds.

Mean arrival rate (packets/sec)	Mean response time (msec)	
	MATE	TA
10,000	8.3241	8.3031
10,500	8.3284	8.3056
11,000	8.3301	8.3091
11,500	8.3322	8.3112
12,000	8.3340	8.3127
12,500	8.3806	8.3113
13,000	8.4370	8.4160
13,500	8.4827	8.4808
14,000	8.5427	8.50300
14,500	8.5475	8.5158
15,000	8.5508	8.5208
15,500	8.5539	8.5230
16,000	8.5557	8.5283
16,500	8.5579	8.5369
17,000	8.6089	8.5879
17,500	8.6261	8.6051
18,000	8.8253	8.8044
18,500	8.8379	8.8169
19,000	8.8401	8.8191
19,500	8.8502	8.8292
20,000	8.8553	8.8397

Table 6.7 Comparison of mean response times for large AF variation

Mean arrival rate (packets/sec)	Mean response time (msec)	
	MATE	TA
10,000	8.3232	8.3022
10,500	8.3265	8.3055
11,000	8.3300	8.3090
11,500	8.3321	8.3111
12,000	8.3340	8.3126
12,500	8.3803	8.3111
13,000	8.4369	8.4159
13,500	8.4826	8.4805
14,000	8.5427	8.5027
14,500	8.5467	8.5157
15,000	8.5506	8.5206
15,500	8.5538	8.5228
16,000	8.5557	8.5284
16,500	8.5578	8.5368
17,000	8.6089	8.5879
17,500	8.626	8.6050
18,000	8.8252	8.8042
18,500	8.8351	8.8152
19,000	8.8400	8.8190
19,500	8.8501	8.8291
20,000	8.8543	8.8381

Table 6.8 Comparison of mean response times for large EF variation

Figures 6.37, 6.38 and 6.39 below consider three different traffic combinations in the network. Figure 6.37 compares the response time at different mean ingress arrival rates when only EF traffic is present in the network and there is an increase in the EF

background traffic. The mean response time ranges from 8.3012 milliseconds to 8.3042 milliseconds for the proposed Traffic Re-assignment algorithm and 8.3211 milliseconds to 8.3235 milliseconds for the MATE algorithm when there is only EF traffic in the network. That is, there is an improvement of 0.024 milliseconds when the proposed Traffic Re-assignment algorithm is used.

Figure 6.37 Mean response times with strictly EF traffic in network

Figure 6.38 below considers the case when there is only AF traffic in the network. Slight differences were obtained for both algorithms when compared to the strictly EF case above. These differences could be due to the methods used for measuring the loss and the delay at

the queues. The mean difference in the response times recorded from this figure is 0.025 milliseconds.

Figure 6.38 Mean response times with strictly AF traffic in network

Finally, figure 6.39 below shows the case when both EF and AF traffic are present in the network. In this case we used an equal increase in the EF and the AF traffic at the same simulation time. Both algorithms show a decrease in the mean response times. This is due to the increase in the number of packets that are being transmitted bi-directionally through the links of the network. However, the Traffic Re-assignment algorithm still shows an improvement over the MATE algorithm of 0.018 milliseconds.

Figure 6.39 Mean response times when both AF and EF traffic are used in the network

6.3.4 Output Analysis of the Simulation Results

The output analysis is the examination of the data generated by the simulations. If the performance of the system is measured by a parameter θ , the result of a set of simulation experiments will be an estimator of θ . The precision of the estimator can be measured by the variance of the estimator, [40]. The purpose of the analysis is to find out if the differences between the MATE algorithm and the proposed Traffic Re-assignment algorithm are statistically significant.

The simulation results in this thesis are all discrete-time data, they are of the form $\{Y_{1,i}, Y_{2,i}, \dots, Y_{n,i}\}$, n is the number of independent replications and i represent the model type. The parameter θ_i is defined as the mean of the output from model i .

Let θ_M represent the mean output from the MATE model and θ_{TA} the mean output from the Traffic Re-assignment model. The differences in the output data for the two models is represented by

$$D_p = Y_{p,M} - Y_{p,TA} \quad (6.1)$$

Here p represents the replication number.

The point estimate is therefore defined by

$$\bar{D} = (\bar{Y}_{p,M} - \bar{Y}_{p,TA}) = \frac{1}{n} \sum_{r=1}^n D_p \quad (6.2)$$

$\bar{Y}_{p,i}$ is the sample mean performance measure for system i .

The sample variance of the differences $\{D_p\}$ is computed by

$$S_D^2 = \frac{1}{n-1} \sum_{r=1}^n (D_p - \bar{D})^2 \quad (6.3)$$

which has $f = n - 1$ degrees of freedom.

The standard error of $\bar{D}$ is estimated by

$$s.e.(\bar{D}) = s.e.(\bar{Y}_{p,M} - \bar{Y}_{p,TA}) = \frac{S_D}{\sqrt{n}} \quad (6.4)$$

The two-sided $100(1-\alpha)\%$ confidence interval is of the form

$$\bar{D} \pm t_{\alpha/2, f} s.e.(\bar{D}) \quad (6.5)$$

$t_{\alpha/2, f}$ is defined as the $100(1-\alpha/2)$ percentage point of a t-distribution with f degrees of freedom. The t-distribution statistic for a 95% confidence interval with 29 degrees of freedom is

$$t_{0.05/2, 29} = 2.0452 \quad (6.6)$$

A confidence interval quantifies the confidence that can be ascribed to the results from a simulation experiment, [41]. A 95% confidence interval means that for 95% of the simulation runs for which an interval is calculated, the actual value for the measure of interest falls within the calculated interval and 5% of occasions the actual value falls outside the calculated value.

