Congestion Control Mechanisms within MPLS Networks

By

Felicia Marie Holness

Submitted for the Degree of Doctor of Philosophy

Supervised by Dr Chris Phillips
Department of Electronic Engineering
Queen Mary and Westfield College
University of London
United Kingdom

September 2000

To God my Saviour,

My mother and sisters Adele and Nicolette, for their continual support and encouragement

Acknowledgements

I would like to express my sincere thanks to my supervisor Dr Chris Phillips for his guidance and encouragement, Professor Laurie Cuthbert for his support and belief in me, and Dr John Schormans for his many comments and technical observations. I would also like to acknowledge the financial funding provided partly by EPSRC and British Telecommunciations (BT) and to Terry Hodgkinson at BT Labs for his technical advice.

Without them this research would not have been possible.

It is impossible for me to name each and every person who contributed to this work, but I would like to thank my fellow researchers namely Babul Miah, Eliane Bodanese and Husam Awadalla for their encouragement and support, and for Ho, Stevie, Rob and Veselin and everyone else who has participated in making my time at QMW enjoyable.

Abstract

Considerable interest has arisen in congestion control through traffic engineering from the knowledge that although sensible provisioning of the network infrastructure is needed, together with sufficient underlying capacity, these are not sufficient to deliver the Quality of Service (QoS) required for new applications. This is due to dynamic variations in load. In operational Internet Protocol (IP) networks, it has been difficult to incorporate effective traffic engineering due to the limited capabilities of the IP technology. In principle, Multiprotocol Label Switching (MPLS), which is a connection-oriented label swapping technology, offers new possibilities in addressing the limitations by allowing the operator to use sophisticated traffic control mechanisms.

Due to the reasons mentioned there is a strong requirement to improve network performance and efficiency. For example, during network transient periods, the efficiency of resource allocation could be increased by routing traffic away from congested resources to relatively under-utilised links. Some means of restoring the Label Switched Paths (LSPs) to their original routes once the transient congestion has subsided is also desirable.

This thesis proposes a novel scheme to dynamically manage traffic flows through the network by re-balancing streams during periods of congestion. It proposes management-based algorithms that will allow label switched routers (LSRs) within the network to utilise mechanisms within MPLS to indicate when flows are starting to experience frame/packet loss and then to react accordingly. Based upon knowledge of the customers' Service Level Agreement (SLAs), together with instantaneous flow information, the label edge routers (LERs) can then instigate changes to the LSP route and circumvent congestion that would hitherto violate the customer contracts.

The scheme has two principle components called FATE (Fast Acting Traffic Engineering) and FATE+. They are a novel extension to the existing CR-LDP (Constraint-based Routed Label Distribution Protocol) signalling protocol and they provide additional functionality that governs the behaviour of an ingress LER and core LSRs when congestion arises. In addition to this scheme, flexible management algorithms can be incorporated into the ingress LER to enable it to respond appropriately to the signalled congestion information, the customer SLAs and the requirements of the network operator. Together, these allow LSRs and LERs to utilise mechanisms within MPLS to react on information received from the network, for example, regarding flows that may be about to experience significant packet loss and then take appropriate remedial action. For example, during transient periods, the efficiency of resource allocation could be increased by routing traffic away from congested resources to relatively under-utilised links.

Based upon knowledge of the customers’ SLAs, together with this instantaneous flow information, the LERs and LSRs can then instigate changes to the LSP route to circumvent congestion that would hitherto violate the customer contracts. Simulation data is provided that shows the efficiency of resource allocation is improved by routing traffic away from congested resources to relatively under-utilised links during transient traffic surges.

In addition, various refinements are discussed to improve the performance of the scheme and the ease with which the scheme can be integrated into an existing MPLS infrastructure. Finally, topics for future research are identified.

Table of Contents

List of Figures

List of Tables

Glossary

AAL	Asynchronous Transfer Mode Adaptation Layer
ATM	Asynchronous Transfer Mode
ATMARP	Asynchronous Transfer Mode Address Resolution Protocol
BGP	Border Gateway Protocol
BUS	Broadcast and Unknown Server
CAC	Connection Admission Control
CBQ	Class Based Queueing
CBS	Committed Burst Size
CDR	Committed Data Rate
CIN	Congestion Indication Notification
CLIP	Classical IP over ATM
CLP	Cell Loss Probability
CPU	Central Processing Unit
CR-LDP	Constraint-based Routed Label Distribution Protocol
CR-LSP	Constraint-based Routed Label Switch Path
Diffserv	Differentiated Services Model
DoD	Department of Defence
DoS	Depth of Search
DWDM	Dense Wavelength Division Multiplexing
ER-LSP	Explicit Routed Label Switched Path
F	Forward
FATE	Fast Acting Traffic Engineering
FEC	Forward Equivalence Class
FIFO	First –In First-Out
FR	Frame Relay
FTN	Forward Equivalence Class – to- Next Hop Label Forwarding Entry
FTP	File Transfer Protocol

GII	Global Information Infrastructure
GSMP	Generic Switch Management Protocol
HOL	Head of Line
ID	Identifier
IETF	Internet Engineering Task Force
ILM	Incoming Label Map
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
IPX	Internetwork Packet Exchange
ISA	Integrated Service Architecture
ITU	International Telecommunication Union
LIB	Label Information Base
LIS	Logical Internet Protocol Subnet
LAN	Local Area Network
LANE	Local Area Network Emulation
LE-ARP	LAN Emulation Address Resolution Protocol
LEC	LAN Emulation Client
LECS	LAN Emulation Configuration Server
LDP	Label Distribution Protocol
LES	LAN Emulation Server
LE Service	LAN Emulation Service
LSP	Label Switch Path
LSR	Label Switching Router
MPOA	Multi-Protocol Over ATM
MPC	MPOA Client
MPS	MPOA Server
NAK	Negative Acknowledgement
NBMA	Non-Broadcast Multiple-Access

NHLFE	Next Hop Label Forwarding Entry
NHRP	Next Hop Resolution Protocol
NHS	Next Hop Server
MAC	Medium Access Control
OSI	Open System Interconnection
OSPF	Open Shortest Path First
OPNET	Optimised Network Engineering Tools
PDU	Protocol Data Unit
PPP	Point-to-Point Protocol
PSTN	Public Switched Telephone Network
QoSR	QoS Routing
QRMM	QoS Resource Management Method
RNG	Random Number Generator
RED	Random Early Detection
RSVP	ReSerVation Protocol
S	Stack
SDH	Synchronous Digital Hierarchy
SLA	Service Level Agreement
SONET	Synchronous Optical Network
SPE	SONET Payload Envelope
TOS	Type of Service
TLV	Type Length Value
TTL	Time-To-Live
U	Unknown
UBR	Unspecified Bit Rate
UDP	User Datagram Protocol
VC	Virtual Circuit
VCI	Virtual Circuit Identifier
VPI	Virtual Path Identifier

VD	Virtual Destination
VN	Virtual Network
VS	Virtual Source
WAN	Wide Area Network
WFQ	Weighted Fair Queueing

List of Mathematical Figures

Ton	mean duration in the on state
Toff	mean duration in the state off
Average rate	The average rate at which the source transmits whilst in the on state
Peak	Peak transmitting rate of the source
r	utilisation of the buffer
c	buffer service capacity
n	number of sources
$\overline{X}(n)$	mean of n samples
X_i	i^{th} sample
s	variance
$p(i)$	probability that there are i packets in the buffer on arrival
$n(i)$	number of times there are i packets in the buffer on arrival
x	number of packets received over the simulation period
R	On rate
a	the probability of generating another excess-rate arrival in the ON state
s	the probability of being silent for another time slot
X	buffer capacity of the queue
P(X)	the excess-rate cell loss probability
h	decay rate
on	mean on duration of the source
off	mean off duration of the source
h	number of packets / second when the source is in the on state
B	ErlangB loss probability
C	service capacity of the buffer
N	number of on off sources
R_{on}	mean rate in the ON state

$Roff$	mean rate in the OFF state
D	probability of a call being delayed
A_P	offered load
N_0	number of servers

Chapter 1: Introduction

1.1 Overview

In this thesis a new congestion control scheme based on the Multiprotocol Label Switching (MPLS) architecture is presented for operation within an autonomous MPLS domain. The approach devised provides a means of evolving existing MPLS network concepts whilst fully supporting coexistence with them in their current form.

The new scheme offers:

- Flexible congestion detection mechanisms for operation within a MPLS domain incorporating features to prevent unstable operation;
- Mechanisms for the selective and rapid re-distribution of traffic along alternative quality of service streams or physical paths once a predetermined congestion trigger condition has arisen;
- Compatibility with the existing label distribution protocol / constraint-based routed label distribution protocol (LDP/CR-LDP).

The transformation of the Internet into an important and ubiquitous commercial infrastructure has not only created rapidly rising bandwidth demands but also significantly changed consumer expectations in terms of performance, security and services. Consequentially as network providers attempt to encourage business and leisure applications on to the Internet, there has been a requirement for them to develop an improved IP network infrastructure in terms of reliability and performance [LI2000].

To satisfy customer demand for quality of service (QoS), network providers need to be able to offer new differentiated services that allow users the ability to flexibly choose the level of service that matches their particular needs [BOR98]. This could range from “toll quality” voice through to traditional “best effort” transport. At the same time it is necessary to control costs and improve profitability, particularly as network access and backbone resources are more tightly provisioned to meet committed levels of service. In response, network providers need to not only evolve their networks to higher speeds, but also need to plan for the introduction of increasingly sophisticated services to address the varied requirements of different customers [AWD99]. In addition, network providers would like to maximise the sharing of the costly backbone infrastructure in a manner that enables them to control usage of network resources in accordance with service pricing and revenue potential.

The current Internet architecture employs control mechanisms such as Transmission Control Protocol (TCP), that were originally implemented to provide flow-control and prevent congestion collapse for non-real-time data transport services [SIY97]. It is these mechanisms that are still in use today.

However:

TCP is an end user mechanism. For a commercial network, relying on end users for congestion control is not desirable. The wilful or inadvertent selfish behaviour of particular user traffic flows cannot be allowed to negatively impact on the service offered to other "conforming" users. Typically, some degree of policing is required to enforce this.

- Although the Internet Protocol (IP) header provides Type of Service classification, the network infrastructure has not fully exploited this for differentiating traffic until recently. See Section 2.2.1.3.
- Over provisioning is not a cost-effective solution and cannot guarantee against congestion at all times due to the stochastic nature of the traffic.
- The differentiated services paradigm provides a mechanism for prioritising traffic, adjusting scheduling and congestion control, and operates on a per class (QoS), but it lacks the ability to respond to transient traffic surges.

Many service providers have responded to the need for congestion control by employing traffic engineering in their networks [SEM2000b]. The increase in interest in this topic has arisen from the knowledge that although sensible provisioning of the network infrastructure is required together with adequate underlying capacity, this is not sufficient to deliver the desired QoS required, due to dynamic variations in load. Traffic Engineering is a powerful concept that can be used by network operators' to balance the traffic load on the various links and routers in the networks so that none are over-utilised or under-utilised [SEM2000a][AWD99]. In operational IP networks, it has been difficult to incorporate effective traffic engineering due to the limited capabilities of the IP technology [SIY97]. Multiprotocol Label Switching (MPLS), a connection-oriented label swapping technology, offers new possibilities in addressing the limitations of IP systems where traffic engineering is considered, by allowing the implementation of sophisticated traffic control mechanisms.

However, as yet the traffic engineering capabilities offered by MPLS have not been fully exploited. For example, once label switched paths (LSPs) have been provisioned through the network operators' network, there are currently no management facilities for dynamic re-optimisation of the traffic flows. The service level agreements (SLAs) between the network operator and the customer are agreed in advance of the commencement of traffic flows, and these are mapped to particular paths throughout the provider's domain typically for the duration of the contract[1]. During transient periods, the efficiency of resource allocation could be increased by routing traffic away from congested resources to relatively under-utilised links. Some means of restoring the LSPs to their original routes once the transient congestion has subsided is also desirable.

1.2 Objectives of the Thesis

The above description serves to highlight the importance of developing a suitable congestion control scheme for MPLS. The aim of the research reported in this thesis has been to:

- Develop a congestion control scheme to detect and alleviate congestion within multi-service, wide area networks;
- To implement the scheme in such a way as to react in a selectably rapid manner to situations of congestion, to minimise their effects;
- To ensure that the proposed scheme interworks with the Internet Engineering Task Force (IETF) "standards" being developed for MPLS networks. This provides a straightforward migration path for network operators.

In order to meet the above objectives, the initial phase of the research concentrated on the need for dynamic re-distribution of traffic flows within MPLS networks using under-utilised links and Label Switched Routers (LSRs). A description of MPLS in general and the related issues is presented in the thesis. In particular, the author stresses the importance of maintaining SLAs between the operator and the customer for the duration of the LSP. The aspect of customer perception of service-guarantees, that have been identified as significant, and which led to the development of the congestion control scheme, is the issue of quality of service (QoS). QoS is a term often used to describe a wide range of meanings such as bandwidth and latency guarantees, and loss probability [CIS97][FER98]. In this thesis the author uses QoS to mean the agreement between the customer and operator in terms of the quantifiable parameters; loss probability and the mean available bitrate for the customers' traffic flows.

A description of the new scheme is then presented in the thesis. The proposed scheme is evaluated using simulations, which provide necessary experimental information to permit a full and comprehensive assessment of the scheme. A discussion and analysis is given.

[1] Time-scales for duration of these SLA contracts are typically weeks, months or even years.

1.3 Summary of Contributions from the Thesis

Starting from the premise that future wide area networks will be multi-service in nature, the author has undertaken a survey of existing traffic engineering mechanisms that are being considered to support them. In particular, given its strong commercial backing, MPLS has been examined. The author determined that although it is a relatively flexible protocol, in its current form, MPLS provides no means of implementing fast-acting congestion control. Similar independently derived conclusions have been reached by other researchers [ASH2000b].

Given this shortcoming, the author focused on the design of a congestion control scheme that would build upon existing MPLS concepts but would provide a means of quickly resolving the effects of transient congestion. This scheme goes far beyond "traditional" provisioning and is likely to provide a significant market advantage to equipment manufacturers and network operators who implement the scheme, or its derivatives.

The proposed scheme, entitled Fast Acting Traffic Engineering (FATE), operates over tightly constrained MPLS domains. A variant of the scheme, known as FATE+, provides a similar congestion control mechanism for use over loosely coupled "abstract nodes" within an MPLS domain.

Both FATE and FATE+ have been implemented as OPNET models and simulated within an MPLS network to ascertain their correct functioning and their satisfactory performance. Several refinements were made to the scheme during the course of the research. These included:

- The use of a loss event counter to trigger the FATE mechanism rather than a fixed-time sliding window;
- The introduction of hysterisis thresholds improved the stability of the scheme limiting the generation of the signalling messages;
- Congestion notifications can now be aggregated within the same signalling message reducing avalanching effects along a congested path.

Finally the author assesses the efficacy of the scheme in a dispassionate manner and considers additional enhancements. Although, there was limited scope for a comparative evaluation of the scheme, a recent proposal by researchers at Nortel / AT&T has been examined. FATE/FATE+ is seen to provide an original fast-acting traffic engineering mechanism that operates over a much shorter time frame than their approach, particularly as their scheme is intended for provisioning.

1.4 Structure of the Thesis

A list of acronyms and terms is provided in the glossary. In addition, figures, tables and mathematical symbols used throughout the thesis are also listed at the beginning of the thesis. The main body of the report consists of seven chapters including the introduction, the discussion and the conclusion. Each chapter begins with a brief description of its scope and ends with a brief summary of the outcome. Details of each chapter are provided below.

This thesis describes in detail a dynamic provisioning mechanism capable of detecting and alleviating congestion within MPLS networks. The focus of the contribution is on the re-distribution of traffic flows along under-utilised LSPs. This introductory chapter has served to summarise the contribution of this thesis in the context of controlling congestion and utilising links within a MPLS network.

Chapter 2 serves as a brief introduction to the QoS capable mechanism currently deployed within the Internet. Then in chapter 3 the MPLS architecture is introduced along with its various components and mechanisms. The congestion control scheme called FATE/FATE+, developed by the author, is described in chapter 4. The signalling messages, mechanisms, procedures and data structures are described.

The simulation models created by the author to investigate the performance of the architecture presented in chapter 4 are described in chapter 5. This chapter details the modelling of the various network components, the assumptions made for the simulation study and the modelling of the congestion control mechanism. The verification and validation of the models are also discussed. The results of the simulation are studied and analysed.

Chapter 6 discusses the characteristics of the scheme and assesses how well is meets its objectives of alleviating the effects of transient congestion. A comparison with the other approaches tackling provisioning within MPLS is made.

Finally Chapter 7 concludes with the merits of this approach highlighted and its limitations identified. Areas for further work are also given.

The authors Publications and References and Appendices are provided in the remaining chapters.

Chapter 2: Traffic Engineering within the Internet

2.1 Overview

IP networks offer the scalability and flexibility for rapid deployment of IP services [DeM]. However, as a result of the rapid growth of the Internet and the increasing demand for services, carriers require a reliable and dependable network offering consistent and predictable network performance [NOR99] [CIS97]. Traffic Engineering is needed to achieve this “mission critical networking” [LI2000] [SHA99].

Traffic Engineering is a powerful concept that can be used by network operators’ to balance the traffic load on the various links, routers, and switches in the networks so that none are over-utilised or under-utilised [SEM2000a][AWD99]. Traffic Engineering targets the ability to efficiently map traffic onto an existing network topology in such a way as to optimise the utilisation of network resources [FEL2000] [SEM2000b]]. This allows network operators the ability to better exploit bandwidth resources across the network. Traffic Engineering should be viewed as augmenting the routing infrastructure by providing additional information that enables traffic to be forwarded along alternate paths across the network.

In order to achieve this, there is a need to consider adaptability to changes in the network, such as topology, traffic loads, failure, etc., as well as the need to adhere to administrator-defined policies. In this way, the traffic engineering function realises the performance optimisation of operational networks, and facilitates efficient and reliable network operations [LI2000], whilst going somewhere towards offering customers the QoS performance they have negotiated for.

In the current evolving networking environment, increasingly more traffic is transported over networks where the IP protocol suite plays a dominant role [ROB97a]. Although the IP networks offer flexibility and scalability, these existing IP networks need to be enhanced in areas of availability, dependability and Quality of Service (QoS), in order to provide a mission critical networking environment [ROS98].

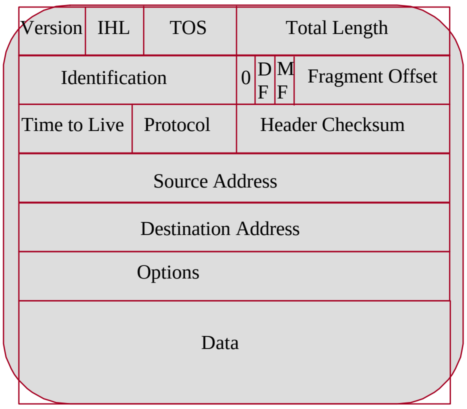
Chapter two provides a background into networks supporting the Internet Protocol, focusing only on components significant in deploying traffic engineering mechanisms.

2.2 Internet Protocol

The US Department of Defence (DoD), developed in the late 1960s [SIY97] a suite of protocols to allow computers on different networks designed by different vendors to share resources across a common inter-network. This computer-communications network had no centralised control, and it also assumed that no one link in the network was reliable. Each message was segmented, packetised, and marked with the address of the sender and receiver. The packets were then forwarded across this network of interconnected *routers*[2] towards their destination using the Internet Protocol (IP).

IP is a connectionless[3] *datagram* based network layer protocol that performs addressing, routing, and control functions for transmitting and receiving packets. As packets are received by the router, IP addressing information, such as the destination address, is used to determine the best "next hop" the packet should take enroute to its final destination.

IP datagrams can have a maximum length of 65535 bytes making it well suited to the transport of non real-time data. Datagrams consist of at least a 20 byte header including the following fields: a version number, the header length, the type of service, the total length of the datagram, the datagram's identification number, fragmentation control information, the “time-to-live” duration, the protocol format of the data field, the source and destination addresses, and possible option field(s) as illustrated in Figure 2-1.



Version - indicates the format of the IP header,
the current version number is 4.
IHL - the Internet Header Length field is the length
of the header in 32-bit words.
TOS - Type of Service field informs the network of the
QoS required, such as precedence,
throughput, and reliability.
Total Length - the length of the IP header and data in bytes.
Identification - a unique value to identify the datagram
Fragmentation Flags - if DF = 1, means the datagram
should not be fragmented.
if DF = 0, indicates the router or host may fragment the
IP datagram. if MF = 1, indicates to the receiver
there are more fragments to come.
if MF = 0, this is the last fragment.
Fragment Offset - the start of the data field relative to the
original unfragmented data.
Time to Live - maximum time in seconds that an IP datagram
can live on the network.
Protocol - indicates the upper-layer protocol that is to receive the IP data.
Header Checksum - used for the IP header only.
Source Address , Destination Address - 32- bit IP addresses of the source and
destination addresses, included in
every datagram.
Options - facilities for the security of a datagram,source routing and
timestamp information.

Figure 2-1 IP Datagram Format

[2] A router is considered to be a device capable of forwarding information based on a network layer header.

[3] IP is classified as a connectionless service because it does not set up a connection with its intended destination before it transmits data. End-to-end connectivity is actually provided by the transport layer Transmission Control Protocol (TCP).

Traditionally IP networks have all user packets competing equally for network resources with a “best effort” service paradigm irrespective of what application they belong to [MON99]. Until recently this model has been sufficient. However, as a result of new applications such as real-time audio and video, there has been an increase in demand for transport services capable of meeting particular QoS requirements [KAU99]. This increase has been paralleled by a rise in user expectations [VAU97][LUO]. Unfortunately, due to limited and inefficient utilisation of network resources (bandwidth, buffer restrictions), sporadic and sustained congestion is a common phenomenon in current IP networks [KUM98]. This sporadic behaviour does not encourage the exploitation of IP networks as a transport medium for real-time and mission critical applications [LI2000]. Enhancing the IP network infrastructure to support these new applications is one of the key challenges of network providers’ [BAK][KEM97].

To introduce QoS capabilities within an IP network there is a need for mechanisms that both implement QoS and signal QoS requirements [KUM98]. This is a radical change from the way the Internet has operated over the past two decades [BAK][SHA98a]. Already there have been many proposed approaches to providing QoS support within IP networks. A number of these *Integrated Architecture* schemes, described in Section 2.2.1, have arisen from the efforts of the International Telecommunications Union (ITU) [ITU] and Internet Engineering Task Force (IETF) [IETF] to provide an overall infrastructure that caters for these differing services.

In this thesis, Section 2.2.1 provides a more detailed examination of a number of mechanisms that can be employed by these *Integrated Architectures*. This section is further subdivided into different sections detailing how QoS is incorporated at various layers of the Open System Interconnect (OSI) model.

2.2.1 Models for Multi-Service Networks

A number of integrated service models have been proposed, perhaps the three that are best known are the ITU’s (International Telecommunications Union) Global Information Infrastructure (GII) [FUM98], The IETF’s (Internet Engineering Task force) Integrated Service Architecture (ISA) and the IETF’s Differentiated Services Model (Diffserv). These three service architectures reflect the views of the future that their initiating organisations hold [PHI][CRO96].

2.2.1.1 Integrated Service Architecture

The Internet Engineering Task Force (IETF)'s Integrated Services work group was formed with a charter of introducing QoS support into IP by defining two service classes: *guaranteed*, for delay sensitive applications, *controlled-load*, for real-time tolerant applications requiring reliability but no fixed delay, in addition to *best effort* for applications like ftp and email. [RFC1633][RFC2210] describes the mechanisms employed within the integrated services paradigm.

The components of integrated services are an admission control algorithm, packet scheduler, classifier and a resource reservation protocol. The admission control algorithm will determine whether a request for resources can be granted, ReSerVation Protocol (RSVP) [RFC2205][RFC2205][MET99a][[MAN97] is used as an IP signalling protocol for reserving resources in LANs. It consists of PATH messages from source to destination followed by the recipient sending reservation requests to routers along the path back to the source. The classifier determines what classes the packets should be placed in based on the information in the header, such as IP and port source and destination addresses, the scheduler determines the order in which the packets should be serviced by placing them into priority queues.

The IETF's ISA was the first attempt to develop a multi-service architecture based on logical extensions of the Internet's *best effort* services. The two main proposed additional services are a *guaranteed worst case bandwidth and delay* service, and a *controlled load* service that does its best to deliver the same Quality of Service (QoS) that an uncongested packet network would deliver. It achieves this by reserving bandwidth and buffer resources through routers using a connection admission control system as illustrated in Figure 2-2.

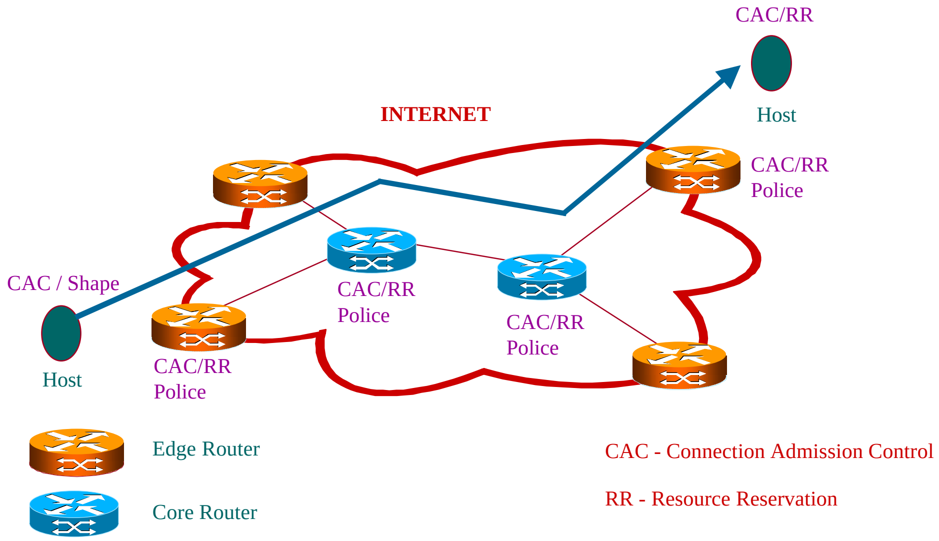


Figure 2-2 The Integrated Service Architecture

2.2.1.2 Global Information Infrastructure

The ITU's response has been the development of the Global Information Infrastructure (GII) [LU98]. This aims to introduce a set of end-to-end QoS services ranging from *best effort*, *available bit rate, deterministic bit rate* and so forth, based on an architecture with ATM in the core and optionally IP in the final access to the desktop as shown in Figure 2-3.

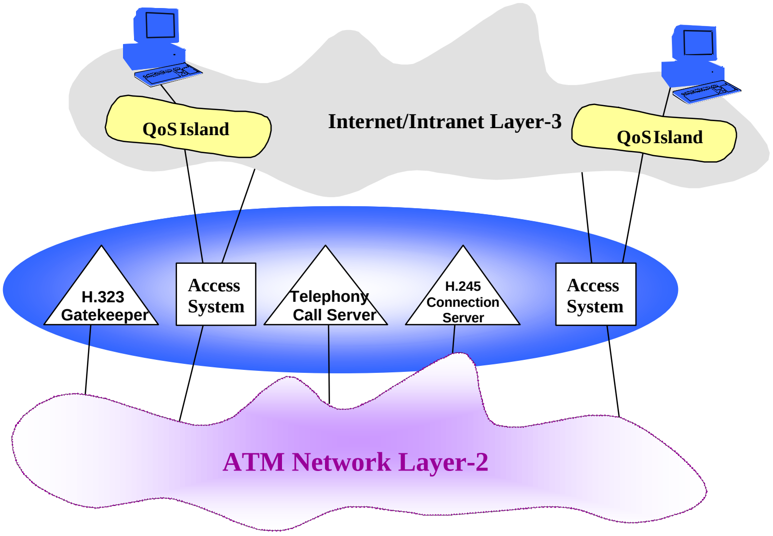


Figure 2-3 Enabling Technologies for the GII Multimedia Architecture

2.2.1.3 Differentiated Services Architecture

The IETF have recently formed a new working group aimed at developing the Differentiated Services Architecture [DIFF], shown in Figure 2-4, a more pragmatic architecture aimed at limited service differentiation, which explicitly avoids modifications to the host protocol stack. Additional services being proposed include *assured service* and *expedited forwarding* (or "premium service") [NIC2000][MER000][BLA][XIA97]. Moreover, the flow classification is simple (i.e. an 8 bit definition) and Connection Admission Control (CAC) and resource-reservation are confined to domain boundaries. This approach assumes that best-effort traffic will remain the dominant loading on the Internet and proposes an evolutionary approach with incremental enhancements. However, resource allocation mechanisms (e.g. Bandwidth Broker) have not yet been addressed.