The following subsections give the confidence intervals for the average response times, the average packet delay and the average loss ratio for the two models. For each simulation model, a total of 30 replications were used and the simulation run time was set at 8 seconds. For each replication, the same simulation seed was used to simulate both algorithms and for the same algorithm each replication was given a different seed.

- Confidence interval for the average response times

The simulation results for the average response times for the MATE algorithm and the Traffic Re-assignment algorithm are collected for analysis. These results are collected under three different ingress traffic rates. Tables 6.9 and 6.10 show the simulation results for the MATE algorithm and the Traffic Re-assignment algorithm respectively. Ingress traffic rates of 12,000, 16,000 and 20,000 packets/sec were used in the simulation. The tables also give the sample mean and the sample variance for the collected data. Lower values of the average response times were recorded for the Traffic Re-assignment algorithm than the values recorded for the MATE algorithm.

Simulation Sequence number	Response times under different ingress traffic rates (msec)		
	12,000 packets/sec	16,000 packets/sec	20,000 packets/sec
1	8.3326	8.5622	8.8524
2	8.3327	8.5598	8.8542
3	8.3329	8.5511	8.8513
4	8.3324	8.5422	8.8532
5	8.3324	8.558	8.8523
6	8.3327	8.5512	8.8534
7	8.3321	8.5531	8.8514
8	8.3325	8.5513	8.8543
9	8.3362	8.5565	8.8556
10	8.3313	8.5581	8.8524
11	8.3321	8.5515	8.8607
12	8.3311	8.5602	8.8606
13	8.3313	8.5561	8.8603
14	8.3324	8.5513	8.8526
15	8.3312	8.5514	8.8518
16	8.3334	8.5514	8.8523
17	8.3401	8.5602	8.8602
18	8.3431	8.5601	8.8543
19	8.333	8.5545	8.8568
20	8.3402	8.5569	8.8547
21	8.343	8.5585	8.8604
22	8.3361	8.5601	8.8532
23	8.3344	8.5579	8.8542
24	8.3312	8.5587	8.8536
25	8.3316	8.5578	8.8544
26	8.3313	8.5602	8.8549
27	8.3341	8.5602	8.8555
28	8.3409	8.5563	8.8603
29	8.3321	8.5531	8.8604
30	8.3319	8.5519	8.8543
Sample Mean	8.3340	8.5557	8.8552
Sample Variance	1.31419E-05	1.94793E-05	1.01014E-05

Table 6.9 Response times for the MATE algorithm

Simulation Sequence number	Response times under different ingress traffic rates (msec)		
	12,000 packets/sec	16,000 packets/sec	20,000 packets/sec
1	8.3127	8.5312	8.8323
2	8.3125	8.5413	8.8412
3	8.3128	8.5212	8.8212
4	8.3122	8.5321	8.8321
5	8.3127	8.5381	8.8722
6	8.3126	8.5211	8.8367
7	8.3121	8.5391	8.8315
8	8.312	8.5312	8.8243
9	8.3161	8.5271	8.8453
10	8.3103	8.5283	8.8335
11	8.3162	8.5315	8.8567
12	8.3114	8.5332	8.8056
13	8.3213	8.5281	8.8673
14	8.3124	8.5214	8.8135
15	8.3112	8.5317	8.8238
16	8.3134	8.5233	8.8245
17	8.3201	8.5212	8.8673
18	8.3131	8.5284	8.8446
19	8.3105	8.5361	8.8373
20	8.3112	8.5372	8.8566
21	8.3102	8.5281	8.8673
22	8.3111	8.5003	8.8015
23	8.3218	8.5299	8.7998
24	8.3101	8.5346	8.8499
25	8.3102	8.5298	8.8567
26	8.3112	8.5018	8.8349
27	8.3101	8.5388	8.8569
28	8.3109	8.5363	8.8678
29	8.3101	8.5313	8.8634
30	8.3114	8.5211	8.8245
Sample Mean	8.3127	8.5284	8.8396
Sample Variance	1.02162E-05	8.93213E-05	0.000429818

Table 6.10 Response times for the Traffic Re-assignment algorithm

The simulation result analysis for the response times for the MATE and the Traffic Re-assignment algorithms can be represented as follows:

When the ingress traffic rate is 12,000 packets/sec, using equations 6.2 and 6.4, the point estimate is $2.1281\text{E-}02$ and the standard error is $8.463\text{E-}04$. The 95% confidence interval for the difference in the expected response times is represented as:

$$2.1281\text{E-}02 - 1.7308\text{E-}03 \leq \theta_M - \theta_{TA} \leq 2.1281\text{E-}02 + 1.7308\text{E-}03$$