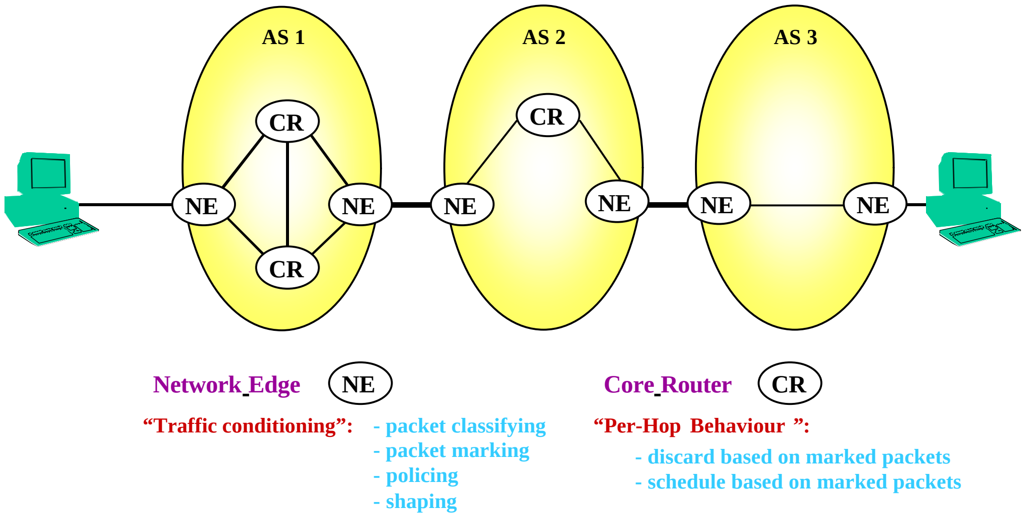


Figure 2-4 The Differentiated Services Architecture

The IETF's Differentiated Services work group has concentrated on offering services other than best effort; it does this by using a field in the IP header called the Differentiated Service (DS) field [RFC2474]. This architecture currently offers three levels of service [RFC2638]; *Premium* for real-time applications; *Assured* for applications requiring reliability but no delay constraints guarantees; *Best Effort* for delay tolerant applications.

The mechanisms of this architecture are classification, marking, policing and shaping of flows. A customer has to indicate what level of service they require by marking the DS field of the packet and will be charged accordingly; the customer will have an agreement with its service provider about the service classes supported, and the amount of traffic allowed in each class. The Premium Service will be more expensive than the Assured Service with Best Effort being the cheapest [RFC2836].

Differentiated Services buffers packets and services them at a rate that depends on the value assigned in the type of service field, i.e., no resources are reserved before packets are sent into the network.

Having now considered a number of integrated service architectures, the following sections describe various QoS mechanisms that can be used to support them. The sections are further subdivided in accordance with the layers of the OSI reference model they operate at.

2.2.2 QoS Support at the IP Network Layer

The following section describes how the IP protocol has implemented QoS mechanisms at the network layer. It starts in Section 2.2.2.1 by addressing schemes that discriminate between different classes of service. These have the benefit that they can be applied to large-scale networks, as they don't require any form of per-user information processing or state storage. Conversely, Section 2.2.2.2 provides a summary of schemes that support QoS on a per-flow basis. Although these provide greater control they are hindered by their lack of scalability.

2.2.2.1 Per Class Routing

Per Class Routing mechanisms provide multi-level service support based on generic service classifications rather than information pertaining to customer flows. The benefit of this methodology is that it can be used in large-scale networks unlike per flow routing [ROB97b]. Routers supporting per-class routing do not need to maintain state information related to individual flows. Instead these schemes use a simple form of service type marking in conjunction with an appropriate form of queue management and scheduling.

Using the IP *Type of Service* field in the IP header provides a method of marking and distinguishing between different service classes as packets traverse the network [RFC1349]. The *Type of Service* field is composed of 3 precedence bits and 5 ToS bits in IPv4 packet headers. The ToS bits are generally regarded as describing the type or class of service required. Use of the precedence bits is undefined, but the prevalent view is that they should be used to indicated importance (e.g. drop priority) [FER98], and that they should be under the control of the network, rather than the end system, as a tool for providing different quality of service to different users. However, the role of both fields is currently under discussion [XIA99][NOR2000], particularly in the light of the ongoing 'differential services' activity.

The format and positioning of the TOS and IP precedence fields in the IP packet header are shown in Figure 2-5.

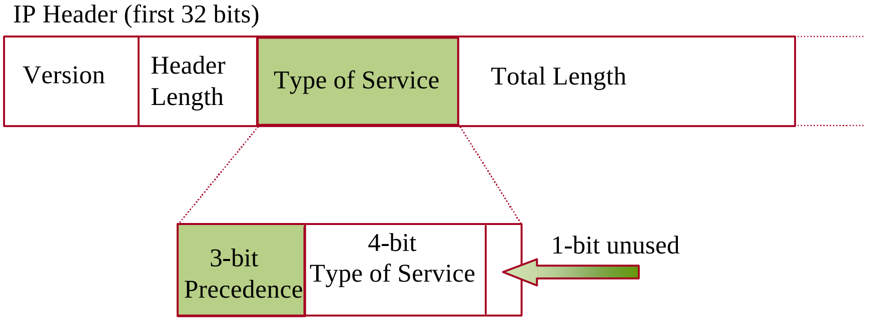


Figure 2-5 The IP TOS field and IP precedence

With per class routing, IP packets are marked with the desired QoS identifier when the packet enters the network. On all subsequent interior routers, the required action is to look up the ToS field and apply the associated QoS action to the packet. This approach can scale quickly and easily, given that the range of QoS actions is a fixed number and does not grow with traffic volume, whereas flow identification is a computational task related to traffic volume.

Another component of per class routing is queue management [KLE96][KLE76]. When used in conjunction with a suitable buffer scheduling discipline, by examination of the ToS bits, a Queue manager can determine the appropriate buffer an incoming packet should be stored in.

Queue Management can be supported in terms of selective discarding and queueing strategies. For example, Random-Early Detection (RED)[JAC98][FLO93] is an example of a discard mechanism that monitors average queue length and compares this with two threshold values to control the rate of random packet discards.

The following sections describe various queuing mechanism that can be used within switches and routers to prioritise the forwarding of packets [FLO95][JON98].

2.2.2.1.1 FIFO Queueing

FIFO (First In, First Out) queuing is considered to be the standard method for store-and-forward handling of traffic from an incoming interface to an outgoing interface. As packets enter the input interface, they are placed into the appropriate output interface queue in the order in which they are received – thus the name *first-in-first-out*.

When a network operates in a mode with a sufficient level of transmission capacity and adequate levels of switching capability, queuing is necessary only to ensure that short-term highly transient traffic bursts do not cause packet discard. In this environment, FIFO queuing is adequate because, as long as the queue depth remains sufficiently short, the average packet-queuing delay is an insignificant fraction of the end-to-end packet transmission time.

As the load on the network increases, the transient bursts cause significant queuing delay (significant in terms of the growth of the queue over longer periods of time), and when the queue is fully populated, all subsequent packets are discarded. When the network operates in this mode for extended periods, the offered service quality inevitably degenerates.

2.2.2.1.2 Priority Queuing Mechanisms

Priority Queuing has many variants although they can all be regarded as schemes that use a scheduler to service certain types of traffic in preference to others. Multiple output queues are used to buffer traffic according to class. The precedence with which each of the queues will be serviced in a cycle is determined by the portion of scheduling time assigned to that queue. This can be governed by a scheduling template.

This servicing algorithm provides “fairness” by prioritising queuing services for certain types of traffic. In an extreme form, lower priority buffers receive no service whilst packets in the higher priority buffers are continually served. This can lead to buffer starvation [PAR94][JAC98].

An example of simple priority queueing would be to sort traffic (i.e. via the ToS bits) into two or possibly more queues, which represent different priorities. Scheduling is done in strict priority order, i.e. traffic is only served from a lower priority queue when the higher priority queues are empty. A more refined mechanism is Weighted Fair Queueing (WFQ) where traffic is classified into a number of flows that are placed into virtual queues. These are then serviced using a scheduling algorithm that takes account of packet sizes, arrival times, and current backlogs to ensure that each flow has an appropriate level of performance. An alternative use is in controlling the proportion of scheduling resources given to different service classes; this gives each class a guaranteed share of resources and avoids the potential for 'starvation' that is a feature of simple priority queueing. Finally, Class Based Queueing (CBQ) [JON98] [FER98] is a bandwidth-management technique based on multi-queue scheduling, in which bandwidth allocations across a link may be assigned to different traffic categories according to a hierarchical structure. A broad range of traffic classifications are possible, including by address, protocol, port number or ToS field.

2.2.2.2 Per Flow Routing

Per flow routing eliminates the need for a router to perform an initial classification of a packet into one of potentially many thousands of active flows and then apply a QoS rule that applies to that form of flow, the IP precedence bits can be used to reduce the scope of the task considerably.

Per flow routing provides a method of extracting information from the IP packet header and associating it with previous packets. The intended result is to identify the end-to-end application stream of which the packet is a member. Once a packet can be assigned to flow, the packet can be forwarded with an associated class of service that may be defined on a per-flow basis.

2.2.2.2.1 Differentiation per-Flow

The purpose of per-flow differentiation is to be able to provide similar QoS characteristics to a similar IP end-to-end session, e.g., allowing real-time flows to be forwarded with QoS parameters different from non-real-time flows.

This is similar in concept to assigning QoS characteristics to Virtual Circuits (VCs) within a frame relay or ATM network. However, given the number of flows that may be active in the core of a network, this approach is widely considered to be impractical as far as scalability is concerned [JAC98]. Maintaining state and manipulating flow information for this large number of flows would require considerable computational overhead. This is primarily the approach that the ReSerVation Protocol (RSVP) takes, and as a result may not scale sufficiently well in a large network [NOR2000]. Thus a simpler and more scalable approach may be necessary for larger networks.

2.2.2.2.2 QoS Routing

QoS Routing (QoSR) [RFC2386][RFC2676] is an enhanced form of differentiation per flow. It determines the path for a flow based on knowledge of the resource availability in the network, as well as the QoS requirements of the flow. QoSR extends the current routing paradigm in three basic ways [SHA98b]. Firstly, it supports traffic using integrated-services class of service so multiple paths between node pairs will have to be calculated. Some of these new classes of service will require the distribution of additional routing metrics, e.g., delay, and available bandwidth. If any of these metrics change frequently, routing updates can become more frequent thereby consuming network bandwidth and router CPU cycles. Secondly, opportunistic routing will shift traffic from one path to another as soon as a “better” path is found. The traffic will be shifted even if the existing path can meet the service requirements of the existing traffic. If routing calculation is tied to frequently changing consumable resources (e.g. available bandwidth) this change will happen more often and can introduce routing oscillations as traffic shifts back and forth between alternative paths. Furthermore, frequently changing routes can increase the variation in the delay and jitter experienced by the end users. Thirdly, the optimal routing algorithms existing in routing protocols do not support alternate routing. If the best existing path cannot admit a new flow, the associated traffic cannot be forwarded even if an adequate alternate path exists.

2.2.3 QoS Support for IP Services at the Data-Link Layer

Originally, the idea of running IP over ATM was necessary in order to gradually phase ATM into the Internet architecture for an eventual takeover of IP [JOH97][SAL97]. However, the possibility of running IP over ATM is already being used as ATM provides a fast switching technology and running IP over ATM can resolve the network layer router bottleneck problem [TES99].

A problem with this type of integration stems from the fact that IP is a connectionless protocol while ATM is a connection-oriented protocol. This mismatch has led to complexity, inefficiency, and duplication of functionality in attempting to integrate IP with ATM [LUK97][KAT97][WHI98]. One obvious example of inefficiency lies in the fact that TCP is often the transport protocol of choice for IP. Since the ATM layer guarantees that cells are always delivered in order, the fact that the TCP transport layer handles the reordering of out-of-order packets is clearly one case of duplication of functionality.

Most of the IP over ATM approaches proposed in the literature and standards bodies to-date treat the physical network as a large opaque cloud. The real topology of the underlying ATM network is thus obscured from the network layer, allowing ATM to be used as a means of bandwidth i.e., transporting frames between two points. It also decouples the functionality of IP from ATM which can be seen as either an advantage or disadvantage.

The following sections focus on protocols that have been proposed by the IETF and the ATM Forum. The protocols and functions discussed below introduce services such as QoS to the basic internetworking functions of TCP/IP.

The classical suite of IP over ATM technologies is attributed to the functions described in [RFC1577][RFC2336] and updated in [RFC2225] includes CLIP and the NHRP [RFC2332], they are classical in the sense that they support traditional IP services and behaviours and are transparent to the applications running above them.

2.2.3.1 Classical IP over ATM

The Classical IP model (CLIP) refers to a network where hosts are organised in subnetworks sharing a common IP address prefix, where ARP is used for IP address to Medium Access Control (MAC) address resolution and where communication across subnetworks goes through routers.

The main advantage of CLIP is its compatibility with IP, enabling higher layer protocols and applications to run transparently over ATM while making use of ATM’s high bandwidth availability. Also CLIP allows easy integration of IP based services with other ATM services [DUM98][ESA95]. This model does not change IP in any way and uses ATM as an underlying technology. The main ATM characteristics that have been exploited by IP are VC switching, Unspecified Bit Rate (UBR) support that is well suited for best effort traffic, and high speed at the user terminal and router interfaces [SCH96].

The major disadvantage of CLIP is that it cannot benefit from ATM’s inherent end-to-end QoS guarantees for the following reasons:

Direct ATM connections can only be established inside a LIS (Logical Internet Protocol Subnet) but not across LIS borders;

Owing to the use of the classical IP routing model (address resolution is limited to a LIS);

IP traffic between hosts on differing LISs always flows via one or more intermediate IP routers that can only provide best effort delivery at the IP level. This results in a concatenation of ATM connections, even though it may be possible to open a direct ATM connection between two hosts, thus denying end-to-end QoS guarantee. In other words, IP packets across a LIS border hop several times through the ATM network instead of using one single hop;

All IP data flow between two hosts' share the bandwidth of a single VC. Having only one shared VC between two hosts makes it impossible for individual applications to get a QoS guarantee for their data flow;

Unlike LAN Emulation (LANE) described later in Section 2.2.3.3, CLIP does not allow the use of the legacy LAN equipment. So deployment of CLIP is only feasible where new LAN networks with ATM to the desktop are built.

2.2.3.2 Next Hop Resolution Protocol

The next protocol from the suite of classical IP/ATM solutions is the Next Hop Resolution Protocol (NHRP). NHRP extends the notion of the CLIP ATMARP protocol beyond the boundaries of the LIS [PAR97].

For NHRP operation there has to be one Next Hop Server (NHS) in every LIS as illustrated in Figure 2-6. All hosts on a LIS, register their (Non-Broadcast Multiple-Access) NBMA and internetwork layer (e.g. IP) address with their Next Hop Server (NHS) when booting.

Assume a Source wants to send an IP packet to a Target that lies outside its Logical IP Subnetwork (LIS). To resolve the NBMA address of the target, the source sends a next hop Resolution Request to its NHS. The NHS checks whether the target lies in the same LIS. If the NHS does not serve the target, the NHS forwards the request to the next NHS along the routed path. This forwarding process continues until it reaches the NHS that serves the target. This NHS can resolve the target's NBMA address and sends it back along the routed path. The intermediate NHS can store the address mapping information for the target contained in the Resolution Reply to answer subsequent Resolution Requests.

The main advantage of the NHRP is that it can solve the multiple-hop problem through NBMA networks by offering inter-LIS address resolution, thus enabling the establishment of a single-hop connection through the NBMA network. If the network is ATM, a single direct VC can be established across several LIS, bringing QoS in terms of traffic contract guarantees to the IP data flow between the VCs endpoints. Also if a direct connection can be established through the NBMA network, it will be shared by all IP traffic between the two endpoints. This means that it is not capable of providing QoS to an individual application as the resources are shared among the individual flows.

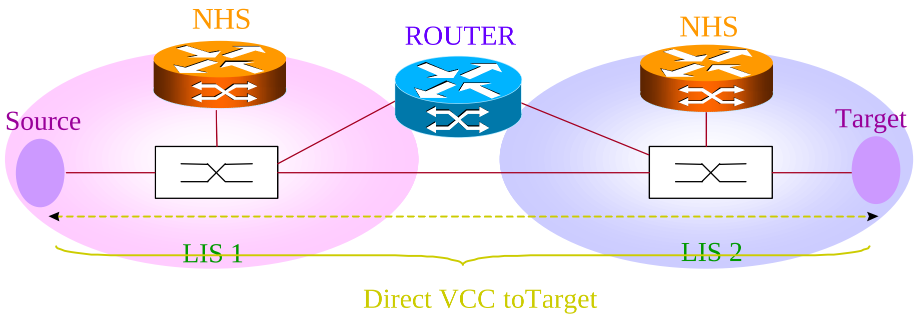


Figure 2-6 Next Hop Resolution Protocol

Using IP, LANE or CLIP means that there is no quality of service supported at the application level, as the IPv4 layer is decoupled from all features of the underlying ATM network. For QoS, some form of IP signalling interworking would be required as this would allow an application to request a level of quality of service.

2.2.3.3 LAN Emulation

Current Local Area Network (LAN) implementations are inherently different to ATM in that they offer a connectionless service where multicast and broadcast message transfers are easily accomplished. LAN emulation (LANE) [LAN21][LAN93][DRI97] provides a means of supporting these conventional LAN services over an ATM network in such a way that existing software applications[4] are able to transfer data frames between endstations as if they were attached to a conventional LAN. LANE does not replace routers or routing; it provides a complementary MAC-level service.

Each emulated LAN is composed of a set of LAN Emulation Clients (LECs) and a single LAN Emulation Service (LE Service) facility. A LEC is typically an entity within ATM-based terminal equipment or a bridge through which a set of users, identified by their MAC addresses, connect to the LE Service. The LE Service provides the LE layers with a connectionless service for the transfer of LAN messages between destinations. It acts as the shared medium for the emulated LAN in the inherently connection-oriented ATM network.

The LE Service is composed of three entities; the LAN Emulation Configuration Server (LECS), the LAN Emulation Server (LES) and the Broadcast and Unknown Server (BUS) as shown in Figure 2-7 together with a number of LEC entities. The LE Service provides a virtual shared medium connecting all the endstations, supporting connectionless services for the transfer of LAN messages between destinations in the inherently connection-oriented ATM

[4] These ‘existing applications’ are considered to operate at OSI layer 3 and above.

network. The LE Service also ensures that frame ordering is preserved when transferring data between them.

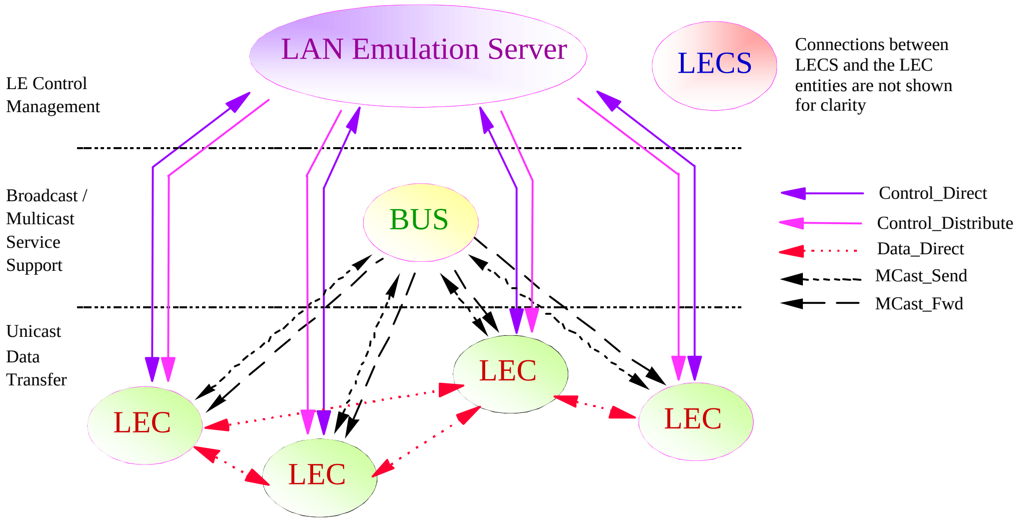


Figure 2-7 LAN Emulation Service Configuration

The LAN Emulation Configuration Server (LECS) controls the assignment of different emulated LANs to individual LECs. Based upon its own policies, configuration database and information provided by the clients, the LECS provides a LEC with the appropriate LES ATM address, upon request.

The LES implements the control co-ordination functions for the emulated LAN. It provides a facility for registering MAC addresses. A LEC wishing to join the emulated LAN must first attempt to set-up a bi-directional Control_Direct VCC to the LES using the ATM address provided by the LECS. The LES verifies whether the LEC may join the LAN. If so the LEC may then register the LAN destination(s) it represents. The joining LEC must also be prepared to accept the uni-directional point-to-multipoint Control_Distribute VCC from the LES. Messages from the LES to the LECs can be transferred across either the Control_Direct or Control_Distribute VCCs.

The LES also supports a LAN Emulation Address Resolution Protocol (LE-ARP) processing function. LECs may query the LES using the Control_Direct VCC when they wish to resolve a MAC address to an ATM address. The LES either responds directly to the LEC or forwards the query to other LECs so that they may respond. If the LES forwards LE_ARP request to LEC(s), it must forward any LE_ARP response back to at least the originating LEC via the Control_Direct or Control_Distribute VCCs.

LECs are required to connect to the BUS once they have established a connection to the LES. A LEC obtains the address of the BUS by sending an LE_ARP-REQUEST message to the LES, in order to resolve the broadcast MAC address (Hexadecimal: FFFFFFFFFFFF). Once the ATM address of the BUS is known to the LEC, the LEC must attempt to establish a bi-directional Multicast_Send VCC with the BUS. It must then be prepared to accept connection to the uni-directional point-to-multipoint Multicast_Forward VCC from the BUS.

The BUS handles data sent by a LEC to the broadcast MAC address, all multicast traffic and initial unicast frames, which are sent by a LEC before the Data_Direct TARGET_ATM_ADDRESS has been resolved. As the LE service employs the ATM Adaptation Layer 5 (AAL5), the BUS must also support serialisation, so that cells received by it from distinct origins to the same destination LEC do not become interleaved.

2.2.3.4 Multiprotocol over ATM (MPOA)

Two possible means of scaling IP to meet the existing and anticipated demand are to either replace the routed infrastructure that exists with newer, faster, cheaper routers that route every single packet at high speed, or to implement a "route once, switch many" strategy. Both IP Switching [MET99b][DAV][AHM97] and MPOA [MPO114][MPO87] adopt this latter approach and reduce the latency associated with moving Layer 3 traffic from one subnet to another, although there is considerable debate as to which is preferable[TIN96]. This process involves performing destination address discovery and resolution (routing) before forwarding over a Layer 2 (switched) environment.

MPOA operates at both layer 2 and layer 3, and is capable of providing direct layer 3 connectivity across ATM when using *shortcut flows* in order to fully exploit ATM’s QoS features. However, it is also capable of using a *default flow* using LAN Emulation when a suitable *shortcut flow* does not exist and the Layer 2 traffic is to be transferred within the same subnet or emulated LAN. For destinations outside the subnet, MPOA uses the NHRP to discover the ATM address of the intended destination.

MPOA is intended to support the efficient transfer of inter-subnet unicast data in a LANE environment. MPOA integrates LANE and NHRP to preserve the benefits of LAN Emulation, while allowing inter-subnet, internetwork layer protocol communication over ATM VCCs without requiring routers in the data path [GUA98].

MPOA provides MPOA Clients (MPCs) and MPOA Servers (MPSs) and defines the protocols that are required for MPCs and MPSs to communicate. MPCs issue queries for shortcut ATM addresses and receive replies from the MPS using these protocols. MPOA also ensures interoperability with the existing infrastructure of routers. MPOA Servers make use of routers that run standard internetwork layer routing protocols, such as OSPF, providing a smooth integration with existing networks.

MPOA makes route servers appear as conventional routers to the external IP or other Layer 3 network, and would only be responsible for route calculations. The traditional router function of packet forwarding would be performed by edge devices connected to a route server, to the Layer 3 network, and to the ATM network. At the start of the session, the route server calculates an appropriate end-to-end route, and hand sover to an edge device to compute the best path through the ATM network. The path then emerges from the ATM cloud at an exit point as close as possible to the destination. For short packet flows, the route server would be involved with every packet just like conventional routers. However, for longer flows, such as with a FTP session, or an email transmission, a cut through path can be set up through the ATM network directly via edge devices, bypassing the router server. In this respect MPOA is similar to Ipsilon's IP switching [RFC2297], which also routes short flows conventionally but sets up cut through for longer flows. However different protocols are used to set up the path through the ATM network, NHRP in the case of MPOA, and generic switch management protocol (GSMP) [RFC1987] for IP switching.

At the moment, MPOA cuts paths through the ATM network in the same way, using the Next Hop Routing Protocol (NHRP). But the route server has been replaced by the much simpler NHRP server, sometimes called MPS (Multiprotocol server). The MPS server is merely responsible for computing the path through the ATM network that corresponds best with the Layer 3 hop calculated by a conventional router, using standard routing protocols such as OSPF. So now MPOA consists just of LAN emulation (LANE) and NHRP, plus a few other internal signalling protocols.

It is worth noting that the NHRP is not an end-to-end route calculation protocol, and does not decide which subnet to forward traffic to; this is left to conventional routers.

Although MPOA is comprehensive in its attempt to provide an efficient means of transporting any type of layer 3 data across an ATM network, whilst remaining compatible with LANE, it has been suggested [TIN96] that it is too complex and doesn't scale well.

2.2.4 QoS Support for IP Services at the Physical Layer

2.2.4.1 IP over SONET

Synchronous Optical Network (SONET) / Synchronous Digital Hierarchy (SDH) is a physical layer technology designed to provide a universal transmission and multiplexing scheme, with transmission rates in the gigabit per second range, and a sophisticated Operations and Management system [MAN98][PAG].

SONET is a standard for connecting fibre-optic transmission systems and transports user data in containers. It defines interface standards at the physical layer of the OSI model.

SONET establishes OC levels from 51.8 Mbps to 2.48 Gbps and beyond. However, SONET was originally designed to facilitate the transport of existing plesiochronous signals and so many of the container capacities are dimensioned accordingly [McD].

As SONET is a physical layer protocol and IP is a network layer protocol, according to the OSI 7-layer model, a mediating data link layer protocol is required between the two.

The physical layer is responsible for transmitting raw bits over a communication channel. The data link layer is responsible for converting the raw bit stream offered by the physical layer into a stream of frames for use by the network layer. The network layer is concerned with getting packets from the source to the destination and deals with such issues as packet routing and congestion control in a subnet.

Although SONET is logically described as a series of "frames", from a data link layer perspective it is nothing more than a series of octets. In order to be able to map IP packets into an SONET Payload Envelope (SPE) [TRI98] we need to be able to clearly determine the beginning and end of a packet (or multiple packets) within the SPE. The Point-to-Point Protocol (PPP) data link protocol is normally used to provide IP packet delineation [RFC1661].

SONET has the ability to offer very simple traffic engineering mechanisms although at present their implementation is proprietary. Traffic can be 'fairly' transported along physical links within containers. At Add and Drop Multiplexers traffic can be 'switched' onto different paths as shown in Figure 2-8.

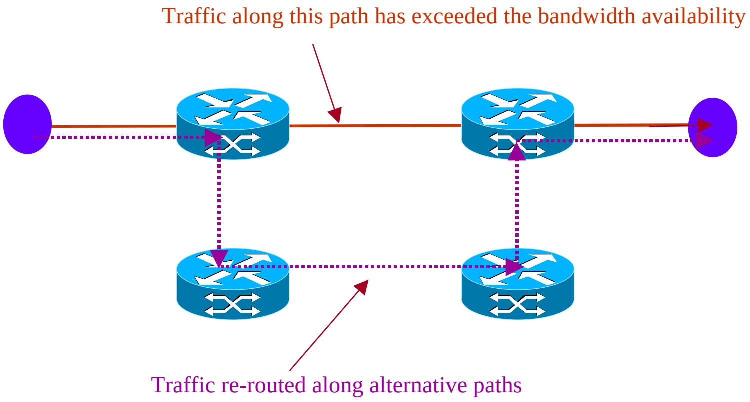

Traffic along this path has exceeded the bandwidth availability
Traffic re-routed along alternative paths

Figure 2-8 Traffic Engineering within SONET

Here, the network operator has agreed to transport data between two customer sites. Traffic can either be transported across the network infrastructure along multiple paths for load balancing, or a more "service aware" scheme can be employed. For example, traffic that is delay sensitive can be transported in a container taking a short physical path to the destination, whilst delay tolerant traffic is transported along a more circuitous route.

2.2.4.2 IP over DWDM

Dense Wavelength Division Multiplexing (DWDM) [GAN] has the ability to increase the capacity of existing networks without the need for expensive re-cabling and can reduce the cost of network upgrades. DWDM is the process of multiplexing signals of different wavelengths onto a single fibre. By doing this, it creates many virtual fibres each capable of carrying a different physical layer signal.

IP over DWDM [SEE2000] [JAIN] is the concept of sending data packets over an optical layer using DWDM for its capacity and other operations. The optical layer has been supplemented with more functionality, which were once in the higher layers. This creates an all-optical network where all management is carried out in the network and photonic layers [CRO99]. Where possible, the optical network can provide end-to-end services completely in the optical domain, without having to convert the signal to the electrical domain during transit. Transmitting IP directly over DWDM is able to support bit rates of OC-192 (Optical Carrier).

DWDM's ability to transport traffic along different wavelengths along the same physical path allows DWDM to offer a basic form of traffic engineering. By assigning different traffic classes to different wavelengths, traffic can be routed along particular nodes and links within the network as illustrated in Figure 2-9 where two wavelengths are supported on each link.

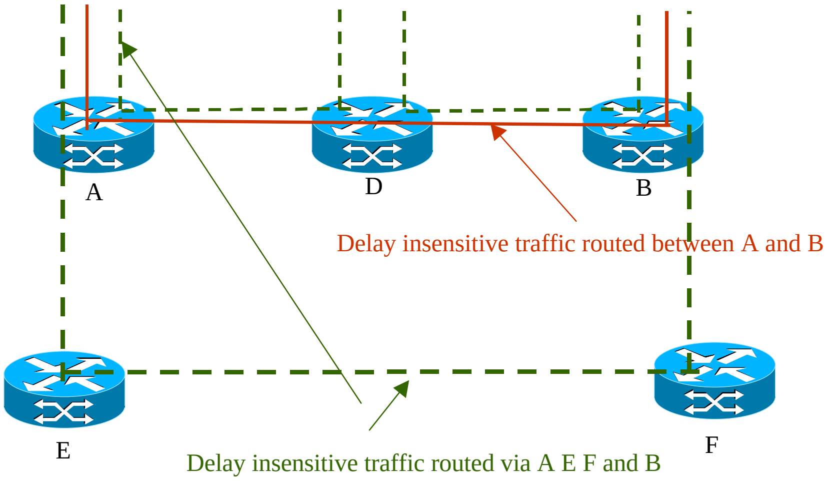

A
D
B
Delay insensitive traffic routed between A and B
E
F
Delay insensitive traffic routed via A E F and B

Figure 2-9 Traffic Engineering within DWDM

In this figure a considerable amount of traffic is to pass between nodes A and B. However, the "green" wavelength is already in use for transporting other traffic between nodes A and D and D and B. The operator thus chooses to send some of the traffic from A, destined for B, using the "red" wavelength via nodes A-D-B. Additional traffic from A to B is sent via a "green" wavelength path along a longer alternative route. By using a suitable packet filter at node A, premium service packets can be transported along the short "red" route to node B, whilst "best effort" services take the longer green route.

2.2.5 Summary

Chapter 2 has described a variety of mechanisms and procedures that allow traffic engineering philosophies to be deployed within the Internet. It has described various schemes capable of discriminating between different classes of service. Some or these can be applied to large-scale networks, as they do not require any form of per-user information processing or state storage. Whilst others support QoS on a per-flow basis. The chapter has also described service architectures for the transport of IP over wide area networks. However, many of these schemes involve simply overlaying IP over lower layer technology such that the underlying layers are unable to discriminate between different network layer service classes. The following Chapter 3 describes a protocol called Multiprotocol Label Switching that provides a new “IP aware” datalink layer technology with inherent traffic engineering capabilities.

Chapter 3: Multiprotocol Label Switching

3.1 Overview

Multi-Protocol Label Switching (MPLS) [ROS99b] has evolved from the fast IP switching solutions proposed in the mid-1990s from a plethora of companies such as Ipsilon, Cisco and IBM [RFC1987]. In traditional network layer routing, as a router receives a packet it makes an independent forwarding decision for that packet. Each router analyses the packet's header and performs a best match routing table lookup to make an independent decision as to what the next hop for the packet should be.

MPLS emerged from the IETF's effort to standardise these proprietary solutions, with the primary objective of integrating label switched forwarding with network layer routing [SEM2000a]. Label switching is expected to:

Address scalability issues and overheads that were associated with IP-over-ATM overlay networks.

Enable forwarding to occur at terabit speeds by simplifying the operation in the core of the network.

Provide support for new routing capabilities that enhance conventional IP routing techniques, by offering connection-like benefits with traffic classification capabilities.

MPLS has a number of potential benefits over existing pure datagram routed networks [CAL99a]:

Simplified Forwarding: Label switching allows packet forwarding to be based on an exact match for a fixed length label, rather than a longest match algorithm applied to a longer address as used in normal datagram forwarding.

Efficient Routing: MPLS allows the explicit route to be carried only at the time that the label switched path is set up, and not with each packet. Whereas with datagram routing the explicit route has to be carried in each packet and this causes a large overhead.

Traffic Engineering: Enables the loading on links and routers to be balanced throughout the network, this is an important concept in networks where alternative paths are available. Traffic Engineering can be achieved somewhat by adjusting the metrics associated with network links in datagram routing. However, in a network with a large number of alternative paths between any two points, balancing traffic levels on all links is difficult to achieve solely by adjusting the metrics used with hop by hop datagram routing.

Mapping IP packets to Forwarding Equivalence Classes: MPLS allows the mapping of IP packets to FECs to occur only once at the ingress to an MPLS domain. In the case of datagram routing, IP packets would be mapped to a service level that would require packet filtering based on source and destination addresses and incoming interface, etc. Also some information such as the incoming interface is only available at the ingress node to the network. This implies that the preferred way to offer provisioned QoS is to map the packet at the ingress point to the preferred QoS level, and then label the packet in a way to acknowledge that. MPLS offers an efficient method to label the QoS class associated with any packet.

Simple Forwarding Paradigm: MPLS offers a simple forwarding paradigm that enables the support of multiple types of service on the same network, regardless of the control plane protocols used to formulate the forwarding tables. MPLS can be deployed within switches that are not capable of analysing the network layer header, but are able to do a label lookup and replacement. Where each label is a short fixed length size, lookup and swapping can be efficiently performed in hardware.

Initially MPLS was focused its efforts on IPv4 over ATM, however the aim is to extend this to cater for multiple network layer protocols, i.e., IPv6, IPX, etc., over any data link technology.

It should be noted here that although many researchers within the IETF MPLS WG refer to MPLS packets, and the author will continue to follow this convention, they are actually MPLS frames containing layer 3 protocol data units (PDU) such as IP packets.

In traditional network layer routing, as a router receives a packet it makes an independent forwarding decision for that packet. Each router analyses the packet's header and performs a routing table lookup to make an independent decision as to what the next hop for the packet should be.

In MPLS, packets are assigned to a Forwarding Equivalence Class (FEC) at the ingress router located at the edge of the MPLS domain. The FEC to which they are assigned to can be dependent on a number of attributes including the address prefix in the packet's header, or the port the packet arrived on. However, the assignment of a packet to a FEC is done just once, as the packet enters the MPLS domain. The FEC to which the packet is assigned is encoded as a label, and is sent along with the packet when it is forwarded to its next hop, along with all the packets within the flow. A flow within an MPLS domain is defined as a stream of packets belonging to an application. Intermediate routers receiving this packet do not examine the network layer header. The label is used as an index into a table specifying the next hop and a new label with which to replace the old incoming label, as illustrated in Figure 3-1.

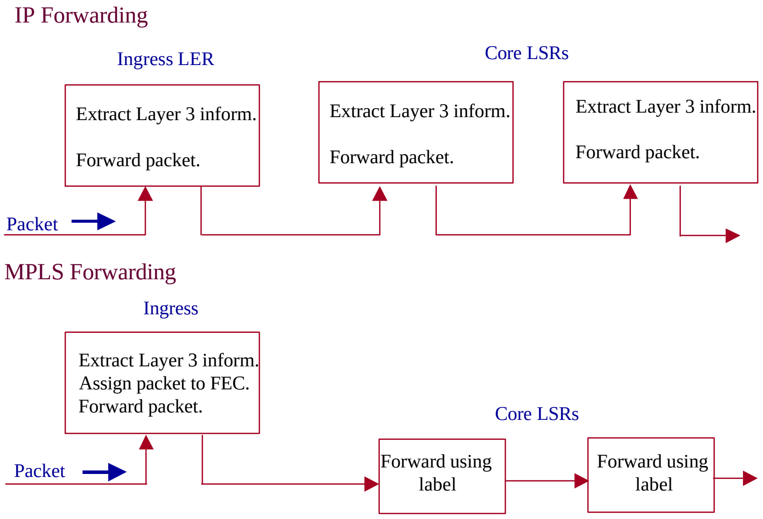


Figure 3-1 IP and MPLS Forwarding Techniques Compared

3.2 Principal Concepts in MPLS

A key concept in MPLS is the separation of an IP router's functions into two parts: forwarding and control [CHE99] The forwarding part is responsible for how data packets are relayed between IP routers, using label swapping similar to ATM switching's virtual path/ virtual channel identifier. The control part consists of network layer routing protocols to distribute routing information between routers, and label binding procedures for converting this routing information into the forwarding tables needed for label switching, as shown in Figure 3-2. It should be noted that MPLS is not a routing protocol – but is a fast forwarding mechanism that is designed to work with existing Internet routing protocols such as Open Shortest Path First (OSPF) [RFC1247], or the Border Gateway Protocol (BGP) [RFC1771]. This separation of the two components enables each component to be developed and modified independently.

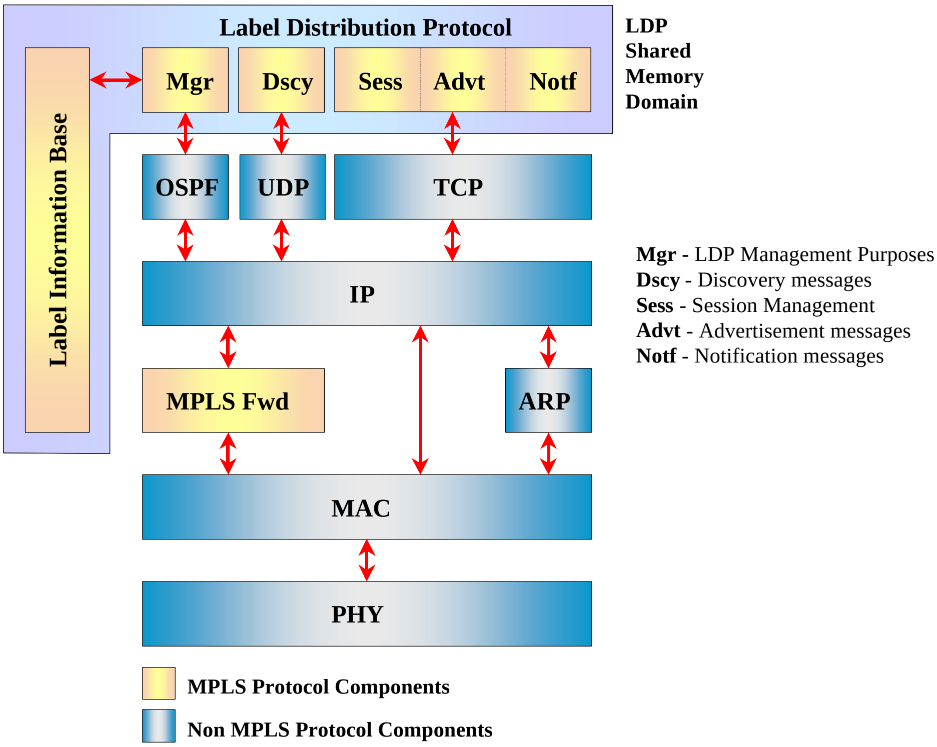

Label Distribution Protocol
LDP
Shared
Memory
Domain
Label Information Base
Mgr
Dscy
Sess
Advt
Notf
OSPF
UDP
TCP
IP
MPLS Fwd
ARP
MAC
PHY
Mgr - LDP Management Purposes
Dscy - Discovery messages
Sess - Session Management
Advt - Advertisement messages
Notf - Notification messages
MPLS Protocol Components
Non MPLS Protocol Components

Figure 3-2 Control and Forwarding Components

3.2.1 LDP Components and Mechanisms

A fundamental concept in MPLS is that two Label Switching Routers (LSRs) must agree on the meaning of the labels used to forward traffic between them and through them. This understanding is achieved by employing a set of signalling procedures, called a label distribution protocol [AND99], by which one LSR informs another of the label / FEC bindings it has made. Two LSRs which use a label distribution protocol to exchange label / FEC binding information are known as “label distribution peers” with respect to the binding information they exchange. If two LSRs are label distribution peers, a “label distribution adjacency” exists between them. This thesis refers to the Label Distribution Protocol (LDP) although the author appreciates that there a number of different label distribution protocols being standardised as described in [JAM99a], [REK99] and [AWD2000].

The LDP is a protocol defined for distributing labels. It is a set of procedures and messages by which LSRs establish Label Switched Paths (LSPs) through a network by mapping network layer routing information directly to data-link layer switched paths.

LDP associates a FEC with each LSP it creates. The FEC associated with an LSP specifies which packets are mapped to that LSP. LSPs are extended through a network as each LSR maps incoming labels for a FEC to the outgoing label to the next hop for the given FEC.

The following sections outline the various components, mechanisms and procedures deployed within the MPLS architecture, however more information is provided in [AND99] and Figure 3-3 illustrates a typical MPLS domain.

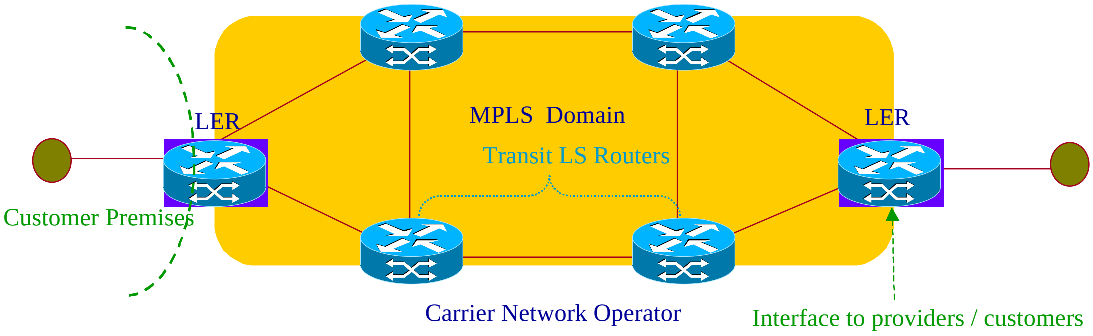


Figure 3-3 A MPLS Domain

3.2.1.1 Label Switch Router

A Label Switch Router (LSR) is a device that is capable of forwarding packets at layer 3 and forwarding frames that encapsulate the packet at layer 2. The label swapping mechanism is implemented at layer 2.

3.2.1.2 Label Edge Router

A Label Edge Router (LER) is both a router and a layer 2 switch that is capable of forwarding MPLS frames to and from an MPLS domain. It performs the IP to MPLS FEC binding including the aggregation of incoming flows. It also communicates with interior MPLS LSRs to exchange label bindings. Often referred to as an ingress or egress LSR, because it is situated at the edge of a MPLS domain.

3.2.1.3 Label Switch Path

A Label Switch Path (LSP) is an ingress-to-egress switched path built by MPLS nodes to forward the MPLS encapsulated packets of a particular FEC using the label swapping forwarding mechanism. It is similar to the concept of Virtual Channels within an ATM context.

3.2.1.4 Forwarding Equivalence Classes

A Forwarding Equivalence Class (FEC) is a set of packets that are treated identically by a router, i.e., forwarded out the same interface with the same next hop and label, and assigned the same class of service. When a packet enters the MPLS domain at the ingress node, it is mapped into a FEC. The mapping can be done according to a number of factors, i.e., the address prefix, source/destination address pair, or ingress interface. At the current moment there are three defined FEC elements, an address prefix, router ID and flow (source/destination port and IP addresses). A group of IP packets that are forwarded over the same path and treated in the same manner and can be mapped to a single label by a LSR, as shown in Figure 3-4.

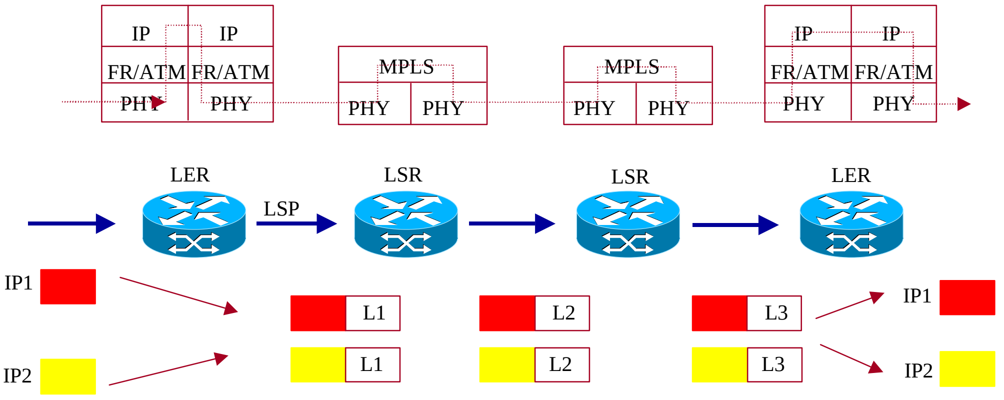


Figure 3-4 Forwarding Equivalence Classes

3.2.1.5 Label

A label is a short, fixed length, locally significant identifier that is used to identify a FEC. A packet may be assigned to a FEC based on its network layer destination address; however, the label does not directly encode any information from the network layer header. A labelled packet is a packet into which a label has been encoded. The label may reside in an encapsulation header that exists specifically for this purpose, called an MPLS 'shim' header, or a stack entry as it is often referred as, shown in Figure 3-5[5]. Alternatively, it may reside in an existing data link as long as there is a field that is available for that purpose [ROS2000a][WOR2000]. The 32-bit MPLS header contains the following fields:

Label field (20-bits), carries the actual label value.

Experimental field (3-bits), not yet defined.

S (1-bit), Stack supports a hierarchical label stack.

TTL (8-bits) Time-To-Live an inherent part of IP functionality.

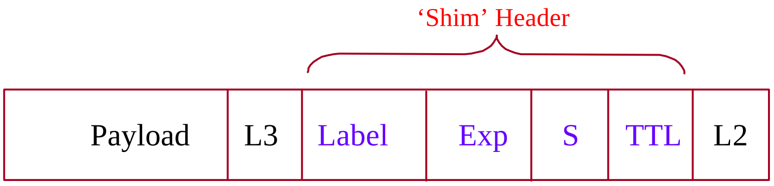


Figure 3-5 MPLS 'shim' header

In assigning and distributing labels, there are two possible label spaces: per interface label space and per platform label space.

Per interface label space allows labels are used for interfaces that use interface resources for labels, e.g., an ATM interface that uses VCIs as labels.

[5] This diagram illustrates a generic encapsulation, however it would differ for alternative data link frames.

Platform label spaces are used when a single label space is partitioned across multiple interfaces.

3.2.1.6 Label Stack

A Label Stack is an ordered set of labels appended to a packet that enables it to explicitly carry information about more than one FEC that the packet belongs to and the corresponding LSPs that the packet may traverse. The label stack is represented as a sequence of "label stack entries" appearing after the data link layer headers, but before any network layer headers. The entries can be 'pushed' i.e., placed onto the stack or 'popped' i.e., removed from the stack. LSP1 and LSP2 labels can be stacked inside MPLS frames with label LSP3, as shown in Figure 3-6. Label stacking enables 'tunnelling' to occur within an MPLS domain as explained in Section 3.2.1.17.

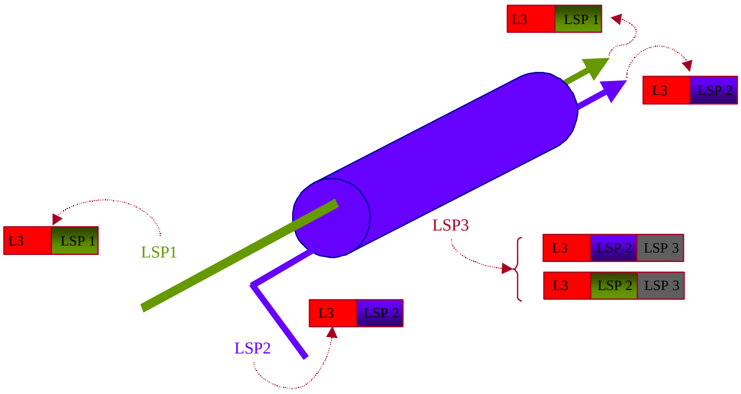


Figure 3-6 Label Stacking

3.2.1.7 Label Encapsulations

MPLS is intended to run over multiple data link layers such as:

ATM: where the label is contained in the VPI/VCI field of the ATM header.

Frame Relay: where the label is contained in the DLCI field in the FR header.

PPP/LAN: where the 'shim' header is inserted between the layer two and three headers.

Additional 'shim' labels can be stacked.

3.2.1.8 Upstream and Downstream LSRs

R1 and R2 are LSRs as shown in Figure 3-7, where they have agreed to bind a label L to FEC F, for packets sent from R1 to R2. Therefore with respect to this binding R1 is said to be the 'upstream LSR' and R2 is the 'downstream LSR'. R2 informs R1 of the binding 🕙 and R1 keeps a record of it, and uses it to forward frames with the new binding 🕚.

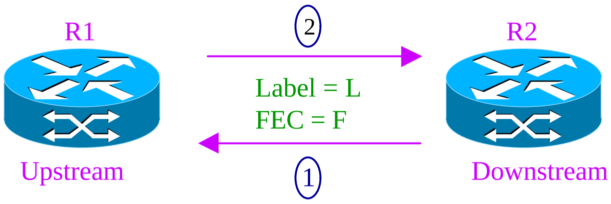


Figure 3-7 Upstream and Downstream LSRs

3.2.1.9 LDP Discovery

LDP discovery is a mechanism that enables a LSR to discover potential LDP peers. There are two types of discovery mechanisms, Basic Discovery and the Extended Discovery mechanism. This section describes the Basic approach. The Extended approach falls outside the scope of this thesis but is described in [[AND99]].

To engage in LDP Basic Discovery a LSR periodically sends LDP Link Hello messages. Hello messages are sent as UDP packets addressed to the 'well-known' LDP discovery port for the "all routers on this subnet" group multicast address as shown in Figure 3-8. A Hello message contains the LDP Identifier for the label space the LSR intends to use for the interface. Receipt of a Hello message on an interface identifies a 'Hello Adjacency' with a potential LDP peer reachable at the link level on the interface as well as the label space the peer intends to use for that interface.

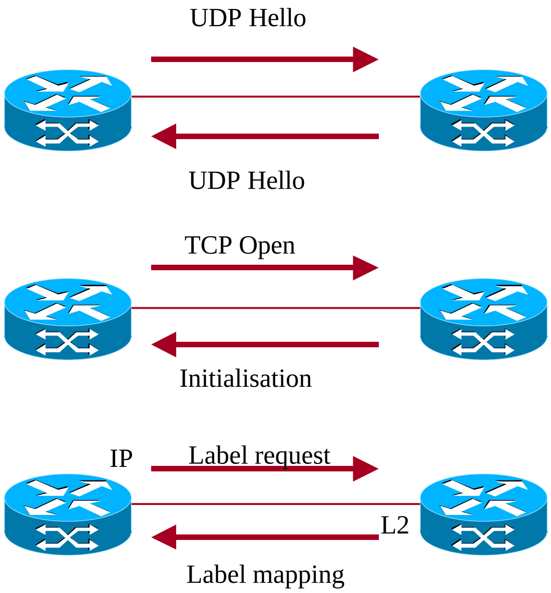


Figure 3-8 Discovery Procedures

3.2.1.10 LDP Session and Establishment

The exchange of Discovery Hello messages between two LSRs triggers the LDP session establishment. This is a two step process:

Transport connection establishment

Session Initialisation

Transport Connection Establishment

The exchange of hellos results in the creation of a Hello Adjacency at LSR1 that serves to bind the link L and the label spaces LSR1:a and LSR2:b.

If LSR1 does not already have an LDP session for the exchange of label spaces LSR1:a and LSR2:b it attempts to open a TCP connection for a new LDP session with LSR2.

LSR1 determines the transport addresses to be used at its end (A1) and LSR2's end (A2) of the LDP TCP connection. Address A1 is determined as follows:

a) If LSR1 uses the address field in the Hello's it sends to LSR2 to advertise an address, A1 is the address LSR1 advertises.

b) If LSR1 does not use the address field, A1 is the source IP address used in Hellos it sends to LSR2.

Similarly, address A2 is determined as explained above.

2. LSR1 determines whether it will play the active or passive role in session establishment by comparing addresses A1 and A2 as unsigned integers. If A1 > A2, LSR1 plays the active role; otherwise it is passive.

3. If LSR1 is active, it attempts to establish the LDP TCP connection by connecting to the well-known LDP port at address A2. If LSR1 is passive, it waits for LSR2 to establish the LDP TCP connection to its well-known LDP port.

3.2.1.11 Session Initialisation

After LSR1 and LSR2 establish a transport connection they negotiate session parameters by exchanging LDP Initialisation messages. The parameters negotiated include LDP protocol version, label distribution method, timer values and VPI/VCI ranges for ATM.

Successful negotiation completes establishment of an LDP session between LSR1 and LSR2 for the advertisement of label spaces LSR1:a and LSR2:b.

After a connection is established, if LSR1 is playing the active role, it initiates negotiation of session parameters by sending an Initialisation message to LSR2. If LSR1 is passive, it waits for LSR2 to initiate the parameter negotiation.

By waiting for the Initialisation message from its peer the passive LSR can match the label space to be advertised by the peer with a Hello adjacency previously created when Hellos were exchanged.

1. When LSR1 plays the passive role:

a) If LSR1 receives an Initialisation message it attempts to match the LDP Identifier carried by the message PDU with a Hello adjacency.

b) If there is a matching Hello adjacency, the adjacency specifies the local label space for the session. Next LSR1 checks whether the session parameters proposed in the message are acceptable. If they are, LSR1 replies with an Initialization message of its own to propose the parameters it wishes to use and a KeepAlive message to signal acceptance of LSR2's parameters. If the parameters are not acceptable, LSR1 responds by sending a Session Rejected/Parameters Error Notification message and closing the TCP connection.

c) If LSR1 cannot find a matching Hello adjacency it sends a Session Rejected/No Hello Error Notification message and closes the TCP connection.

d) If LSR1 receives a KeepAlive in response to its Initialization message, the session is operational from LSR1's point of view.

e) If LSR1 receives an Error Notification message, LSR2 has rejected its proposed session and LSR1 closes the TCP connection.

2. When LSR1 plays the active role:

a) If LSR1 receives an Error Notification message, LSR2 has rejected its proposed session and LSR1 closes the TCP connection.

b) If LSR1 receives an Initialization message, it checks whether the session parameters are acceptable. If so, it replies with a KeepAlive message. If the session parameters are unacceptable, LSR1 sends a Session Rejected/Parameters Error Notification message and closes the connection.

c) If LSR1 receives a KeepAlive message, LSR2 has accepted it proposed session parameters.

d) When LSR1 has received both an acceptable Initialization message and a KeepAlive message the session is operational from LSR1's point of view.

As described it would be possible for a pair of incompatibly configured LSRs that disagree on session parameters to engage in an endless sequence of messages as each Negative Acknowledgements (NAKs) the other's Initialization messages with Error Notification messages.

However, an LSR must throttle its session setup retry attempts with an exponential backoff in situations where Initialization messages are being NAK'd. It is also recommended that an LSR detecting such a situation take action to notify an operator. The session establishment setup attempt following a NAK'd Initialization message must be delayed no less than 15 seconds, and subsequent delays must grow to a maximum delay of no less than 2 minutes. The specific session establishment action that must be delayed is the attempt to open the session transport connection by the LSR playing the active role.

The throttled sequence of Initialization NAKs is unlikely to cease until operator intervention reconfigures one of the LSRs. After such a configuration action there is no further need to throttle subsequent session establishment attempts (until their initialization messages are NAK'd). Due to the asymmetric nature of session establishment, reconfiguration of the passive LSR will go unnoticed by the active LSR without some further action. Section 3.2.1.14 describes an optional mechanism an LSR can use to signal potential LDP peers that it has been reconfigured.

3.2.1.12 Maintaining Hello Adjacencies

An LDP session with a peer has one or more Hello adjacencies. An LDP session has multiple Hello adjacencies when a pair of LSRs is connected by multiple links that share the same label space; for example, multiple PPP links between a pair of routers. In this situation the Hellos an LSR sends on each such link carry the same LDP Identifier. LDP includes mechanisms to monitor the necessity of an LDP session and its Hello adjacencies. LDP uses the regular receipt of LDP Discovery Hellos to indicate a peer's intent to use the label space identified by the Hello. An LSR maintains a hold timer with each Hello adjacency that it restarts when it receives a Hello that matches the adjacency. If the timer expires without receipt of a matching Hello from the peer, LDP concludes that the peer no longer wishes to label switch using that label space for that link (or target, in the case of Targeted Hellos) or that the peer has failed. The LSR then deletes the Hello adjacency. When the last Hello adjacency for a LDP session is deleted, the LSR terminates the LDP session by sending a Notification message and closing the transport connection.