When the ingress traffic rate is 16,000 packets/sec, the point estimate is $2.7233\text{E-}02$ and the standard error is $1.9407\text{E-}03$. The 95% confidence interval for the difference in expected response times is represented as:

$$2.7233\text{E-}02 - 3.9691\text{E-}03 \leq \theta_M - \theta_{TA} \leq 2.7233\text{E-}02 + 3.9691\text{E-}03$$

For an ingress traffic rate of 20,000 packets/sec, the point estimate is $1.5512\text{E-}02$ and the standard error is $3.5371\text{E-}03$. The 95% confidence interval for the difference in expected response times is represented as:

$$1.5512\text{E-}02 - 7.2341\text{E-}03 \leq \theta_M - \theta_{TA} \leq 1.5512\text{E-}02 + 7.2341\text{E-}03$$

For all three ingress traffic rates used, the observed differences are statistically significant since the 95% confidence intervals do not contain zero. The estimate $\theta_M - \theta_{TA}$ for the ingress traffic of 12,000 packets/sec is more precise than the estimates for ingress traffic rates of 16,000 and 20,000 packets/sec. The observed difference is lower at 20,000 packets/sec and is the highest at 16,000 packets/sec.

- Confidence interval for the average packet delay

The simulation results for the average packet delay for the MATE and the Traffic Re-assignment algorithms are collected for analysis. These results are collected at three specific simulation times after the traffic shift to the alternate LSP. Tables 6.11 and 6.12 show the simulation results for the MATE algorithm and the Traffic Re-assignment algorithm respectively. The average packet delays for both algorithms were recorded at simulation times 5.0, 6.0 and 7.0 seconds. The tables also give the sample mean and the sample variance for the collected data. The Traffic Re-assignment model recorded lower values for the average packet delay than the MATE model.

Simulation Sequence number	Average packet delay at specific simulation times (msec)		
	5.0 seconds	6.0 seconds	7.0 seconds
1	0.0982	0.0895	0.0793
2	0.098	0.0889	0.0784
3	0.0983	0.0883	0.0729
4	0.0997	0.0881	0.0763
5	0.0971	0.0879	0.0785
6	0.0966	0.0895	0.0787
7	0.0978	0.0891	0.0799
8	0.0971	0.0888	0.0788
9	0.0975	0.0893	0.0741
10	0.0978	0.0897	0.0797
11	0.0999	0.0889	0.0789
12	0.0988	0.0884	0.0804
13	0.0985	0.0892	0.0796
14	0.0967	0.0887	0.0798
15	0.0984	0.0885	0.0803
16	0.0987	0.0897	0.0801
17	0.0971	0.0886	0.0732
18	0.0983	0.0897	0.0796
19	0.0978	0.0889	0.0801
20	0.0989	0.0886	0.0806
21	0.0993	0.0895	0.0799
22	0.0979	0.0889	0.0745
23	0.0971	0.0887	0.0777
24	0.0982	0.0881	0.0799
25	0.0995	0.0894	0.0794
26	0.0971	0.0887	0.0801
27	0.0979	0.0882	0.0779
28	0.0982	0.089	0.0805
29	0.0983	0.0881	0.0772
30	0.0991	0.0892	0.0794
Sample Mean	0.0981	0.0888	0.0785
Sample Variance	7.58575E-07	2.79414E-07	4.80185E-06

Table 6.11 Average packet delay for the MATE algorithm

Simulation Sequence number	Average packet delay at specific simulation times (msec)		
	5.0 seconds	6.0 seconds	7.0 seconds
1	0.0942	0.0852	0.0773
2	0.0953	0.0847	0.0761
3	0.0963	0.0853	0.0698
4	0.0977	0.0841	0.0723
5	0.0931	0.0849	0.0745
6	0.0926	0.0854	0.0757
7	0.0938	0.0861	0.0779
8	0.0971	0.0858	0.0778
9	0.0935	0.0863	0.0695
10	0.0922	0.0857	0.0757
11	0.0989	0.0848	0.0699
12	0.0918	0.0861	0.0724
13	0.0915	0.0891	0.0756
14	0.0927	0.0853	0.0757
15	0.0924	0.0863	0.0763
16	0.0967	0.0867	0.0788
17	0.0961	0.0856	0.0634
18	0.0973	0.0867	0.0686
19	0.0938	0.0869	0.0721
20	0.0939	0.0855	0.0796
21	0.0933	0.0859	0.0736
22	0.0959	0.0848	0.0689
23	0.0901	0.0854	0.0718
24	0.0972	0.0847	0.0789
25	0.0955	0.0858	0.0754
26	0.0931	0.0857	0.0738
27	0.094	0.0849	0.0689
28	0.0922	0.0854	0.0795
29	0.0993	0.0851	0.0673
30	0.0971	0.0855	0.0734
Sample Mean	0.0946	0.0856	0.0736
Sample Variance	5.5389E-06	8.54954E-07	1.64456E-05

Table 6.12 Average packet delay for the Traffic Re-assignment
algorithm

The simulation result analysis for the average packet delay for the MATE and the Traffic Re-assignment algorithms can be represented as follows:

When the simulation time is 5.0 seconds, using equations 6.2 and 6.4, the point estimate is 3.5066E-03 and the standard error is 3.9231E-04. The 95% confidence interval for the difference in the expected packet delay is represented as:

$$3.5066E-03 - 8.0235E-04 \leq \theta_M - \theta_{TA} \leq 3.5066E-03 + 8.0235E-04$$