3.2.1.13 Maintaining LDP Sessions

LDP includes mechanisms to monitor the integrity of the LDP session. LDP uses the regular receipt of LDP PDUs on the session transport connection to monitor the integrity of the session. An LSR maintains a KeepAlive timer for each peer session which it resets whenever it receives an LDP PDU from the session peer. If the KeepAlive timer expires without receipt of an LDP PDU from the peer the LSR concludes that the transport connection is bad or that the peer has failed, and it terminates the LDP session by closing the transport connection. After an LDP session has been established, an LSR must arrange that its peer receive an LDP PDU from it at least every KeepAlive time period to ensure the peer restarts the session KeepAlive timer.

The LSR may send any protocol message to meet this requirement. In circumstances where an LSR has no other information to communicate to its peer, it sends a KeepAlive message. An LSR may choose to terminate an LDP session with a peer at any time. Should it choose to do so, it informs the peer with a Shutdown message.

3.2.1.14 LDP messages

There are four categories of LDP messages that are considered to be essential in establishing and maintaining LSPs:

Discovery messages advertise the presence of LSRs.

Session messages establish and maintain LDP messages.

Advertisement messages create, change, and delete label mappings for FECs.

Notification messages carry advisory and error information.

All LDP messages have the format shown in Table 3-1. If an LSR does not recognise a message, the U (unknown message) bit tells the LSR whether to notify the sender. The 15-bit message type field identifies an LDP as one of 10 defined types:

Hello message for LDP discovery.

Initialisation message for LDP session establishment.

Keep Alive message to maintain the continuity of an LDP session in absence of other messages.

Address message to advertise interface addresses.

Address Withdraw message to withdraw previously advertised interface addresses.

Label Mapping message to advertise label bindings.

Label Request message to request a label binding for an FEC.

Label Withdraw message to break a previously established FEC label mapping.

Label Release message to free an FEC label mapping.

Notification message to give advisory or error information about various events.

U	Message Type	Message Length	Message ID	Mandatory Parameters	Optional Parameters

Table 3-1 LDP message format

The 16-bit message length field is the total length of the message in bytes. The 32-bit message ID is a number that uniquely identifies the particular message. The mandatory and optional parameters use a type-length-value (TLV) encoding with the format shown in Table 3-2. If an LSR does not recognise the TLV, the U (unknown TLV) bit tells the LSR whether to notify the sender and ignore the entire message, or ignore the TLV and process the remainder of the message. If an LSR does not recognise the TLV and the message is to be forwarded, the F (forward unknown TLV) bit tells the LSR whether to forward the unknown TLV. The 14-bit type field indicates the type of message. The 16-bit length field is the length of the value field in bytes.

U	F	Type	Length	Value

Table 3-2 TLV encoding

3.2.1.15 Data Structures

This section describes the various data structures that are required for the establishment and maintenance of LSPs they are namely the *Next Hop Label Forwarding Entry (NHLFE), FEC-to-NHLFE (FTN) and the Incoming Label Map (ILM).* Figure 3-9 illustrates the various structures within an MPLS domain.

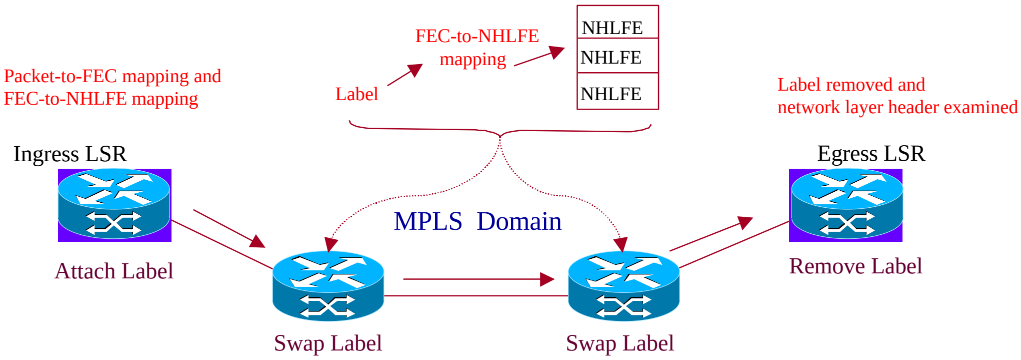


Figure 3-9 Forwarding packets in an MPLS domain

3.2.1.15.1 Next Hop Label Forwarding Entry

The Next Hop Label Forwarding Entry (NHLFE) is used when forwarding a packet encapsulated in a MPLS frame i.e., labelled packet. It contains the following information:

The packet’s next hop

The operation to be performed on the packet:

Replace the top label with a new label

Pop the label

Replace the top label with a new label and push one or more new labels onto the stack.

The data link encapsulation to use when transmitting the packet

The way to encode the label stack when transmitting the packet

Assigns a value to the LSP called the LSP-ID.

Any additional information needed in order to properly dispose of the packet.

An example of a NHLFE table is shown in Table 3-3.

Input Label	Input Port	Destination	Operation	Output Label	LSP ID		Output Port
					Ingress ID	Value	
0:54	3	138.43	Replace 0:	0:81	1	10	2

Table 3-3 NHLFE

3.2.1.15.2 FEC-to-NHLFE

The FEC-to-NHLFE (FTN) table is used to forward unlabelled packets by translating and binding a layer 3 entity to a layer 2 MPLS LSP, it maps each incoming label to a set of NHLFEs, and can be located at the LERs, where it provides an initial label for the first hop. Table 3-4 shows the entries in a typical FTN.

Input Port	Destinatio n	Output Port	Output Label
1	138.43	2	0:54

Table 3-4 FTN

3.2.1.15.3 Incoming Label Map

The Incoming Label Map (ILM) is located within the core of an MPLS network on the LSRs, and maps each incoming label to a set of NHLFEs, in order to forward labelled packets, as shown in Table 3-5.

Destination	Input Port	Input Label
138.43	3	0:54

Table 3-5 ILM

3.2.1.15.4 LDP Identifiers and Next Hop Addresses

A LSR identifies itself by a six octet LDP identifier, where the first four octets encode an IP address assigned to the LSR, and the last two octets identify a label space within the LSR. A LSR uses a different identifier for each set of label spaces it decides to use. Before an LSR can exchange labels with a peer, a LDP session must exist. LDP uses TCP as a reliable transport for sessions.

An LSR maintains learned labels in a Label Information Base (LIB). When operating in Downstream Unsolicited mode, the LIB entry for an address prefix associates a collection of (LDP Identifier, label) pairs with the prefix, one such pair for each peer advertising a label for the prefix.

When the next hop for a prefix changes the LSR must retrieve the label advertised by the new next hop from the LIB for use in forwarding. To retrieve the label the LSR must be able to map the next hop address for the prefix to an LDP Identifier. Similarly, when the LSR learns a label for a prefix from an LDP peer, it must be able to determine whether that peer is currently a next hop for the prefix to determine whether it needs to start using the newly learned label when forwarding packets that match the prefix.

To make that decision the LSR must be able to map an LDP Identifier to the peer's addresses to check whether any are a next hop for the prefix. To enable LSRs to map between a peer LDP identifier and the peer's addresses, LSRs advertise their addresses using LDP Address and Withdraw Address messages. An LSR sends an Address message to advertise its addresses to a peer. An LSR sends a Withdraw Address message to withdraw previously advertised addresses from a peer.

3.2.1.16 Label Swapping

Label Swapping uses the following procedures to forward labelled and unlabelled packets.

In forwarding a labelled packet, a LSR examines the label at the top of the label stack, and examines the ILM to map this label to an NHLFE. With this information it determines where this packet needs to be forwarded to, and performs an operation on the packet’s label stack. This operation may mean swapping the incoming label with a new output label, or replacing the label with a new label and pushing a new label on top, or simply ‘popping’ the label and examining network layer header. It then encodes the new label stack into the packet and forwards it.

In forwarding a packet that is received unlabelled – unlabelled packet, a LSR determines which FEC to assign the packet to by examining the packet’s network layer header. Once the packet has been assigned to a FEC, the LSR uses the FTN to map this to an NHLFE. Using the information in the NHLFE, it determines where to forward the packet, and performs an operation on the packet’s label stack. It then encodes the new label stack into the packet and forwards it.

3.2.1.17 Tunnelling

A LSR, R1 can cause a packet to be delivered to another LSR R4, even though R1 and R4 are not consecutive routers on a hop-by-hop path for that packet.

Consider Figure 3-10 where a LSP is formed by {R1, R2, R3, R4}. Suppose R1 receives an unlabelled packet P, and pushes on its label stack the label to cause it to follow this path. Suppose though that R2 and R3 are not directly connected, but are connected via LSRs {R21, R22, R23} so in fact the LSP is {R1, R2, R21, R22, R23, R3, R4}.

P will have a label stack of depth 1 when it travels from R1 to R2, at R2 it is determined that P must enter a tunnel. R2 first replaces the incoming label with a label that is meaningful to R3 and pushes another label on that is recognised by R21. This new label is now at a depth of 2, resulting in switching occurring at level 2 for R21, R22, and R23. When P arrives at R23, the top label will be popped and forwarded on the level 1 label that will be recognised by R3, and used to forward the packet unto R4. This process of label stacking allows LSP tunnelling to occur to any depth.

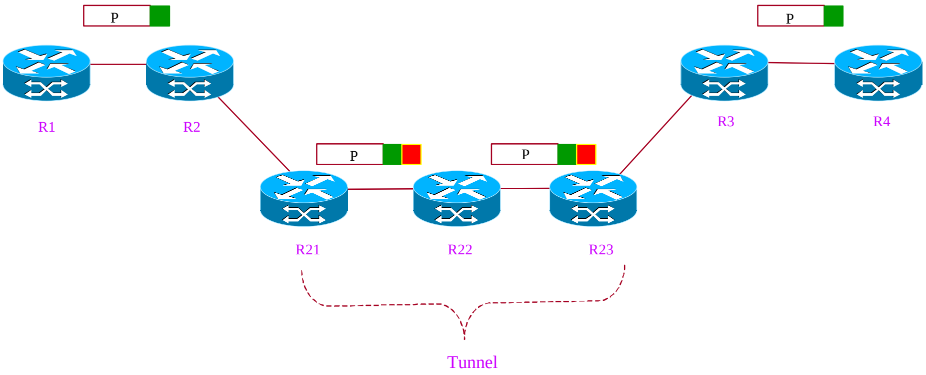


Figure 3-10 Tunnelling

3.2.2 Route Selection

Route selection refers to the method used for selecting the LSP for a particular FEC. The proposed MPLS protocol architecture supports two options for Route selection: hop by hop routing, and explicit routing.

Hop by hop routing allows each node to independently choose the next hop for each FEC. This is the usual mode today in existing IP networks. A “hop by hop routed LSP” is an LSP whose route is selected using hop by hop routing.

In an explicitly routed LSP, each LSR does not independently choose the next hop; rather, a single LSR, generally the LSP ingress or the LSP egress, specifies several (or all) of the LSRs in the LSP. If a single LSR specifies the entire LSP, the LSP is “strictly”explicitly routed. If a single LSR specifies only some of the LSP, the LSP is “loosely” explicitly routed.

The sequence of LSRs followed by an explicitly routed LSP may be chosen by configuration, or may be selected dynamically by a single node.

Explicity routing may be useful for a number of purposes, such as policy routing or traffic engineering. In MPLS, the explicit route needs to be specified at the time that labels are assigned, but the explicit route does not have to be specified with each IP packet. This makes MPLS explicit routing much more efficient than the alternative of IP source routing.

3.2.3 Label Distribution Modes

The MPLS architecture allows two modes of operation for LSRs distributing label mappings; there is the *Downstream on Demand* label distribution mode and the *Downstream Unsolicited* label distribution mode.

3.2.3.1 Downstream on Demand Label Advertisement

The MPLS architecture allows a LSR to explicitly request a label from its next hop for a particular FEC. This is referred to as the Downstream on Demand mode, where the upstream LSR is invariably responsible for requesting label mappings. Figure 3-11 illustrates LSR1 requesting a label from LSR2.

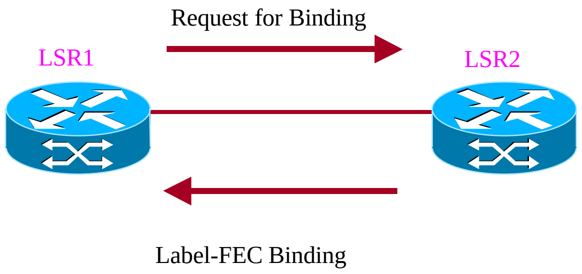


Figure 3-11 Downstream on Demand Label Advertisement

3.2.3.2 Downstream Unsolicited Label Advertisement

The MPLS architecture allows a LSR to distribute bindings to LSRs that have not explicitly requested for them. This is referred to as the Downstream Unsolicited mode, where the downstream LSR is responsible for advertising a label mapping to the upstream LSR. Figure 3-12 illustrates LSR2 delivering a label binding to LSR1 without LSR1 requesting one.

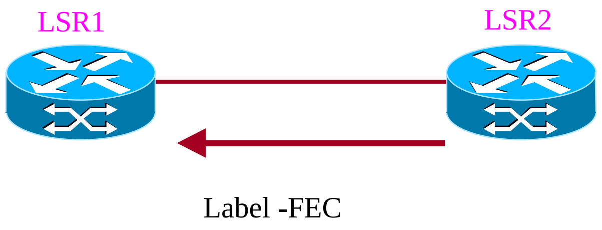


Figure 3-12 Downstream Unsolicited Label Advertisement

3.2.4 LSP Control Modes

There are two control modes that a LSR can exhibit in when establishing a LSP: they are *Independent LSP Control* and the *Ordered LSP Control*.

3.2.4.1 Independent Label Distribution Control

In Independent Label Distribution Control, each LSR may advertise a label mapping to its neighbours whenever it chooses. If a LSR is operating in the Independent Downstream on Demand mode, it may reply to a request for label mappings without waiting to receive a label mapping from the next hop. When operating in the Independent Unsolicited mode, a LSR may advertise a label mapping for a FEC to its neighbours whenever it is prepared to label-switch that FEC.

3.2.4.2 Ordered Label Distribution Control

In this mode of operation, a LSR may only transmit a label mapping for a FEC for which it has received a label mapping for the FEC from the next hop, or for which it is the egress. Before the LSR can pass label mappings to upstream LSRs, it must wait to receive a label from a downstream LSR in the case where the LSR is not the egress or where no mapping hitherto exists.

3.2.5 Label Retention Modes

There are two modes of retaining received labels employed within the MPLS architecture; they are the *Conservative Label Retention* and *Liberal Label Retention* modes. Consider Figure 3-13, where LSR1 has received label bindings from all its possible next hops to LSR5.

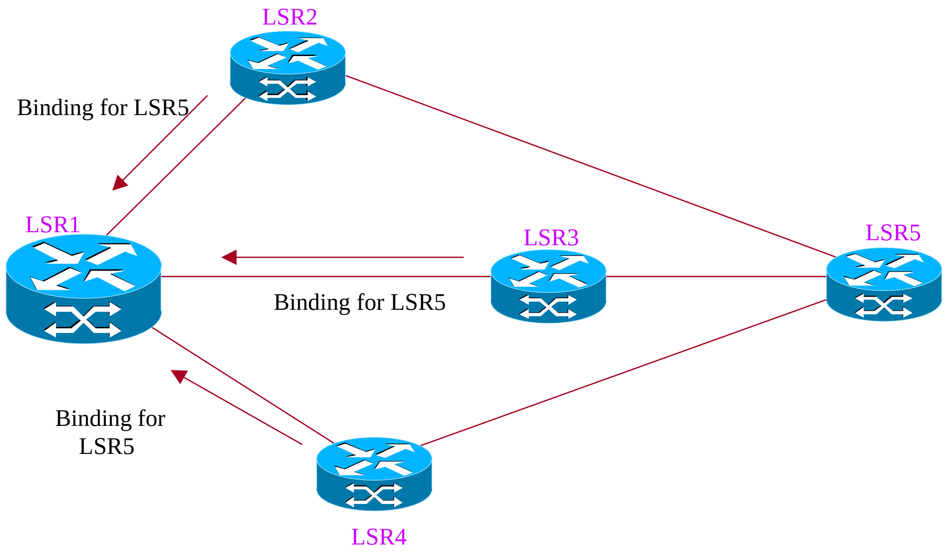


Figure 3-13 Label Mapping Distribution

3.2.5.1 Conservative Label Retention Mode

When operating in the Downstream Unsolicited Advertisement mode it is possible to receive label mappings for all routes from all peer LSRs. In the conservative label retention mode, label mappings are only retained if they are received from a valid next hop and will be used to forward packets. If Downstream on Demand is being used, a LSR will request a label only from the next hop LSR according to the routing protocol. In the case were label conservation is required, as in an ATM switch, Downstream on Demand is favoured along with the conservative label retention mode. The conservative mode is advantageous in that only labels needed for forwarding packets are allocated and maintained. This is particularly important in LSRs where the label space is limited as in an ATM switch. In Figure 3-14 LSR1 retains only the label binding it has received from the valid next hop to LSR5, i.e., LSR4.

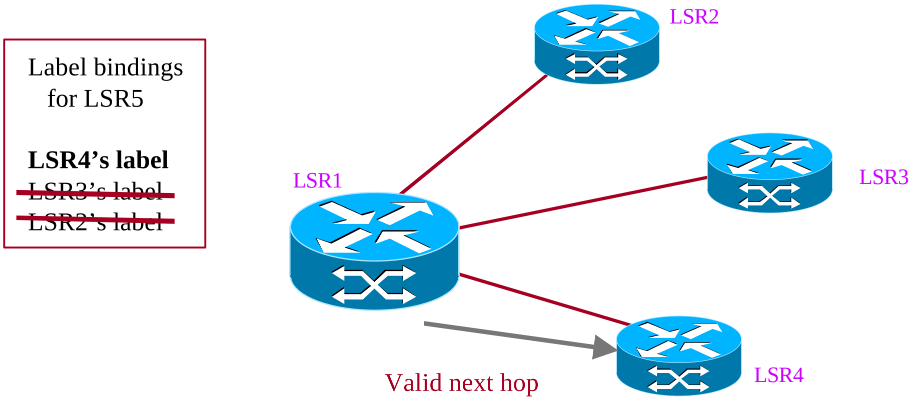


Figure 3-14 Conservative Label Retention

3.2.5.2 Liberal Label Retention Mode

In this mode every label mapping received from a peer LSR is retained regardless of whether the LSR is the next hop for the advertised mapping. The principle advantage of the liberal label retention mode is that reacting to routing changes can be quick as the labels already exist. The main disadvantage is that unneeded labels are distributed and maintained. In Figure 3-15 LSR1 retains all the label bindings from all its possible hops to LSR5.

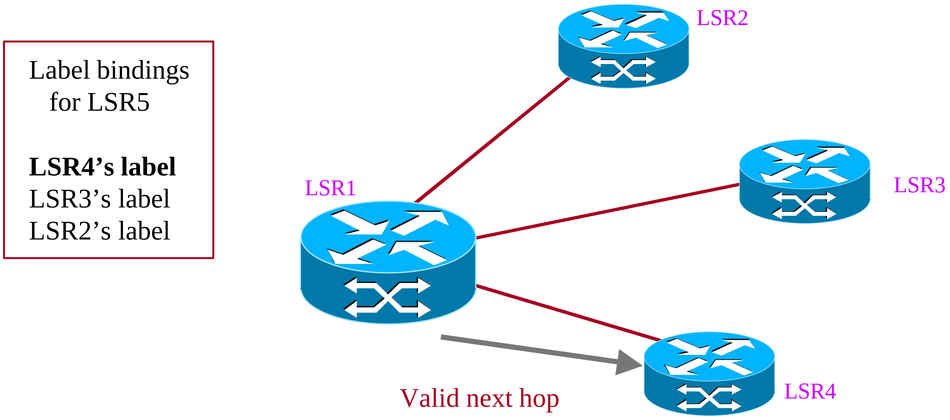


Figure 3-15 Liberal Label Retention

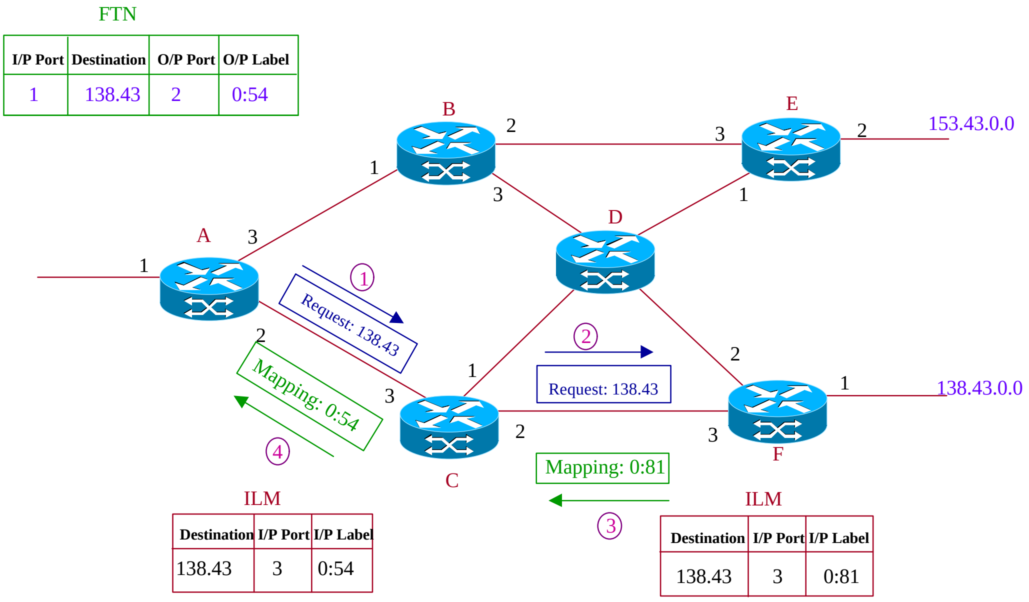
3.2.6 LSP Establishment and Maintenance

An ingress LER may decide that it does not want to set up a LSP for every possible destination in its routing table, and instead may forward some packets using conventionally network layer routing. The destinations to which it chooses to establish a LSP will be assigned to a FEC. It may choose to assign all possible destination addresses reachable from it to FECs, each being encoded as a short fixed length label, as shown in Table 3-6. Once the ingress LER has assigned all the packets to a FEC, it needs to obtain a label for each FEC entry in its table using a method called label distribution.

Destination Address	**FEC**	**Label**
153.43.17.6	153.43	0:54
138.54.32.2	138.43	0:35

Table 3-6 FEC-to-Label Binding

The mode of label distribution used within this research is Downstream on Demand Conservative Label Retention, where a LSR explicitly requests a label binding for a FEC from its next hop. An awareness of its neighbours is available from the hello protocol [AND99]. Once a path has been identified using a standard routing protocol such as OSPF, the ingress node formulates a Label Request message for the particular FEC, and forwards it to the next hop along the path that it wishes to establish, 🕙, as shown in Figure 3-16. When a LSR receives the messsage and determines that it is not the egress LER it forwards it to the next hop. When a LSR determines that it is the egress LER for that particular FEC it assigns a label and sends a Label Mapping message, 🕛, whose format is shown in Figure 3-19, with the label enclosed to the source of the Label Request message. When a LSR receives a Label Mapping message from the next hop, a Label Mapping message is propagated upstream until the ingress LSR for that FEC is reached, ⇙. Once the label has been received, a LSP has been set up between the ingress and egress LSRs, as shown Figure 3-20.



FTN

I/P Port	Destination	O/P Port	O/P Label
1	138.43	2	0:54

ILM

Destination	I/P Port	I/P Label
138.43	3	0:54

ILM

Destination	I/P Port	I/P Label
138.43	3	0:81

Figure 3-16 LSR Label Distribution

This label request message, whose format is shown in Figure 3-17 propagates along a number of LSRs until it reaches the egress for that FEC. A LSR determines it is the egress for that FEC if any of the following conditions apply:

If its next hop is not MPLS aware, or

If the operation is to remove the label and examine the network layer header,

If the destination address matches the same subnet as one that is directly connected to one of its interfaces.

Upon receipt, a LSR determines if it has previously sent a Label Request message for that FEC to its next hop. Figure 3-18 illustrates a flow diagram of the process a LSR undergoes on receiving a Label Request message.

0	Label Request (0x401)		Message Length
Message ID			
0	0	FEC TLV	Length
Prefix (2)		Address Family	PreLen
Prefix			

FEC TLV - FEC for which a label is being requested
FEC
Prefix - An address prefix encoded according to the Address Family field.
Address Family - encoding of the address family for the address prefix in the Prefix field.
PreLen - Length of the address prefix that follows in bits.
Prefix - An address prefix encoded according to the Address Family field

Figure 3-17 Label Request TLV

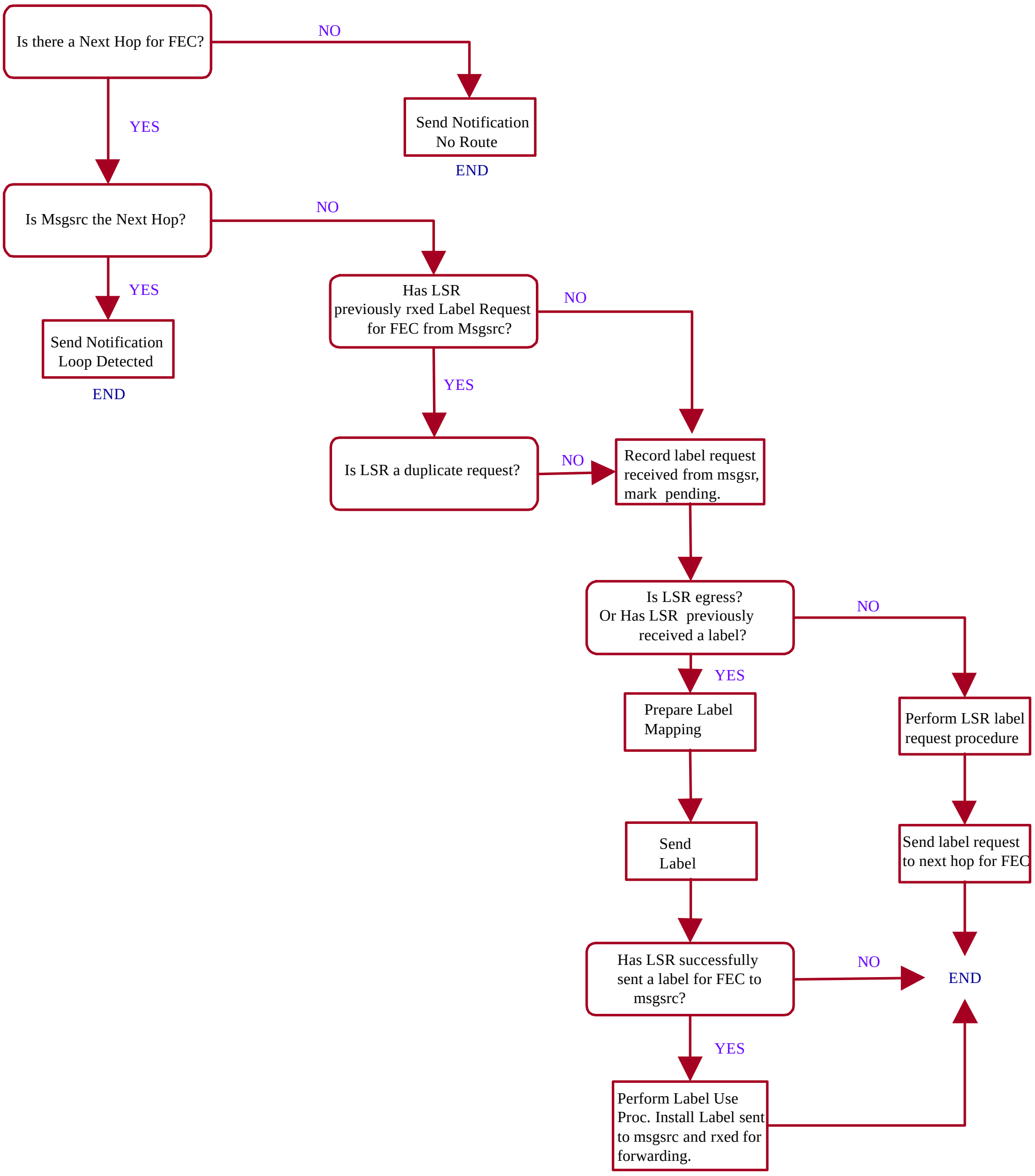
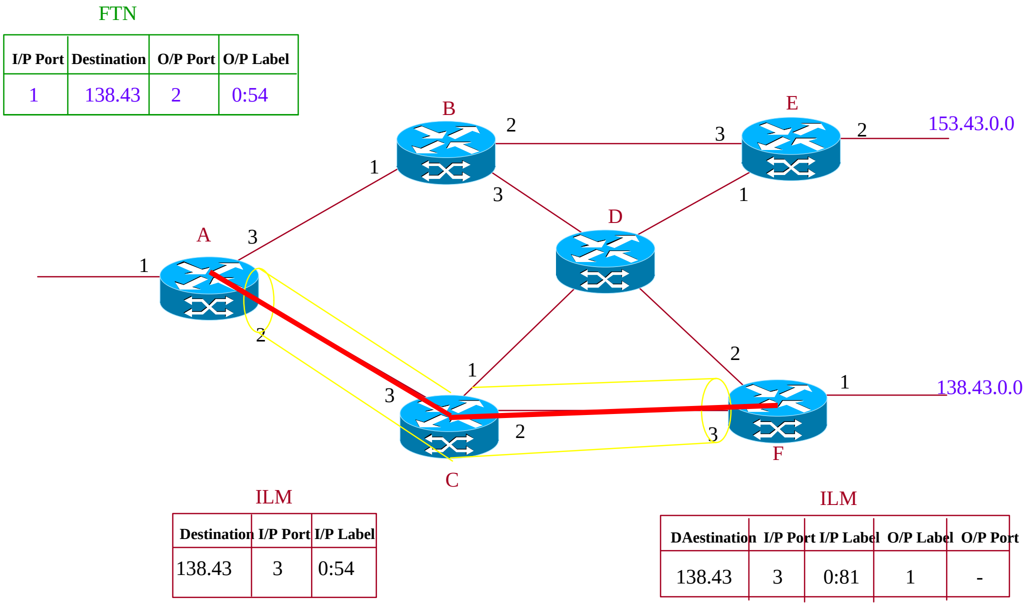


Figure 3-18 Flow Diagram of the Label Request Procedures

<table>
<tr><td>0</td><td colspan="3">Label Mapping (0x400)</td><td colspan="2">Message Length</td></tr>
<tr><td colspan="6">Message ID</td></tr>
<tr><td>0</td><td>0</td><td colspan="2">FEC (0x100)</td><td colspan="2">Length</td></tr>
<tr><td colspan="3">Prefix (2)</td><td colspan="2">Address Family</td><td>PreLen</td></tr>
<tr><td colspan="6">Prefix</td></tr>
<tr><td>0</td><td>0</td><td colspan="2">Generic Label</td><td colspan="2">Length</td></tr>
<tr><td colspan="6">Label</td></tr>
</table>

Generic Label - used to encode labels for use on links for which label values are independent of the underlying link technology. E.g., PPP and Ethernet.
Label - 20 bit label value

Figure 3-19 Label Mapping message format



FTN

I/P Port	Destination	O/P Port	O/P Label
1	138.43	2	0:54

ILM

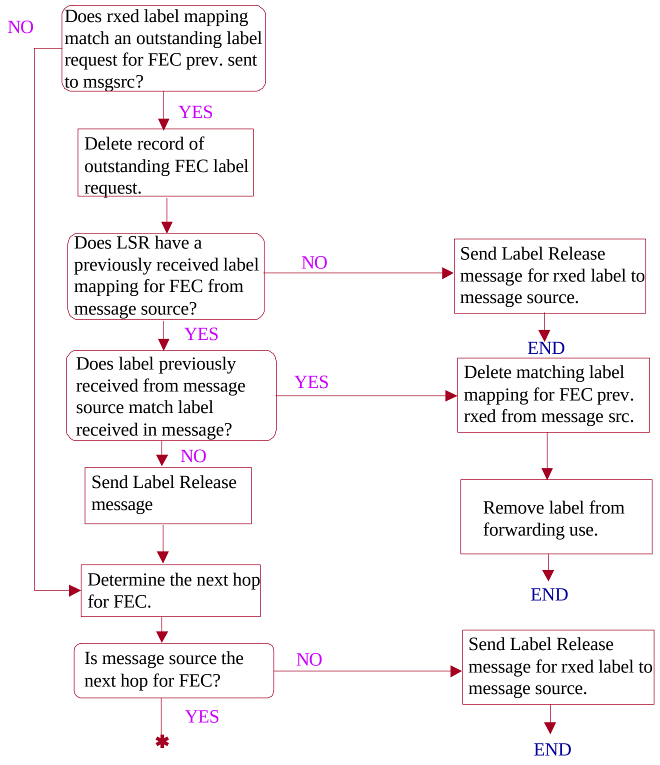
Destination	I/P Port	I/P Label
138.43	3	0:54

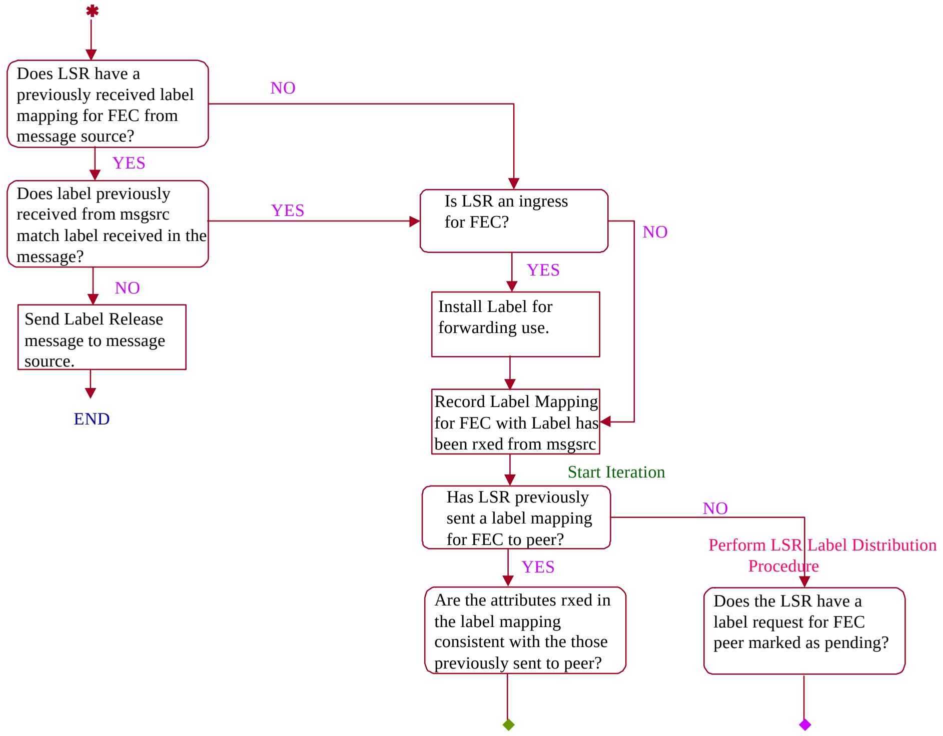
ILM

DAestination	I/P Port	I/P Label	O/P Label	O/P Port
138.43	3	0:81	1	-

Figure 3-20 LSP Established

A flow diagram of the process a LSR undergoes on receiving a Label Mapping message is shown in Figure 3-21.


NO
Does rxed label mapping match an outstanding label request for FEC prev. sent to msgsrc?
YES
Delete record of outstanding FEC label request.
Does LSR have a previously received label mapping for FEC from message source?
NO
Send Label Release message for rxed label to message source.
END
YES
Does label previously received from message source match label received in message?
YES
Delete matching label mapping for FEC prev. rxed from message src.
NO
Send Label Release message
Remove label from forwarding use.
END
Determine the next hop for FEC.
Is message source the next hop for FEC?
NO
Send Label Release message for rxed label to message source.
YES
✱
END


✱
Does LSR have a previously received label mapping for FEC from message source?
NO
YES
Does label previously received from msgsrc match label received in the message?
YES
Is LSR an ingress for FEC?
NO
NO
YES
Send Label Release message to message source.
Install Label for forwarding use.
END
Record Label Mapping for FEC with Label has been rxed from msgsrc
Start Iteration
Has LSR previously sent a label mapping for FEC to peer?
NO
Perform LSR Label Distribution Procedure
YES
Are the attributes rxed in the label mapping consistent with the those previously sent to peer?
Does the LSR have a label request for FEC peer marked as pending?

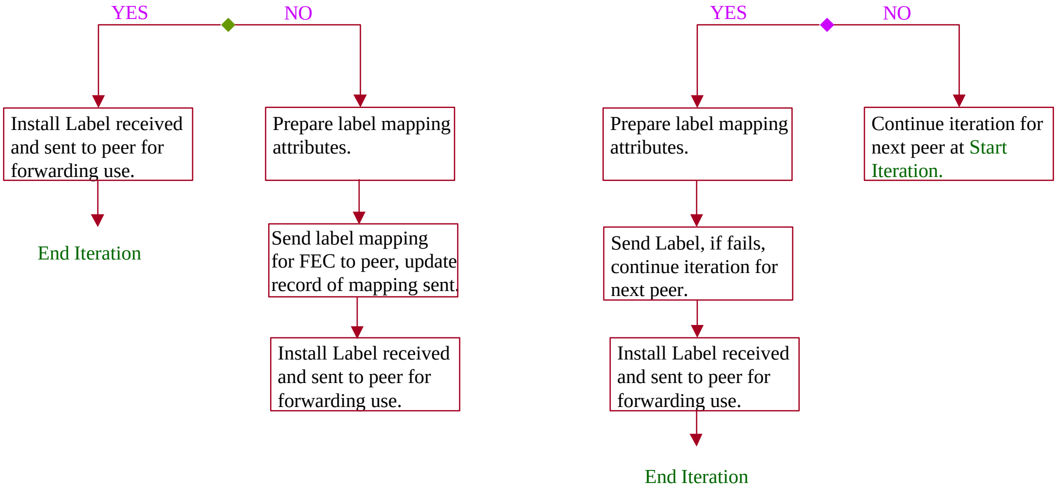

YES
NO
Install Label received and sent to peer for forwarding use.
End Iteration
Prepare label mapping attributes.
Send label mapping for FEC to peer, update record of mapping sent.
Install Label received and sent to peer for forwarding use.
YES
NO
Prepare label mapping attributes.
Send Label, if fails, continue iteration for next peer.
Install Label received and sent to peer for forwarding use.
End Iteration
Continue iteration for next peer at Start Iteration.

Figure 3-21 Flow Diagram of the Label Mapping Procedures

The Label Mapping message is used by an LSR to distribute a label mapping for a FEC to an LDP peer. If an LSR distributes a mapping for a FEC to multiple LDP peers, it is a local matter whether it maps a single label to the FEC, and distributes that mapping to all its peers, or whether it uses a different mapping for each of its peers.

An LSR receiving a Label Mapping message from a downstream LSR for a Prefix or Host Address FEC element should not use the label for forwarding unless its routing table contains an entry that exactly matches the FEC element.

3.2.6.1 Label Request Procedures

The Request message is used by an upstream LSR to explicitly request that the downstream LSR assign and advertise a label for a FEC.

A LSR may transit a Request message under any of the following conditions:

The LSR recognises a new FEC via the forwarding table, and the next hop is an LDP peer, and the LSR does not have a mapping from the next hop for the FEC.

The next hop to the FEC changes, and the LSR does not have mapping from that next hop for the given FEC.

The LSR receives a Label Request for a FEC from an upstream LDP peer, the FEC next hop is an LDP peer, and the LSR does not have a mapping from the next hop.

The receiving LSR should respond to a Label Request message with a Label Mapping for the requested label or with a Notification message indicating why it cannot satisfy the request.

When the FEC for which a label is requested is a Prefix FEC element or a Host Address FEC element, the receiving LSR uses its routing table to determine its response. Unless its routing table includes an entry that exactly matches the requested Prefix or Host Address, the LSR must respond with a No Route Notification message.

The message ID of the Label Request message serves as an identifier for the Label Request transaction. When the receiving LSR responds with a Label Mapping message, the mapping message must include the message ID of the Label Request message.

3.3 MPLS Traffic Engineering Mechanisms

Traffic Engineering within MPLS arises from the network operators' requirement to provide a network infrastructure that is dependable and offers consistent network performance. Traffic Engineering allows network operators the ability to re-route traffic flows away from the "least cost" path calculated by routing protocols and onto potentially less congested physical paths through the network. As a result of the unprecedented growth in demand for network resources and the competitiveness amongst providers', traffic engineering has become the primary application for MPLS. The goal of traffic engineering is to efficiently use the limited network resources such that no individual component i.e., router or a link is overutilised or underutilised [SEM2000b].

The ability of MPLS to support explicit routes, operate over any media infrastructure and to be able to collect statistics regarding LSPs, suggests it is well suited to provide traffic engineering capabilities.

The IETF have proposed two different protocols for reserving resources within MPLS namely, Constraint-Based routing, using LDP (CR-LDP) [JAM99a], and the Resource ReSerVation Protocol (RSVP) with extensions [AWD2000], to cater for traffic engineering within an MPLS domain.

The route for a given LSP can be established in two ways, control driven (i.e., hop-by-hop LSP), or explicitly routed (ER-LSP). When setting up a hop-by-hop LSP, each LSR determines the next interface to route the LSP based on its layer 3 routing topology database, and sends the label request to the L3 next hop. When setting up an ER-LSP, the route for the LSP is specified in the "setup" message itself, and this route information is carried along the nodes the setup message traverses. All the nodes along the ER-LSP will follow the route specification and send the label request to the next indicated interface. While the hop-by-hop LSP follows the path that normal layer 3 routed packets will take, the ER-LSP can be specified and controlled by the network operators or network management applications to direct the network traffic, independent of the L3 topology. In this way ER-LSP may be used to achieve network traffic engineering.

MPLS also provides flexibility for network operators to manage their traffic with both strict and loose explicit routing. In the case of a strict ER-LSP, a network operator specifies the exact full route (nodes and interfaces) that the ER-LSP will traverse. Loose ER-LSPs allow some flexibility for routing and rerouting options, and minimises configuration overhead. In addition, a loose segment can be adaptive by moving to a new route according to the changes incurred in the layer 3 routing table. However, this kind of route change is not always desirable due to stability and control requirements of the operators. In this case, the loose segment provides a "pinning" mechanism, meaning that an alternative route will only be tried when failure happens. The following subsections describe how RSVP and CR-LDP can be used in MPLS networks to reserve resources.

3.3.1 RSVP

Classical RSVP as specified in RFC 2205 [BRA97], allows routers the flexibility to retain their connectionless transport behaviour, however RSVP has scalability issues when the number of sessions increases in the network as stated in RFC 2208 [MAN97]. To enable RSVP to be deployed within an MPLS environment, the existing protocol needs to be augmented. RSVP protocol messages are augmented with new objects to support label allocation, distribution and binding, along with explicit routes. Notable changes have been introduced to the existing RSVP protocol infrastructure including modification to the "soft state" mechanism; where messages are sent periodically to maintain the path, and the refresh mechanisms amongst others to enable RSVP to support ER-LSPs, all of which have been documented in [AWD2000]. Figure 3-22 illustrates the flow of RSVP messages in establishing a LSP.

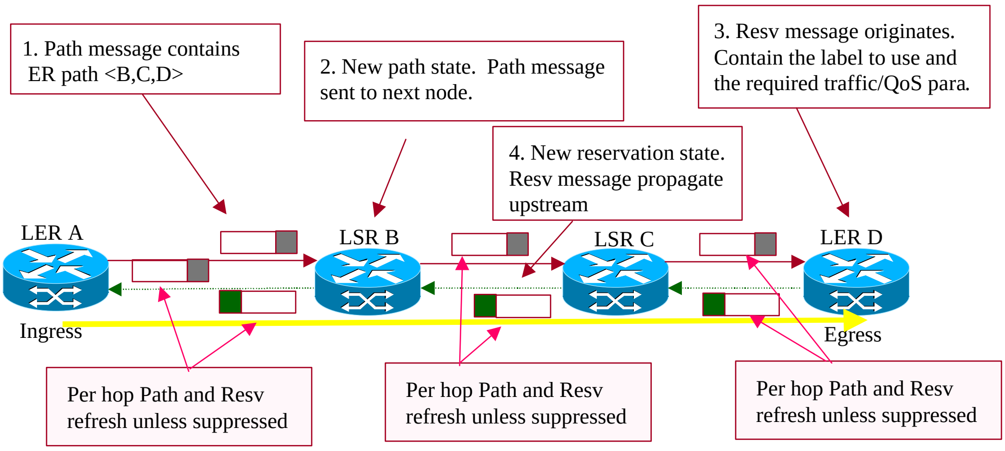

1. Path message contains ER path <B,C,D>
2. New path state. Path message sent to next node.
3. Resv message originates. Contain the label to use and the required traffic/QoS para.
4. New reservation state. Resv message propagate upstream
LER A
LSR B
LSR C
LER D
Ingress
Egress
Per hop Path and Resv refresh unless suppressed
Per hop Path and Resv refresh unless suppressed
Per hop Path and Resv refresh unless suppressed

Figure 3-22 Extensions to RSVP to establish an ER-LSP

3.3.2 CR-LDP

CR-LDP has its foundations in the existing LDP protocol, and is extended to incorporate the explicit route information. An explicit route is represented in a Label Request message as a list of nodes along a constraint-based route. If the requested path can satisfy the resources required, labels are allocated by means of Label Mapping messages. Further details of CR-LDP procedures and features can be found in [JAM99a]. Figure 3-23 illustrates the flow of messages when CR-LDP is used to establish a LSP.

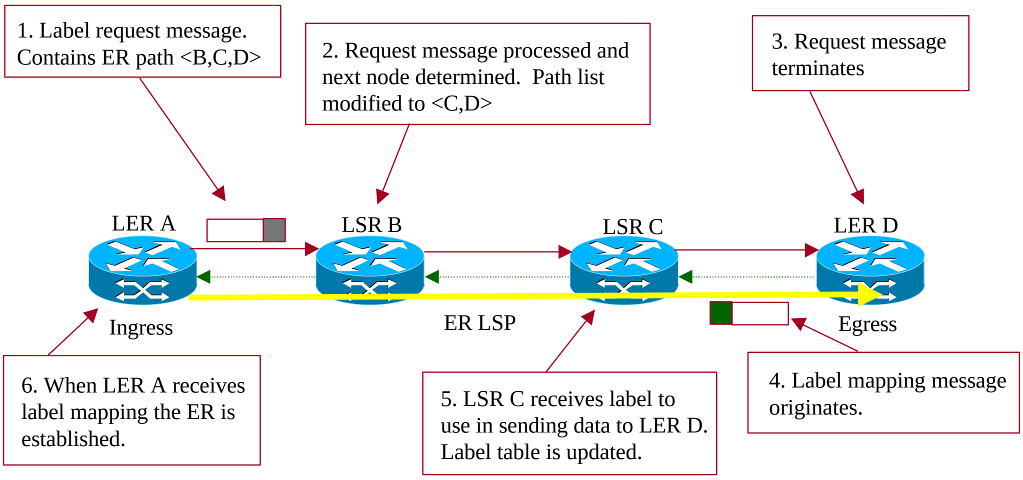

1. Label request message. Contains ER path <B,C,D>
2. Request message processed and next node determined. Path list modified to <C,D>
3. Request message terminates
LER A
LSR B
LSR C
LER D
Ingress
ER LSP
Egress
6. When LER A receives label mapping the ER is established.
5. LSR C receives label to use in sending data to LER D. Label table is updated.
4. Label mapping message originates.

Figure 3-23 CR-LDP Path Establishment

3.3.3 Comparative Analysis of RSVP and CR-LDP

CR-LDP is a part of LDP and employs the same mechanisms and messages as LDP for discovery, session, establishment, maintenance, label distribution and error handling. Enabling LDP/CR-LDP to provide network providers with a unified distribution and path setup mode for MPLS, thus maximising operational efficiency.

RSVP with the appropriate extensions is able to operate in the downstream on demand label allocation mode. However, if other MPLS modes are required, i.e., downstream unsolicited, then both LDP and RSVP protocols must be present in the network. This adds complexity and has a negative impact on the planning and operational costs. Another disadvantage of this situation is the need to manage more than one network, and the objective of MPLS was to move away from that.

CR-LDP uses the reliable transport of TCP, so error notification messages are delivered in a an orderly manner. RSVP runs on raw IP transport and cannot guarantee fast failure notification, as a result of this traffic may not be re-routed until the 'clean up timeout' interval has expired which is undesirable in communication networks [LI2000].

CR-LDP uses “hard state” controlled paths that scale well as the number of ER-LSPs increases in the network. This is because unlike the soft state case, once a path is set up there are no additional messages needed to maintain the path, keeping the number of messages needed to establish, maintain and release ER-LSPs to a minimum, thus allowing CR-LDP to scale well. RSVP on the other hand has a scalability problem, as documented in RFC 2208. As the number of paths through a node increases, the number of soft-state refresh messages to maintain the paths also increases. As documented in RFC 2208 the computational requirement on the routers increase proportionally with the number of sessions.

In summarising, CR-LDP is an open standard protocol, proposed and accepted by the IETF MPLS working group and also the ITU SG13 [SMI99]. It does not depend on other protocols that are outside the range of the MPLS WG, thus providing a few advantages. It can be enhanced to adopt new network requirements, and it promotes interoperability as documented in [LI2000]. In contrast, RSVP extensions have not yet shown a clear solution for interoperability in the networks. In terms of traffic engineering, technically, both CR-LDP and RSVP provide similar signalling functionality. However as noted by [BRI2000], major modifications to make RSVP applicable for traffic engineering has reduced its feasibility in a MPLS network. Only Cisco is a strong proponent of the RSVP with extensions approach. As such, this thesis focuses on the former signalling mechanism. Figure 3-24 illustrates the principle difference between RSVP and CR-LDP.

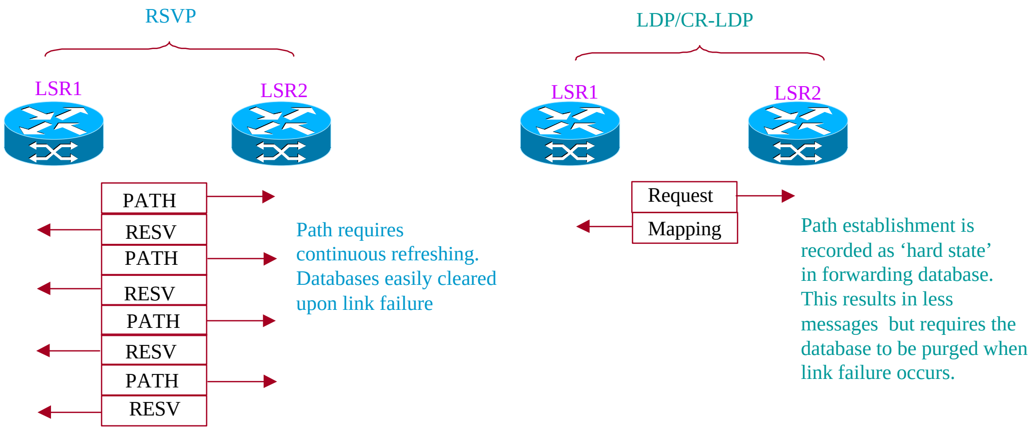
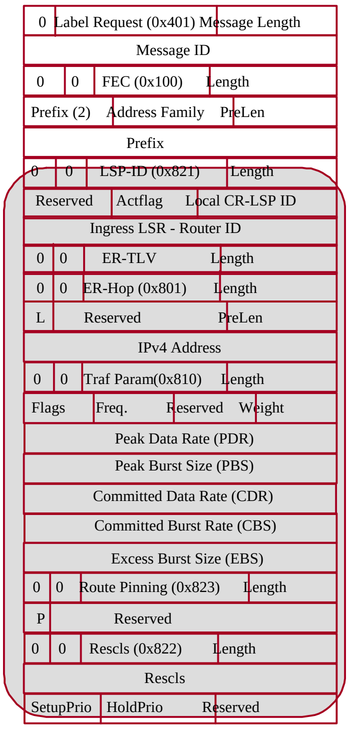

RSVP
LSR1
LSR2
PATH
RESV
PATH
RESV
PATH
RESV
PATH
RESV
Path requires continuous refreshing. Databases easily cleared upon link failure
LDP/CR-LDP
LSR1
LSR2
Request
Mapping
Path establishment is recorded as ‘hard state’ in forwarding database. This results in less messages but requires the database to be purged when link failure occurs.

Figure 3-24 Fundamental difference between RSVP and CR-LDP

3.3.4 CR-LSP Establishment and Maintenance

Similar data structures used in establishing and maintaining LSPs are also used in CR-LSP establishment [JAM99a]. Traffic flows destined for a particular destination are assigned to a FEC that is encoded as a LSP expressed as a series of labels. Once a label has been received by the ingress it is treated as if were received for a packet requiring no special constraints.

A request at the ingress LSR to set up a CR-LSP might originate from a management system or an application, the details may be implementation specific. The ingress LSR uses information provided by the management system or the application, possibly together with information from the routing database, to calculate the explicit route and to create the Label Request message. A CR-LSP is initiated by the ingress LSR along a pre-determined path. The route for a given LSP is explicitly routed as specified in the Label Request message, allowing the route information to be carried along the nodes the Label Request message traverses, from the ingress to the egress LER. The CR-LSP can be specified and controlled by the network operators or network management applications to direct the network traffic, independent of the layer 3 topology. Figure 3-25 illustrates the format of the Label Request message when it is used to set up a CR-LSP.



0 Label Request (0x401) Message Length
Message ID
0 0 FEC (0x100) Length
Prefix (2) Address Family PreLen
Prefix
0 0 LSP-ID (0x821) Length
Reserved Actflag Local CR-LSP ID
Ingress LSR - Router ID
0 0 ER-TLV Length
0 0 ER-Hop (0x801) Length
L Reserved PreLen
IPv4 Address
0 0 Traf Param(0x810) Length
Flags Freq. Reserved Weight
Peak Data Rate (PDR)
Peak Burst Size (PBS)
Committed Data Rate (CDR)
Committed Burst Rate (CBS)
Excess Burst Size (EBS)
0 0 Route Pinning (0x823) Length
P Reserved
0 0 Rescls (0x822) Length
Rescls
SetupPrio HoldPrio Reserved

LSP ID - a unique identifier of a CE-LSP within an MPLS network. It is composed of the ingress router ID and locally unique CR-LSP ID to that LSR.
Actflag - Action Indicator Flag that indicates the action that should be taken if the LSP already exists on the LSR receiving the message.
ER -Hop - IP address of LSR , L bit indicates whether this is a loose hop
Flag - 8 bit field indicating whether or not the traffic parameters are negotiable.
Freq. - the delay that may be introduced
Weight - determines the CR-LSP relative share of the possible excess bandwidth above its committed rate.
Peak Rate - maximum rate at which traffic should be sent to the CR-LSP, defined in terms of PDR + PBS.
Committed Rate - rate that the MPLS domain commits to be available to the CR-LSP, defined in terms of CDR+ CBS.
Excess Burst Rate - used to measure the rate at which the traffic sent on a CR-LSP exceeds the committed rate.
Route Pinning - applicable to segments of an LSP that are loosely routed, indicated by the L bit being set, P = 1, indicates route pinning is requested.
Rescls - the network operator may classify network resources in various ways. These classes are known as colours or administrative groups. When CR-LSP is being established, its necessary to indicate which resource class the CR-LSP can draw from.
SetupPrio / HoldPrio - specify the priority of the existing CR-LSP.

Figure 3-25 CR-LDP Label Request message format

When a LSR receives a Label Request message containing an Explicit Route (ER), it must determine the next hop for this path. Selection of this next hop may involve a selection from a set of possible alternatives. The LSR receiving the Label Request message must first evaluate the first ER-Hop. If the L bit is set in the first ER-HOP indicating this is a loose LSP (i.e., the next hop does not have to follows a strict route). If the node is not part of an abstract node (i.e., a collection of nodes described by a symbolic node) described by the first ER-Hop, it has received the message in error, and should return a Bad-Initial-ER-Hop error. If the L-bit is set and the local node is not part of the abstract node described by the first ER-Hop, the nodes selects a next hop that is along the path to the abstract node described by the first ER-Hop. If there is no first ER-Hop, the message is in error and the system should return a Bad-Explicit Routing error. If there is no second ER-Hop, this indicates the end of the explicit route. The explicit route TLV should be removed from the message.

When the node recognises itself as the egress for the CR-LSP, it returns a Label Mapping message that will traverse the same path as the LSR message in the opposite direction. The LSR receiving it determines its response to a Label Request message that is still pending. A Label Mapping message is generated and sent to the next upstream LSR. When the ingress node receives a Label Mapping the CR-LSP is set up.

3.3.4.1 CR-LDP Preemption

Each CR-LSP carries an LSP priority. This priority can be used to allow new LSPs to 'bump' or tear down, existing LSPs of lower priority in order to steal their resources. This is especially useful during times of failure and allows the network operator to rank the LSPs such that the most important obtain resources before less important LSPs. These are called the *setupPriority* and a *holdingPriority* and 8 levels are provided.