When the simulation time is 6.0 seconds, the point estimate is 3.2133E-03 and the standard error is 1.5160E-04. The 95% confidence interval for the difference in expected packet delay is represented as:

$$3.2133E-03 - 3.1005E-04 \leq \theta_M - \theta_{TA} \leq 3.2133E-03 + 3.1005E-04$$

For the 7.0 seconds simulation time, the point estimate is 4.8401E-03 and the standard error is 5.4180E-04. The 95% confidence interval for the difference in expected packet delay is represented as:

$$4.8401E-03 - 1.1079E-03 \leq \theta_M - \theta_{TA} \leq 4.8401E-03 + 1.1079E-03$$

The observed differences at all three simulation times are statistically significant since the 95% confidence intervals do not contain zero. An observed difference of 4.8401E-03 was obtained for the simulation time of 7.0 seconds. Smaller observed differences of 3.5066E-03 and 3.2133E-03 were obtained for the 5.0 and 6.0 seconds simulation times respectively. The estimates $\theta_M - \theta_{TA}$ for the three simulation times are precise.

- Confidence interval for the Loss ratio

The simulation results for the loss ratio for the MATE and the Traffic Re-assignment algorithms were also collected for analysis. These results are collected at three specific simulation times after the traffic shift to the alternate LSP. Tables 6.13 and 6.14 show the simulation results for the MATE algorithm and the Traffic Re-assignment algorithm respectively. The Loss ratios for both algorithms were recorded at simulation times of 4.5, 5.5 and 6.5 seconds. The tables also give the sample mean and the sample variance for the collected data. The values for the loss ratio for the Traffic Re-assignment model were lower than the values collected for the MATE model.

Simulation Sequence number	Loss ratio at specific simulation times		
	4.5 seconds	5.5 seconds	6.5 seconds
1	0.0182	0.0122	0.0114
2	0.0183	0.0123	0.0115
3	0.0181	0.0121	0.0113
4	0.0179	0.0119	0.0111
5	0.018	0.012	0.0112
6	0.0178	0.0118	0.011
7	0.0184	0.0124	0.0116
8	0.0183	0.0123	0.0115
9	0.0187	0.0127	0.0119
10	0.0188	0.0128	0.012
11	0.018	0.012	0.0112
12	0.0179	0.0119	0.0111
13	0.0175	0.0115	0.0107
14	0.0186	0.0126	0.0118
15	0.0183	0.0123	0.0115
16	0.0182	0.0122	0.0114
17	0.0181	0.0121	0.0113
18	0.0179	0.0119	0.0111
19	0.0178	0.0118	0.011
20	0.0176	0.0116	0.0108
21	0.0179	0.0119	0.0111
22	0.0181	0.0121	0.0113
23	0.0187	0.0127	0.0119
24	0.0185	0.0125	0.0117
25	0.0189	0.0129	0.0121
26	0.0184	0.0124	0.0116
27	0.0187	0.0127	0.0119
28	0.018	0.012	0.0112
29	0.0179	0.0119	0.0111
30	0.0176	0.0116	0.0108
Sample Mean	0.0182	0.0122	0.0114
Sample Variance	1.40103E-07	1.35433E-07	1.40103E-07

Table 6.13 Loss ratio for the MATE algorithm

Simulation Sequence number	Loss ratio at specific simulation times		
	4.5 seconds	5.5 seconds	6.5 seconds
1	0.0162	0.0102	0.0094
2	0.0168	0.0108	0.01
3	0.0163	0.0103	0.0095
4	0.0166	0.0106	0.0098
5	0.0166	0.0106	0.0098
6	0.0167	0.0107	0.0099
7	0.0159	0.0099	0.0091
8	0.0163	0.0103	0.0095
9	0.0168	0.0108	0.01
10	0.0169	0.0109	0.0101
11	0.0162	0.0102	0.0094
12	0.0165	0.0105	0.0097
13	0.0163	0.0103	0.0095
14	0.0169	0.0109	0.0101
15	0.0164	0.0104	0.0096
16	0.0168	0.0108	0.01
17	0.0164	0.0104	0.0096
18	0.0162	0.0102	0.0094
19	0.0161	0.0101	0.0093
20	0.016	0.01	0.0092
21	0.0158	0.0098	0.009
22	0.0156	0.0096	0.0088
23	0.0167	0.0107	0.0099
24	0.0165	0.0105	0.0097
25	0.0169	0.0109	0.0101
26	0.0161	0.0101	0.0093
27	0.0166	0.0106	0.0098
28	0.0162	0.0102	0.0094
29	0.0165	0.0105	0.0097
30	0.016	0.01	0.0092
Sample Mean	0.0163	0.0104	0.0096
Sample Variance	1.20644E-07	1.20644E-07	1.20644E-07

Table 6.14 Loss ratio for the Traffic Re-assignment algorithm

The simulation result analysis for the loss ratio for the MATE and the Traffic Re-assignment algorithms can be represented as follows:

When the simulation time is 4.5 seconds, using equations 6.2 and 6.4, the point estimate is $1.7166\text{E-}03$ and the standard error is $6.4321\text{E-}05$. The 95% confidence interval for the difference in the expected loss ratio is represented as:

$$1.7166\text{E-}03 - 1.3154\text{E-}04 \leq \theta_M - \theta_{TA} \leq 1.7166\text{E-}03 + 1.3154\text{E-}04$$

When the simulation time is 5.5 seconds, the point estimate is $1.7766\text{E-}03$ and the standard error is $6.4182\text{E-}05$. The 95% confidence interval for the difference in expected loss ratio is represented as:

$$1.7766\text{E-}03 - 1.3127\text{E-}04 \leq \theta_M - \theta_{TA} \leq 1.7766\text{E-}03 + 1.3127\text{E-}04$$

For the 6.5 seconds simulation time, the point estimate is $1.7822\text{E-}03$ and the standard error is $6.8320\text{E-}05$. The 95% confidence interval for the difference in expected loss ratio is represented as:

$$1.7822\text{E-}03 - 1.3972\text{E-}04 \leq \theta_M - \theta_{TA} \leq 1.7822\text{E-}03 + 1.3972\text{E-}04$$

The observed differences for the specific simulation times are statistically significant since the 95% confidence intervals do not contain zero. The results show very similar observed differences for the specific times used in the simulations. The estimates $\theta_M - \theta_{TA}$ for the three simulation times are precise.

In all three of the performance measures used in the simulations, the 95% confidence interval for $\theta_M - \theta_{TA}$ is greater than zero. The differences between the MATE algorithm and the Traffic Re-assignment algorithm are statistically significant. The proposed scheme is considered a better model than the MATE scheme.

Chapter 7 Discussion

Network operators face the challenge of maintaining customer guarantees whilst sustaining profitability. Providers have adapted the approach of mapping traffic onto a physical topology by using list cost metrics evaluated by IGP. This approach has many limitations, some of which were often resolved by over-provisioning bandwidth along links that were anticipated to be heavily loaded or by employing limited forms of Load-sharing. As networks continue to grow larger, and the demands of customers become greater (in terms of required bandwidth and QoS), the mapping of traffic onto physical topologies needs to be approached in a way that will allow the offered load to be supported in a controlled and efficient manner.

The aim of this research has been to develop a scheme that can be used to reduce the delay of EF class traffic and the loss of AF class traffic during congestion periods along LSPs in the MPLS domain. In achieving this, we have developed mechanisms that can monitor the delay, the loss and the variation of traffic at the intermediate LSRs. We have also developed a novel method for temporarily shifting traffic at the ingress router of the MPLS domain.

The proposed Traffic Re-assignment scheme employs mechanisms for reacting to congestion at the intermediate LSRs and at the ingress router. In the MPLS domain, there

are two types of LSPs that can be established: explicitly routed and loosely routed. The proposed scheme uses the explicitly routed LSPs for performing traffic shifts.

Other mechanisms that have been proposed for the detection and reaction to congestion in the MPLS domain are the FATE, FATE+ and MATE schemes. In the FATE scheme, upon detecting congestion, the congested LSR formulates a notification message and transmits it to the ingress router that the LSP originates from. This method places the responsibility of what appropriate action that should be taken onto the ingress router. The FATE scheme uses the explicitly routed LSPs and therefore the ingress router may have a specific route it wishes its traffic to traverse. Thus, by informing the ingress router of potential problems, the scheme allows the ingress router to decide on an appropriate reaction to resolve the problem.

The FATE+ scheme uses the loosely routed LSP where the ingress router has no knowledge of the actual links/LSRs that the traffic is traversing. Upon detection of congestion, the appropriate action is taken by the congested LSR. The LSR can transfer the traffic into a higher QoS stream via a different downstream LSR or via an alternative upstream LSR.

The MATE scheme avoids network congestion by adaptively balancing the load among multiple LSPs based on measurement and analysis of path congestion. The scheme does not require the intermediate LSRs to perform traffic engineering or measurements.

The measurement of losses or delays is done at the ingress or egress router. The shifting of traffic occurs at the ingress router.

There are two fundamental differences between the proposed Traffic Re-assignment scheme and the three schemes discussed. Firstly, the proposed scheme uses mechanisms to compute the amount of each class of traffic that should be shifted away from a congested LSP. This computation is done at the intermediate LSRs. Secondly, at the ingress router, the shifting of individual classes of traffic is done. The advantage of such a method is that the alternate LSP as well as the congested LSP are prevented from being over-utilized or under-utilized after the traffic is shifted. In the FATE, FATE+ and the MATE schemes there is no guarantee that after the traffic is re-routed the LSPs will not be over or under-utilized.

The Traffic Re-assignment scheme uses explicit routed LSPs. In comparison to the FATE+ scheme that uses the loosely routed LSPs, there is no need for LSP setup during congestion periods since they have already been established. This therefore leads to an overall improvement in the response time in the Traffic Re-assignment scheme over the FATE+ scheme.