When an LSR is established its setupPriority is compared with the holdingPriority of existing LSPs, any with a lower holdingPriority may be bumped to obtain their resources. This process may continue in a domino fashion until the lowest holdingPriority LSPs either clear or their flows are reassigned onto less desirable routes.

However pre-emption in CR-LDP does not allow the operator to determine which connection can be torn down.

3.3.5 Establishing a CR-LSP to support Loss Sensitive Applications

The network operator and the customer first set up a service level agreement. Two parameters of interest are the Peak Data Rate (PDR) and the Peak Burst Size (PBS) which are chosen by the user based on its requirements and are included in the signalling message [JAM99a]. An indication of whether the parameter values are subject to negotiation is flagged. Also of importance is the need for the customer and the provider to agree on the actions to be enforced at the network edge. The specification of these actions may be part of the SLA, but is not included in the signalling message. To enforce the traffic contract between user and network, a policing function carried out in the edge device may be tag or drop packets that exceed the specified PDR and PBS specifications.

The signalling message is sent in the direction of the ER path and the LSP is established following the normal LDP procedures. Each LSR applies its admission control rules. If sufficient resources are not available and the parameter values are subject to negotiation, then the LSR could negotiate down the PDR, the PBS or both. The new parameter values are echoed back in the Label Mapping message. LSRs might need to re-adjust their resource reservations based on the new traffic parameter values.

3.3.6 Establishing a CR-LSR to support Loss Insensitive Applications

The user assigns values for the PDR, PBS, Committed Data Rate (CDR), and Committed Burst Size (CBS), along with the negotiation flag set to indicate whether or not the respective values are negotiable. As the service is loss insensitive the frequency is set to unspecified, i.e., there is no limit to the number of packets that can be buffered however there is a need to ensure the packets are not dropped. The actions to be enforced at the network edge are not included in the signalling message but are specified in the SLA between the user and provider [JAM99a]. The edge rules might include marking to indicate high discard precedence for packets that exceed CDR and CBS. The rules may also include dropping packets that do not conform to the PDR and PBS values. The ingress LER of the CR-LSP is expected to run its admission control rules and negotiate traffic parameters down if sufficient resources do not exist. The new parameter values are echoed back in the Label Mapping message. LSRs might need to re-adjust their resources based on the new traffic parameter values.

3.3.7 Summary

Chapter 3 has described the mechanisms and procedures that are encompassed by the MPLS protocol. It has explained how packets can be transported through a network along paths using metrics other than the ‘least cost path’. However, although the protocol allows for routes guaranteeing particular ‘constraints’, as of yet there is no provision to permit dynamic redistribution of traffic flows along under-utilised LSPs through the network during transient periods of congestion. The following Chapter 4 describes a scheme called Fast Acting Traffic Engineering (FATE) which solves the problem of re-mapping flows onto alternative paths or higher QoS streams during such busy periods.

Chapter 4: Functionality of Congestion Control Schemes

4.1 Motivation for Fast Acting Traffic Engineering

Considerable interest in congestion control through traffic engineering has arisen from the knowledge that although sensible provisioning of the network infrastructure is needed together with sufficient underlying capacity, these are not sufficient to deliver the QoS required [BOR98][[SEM2000b]]. This is due to dynamic variations in load resulting in transient surges in traffic converging on a particular network component. In operational IP networks, it has been difficult to incorporate effective traffic engineering due to the limited capabilities of the IP technology. In principle, Multiprotocol Label Switching (MPLS), a connection-oriented label swapping technology, offers new possibilities in addressing the limitations by allowing the operator to use sophisticated traffic control mechanisms i.e., load responsive routing.

However, as yet, the traffic engineering capabilities offered by MPLS have not been fully exploited. Once label switched paths (LSPs) have been provisioned through the service providers' network, there are currently no management facilities for dynamic re-optimisation of traffic flows. The service level agreements (SLAs) between the network operator and the customer are agreed in advance of the commencement of traffic flow, and these are mapped to particular paths throughout the provider's domain and may be maintained for the duration of the contract. During transient periods, the efficiency of resource allocation could be increased by routing traffic away from congested resources to relatively under-utilised links. Some means of restoring the LSPs to their original routes once the transient congestion has subsided is also desirable.

Today's network operators require the flexibility to dynamically renegotiate bandwidth once a connection has been set up [SEM2000a], [LI2000] preferably using automated solutions to manage an access switch management algorithm and route connections. Although these services are already provided to some extent with provisioning, they tend to occur relatively infrequently (several times in a day) using prior knowledge and manual intervention. There are currently no mechanisms in place within the network to allow the operator to rapidly change the traffic paths in response to transient conditions.

This thesis proposes Fast Acting Traffic Engineering (FATE) scheme that solves the problem of dynamically managing traffic flows through the network by re-balancing streams during periods of congestion. It proposes mechanisms and procedures that will allow label switched routers (LSRs) in the network to utilise mechanisms within MPLS to indicate when flows may be about to experience possible frame/packet loss and to react to it. Based upon knowledge of the customers' SLAs, together with instantaneous flow information, the label edge routers (LERs) can then instigate changes to the LSP route to circumvent congestion that would hitherto violate the customer contracts.

4.2 Traffic Engineering Context

Consider the scenario depicted below in Figure 4-1. It illustrates an example of an existing scheduling mechanism i.e.; WFQ with a number of buffers catering for traffic streams with different QoS constraints. The amount of time the scheduler devotes to each buffer is dependent on the loss probability threshold predefined for that QoS buffer and its current provisioned utilisation. This is a very simple demonstration of a variant of WFQ.

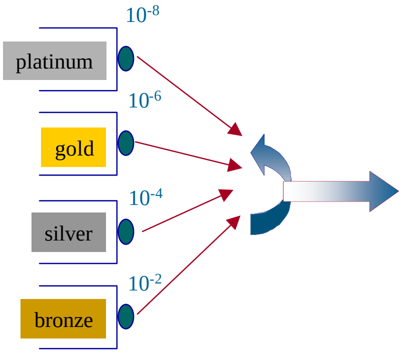


Figure 4-1 Buffer Configuration

4.2.1 Multi-Service Provisioning Environment

At present the Internet has a single class of service - "best effort". As a result of this single service all traffic flows are treated identically, there is no priority servicing regardless of the requirements of the traffic. There have been many attempts to addressing this issue as described previously in Chapter 2. An existing provisioning scheme that can be applied within an MPLS environment can be described as follows:

Consider Figure 4-2 which shows a scheduler at each egress port of a LSR. The scheduler has been programmed to visit each class-based buffer at a rate commensurate with the loading of that particular buffer and its identified QoS constraint(s) i.e., long-term guaranteed loss limit.

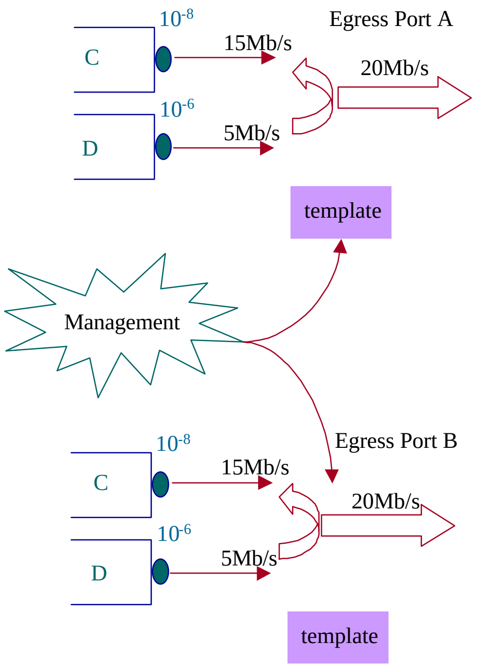


Figure 4-2 Multi-Service Provisioning

The order and frequency with which the scheduler services each of the buffers is determined by a port template that may be programmed by the management module and read by the scheduler. In the scenario depicted, where buffer C would be serviced three times more than buffer D, the template would take the format shown below in Figure 4-3.

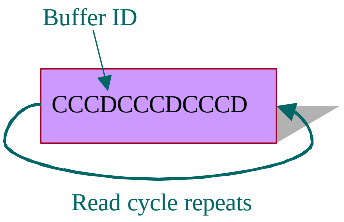


Figure 4-3 Example Scheduling Template

A scheduling template is programmed according to a predetermined loss probability threshold for each buffer for a given anticipated load. In a situation where the traffic loading through a buffer stream increases, the management function has the ability to change the template of the scheduler if deemed appropriate. The management function uses its knowledge of the traffic characteristics and the current loading to determine whether a new CR-LSP can be routed through a particular buffer. Whilst attempting to maintain the buffer's traffic engineering constraints below the specified loss probability. Over a specified time period the loss probabilities through each buffer stream are recorded and in the event that the loss exceeds a predefined threshold the management function may decide to alter the template i.e. the scheduler's rate, to accommodate for the additional loading.

Scheduling reallocation is permissible provided the contracted QoS requirements of the LSPs traversing the buffer(s) are not violated.

As this method is based on predicted behaviour (estimated over the last time period) it does not cater for transient fluctuations in load. It is a provisioning mechanism. The proposed scheme FATE on the other hand provides a means of dynamically redistributing existing CR-LSPs between buffers or via alternative paths in response to short term congestion events.

4.2.2 Traffic Flow Provisioning

So far basic provisioning has been considered i.e., mapping LSPs to buffers according to their particular QoS requirements and the long term loading situation. This mechanism allows for readjustment of the scheduling templates in response to predicted loading variations. It is relatively slow and operates on buffers – it does not provide granularity to respond to issues associated with individual LSPs.

FATE provides a vital fast acting mechanism. It allows individual LSPs to be dynamically remapped to QoS buffers providing a higher service class along a specified path in response to transient congestion situations.

Consider the situation where a request for the *silver* service class, introduced in Section 4.2 is made as depicted in Figure 4-4. On receipt of a Label Request each LSR decides:

- ♦ Whether it can allocate the bandwidth;
- ♦ Given the current situation would it be able to alter the scheduling template.

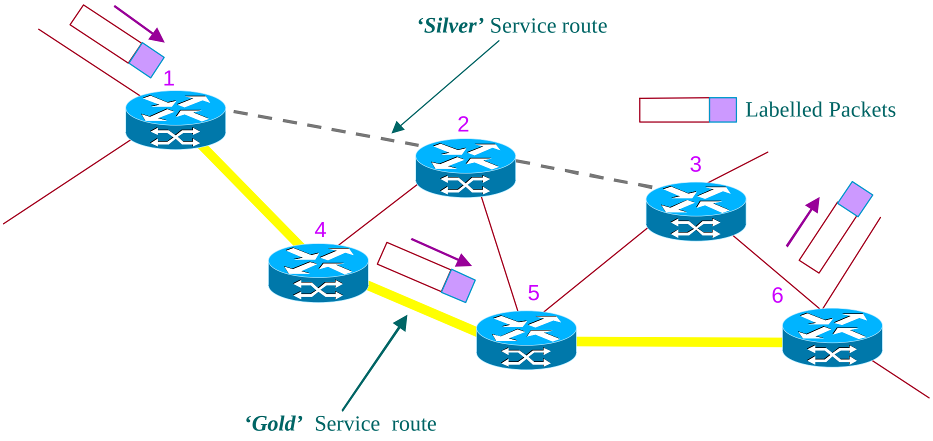


Figure 4-4 Traffic Establishment

If LSR 5 is not able to meet either one of those requirements it returns a notification message to the ingress LER, who on receipt may either decide to request a higher service class i.e., *gold* along the same LSP, or to select an alternative route. The buffer configuration for the situation previously considered is illustrated below in Figure 4-5. The LSRs within the MPLS domain are identically dimensioned[6] as shown previously in Figure 4-1. Each LSR has four buffer streams each catering for a minimum loss probability which equates to a specific service class e.g., *bronze*, *silver*, *gold and platinum*.

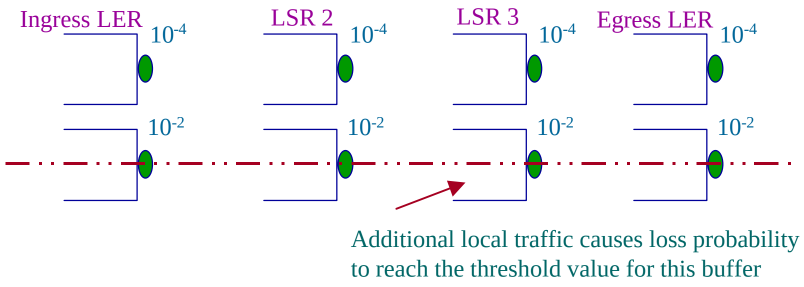


Figure 4-5 Identical buffer streams along a CR-LSP

A class selected at the ingress LER must then be supported by the same buffer streams within each LSR along the chosen CR-LSP as shown in Figure 4-5. Configuring the buffers in this way may lead to inefficiencies in situations of congestion. E.g., if LSR 3 receives a Label Request whose loss probability requirement it cannot cater for at that moment within its lower service class but that can be satisfied in a higher class, it has to be rejected. In this situation the ingress LER may decide to renegotiate for a higher service class. Thus allowing for a simplistic approach to QoS provisioning with the local congestion at LSR 3 resulting in a re-mapping of the CR-LSP along the entire path, as shown in Figure 4-6.

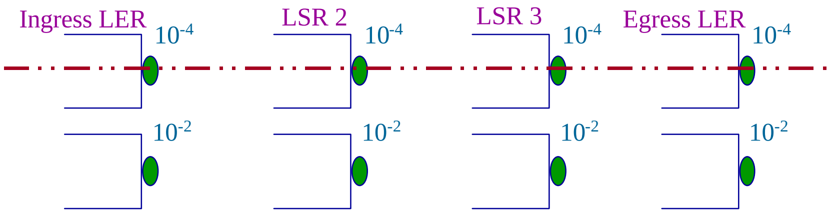


Figure 4-6 CR-LSP established through a higher service class

[6] This is in terms of their minimum loss probability threshold for a given service class.

However, in the situation where the congested LSR had the flexibility to temporarily map the traffic flows into a higher service class as illustrated in Figure 4-7 may result in a more complex QoS provisioning but offers a guaranteed service at a price[7].

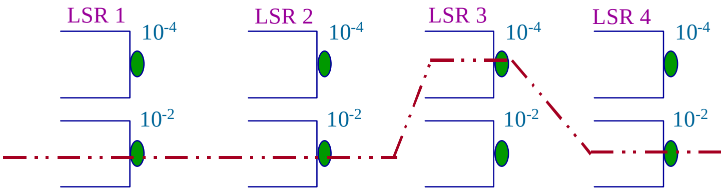


Figure 4-7 CR-LSP experiencing more than one type of service

The occurrence of this type of situation is not addressed in the FATE scheme, as the traffic flows would always experience the same type of service regardless of whether a particular LSR could cater for it in a lower buffer stream than another LSR along the route. However FATE+ described in Section 4.4 addresses the issue.

4.3 Description of the FATE Scheme

This section describes the mechanisms and procedures that are employed within the FATE (Fast Acting Traffic Engineering) scheme.

4.3.1 Congestion Detected in a CR-LSP

An ingress LER can determine the contribution it makes to the utilisation of the LSRs along each LSP, and can set up CR-LSPs[8] with that limited knowledge. However, it currently has no knowledge of how those same LSRs are being utilised by other LERs. It is this lack of information when deciding which LSP may meet an application's requirement that can lead to congestion occurring within a downstream LSR. This section proposes a fundamental set of novel mechanisms that can be employed in the effort to either pre-empt congestion or respond to its occurrence in a LSR along which a CR-LSP has already been established.

[7] Pricing is beyond the scope of this thesis, however researchers at QMWC have produced pricing and charging solutions that can be considered for this situation [MIA99].

[8] The terms CR-LSP and LSPs are used interchangeable, the main difference is that CR-LSPs are established based on constraints, e.g., explicit route constraints, QoS constraints, etc.

Assume there are a small number of LSPs maintained in an MPLS domain, such that the network provider is able to monitor them individually. These LSPs will typically be the aggregation of a number of individual connections or flows from a customer site[9].

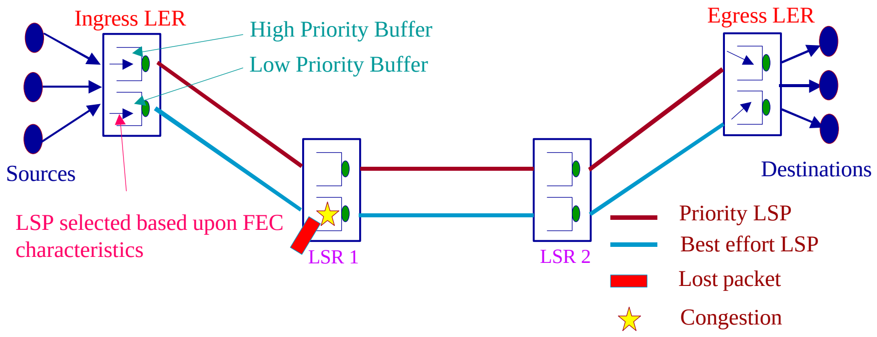


Figure 4-8 Congestion detected in the Best Effort Buffer

Consider Figure 4-8 it depicts traffic flowing between *sources* and *destinations* passing through two buffers at each LSR, one catering for *priority* traffic, i.e., loss stringent applications, and one for *best effort* traffic, i.e., ftp or email. Priority traffic would be assigned the gold or silver service class and best effort the bronze service class. Although the diagram shows an individual LSP from a source, that could quite easily represent aggregate LSPs from a number of customer sites. There are typically many other LSPs passing through each buffer but this cross traffic is omitted for simplicity in this illustration. Although hop by hop routing could be used when setting up LSPs, for traffic engineering purposes there is a need to ensure the traffic follows a specified path[10]. Explicit routing is used in the route selection as this allows the ingress LER to specify all the LSRs along the LSP. The sequence of LSRs followed by an explicitly routed LSP is chosen by the ingress LER using topological information learned from a link state database[11] in order to compute the entire path pending at the egress LER. Additional to the path selection process the FATE scheme allows the ingress LER to identify the class of LSP and so enable it to be directed through the appropriate buffers at each LSR. With reference to Figure 4-8 assumes that over time, as a result of the increased load through LSR1, it starts to lose packets from the best effort LSP. The author proposes the buffers[12] be configured as shown in Figure 4-9 and that each LSR periodically calculates the packet loss in the individual LSPs passing through each input buffer stream. If the value exceeds a given

[9] It is highly unlikely that the network provider would be able to manage and control a large number of connections or flows individually.

[10] Not always the "least cost "path as indicated by routing protocols i.e., OSPF.

[11] MPLS requires the presence of a separate routing algorithm to construct the link state database.

[12] Although the figure shows two buffers, it is possible to increase the number within a LSR.

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 that LSR.

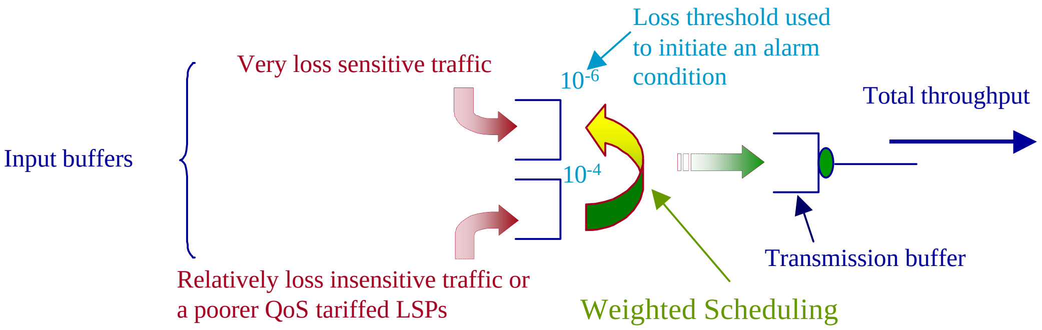


Figure 4-9 QoS Buffer Configuration

At the input buffer the number of packets belonging to the best effort LSP and the number of packets destroyed due to failed insertion are recorded. They are used to calculate the packet loss in the LSP just before the packets are forwarded to the output buffer, whose sole purpose is to act as a transmission buffer through which no loss occurs, as illustrated in Figure 4-10.

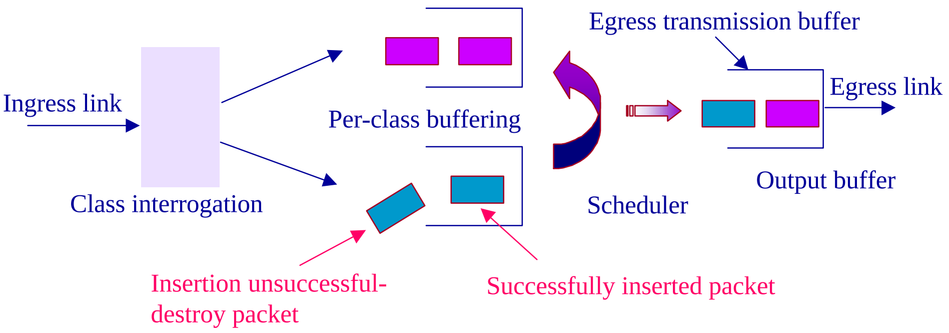


Figure 4-10 Packet Loss Calculated at the input buffers

Once the packet loss in the best effort LSP has risen above the predetermined threshold value, for an extended time period the LSR creates a LDP notification message containing the proposed *Congestion Indication* TLV with the format shown in Figure 4-13. In order to determine what format the TLV should take, there is a need to determine what information needs to be forwarded and to whom. The aim of sending the *Congestion Indication* notification (CIN) message is to indicate to the ingress LER that there are packets being lost from a particular CR-LSP originating from it, allowing the ingress LER to either:

1.Decide that the packet loss it is currently experiencing remains sufficiently low for it to continue to meet its SLA requirements, allowing/permitting no further action to be taken at this time.
2.Renegotiate for new quality requirements along the existing LSP[13].
3.Negotiate for new quality requirements along an alternative LSP.

Table 4-1 Decisions made by LER on receiving a Congestion Indication Notification Message

Figure 4-11and Figure 4-12 depict flow diagrams showing how congestion is detected and the congested LSR's response to it.

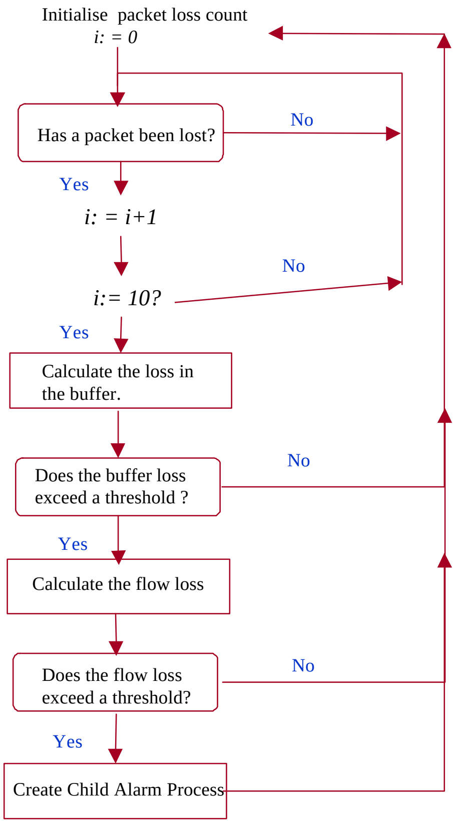

Initialise packet loss count
i: = 0
Has a packet been lost?
No
Yes
i: = i+1
No
i:= 10?
Yes
Calculate the loss in the buffer.
Does the buffer loss exceed a threshold ?
No
Yes
Calculate the flow loss
Does the flow loss exceed a threshold?
No
Yes
Create Child Alarm Process

Figure 4-11 LSR Monitoring Parent Congestion Detection

[13] Request that the LSP be promoted to pass through a higher priority buffer along the same path, and within the same LSR.

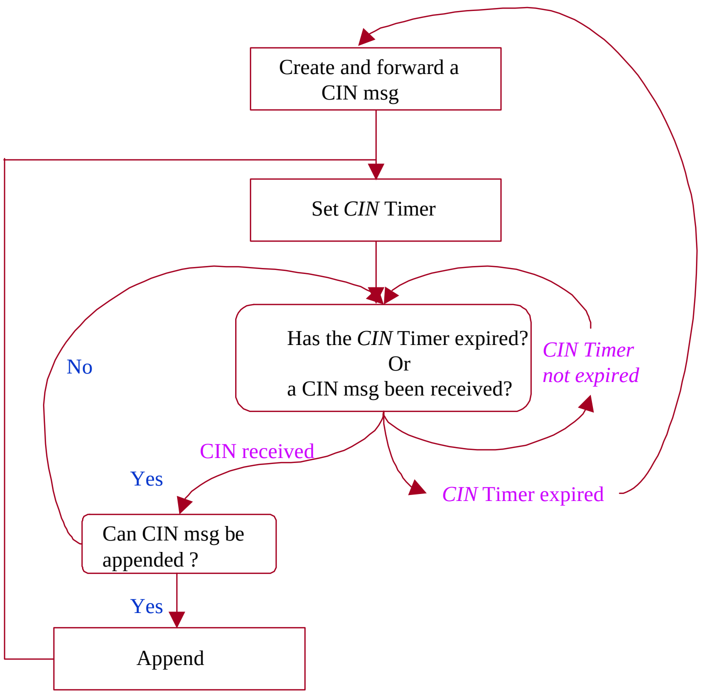

Create and forward a CIN msg
Set CIN Timer
Has the CIN Timer expired? Or a CIN msg been received?
CIN Timer not expired
No
CIN received
Yes
CIN Timer expired
Can CIN msg be appended ?
Yes
Append

Figure 4-12 LSR Child Alarm Process

In order for the ingress LER to act on the received information, it needs to know the following:

- ♦ The identity of the LSP that is experiencing congestion.
- ♦ The current loss in the buffers the LSP is traversing.[14]
- ♦ The LSRs this loss is occurring in.
- ♦ The current loss the LSP is experiencing.

0	Congestion Indication(0x500)		Msg Length
Message ID			
0	0	LSP-ID TLV	Length
Reserved	ActFlag		Local CR-LSP
Ingress LSR Router ID			
0	0	Congestion Modification TLV	Length
LSR Router ID			
Strm	Loss Buffer Currently Experiencing		
Loss LSP Currently Experiencing			

LSR Router ID - The IP address of the LSR experiencing congestion.
Strm - Index of the buffer stream experiencing loss.
Loss Buffer Currently Experiencing - Loss the buffer stream given, is currently experiencing.
Loss Flow Currently Experiencing - Loss the LSP identified by the Local CR-LSP value is currently experiencing.

Figure 4-13 Congestion Indication TLV

[14] Although each buffer and its servicing scheduling are dimensioned for a specific CLP, at any time due to traffic loading the current available packet loss within the buffer may have increased or decreased.

As a result of this information, the congested LSR generates a *CIN* message. This must contain the identity of the LSR that is experiencing loss, the identity of the CR-LSP along which packet loss was detected, and the packet loss the LSP and buffer are currently experiencing. The congested LSR uses the input port that the packet was received on and the input MPLS label, as an index into the NHLFE table to obtain the CR-LSP ID. The CR-LSP ID identifies the *Ingress LSR Router ID* i.e., originating LER, and the local value assigned by that ingress LER to identify the CR-LSP initiated by it. The buffer the LSP traverses through at the congested LSR is obtained using the CR-LSP ID to index a separate *Buffer Table* whose role is explained in Section 4.3.3.1. The LSR's own IP address is included in the message along with the current packet loss both the LSP and the buffer are experiencing. The *CIN* message is then forwarded to the next hop LSR towards the ingress LER.

Rather than all congested LSRs always generating CIN messages, intermediate LSRs upon receipt of a *CIN* message may append relevant information to it concerning their status if they are also experiencing congestion. If a LSR receives a *CIN* message shortly after sending one, it checks the *Congestion Indication Table* whose format is shown in Table 4-2, to see if the timer it has set has expired. If it has not expired, it will simply forward the message without appending its own information, otherwise it will include its information before forwarding. Figure 4-14 depicts the flow diagram of the behaviour of a LSR on receiving a CIN message.

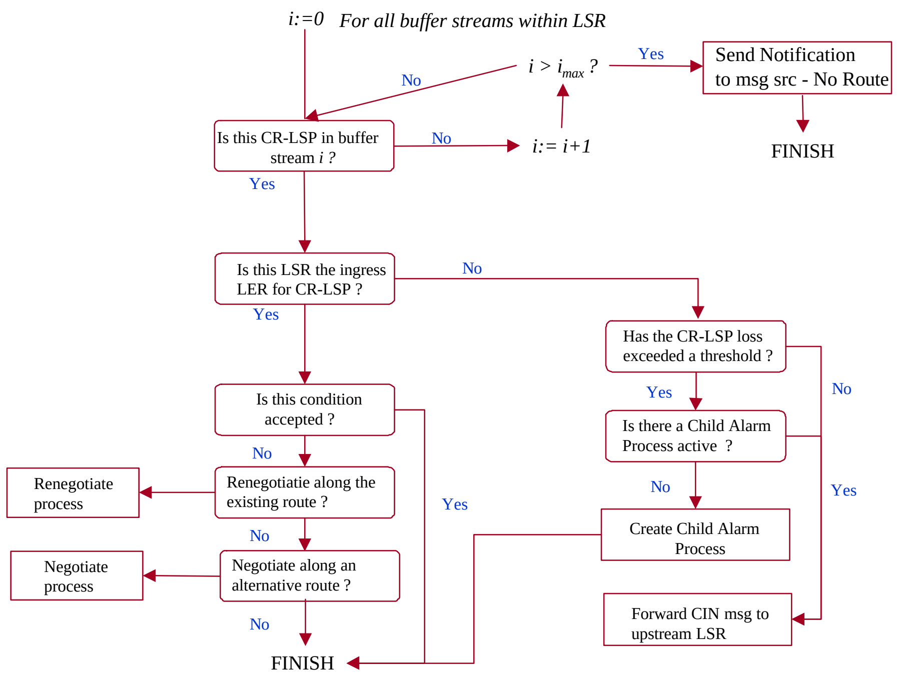

i:=0
For all buffer streams within LSR
i > i_max ?
Yes
Send Notification to msg src - No Route
No
FINISH
Is this CR-LSP in buffer stream i ?
No
i:= i+1
Yes
Is this LSR the ingress LER for CR-LSP ?
No
Yes
Has the CR-LSP loss exceeded a threshold ?
Yes
No
Is this condition accepted ?
Is there a Child Alarm Process active ?
No
No
Yes
Renegotiate process
Renegotiatie along the existing route ?
Yes
Create Child Alarm Process
No
Negotiate process
Negotiate along an alternative route ?
No
Forward CIN msg to upstream LSR
FINISH

Figure 4-14 LSR CIN Message Receipt

Timers are used to control the responsiveness of the FATE scheme to traffic loading transients. For example, when a LSR is congested it can issue a *CIN* message. In doing so it sets a retransmission timer. It is not permitted to issue another message until the timer expires, thus avoiding signalling storms whilst improving the robustness of the protocol. Alternatively, if it receives a *CIN* message on route to the ingress LER from another congested LSR, it can simply append its own congestion information and set the timer accordingly. In doing so, avalanches of congestion notification messages towards the ingress LER are prevented. In addition, stability is improved by averaging the observed traffic parameters at each LSR and employing threshold triggers.

Each LSR experiencing congestion records in its *Congestion Indication Table* the CR-LSP ID, and the current LSP and buffer loss probability, as shown in Table 4-2. A timer is set. If when the timer expires, if the LSR is still suffering from congestion, the LSR will send another *CIN* message with the updated calculated loss values and reset the timer.

Buffer Stream	***CR-LSP ID***		***Current Loss in CR-LSP***	***Current Loss in Buffer***	***Timer***
	Ingress ID	***Value***			
1	138.37.32.75	2	3.78e-03	5.63e-04	▤

Table 4-2 Congestion Indication Table

Associated with each *Congestion Indication* TLV is a *Congestion Modification* TLV parameter as highlighted in Figure 4-13. It allows the congested LSR formulating the *CIN* message to include the current losses experienced in the specified LSP and buffer.

When the ingress LER receives a *CIN* message, it may do any of the actions previously outlined in Table 4-1.

The motivation behind monitoring individual LSPs through a particular buffer stream stems from the ingress LER's need to ensure the SLAs between the customers and the MPLS network are maintained at all times. To enable it to do this, it needs to have knowledge of the loss encountered by the individual LSPs originating from it. Individual LSPs from a customer site are aggregated into LSPs that share class based buffer resources. As a result of this, the LSP loss rather than the individual flow losses is reported back to the ingress LER, who has knowledge of which flows are affected via the flow/LSP binding information.

By monitoring both the losses experienced by individual LSPs and buffer streams, it gives the ingress LER two averages to consider when deciding whether to renegotiate QoS requirements along an existing path or a different path, or whether to accept the current condition. For example, consider when an ingress LER receives an indication that the loss in a buffer its LSPs are passing through is experiencing a particularly poor loss probability (1 in 10^{-}

^{2}). However the loss probability experienced by the buffer it is traversing (1 in 10^{-5}) is acceptable.

The ingress LER has may decide to set a number of CIN messages it should receive before responding to the congestion notification.

Some means of averaging the loss statistics provides a useful dampening factor. To prevent an avalanche of CIN messages being sent to a single ingress LER, the congested LSR when it determines that more than one CR-LSP traversing its buffers is experiencing a particularly poor loss probability, will aggregate the CIN messages for those individual buffers.

4.3.2 Scalability

Monitoring losses in individual LSPs is not very scalable, even if those LSPs represent the aggregation of individual connections or flows from a customer site. It is quite possible that at any instance in time, a LSR could be expected to handle a very large number i.e., thousands of these LSPs. As a result of this scalability issue, detecting losses in individual LSPs described previously, may not be a viable option in an MPLS domain expected to maintain a large volume of LSPs[15]. This immediately poses two questions.

How is it possible for an autonomous MPLS network to apply congestion control mechanisms in a situation were it has numerous flows, some of which may be entering the domain just after exiting a customers premises, and others on route from or to another autonomous domain?

How can this service provider ensure the customers SLA is met whilst traversing this network?

In monitoring a single LSP or a number of LSPs that connect between a specific source and destination, connected within a single autonomous system, it is quite easy to identify the ingress and egress LERs, and the exchange of messages can be easily handled under the control of the operator.

Consider the case when the source and destination are not within the same domain, or where the MPLS domain is as an intermediary transport 'pipe'. It is not possible or desirable for the operator to determine the absolute source and destination of each LSP.

The author proposes assigning a *Virtual Source/Virtual Destination* (VS/VD) [SPO98] pair for the aggregation of LSPs entering the domain at one point and exiting at another point, using label stacking or tunnelling within the autonomous MPLS domain of interest.

[15] However [SPO98], explains how operation and maintenance (OAM) cells are used in ATM for fault management and network performance on a point to point connection basis, thus implying it is possible to monitor a large number i.e., thousands of flows or connections.

All LSPs arriving at a particular ingress LER and exiting at a particular egress LER are assigned to a FEC. The ingress LER also known as the virtual source, is the entry point to the MPLS domain and it is at this point that an additional label is ‘pushed’ onto the label stack, and used to ‘tunnel’ the packet across the network. On arriving at the egress LER, also known as the virtual destination, the label is ‘popped’ and the remaining label used to forward the packet.

By employing label stacking within the domain and assigning VS/VD pairs, the issue of scalability is removed whilst allowing the operator control of the LSPs traversing its network. It allows for efficient utilisation of the limited network resources and the additional capability of controlling congestion. With the VS/VD paradigm, the congestion control messages need only propagate along as far as the virtual source for the ingress LER and to the virtual destination for the egress LER. With the virtual endpoints of the LSP defined, aggregation of many LSPs can be treated as an individual LSP as described previously Section 4.3.1.

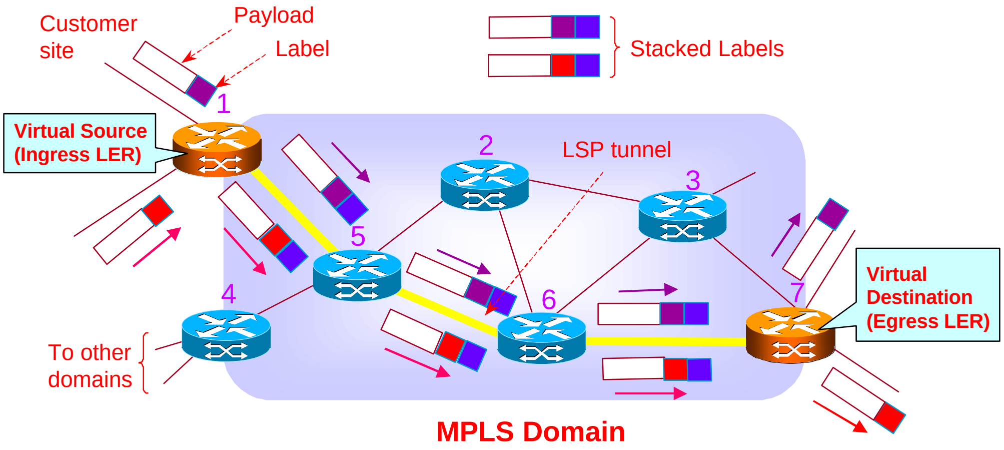


Figure 4-15 Using Label Stacking to confine traffic engineering mechanisms to a particular MPLS domain and permit aggregation

Figure 4-15 illustrates an example of a LSP entering an MPLS domain managed by a single operator. At LSR 1, the virtual source, each packet has a label ‘pushed’ unto its label stack and is transported across the network via LSRs 5 and 6. On arriving at LSR 7, the virtual destination, the label is ‘popped’ and the packet is forwarded based on the next label in the stack. Figure 4-16 illustrates similar class traffic initially assigned to different LSPs, 1 and 2 being mapped to the same LSP 3, by ‘pushing’ an identical additional label onto their respective label stacks.

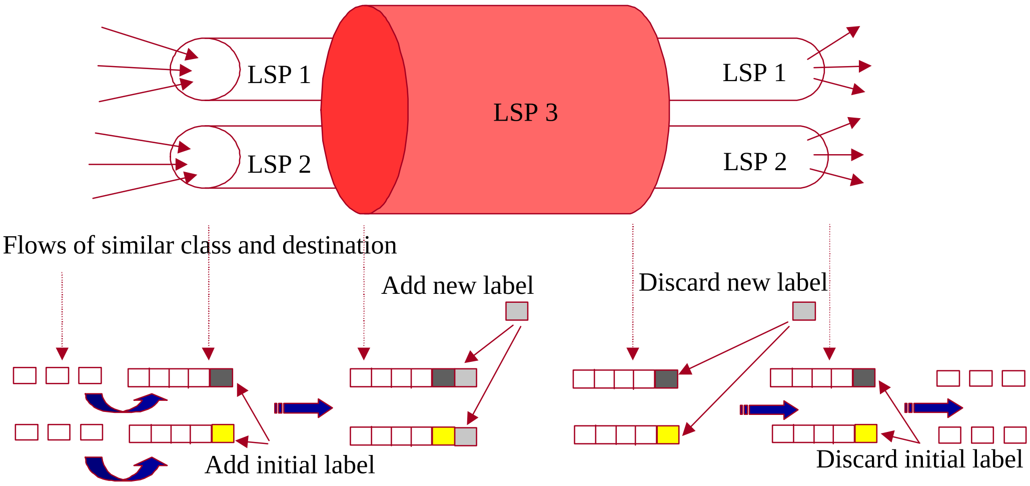


Figure 4-16 LSP Aggregation of similar class flows

4.3.3 Renegotiation

This section describes the procedures that occur during renegotiation, and the various proposed data structures that are required, namely the *Buffer Table* and the *Buffer Requirement Table*.

4.3.3.1 Buffer Table

The *Buffer Table* is maintained by each LSR and used on receiving a Label Request message containing modified traffic parameters. The LSR uses the received CR-LSP ID within the Label Request message, to index an entry within the table, to determine what the initial negotiated parameters were. A *Buffer Table* contains the following information:

- CR-LSP ID (index)
- Buffer stream identity
- Bandwidth reserved for LSP (current and requested)
- Current Loss probability negotiated for the LSP

An example of a *Buffer Table* is shown in Table 4-3. When the CR-LSP is initially established, there is only one entry held in the Bandwidth Reserved field within the *Buffer Table*, i.e. *Current*. During the renegotiation process it is possible that there may be two entries within this field, one for the initial value negotiated i.e., *Current* and the second for the value temporarily assigned, i.e. *Requested*. Once all the LSRs along a CR-LSP have agreed to the negotiated parameter value, the second entry will then be made permanent and assigned to the *Current* field, and the entry will no longer be marked as pending, as explained below in Section 4.3.3.3.

CR-LSP ID		***Buffer Stream***	***Bandwidth Reserved***		***Loss Probability***	***Pending***
Ingress ID	***Value***					
138.37.32.75	5	2	***Current***	***Requested***[16]	***Current***	🗐
			0.5Mb/s	3Mb/s	2.5e-05	

Table 4-3 Buffer Table

4.3.3.2 Buffer Requirement Table

The *Buffer Requirement Table* is maintained by each LSR within an MPLS domain. It contains the following information about individual buffer streams within the LSR through which the LSPs are mapped. As each LSP contains an aggregation of flows this approach is scalable.

- ♦ Buffer stream identity
- ♦ Total bandwidth assigned to buffer stream
- ♦ Available Bandwidth
- ♦ Current Average Loss Probability[17]

The table is used by each LSR to determine whether it can allocate the additional resources on receiving a Label Request message detailing bandwidth requirements and a minimum acceptable loss probability. Table 4-4 shows an example of a typical *Buffer Requirement Table.* In an environment where an MPLS domain supports both CR-LSP's requiring the reservation of bandwidth and LSP's requiring a best effort service, the operator may reserve a portion of its total bandwidth to the requests requiring a best effort service.

Buffer Stream	***Total Bandwidth***	***Available Bandwidth***	***Current Average Loss Probability***
1	10Mb/s	5.0Mb/s	3.5e-05

[16] When the CR-LSP is initially established this value is undefined.

[17] This value is updated periodically.

2	5Mb/s	2.75Mb/s	2.5e-04

Table 4-4 Buffer Requirement Table

4.3.3.3 Renegotiation Procedures

On receiving a *CIN* message the ingress LER extracts the following information: CR-LSP ID that encodes the *Ingress LSR Router ID* and a locally assigned value *Local CR-LSP*, from the LSP-ID TLV. The LSR Router ID experiencing loss, and the value of packet loss the LSP and buffer are currently experiencing, from the *Congestion Modification* TLV. The *Ingress LSR Router ID* along with the *Local CR-LSP* identifies that this message has been received by the correct ingress LER. With this information the ingress LER is able to identify the particular LSP and its traffic parameters.

The ingress LER needs to determine whether it should renegotiate along an existing LSP for a higher buffer stream offering improved servicing or whether it should negotiate for a new LSP route. The decision depends on information gathered from Statistical Control messages explained later in Section 4.3.4. Figure 4-17 illustrates LSPs being switched[18] from an existing path 🕙, on to either a higher buffer stream along the same path 🕚 or along an alternative path 🕛. On determining that a particular buffer stream is suffering loss that exceeds the predetermined threshold, a *CIN* message is sent to the ingress ☜, who upon receipt decides upon remedial action.

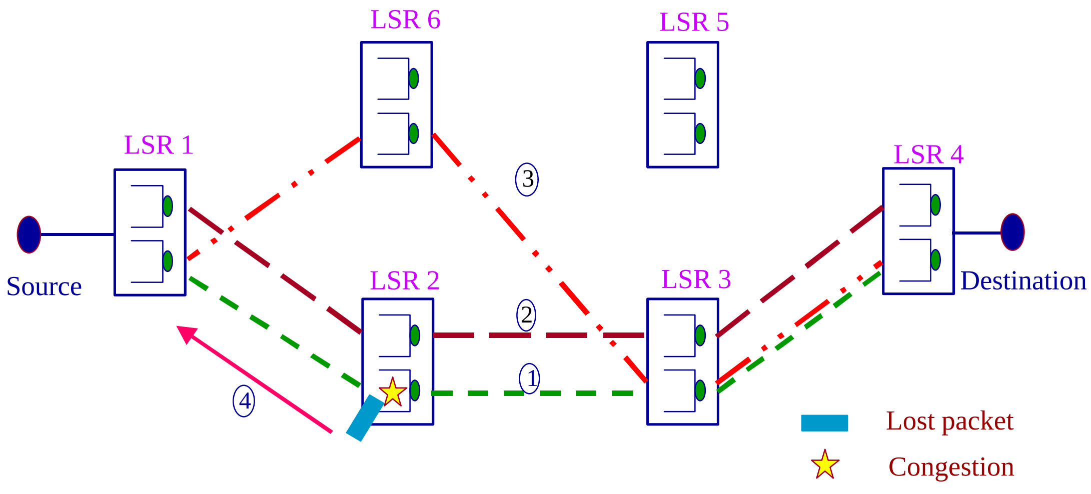


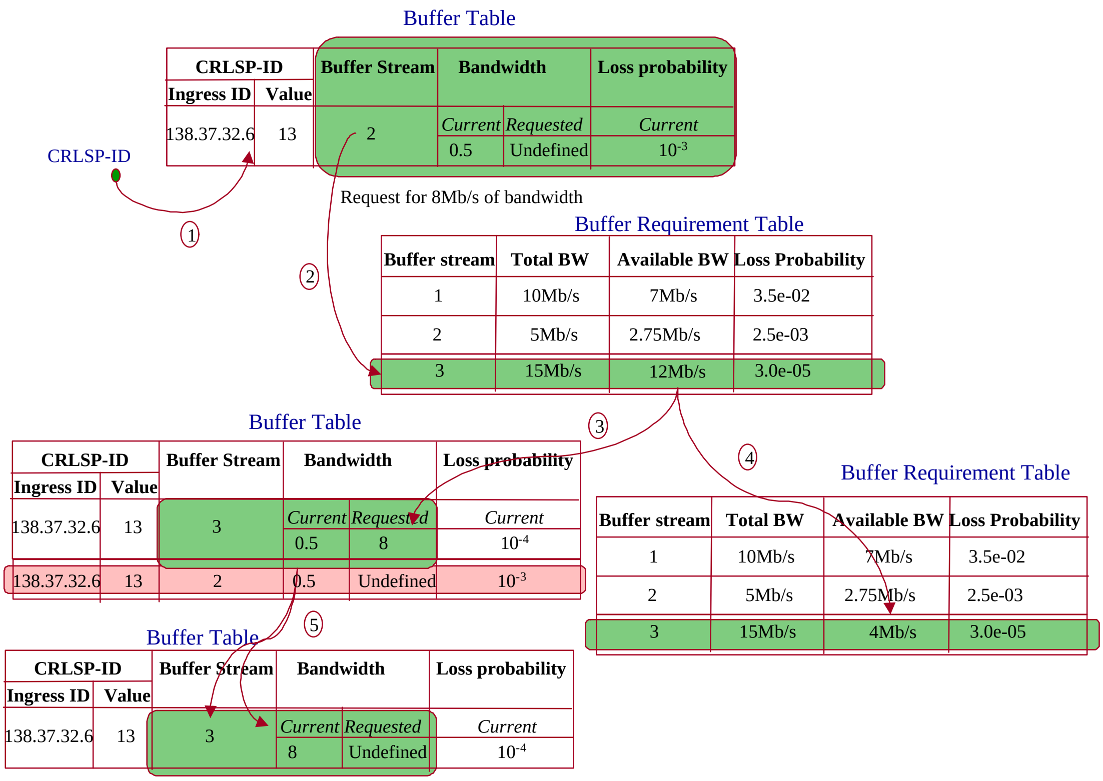
Figure 4-17 Renegotiation Process

[18] Strictly speaking the LSP is not switched – it is the flows it contains that are switched.

4.3.3.3.1 *Renegotiation along an Existing LSP within a Higher Buffer Stream*

If the ingress LER decides to renegotiate along an existing path for a higher service class, it will carry out the following procedure: The ingress LER formulates a Label Request message with the ActFlag set, to indicate that this is an existing CR-LSP along which the traffic parameters need to be modified. The Label Request message contains the newly requested modified traffic parameters along with the service class it requires. When each LSR receives a Label Request message it uses the globally unique CR-LSP ID as an index into the *Buffer Table* as shown in Figure 4-18 to determine which buffer stream the CR-LSP traverses 🕙, the amount of bandwidth initially reserved i.e. *Current*, and the loss probability assigned to that CR-LSP identified by the CR-LSP ID.

The LSR then chooses a higher buffer stream to the one the CR-LSP currently traverses 🕚. It then determines whether it can allocate the bandwidth and the minimum loss probability requested within a higher buffer stream. If it can, it temporarily assigns that amount in the new buffer stream to the *Requested* field in the *Buffer Table* 🕛, whilst maintaining the original entry. It alters the available bandwidth within the *Buffer Requirement Table* ↲ and forwards the Label Request message to the next hop.



CRLSP-ID		Buffer Stream	Bandwidth		Loss probability
Ingress ID	Value				
			Current	*Requested*	*Current*
138.37.32.6	13	2	0.5	Undefined	10^{-3}

Buffer stream	Total BW	Available BW	Loss Probability
1	10Mb/s	7Mb/s	3.5e-02
2	5Mb/s	2.75Mb/s	2.5e-03
3	15Mb/s	12Mb/s	3.0e-05

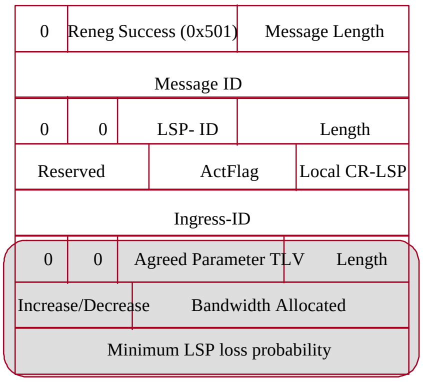
CRLSP-ID		Buffer Stream	Bandwidth		Loss probability
Ingress ID	Value				
			Current	*Requested*	*Current*
138.37.32.6	13	3	0.5	8	10^{-4}
138.37.32.6	13	2	0.5	Undefined	10^{-3}

Buffer stream	Total BW	Available BW	Loss Probability
1	10Mb/s	7Mb/s	3.5e-02
2	5Mb/s	2.75Mb/s	2.5e-03
3	15Mb/s	4Mb/s	3.0e-05

CRLSP-ID		Buffer Stream	Bandwidth		Loss probability
Ingress ID	Value				
			Current	*Requested*	*Current*
138.37.32.6	13	3	8	Undefined	10^{-4}

Figure 4-18 Bandwidth Allocation within a higher buffer stream

If all the LSRs along the CR-LSP are able to meet the requirement on receipt the egress LER will create a LDP notification message containing a proposed *RenegSuccess* TLV (Figure 4-19) indicating the resources have been reserved and send it to the upstream LSR towards the ingress LER of the CR-LSP.



Increase/Decrease - 0 = decrease in bandwidth,
1 = increase in bandwidth.
Bandwidth Allocated - The change in bandwidth agreed.
Minimum LSP loss probability - The minimum loss probability negotiated for this LSP.

Figure 4-19 Reneg Success TLV

On receiving a *RenegSuccess* notification message each LSR will permanently assign the resources to the path, by overwriting the value held in the *Current* field in the *Buffer Table* with the value held in the *Requested* field and remove the original entry ↳. The *RenegSuccess* notification message is then passed upstream. On receipt of a *RenegSuccess* message the ingress LER updates the FEC/label binding to reflect the higher buffer stream through which the CR-LSP will now be routed.

The *Reneg Success* notification message includes the CR-LSP ID, along which the parameters have been agreed[19] on, in terms of bandwidth required and minimum LSP loss probability. The flow diagram depicting the events that occur on receipt of a *RenegSuccess* notification message is shown below in Figure 4-20.

[19] This document assumes the bandwidth is controlled by the operator by possibly using policing and shaping mechanisms, but these mechanisms are beyond the scope of this thesis.

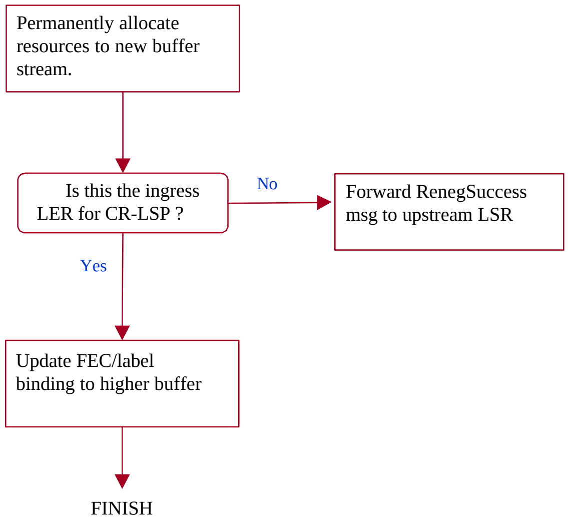

Permanently allocate resources to new buffer stream.
Is this the ingress LER for CR-LSP ?
No
Forward RenegSuccess msg to upstream LSR
Yes
Update FEC/label binding to higher buffer
FINISH

Figure 4-20 LSR RenegSuccess Receipt Procedure

If a LSR cannot allocate the additional resource it will send a proposed *RenegFailure* TLV within a notification message (Figure 4-21) to the message source and not propagate the Label Request message any further. The LSR will append to the *RenegFailure* notification message the maximum current available bandwidth it can allocate within each of its buffer streams that are also capable of meeting the minimum loss probability requested.

<table>
<tr><td>0</td><td colspan="2">Reneg Failure TLV</td><td colspan="2">Message Length</td></tr>
<tr><td colspan="5">Message ID</td></tr>
<tr><td>0</td><td>0</td><td>LSP- ID</td><td colspan="2">Length</td></tr>
<tr><td colspan="2">Reserved</td><td colspan="2">ActFlag</td><td>Local CR-LSP</td></tr>
<tr><td colspan="5">Ingress-ID</td></tr>
<tr><td>0</td><td>0</td><td>Negotiated Parameter</td><td colspan="2">Length</td></tr>
<tr><td>Strm</td><td colspan="4">Maximum Bandwidth Available</td></tr>
</table>

Maximum Bandwidth Available- The maximum amount of bandwidth available whilst meeting the minimum loss probability.
Strm - The buffer stream that is capable of offering the maximum bandwidth

Figure 4-21 Reneg Failure TLV

On receipt of a *RenegFailure* notification message, the LSR will deduce that another LSR further downstream has been unable to allocate resources for a LSP which traverses one of its own buffers. A flow diagram illustrating the events that occur on receipt of a *RenegFailure* notification message are shown below in Figure 4-22.

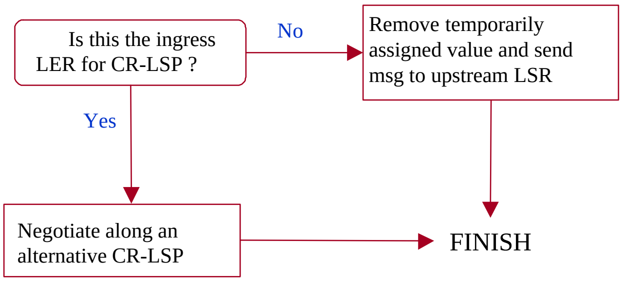
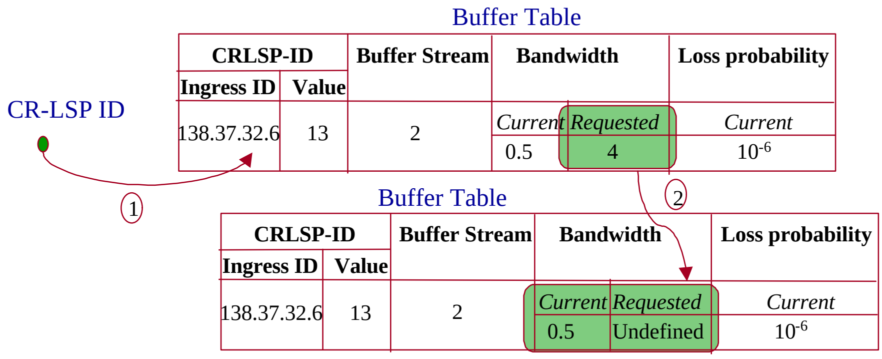

Is this the ingress LER for CR-LSP ?
No
Remove temporarily assigned value and send msg to upstream LSR
Yes
Negotiate along an alternative CR-LSP
FINISH

Figure 4-22 LSR RenegFailure receipt procedure

Using the received CR-LSP ID as an index into its *Buffer Table*, the LSR will find the relevant entry ①. The value held in the *Requested* field is removed ②, leaving the original negotiated value in the *Current* field, as shown in Figure 4-23.