The FATE and FATE+ schemes are designed to measure only the loss of traffic along the LSPs. The proposed Traffic Re-assignment scheme and also the MATE scheme are designed to measure both the loss and delay of Low Priority traffic and High Priority traffic

respectively. As stated previously, customers demand for better QoS are greater and the guarantee for only low loss of traffic is insufficient for today's networks. As networks continue to receive increasingly more mission-critical traffic for transmission, the QoS requirements have also increased to include not only lower loss, but also faster transmission and better quality, especially for Voice and Video traffic. The FATE and FATE+ schemes can only guarantee a lower loss of traffic. This makes the MATE scheme the best model for comparison with the proposed Traffic Re-assignment scheme since it considers the same guarantees as the Traffic Re-assignment scheme. Also both the MATE and the proposed Traffic Re-assignment schemes use the established explicit LSPs. That is, the individual LSPs from both schemes can be used for analysis.

Both the MATE and the proposed Traffic Re-assignment schemes have been evaluated through simulations. The results have shown an improvement in the response times, the loss and the delay for the proposed Traffic Re-assignment scheme. The simulation results have shown a reduction in the mean response time of the order of 10^{-2} when the Traffic Re-assignment algorithm is used instead of the MATE algorithm. In the Traffic Re-assignment algorithm, the notification packets are sent to the ingress router from the first downstream LSR along the congested LSP. In the MATE algorithm, the notification packet is transmitted from the egress router. The difference in the response times for the

two algorithms is due to the difference in the propagation delays experienced by notification packets along the congested LSP.

A reduction of the order of 10^{-3} was also observed for the average packet delay when the Traffic Re-assignment algorithm is used instead of the MATE algorithm. The simulation results also show a reduction in the loss ratio of the order of 10^{-3} for the proposed scheme. Two factors have contributed to the reduction of the loss ratio and the average delay. Firstly, the improvement of the response times of the Traffic Re-assignment algorithm over the MATE algorithm has enabled the shifting of traffic onto the alternate LSPs at a faster rate. This therefore causes lower loss ratios and packet delays along the congested LSP after the traffic shift. Secondly, individual EF and AF traffic are shifted at the ingress router in comparison to the aggregate shift used in the MATE algorithm. This shifting method enables computed amounts of EF and AF traffic to be removed from the congested LSP and therefore reduces individual losses and delays along the LSP.

From the customer's perspective, reduction in losses and delays of their traffic is considered encouraging since there is a greater demand for faster transmission and a more reliable service. From the provider's perspective, the improvement in the mean response times to congestion is considered significant since customers can be guaranteed lower losses and delays of their traffic. Overall the proposed Traffic Re-assignment algorithm

has shown improvements and can be considered a better scheme for reacting to congestion in the MPLS domain.

Chapter 8 Conclusion and Further Work

8.1 Conclusion

The objectives set out for this research have been achieved. A scheme capable of reacting to increases in congestion along LSPs has been developed. It is specifically designed for the MPLS domain.

It is assumed that a set of parallel LSPs has been pre-established between an ingress and egress router and traffic from another source enters this MPLS domain at the intermediate LSRs and exit at the egress router. This traffic contributes to the statistical variation of all the traffic along the LSP, which sometimes leads to congestion at the LSRs. The proposed monitoring and measuring scheme at the intermediate LSRs, is capable of capturing this variation and computing the amount of traffic that should be taken away from the LSRs. Each LSR has assigned to it an upper and lower loss and packet delay threshold. Whenever the loss or delay at a LSR is at the upper threshold, a notification packet is created and the values of the loss, delay and the amount of traffic that should be shifted is sent to an upstream node or the ingress router. This information is used to do the shifting at the ingress router. Also, if the loss and delay at the LSR is below the lower threshold, the LSR generates a notification packet and sends it upstream to the ingress router. This packet contains only the loss and delay values and is used as a signal to the

ingress router that it can shift traffic unto the LSP that this LSR is part of. This ensures that the traffic that was initially shifted to the alternate path can be redistributed back to the original LSP.

We have proposed a novel mechanism at the ingress router that will conserve bandwidth along the LSP the traffic is being switched to and also on the one it is being switched from. At the ingress router, the individual traffic class is shifted. The amount of each class of traffic shifted is dependent on the values obtained from the downstream LSRs. Whenever there is a need to shift a greater amount of EF traffic than the AF traffic, then only the EF traffic is shifted. This shift reduces both the delay and loss along the LSP. If however, more AF traffic needs to be shifted than the EF traffic, the computed values of the EF shift required, is shifted for the EF traffic, and only the difference of the computed amounts of the EF and AF traffic is shifted for the AF traffic. In comparison to shifting the computed amounts of both classes of traffic, this method conserves some of the available bandwidth on the alternate LSP. The amount of bandwidth conserved in the first case, is the computed amount of AF traffic that should be shifted and in the latter, it is the computed amount of EF traffic that should be shifted. Using these two mechanisms described above, we were able to develop the Traffic Re-assignment scheme, which is capable of reducing the loss of AF traffic and the delay of EF traffic along congested LSPs in the MPLS domain.

Simulations have been used to show the performance of the proposed Traffic Re-assignment algorithm. A two-node model was used to compare the performance of the algorithm with the MATE algorithm. Initially, the ingress traffic is sent through the network using a single LSP and background source traffic is injected at the intermediate LSRs. The flows are classified as EF and AF traffic, which means that they require minimum guarantees of delay and loss respectively. At some time during the transmission of the traffic, the mean arrival rate of the background traffic at the LSRs increases. The presence of this increase creates levels of congestion along certain LSRs along the LSP and triggers the Traffic Re-assignment algorithm at the ingress router. The simulation results show that the Traffic Re-assignment algorithm responds to congestion along the LSP faster than the MATE algorithm.

The scheme is specific to a single ingress-egress pair and only considers the case when there is a temporary increase in the background traffic. Also the scheme assumes that there is always an alternate under-utilized LSP into which the traffic can be re-distributed. In comparison to the FATE+ scheme, the proposed Traffic Re-assignment algorithm can only do the traffic shifting at the ingress router.

8.2 Further Work

Further work could evaluate the impact of using multiple ingress-egress pairs in the network and applying the re-assignment algorithm at each ingress router. In such a model,

each intermediate LSR would monitor the variation of traffic from each ingress source and send notification messages to the ingress router whose traffic is responsible for the congestion at a LSR. The scheme will have to be extended to ensure that the shifting mechanism is not triggered in two or more routers at the same time. The performance of this scheme could be evaluated by how quickly the network stabilizes.

Application traffic such as Voice, Video and Email could also be applied to the scheme to observe the actual impact of the delay, loss and throughput of the traffic. Various types of background sources can also be used along different LSPs. That is, instead of using EF and AF background traffic along the LSP, other types of traffic such as TCP or Best Effort traffic could be added to the model.

In this scheme, the notification packet is sent to the ingress router and the traffic is also re-routed from this router. Further work on this scheme could look at the performance of sending the notification packet to the next upstream LSR that shares another LSP. That is, whether there will be any significant improvement in the delay or loss when the traffic is re-routed from the next possible upstream LSR instead of the ingress router. The performance of this scheme could be compared with the FATE+ scheme. More than one notification packet could also be used to determine whether some of the traffic on the LSP should be re-routed. This would give a more accurate confirmation for the switchover.

The simulation results depicting the performance of the Traffic Re-assignment scheme as an individual mechanism have been shown. However, if the scheme has to be implemented into a real network, network operators will be expected to maintain additional explicit LSPs that can be used for the re-route of the traffic. The model can also be advanced to use dynamic LSPs. That is, the LSPs can be established only when they are needed.

Another area of further work for this research involves the use of proactive mechanisms in the MPLS domain. This would include the use of intelligent agents together with Traffic Engineering metrics to predict the onset of congestion prior to it actually arising and to take proactive action accordingly.

References

- [1] L.Li, N.Feldman, L.Anderson et al, "IP Traffic Engineering Using MPLS Explicit Routing in Carrier Networks,"
<http://www.nortelnetworks.com/products/announcements/mpls/source/collateral/whitepaper.html> Feb. 2000.
- [2] R. Doverspike and J. Yates, "Challenges for MPLS in Optical Network Restoration," IEEE Communications Magazine, vol.39, Issue: 2, Feb. 2001, pp.89-96.
- [3] R.Onvural, "Asynchronous Transfer Mode Networks," 2nd ed., 535 pp. 1995.
- [4] G.Armitage, "MPLS: The Magic Behind the Myths," IEEE Communications Magazine, Vol.38, Issue: 1, Jan.2000, pp.124 –131.
- [5] A.Elwalid, C.Jin, S.Low and I.Widjaja, "MATE: MPLS Adaptive Traffic Engineering," INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, Vol.3, 2001, pp.1300-1309.
- [6] E.Rosen, A.Viswanathan and R.Callon, "Multiprotocol Label Switching Architecture," <http://search.ietf.org/internet-draft-ietf-mpls-arch-06.txt>, Jan. 2001.

- [7] Z.Cao, Z.Wang and E.Zegura, "Performance of Hashing-Based Schemes for Internet Load Balancing," INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, Vol.1, 2000, pp.332-341.
- [8] K.Kar, M.Kodialam and T.V.Lakshman, "Minimum Interference Routing of Bandwidth Tunnels with MPLS Traffic Engineering Applications," IEEE Journal on Selected areas in communications, Vol. 18, No. 12, Dec. 2000.
- [9] S.Blake, D.Black, M.Carlson, E.Davies, Z.Wang and W.Weiss, "An Architecture for Differentiated Services," <http://www.ietf.org/rfc/rfc2475.txt>, Dec. 1998.
- [10] M.Shayman and R.Jaeger, "Using ECN to Signal Congestion within an MPLS Domain," <http://www.ietf.org/internet-drafts/draft-shayman-mpls-ecn-00.txt>, Nov.2000.
- [11] R.Rabbit, K.Laberteaux, N.Modi and J.Kenny, "Traffic engineering algorithms using MPLS for service differentiation," Communications, 2000. ICC 2000. 2000 IEEE International Conference, Vol.2, 2000, pp.791-795, vol.2.
- [12] Cisco Systems. "Quality of Service in MPLS Networks," http://www.cisco.com/univercd/cc/td/doc/product/wanbu/bpx8600/mppls/9_3_1/mppls04.htm, May 2001.

- [13] Y. Afek, A. and Bremner-Barr, "Trainet: a new label switching scheme,"
INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and
Communications Societies. Proceedings. IEEE, Vol.2, pp.874-883.
- [14] T. H. Oh, T. M. Chen and J. L. Kennington, "Fault Restoration and Spare Capacity
Allocation with QoS Constraints for MPLS Networks," Global Telecommunications
Conference, 2000. GLOBECOM '00. IEEE, Vol.3, pp.1731-1735.
- [15] T. M. Chen and T. H. Oh, "Reliable Service in MPLS," IEEE Communications
Magazine, vol.37, Issue: 12, December 1999, pp.58-62.
- [16] L. Andersson, P. Doolan, N. Feldman, et al, "LDP Specification,"
<http://search/ietf.org/internet-drafts/draft-ietf-mpls-ldp-06.txt>, Oct. 1999,02/03/00.
- [17] U. Black, "MPLS and Label Switching Networks," Prentice-Hall, 2001.
- [18] D.O. Awduche, J. Malcom, et al, "Requirements for Traffic Engineering over
MLPS," <http://www.kblabs.com/lab/lib/rfcs/2700/rfc2702.txt.html>, Sept.1999.
- [19] P. Bhaniramka, W. Sun and R. Jain, "Quality of Service using Traffic Engineering
over MPLS: An Analysis," Local Computer Networks, 2000. LCN 2000.
Proceedings. 25th Annual IEEE Conference, 2000, pp.238-241.
- [20] R. Collen, P. Dolan, N. Feldman, et al, "A Framework for Multiprotocol Label
Switching," <http://www.globecom.net/ietf/draft/draft-ietf-mpls-framework-02.html>,
Sept.1999.

- [21] B. Davie and Y. Rekhter, "MPLS Technology and Applications," Morgan Kaufmann, 2000.
- [22] X. Xiao, A. Hannan, B. Bailey and L.M. Ni, "Traffic Engineering with MPLS in the Internet," IEEE Network, Vol.14 Issue: 2, March-April2000, pp. 28 –33.
- [23] D.O. Awduche, "MPLS and Traffic Engineering in IP Networks," IEEE Communications Magazine, and Vol. 37 Issue: 12, Dec. 1999 pp.42 –47.
- [24] S. Keshav, "A Control-Theoretic Approach to Flow Control," SIGCOM'91, pp. 3-15, Sept. 1991.
- [25] H. M. Felicia, "Congestion Control Mechanisms within MPLS Networks," PhD Thesis, University of London, Sept.2000.
- [26] E.Dinan, D.O.Awdauche and B.Jabbari, "Analytical Framework for Dynamic Partitioning in MPLS networks," Communications, 2000 ICC 2000. 2000 IEEE International Conference, Vol.3, pp.1604 -1608.
- [27] W.L. Winston, "Operations Research, Application and Algorithms," Duxbury Press, 1994.
- [28] A. Jüttner, B. Szviatovszki, I. Mécs, and Z. Rajkó, "Lagrange Relaxation Based Method for the QoS Routing Problem," Proceedings of IEEE, INFOCOM, vol.2, Alaska, April 2001, pp.859-868.

- [29] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski, "Assured Forwarding PHB Group," <http://www.ietf.org/rfc/rfc2475.txt>, June 1999.
- [30] S. Dharanikota, et al, "OSPF, IS-IS, OSPF, RSVP, and CR-LDP extensions to support inter-area traffic engineering," Work in progress, draft-dharanikota-interarea-mpls-te-ext-01, 2001.
- [31] R.Braden, L.Zang, S.Berson, S.Herzog and S.Jamin, "Resource ReSerVation Protocol (RSVP)," <http://www.faqs.org/rfcs/rfc2208.html>, Sept. 1997.
- [32] D. Bersekas and R. Gallager. "Data networks," Prentice-Hall Inc., 2nd ed. Edition, 1992.
- [33] J. E. Cohen and C. Jeffries, "Congestion Resulting from Increased Capacity in Single-Server Queueing Networks," IEEE/ACM Transactions on Networking, vol. 5, Issue: 2, April 1997, pp.305-310.
- [34] R.W. Wolf, "Time sharing with priorities," SIAM J. Appl. Math, vol.19, no.3, pp. 566-574, 1970.
- [35] N.K. Jaiswal, "Priority Queues," New York: Academic, 1968.
- [36] A.G. Fraser and S.P.Morgan, "Queueing and framing disciplines for a mixture of data traffic types," AT&T Bell Lab. J., Vol. 63, no.6, pp. 1061-1087, 1984.
- [37] J.N. Diagle, "Message Delays at Packet-Switching Nodes Serving Multiple Classes," IEEE Trans. Networking, vol. 38, April 1990 pp. 447-455.

- [38] D. Mitra and K.G. Ramakrishnan, "A case study of Multiservice, Multipriority Traffic Engineering Design for Data Networks," Proceedings, IEEE Globecom'99, Rio de Janeiro, Dec.1999, pp.1077-1083.
- [39] J.N. Daigle, "Message delays with prioritized HOLP and round-robin packet servicing," IEEE Trans. Commun. vol. COM-35, pp.609-619, June 1987.
- [40] J. Banks, J.S. Carson, B.L. Nelson and D.M. Nicol, "Discrete-Event System Simulation," Prentice-Hall inc. 3rd ed. 1999-2001.
- [41] J.M. Pitts, "Cell-Rate simulation modeling of Asynchronous Transfer Mode Telecommunications networks," PHD Thesis, University of London, 1993.