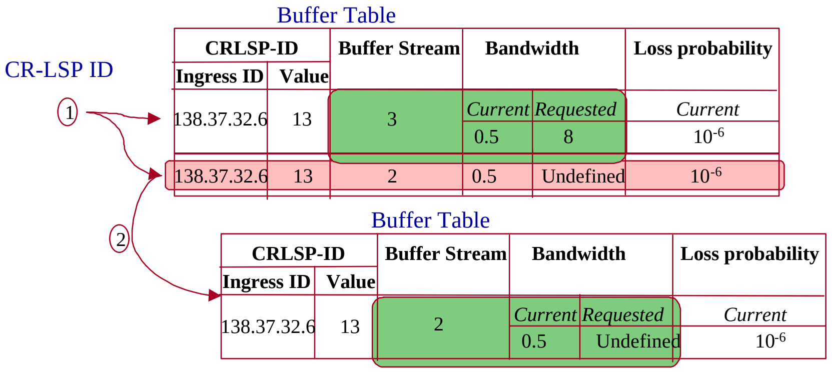


CRLSP-ID		Buffer Stream	Bandwidth		Loss probability
Ingress ID	Value				
138.37.32.6	13	2	*Current* 0.5	*Requested* 4	*Current* 10^{-6}

CRLSP-ID		Buffer Stream	Bandwidth		Loss probability
Ingress ID	Value				
138.37.32.6	13	2	*Current* 0.5	*Requested* Undefined	*Current* 10^{-6}

Figure 4-23 Updating Buffer Table on receipt of a Reneg Failure message

As the request was for a higher buffer stream, there will be more than one entry, with the most recent entry at the top of the table, as shown in Figure 4-24. The entry is removed leaving the original parameters ✸. The *Buffer Requirement Table* is updated accordingly.

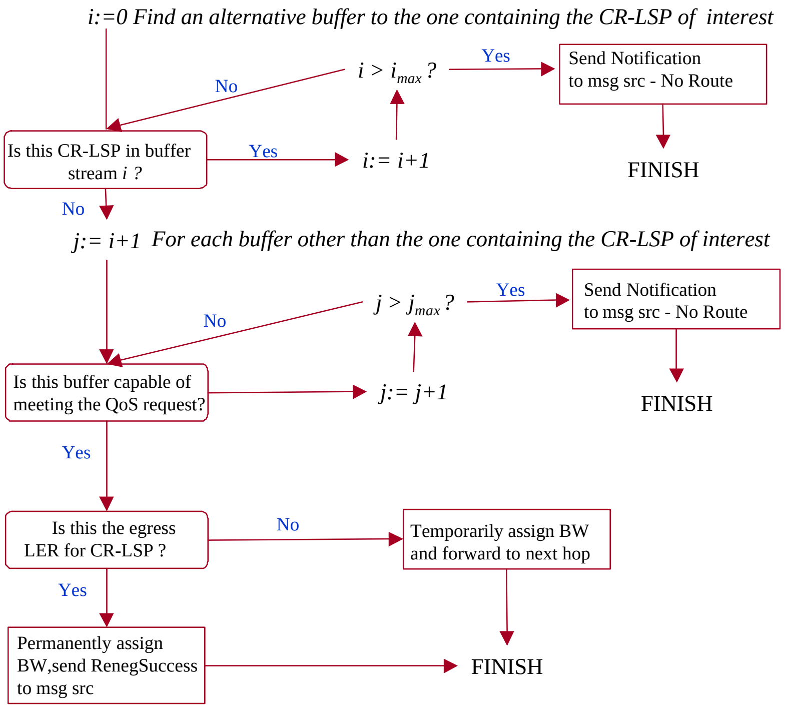


CRLSP-ID		Buffer Stream	Bandwidth		Loss probability
Ingress ID	Value				
138.37.32.6	13	3	*Current* 0.5	*Requested* 8	*Current* 10^{-6}
138.37.32.6	13	2	0.5	Undefined	10^{-6}

CRLSP-ID		Buffer Stream	Bandwidth		Loss probability
Ingress ID	Value				
138.37.32.6	13	2	*Current* 0.5	*Requested* Undefined	*Current* 10^{-6}

Figure 4-24 Removing entry from Buffer Table on receipt of a Reneg Failure message

The ingress LER on receiving a *RenegFailure* message will realise renegotiation along the existing path has failed for that CR-LSP and decide on remedial action. The flow diagram charting the events that occur when renegotiating along an existing CR-LSP is shown below in Figure 4-25.



i:=0 Find an alternative buffer to the one containing the CR-LSP of interest
$i > i_{max}$?
Yes
Send Notification to msg src - No Route
No
Is this CR-LSP in buffer stream *i* ?
Yes
$i := i+1$
FINISH
No
$j := i+1$ *For each buffer other than the one containing the CR-LSP of interest*
$j > j_{max}$?
Yes
Send Notification to msg src - No Route
No
Is this buffer capable of meeting the QoS request?
$j := j+1$
FINISH
Yes
Is this the egress LER for CR-LSP ?
No
Temporarily assign BW and forward to next hop
Yes
Permanently assign BW,send RenegSuccess to msg src
FINISH
FINISH

Figure 4-25 LSR Renegotiation along an existing CR-LSP

The author proposes two new TLV parameters to be used along with the *RenegSuccess* and *RenegFailure* TLV: an *Agreed Parameter* TLV parameter, and a *Negotiated Parameter* TLV parameter. The corresponding message formats are shown in Figure 4-26 and Figure 4-27 respectively.

<table>
<tr><td>0</td><td>0</td><td>Agreed Parameter</td><td>Length</td></tr>
<tr><td colspan="2">Increase/Decrease</td><td colspan="2">Bandwidth Allocated</td></tr>
<tr><td colspan="4">Minimum LSP loss probability</td></tr>
</table>

Figure 4-26 Agreed Parameter TLV

The *Agreed Parameter* TLV includes the traffic parameters that have been allowed by the LSR, in terms of either an increase or decrease in bandwidth and the minimum LSP loss probability. These values are returned to the ingress LER to allow for the case where the ingress LER has specified a range of acceptable values in the Label Request message.

The *Negotiated Parameter* TLV includes the single maximum bandwidth currently available from the network.

0	0	Negotiated Parameter	Length
Strm	Maximum Bandwidth Available		

Figure 4-27 Negotiated Parameter TLV

Figure 4-28 shows the exchange of messages in the case were all the LSRs are able to meet the additional resources. A Label Request message is sent along the LSP from the ingress LER to egress LER ① were all the LSRs are able to allocate the additional resources. At the egress LER a *RenegSuccess* notification message is sent to the ingress LER ② via all the intermediate LSRs. On receipt on a *RenegSuccess* notification message the temporary reservations are now made permanent. Unlike traditional hop-by-hop LSP establishment, the CR-LSP Label Request is not acknowledged hop-by-hop whilst it successfully traverses the intended path. The acknowledgement is initiated from the far-end egress LER and confirms the successful establishment of the path on a hop-by-hop basis until the ingress LER is reached.

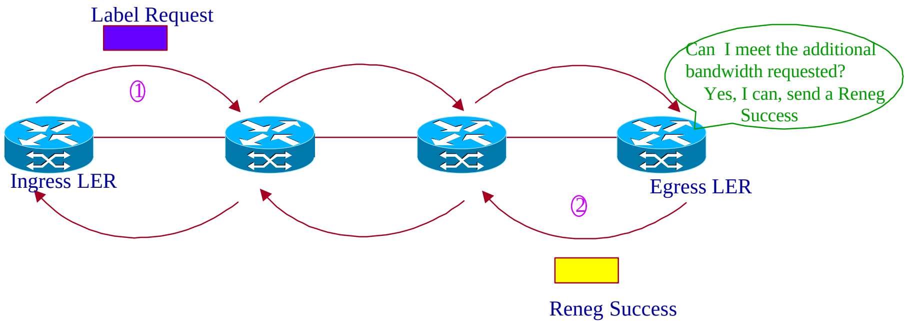

Label Request
①
Can I meet the additional bandwidth requested?
Yes, I can, send a Reneg Success
Ingress LER
Egress LER
②
Reneg Success

Figure 4-28 Successful Resource Allocation

Figure 4-29 shows the case were one of the LSRs along the LSP cannot meet the request. Again a Label Request message is sent along the LSP ① until it meets a LSR that is unable to allocate the request. In this case the LSR formulates a *RenegFailure* notification message, and forwards it back to the ingress LER ②, but does not forward the Label Request message any further.

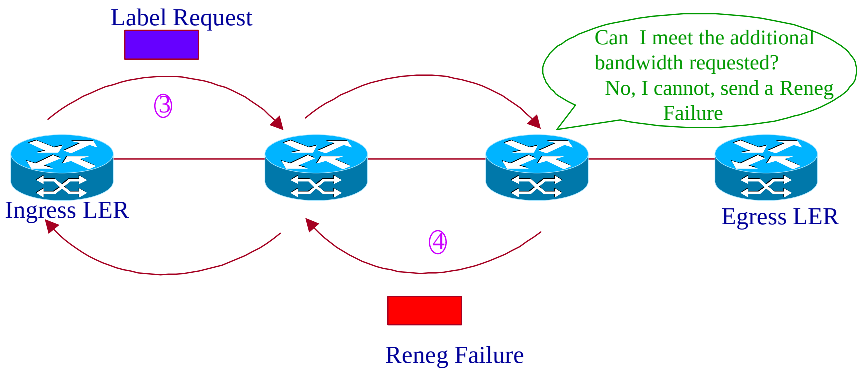

Label Request
③
Can I meet the additional bandwidth requested?
No, I cannot, send a Reneg Failure
Ingress LER
Egress LER
④
Reneg Failure

Figure 4-29 Unsuccessful Resource Allocation

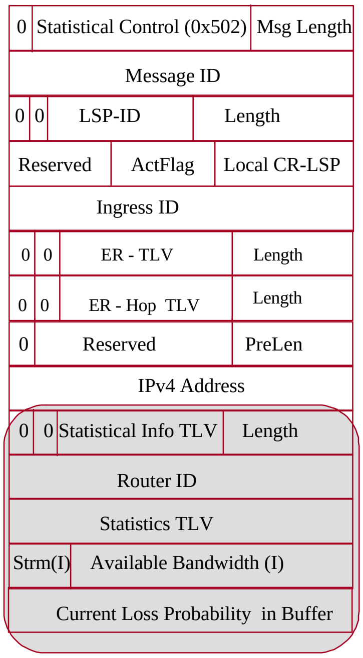
The protocol supports a “crank back” mechanism. For instance, when the ingress LER receives a *RenegFailure* notification message it can select an alternative path either by referring to a topological link cost database maintained by a separate routing protocol or the decision is made by the network management module[20]. It then sends a Label Request message along the revised path. When it receives a Label Mapping confirming a new path has been set up, it replaces the old Label Mapping with the newly received Label Mapping, it can then delete the original label or keep it to send other data along the path it represents. If the decision is to delete the original label, the ingress LER will send a Label Release message [AND99] including the newly replaced Label along the LSP to the egress LER. This procedure results in the label being removed from the pool of “in use” labels. This Label Release message should be sent a few seconds after the last packet is forwarded along that path to ensure the egress LER receives the last packet before it removes the label from forwarding use[21].

4.3.4 Monitoring Procedures

Proposed *Statistical Control* TLVs (Figure 4-30) contained within LDP notification messages, known as *Status Requests*, are either sent into the network periodically from the ingress LER or when the ingress LER receives a *CIN* message.

[20] This is outside the scope of this thesis.

[21] Alternatively a ‘flushing’ mechanism could be used to ensure all data sent along the former path has reached its destination prior to forwarding more data along the new path McD.



Router ID - IPv4 address of LSR
Strm(I) - Index of buffer stream (I)
Available Bandwidth (I) - the remaining bandwidth in buffer stream (I)
Current Loss Probability in Buffer - recorded loss in buffer stream (I)

Figure 4-30 Statistical Control TLV

When the ingress LER chooses to issue a *Status Request*, it uses the CR-LSP ID to determine which CR-LSP it refers to. It then formulates the Status Request message with the explicit route and CR-LSP ID included and transmits it to the next hop in the ER.

As each LSR receives it, it appends its own statistical information to the message. This includes the current losses of all the class-based buffers the CR-LSP could pass through at this LSR along the specified path[22] along with the available bandwidth in each buffer stream. It then forwards the *Status Request* to the next LSR along the CR-LSP. When the message reaches the egress LER, it is sent back to the ingress LER. Upon receipt of a *Status Request* message that it issued earlier, the ingress LER extracts the CR-LSP ID, and records for each LSR along that CR-LSP the bandwidth available and the current losses experienced by each buffer stream. This information is recorded in a *Statistical Buffer Table* for monitoring purposes (Table 4-5).

LSP ID		***Router ID***	***Available BW***	***Current Loss in***	***Current Loss in***
Ingress ID	***Value***		***in Buffer$_i$***	***Flow***	***Buffer$_i$***
138.56.32.76	15	138.56.92.3	10Mb/s	3.76e-04	2.46e-05

Table 4-5 Statistical Buffer Table

[22] In this thesis loss is used as an example statistical parameter, however, this could be easily generalised to a variety of traffic engineering performance metrics.

The *Status Request* messages provide an overall view of the status of the links and LSRs along a particular CR-LSP. It includes the available bandwidth and loss probabilities within every buffer stream within a LSR. The flow diagram tracing the events that occur on receipt of a *StatusRequest* messages is shown below in Figure 4-31.

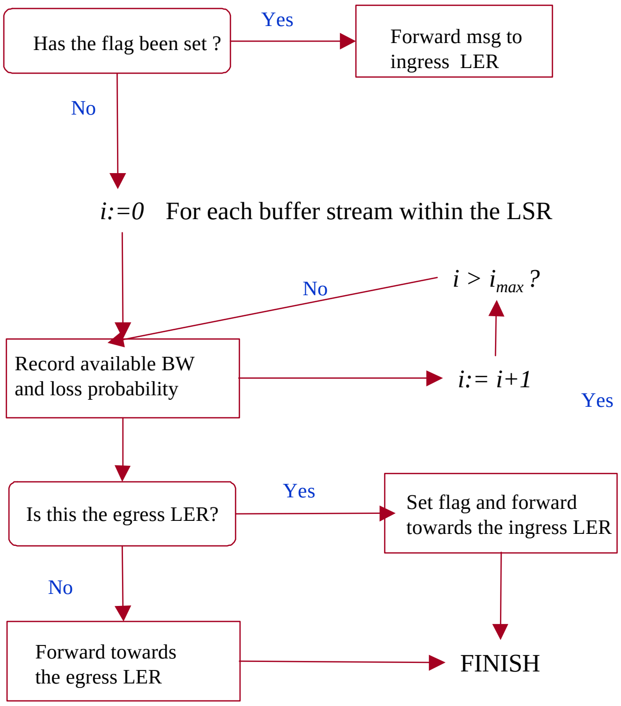

Has the flag been set ?
Yes
Forward msg to ingress LER
No
i:=0 For each buffer stream within the LSR
No
i > i_max ?
Record available BW and loss probability
i:= i+1
Yes
Is this the egress LER?
Yes
Set flag and forward towards the ingress LER
No
Forward towards the egress LER
FINISH

Figure 4-31 LSR Status Request receipt procedure

The CIN message only return status information about the CR-LSP suffering unacceptable losses and the particular buffer it traverses in the congested LSRs between the ingress LER and the initiator of the message i.e., not the entire CR-LSP.

Subsequently, if the ingress LER receives a *CIN* message, it examines the information held in its *Statistical Buffer Table* to help determine whether it should renegotiate along the existing path, as the higher buffer streams seem capable of meeting its QoS requirements. Alternatively, it can choose to negotiate for an alternative path or decide to accept the current condition.

The author proposes two new TLV parameters to be used in the Statistical Control TLV; a *Statistical Information* parameter TLV shown in Figure 4-32 and a *Statistics* parameter TLV shown in Figure 4-33.

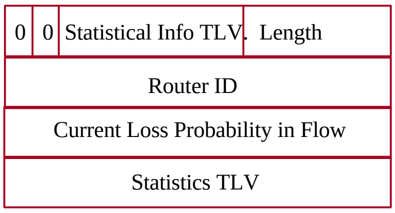


Figure 4-32 Statistical Information TLV

The *Statistical Information* TLV collects the loss probability in the CR-LSP of each LSR along the CR-LSP. It encapsulates the *Statistics* parameter TLV that records the current loss probability and the current available bandwidth in each buffer stream along the CR-LSP within each LSR. The author has chosen these particular statistical parameters; however, they could be generalised to a variety of traffic engineering performance metrics.

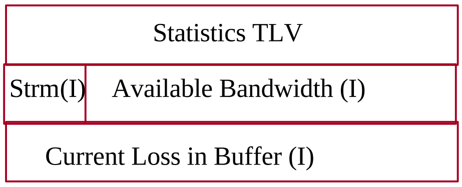


Figure 4-33 Statistical TLV

4.4 Description of the FATE+ Scheme

The previous sections described how the ingress LER monitored and controlled the traffic loading on its CR-LSPs when receiving information feedback from the network. The following sections describe the FATE+ scheme, which is an extension to FATE.

In FATE when a LSR determines it is suffering from congestion along a CR-LSP, it informs the ingress LER by sending a CIN message. Whereupon receipt the ingress LER decides on the appropriate action.

FATE+ permits congested LSRs to negotiate alternative QoS path arrangements in the event of congestion occurring without involving the ingress LER, by redistributing traffic flows from the point of congestion within the network. Whilst fully utilising all available LSRs and links and ensuring the customers' SLAs are met.

Furthermore FATE+ enhances the basic scheme by removing the responsibility of deciding what action should be taken in the event of congestion occurring from the ingress LER, and places it on the congested LSR. Thus removing the need to generate and transmit CIN messages throughout the network and allowing the scheme to scale well in large WANs.

This scheme can only be applied to CR-LSPs that have been 'loosely' configured within the region of interest as explained previously in Chapter 3, as the network operator has been given control on which links/LSRs the customers' traffic flows may traverse. In the case of a 'strictly' configured CR-LSP, re-routing along alternative paths without the customers' approval may result in introducing latency or increased costs at their expense and in that situation the FATE scheme is best suited.

The author acknowledges that path pre-emption exists within the Constraint-Based Routed – Label Distribution Protocol (CR-LDP) [JAM99a]to allow existing paths to be re-routed or torn-down to enable the re-allocation of resources to a new path. However, CR-LDP does not administer for the need to temporarily redistribute CR-LSPs during transient fluctuations of traffic load. In a situation like this, it is paramount to maintain the SLAs associated with customer flows.

The following sections describe the novel mechanisms that can be employed by the network, to eliminate the involvement of the ingress LER in the event of congestion occurring in a LSR along which a CR-LSP has already been established.

This thesis assumes the MPLS domain has been specified as an 'abstract' node, i.e. within this autonomous domain the network operator has complete control over which links/LSRs the CR-LSPs are routed. The initiator of the CR-LSP has specified that within the domain the paths may be 'loosely' routed.

4.4.1 The LSR's reaction to congestion

Consider the case where the packet loss in a particular CR-LSP passing through a buffer located within a congested LSR has risen above a predetermined threshold value as explained previously in Section 4.3.1.

In FATE+ the congested LSR can take any one of the decisions in Table 4-6 to alleviate congestion.

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

Table 4-6 Decisions taken by the congested LSR

The following sections describe the procedures and mechanisms for each of the above decisions.

4.4.1.1 Transfer of flows into a higher buffer stream

When the congested LSR decides to seek an alternative buffer stream, it will carry out the procedure explained previously in Section 4.3.3.3.1 up to the point where the congested LSR determines it is capable of allocating the bandwidth and minimum loss probability initially negotiated for the CR-LSP. Once the congested LSR has found a suitable alternative buffer stream it will carry out the following procedure: It will permanently reserve the bandwidth in the new buffer stream to the *Current* field in the *Buffer Table* and delete the original entry i.e., mapping the traffic flows into the higher buffer stream.

The CR-LSP has now been assigned to a new buffer stream capable of maintaining the initially negotiated traffic parameters, as shown in Figure 4-34.

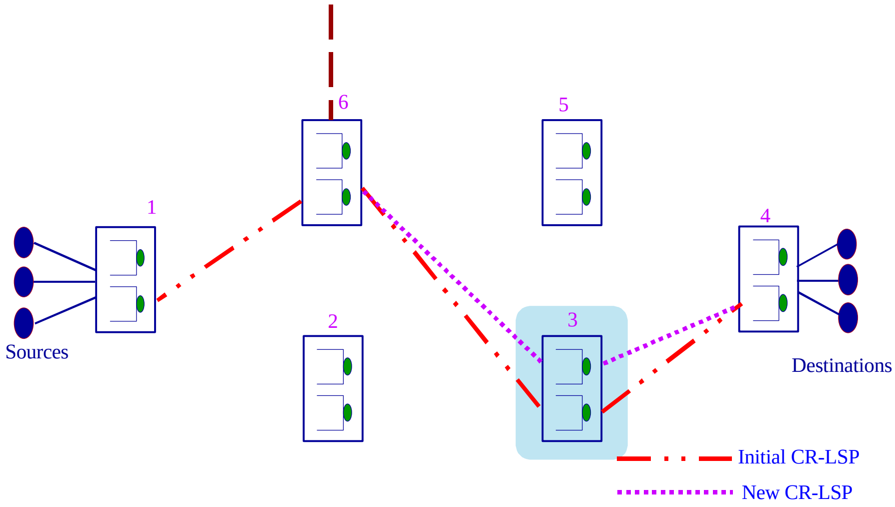


Figure 4-34 CR-LSP re-routed through a different Buffer Stream

4.4.1.2 Re-route traffic flows via an alternative downstream LSR

If the congested LSR decides to re-route the traffic flows[23] via an alternative downstream LSR it carries out the following procedure: The congested LSR assumes the role of an ingress LER and determines what the alternative next hop is eliminating the one currently being used.

It then formulates a Label Request message sent to the next hop for the destination[24] i.e., LSR 5 as illustrated in Figure 4-35. On receiving a Label Mapping from the downstream LSR i.e. 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. By retaining the newly received label, it allows a congested LSR to automatically switch a CR-LSP onto the new route during congested periods. This reduces the number of signalling messages propagating through the network and the time it takes to switch paths from the moment congestion is detected. However unless LSR 3 has knowledge of the SLA it may end up violating the traffic contract as the latency may increase. Also there may be no alternative path – so resorting to FATE where the ingress LER is involved is a necessity. This method will only work for 'loosely' routed CR-LSPs as opposed to 'strictly' routed LSPs, as the ingress LER would no longer have accurate information about the LSRs along a CR-LSP originating from itself, if a LSR were to re-route traffic flows without its knowledge.

[23] This situation may be carried out where there is no alternative buffer stream within the congested LSR.

[24] The author assumes that on receipt of a Label Request message to establish a CR-LSP, the destination was obtained from the explicit route TLV and recorded in a database, thus enabling the LSR to determine the destination for this CR-LSP.

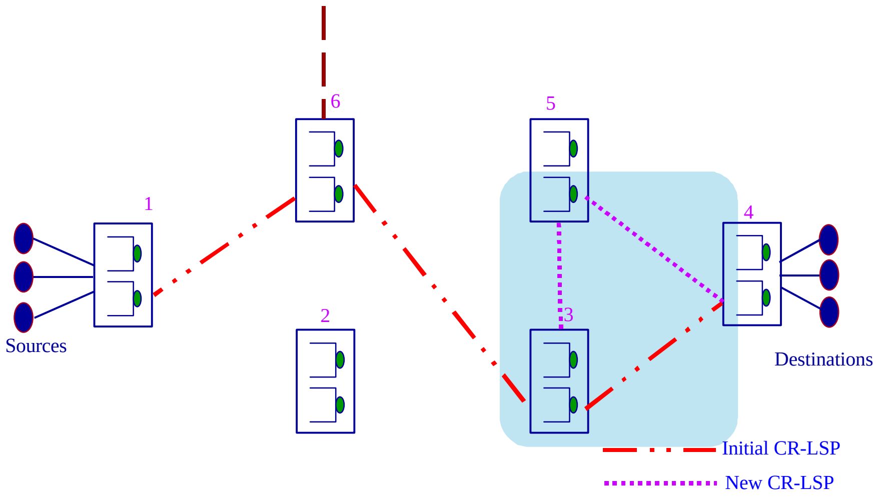


Figure 4-35 CR-LSP re-routed via an alternative downstream LSR

4.4.1.3 Re-route via an alternative upstream LSR

In the 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 would carry out the following procedure: It has to inform the upstream LSR whose identity it gets from the NHLFE table 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 the congested LSR formulates a LDP notification message containing the proposed *No Resources* TLV parameter with the format shown in Figure 4-36. It contains the CR-LSP ID and the label received from the recipient of this message.

<table>
<tr><td>0</td><td colspan="3">No Resources TLV</td><td>Msg Length</td></tr>
<tr><td colspan="5">Message ID</td></tr>
<tr><td>0</td><td>0</td><td colspan="2">LSP-ID TLV</td><td>Length</td></tr>
<tr><td colspan="2">Reserved</td><td colspan="2">ActFlag</td><td>Local CR-LSP</td></tr>
<tr><td colspan="5">Ingress LSR Router ID</td></tr>
<tr><td>0</td><td>0</td><td colspan="2">Generic Label</td><td>Length</td></tr>
<tr><td colspan="5">Label</td></tr>
</table>

Figure 4-36 No Resources TLV

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 it knows is capable of accommodating the path LSR 5 as shown in Figure 4-37. Or it will initiate the Label Request / Label Mapping process for an alternative route. This will only work in the case of 'loosely' routed LSPs as opposed to 'strictly' routed LSPs, as the ingress LER would no longer have accurate information about the LSRs along a CR-LSP originating from itself.

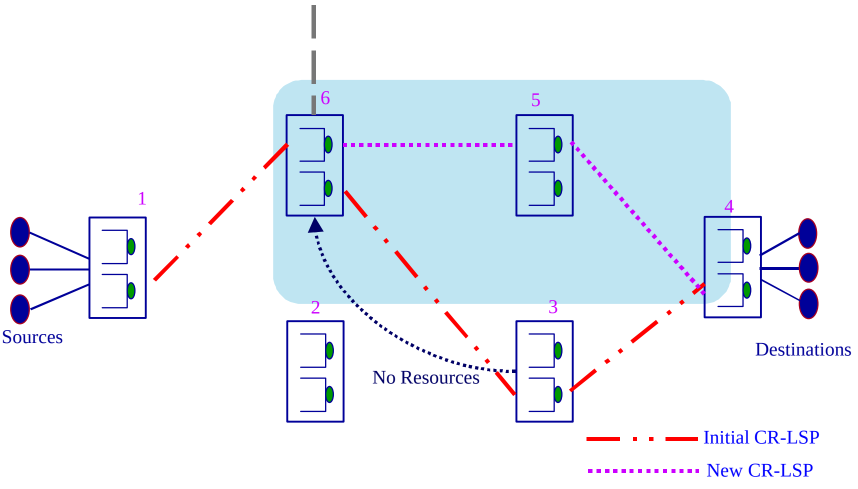
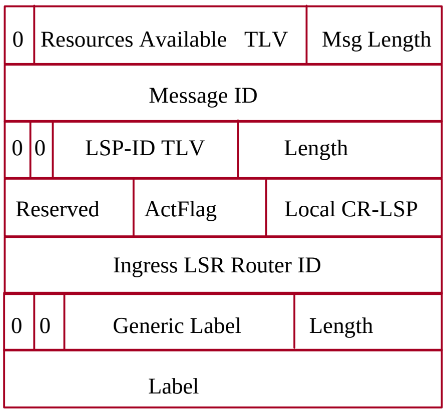


Figure 4-37 CR-LSP re-routed via a different LSR

It is the responsibility of the initiator of the *No Resources* notification message i.e., the congested LSR to inform the recipient when it is able to accommodate traffic flows along the initial CR-LSP again by sending a LDP notification message containing the proposed *Resources Available* TLV parameter, whose format is shown in Figure 4-38. It contains the CR-LSP ID and the label initially received from the recipient from this message.

On receipt of a *Resources Available* notification message, the upstream LSR knows it can continue to send traffic flows along the CR-LSP via the initial choice LSR.



0	Resources Available TLV	Msg Length	
Message ID			
0	0	LSP-ID TLV	Length
Reserved	ActFlag	Local CR-LSP	
Ingress LSR Router ID			
0	0	Generic Label	Length
Label			

Figure 4-38 Resources Available TLV

4.4.2 Summary

FATE+ provides the core network with control over the decision on what action should be taken in situations of congestion. It allows the congested LSR to either 1) re-map the traffic flows along a higher buffer stream 2) via a different downstream LSR or 3) stop flows traversing through its buffers by informing the upstream LSR. In all three situations the control of redistributing traffic flows is limited to the core network. FATE+ also eliminates the need for signalling messages to propagate between the congested LSR and the ingress LER. FATE + is suitably deployed along ‘loosely’ routed CR-LSPs.

In FATE, the indication of congestion is fed back to the ingress LER for the particular CR-LSP and the decision on whether to accept the current condition, or to renegotiate along the existing or an alternative path is made by the ingress LER. FATE is suitably deployed along strictly or ‘explicitly’ routed CR-LSPs.

In the following Chapter 5 the simulation model for FATE/FATE+ is implemented and simulations are run to determine the performance of the proposed schemes against the existing architecture.

Chapter 5: Simulation Model and Results

5.1 Introduction

In the previous chapter a new dynamic congestion control protocol for MPLS networks was proposed including the mechanisms and signalling exchanges in the core network for the FATE and FATE+ schemes. Having described the functional behaviour of these schemes it is necessary to investigate the effect of different physical implementations, traffic loading and characteristics of the functional architecture on the performance of FATE and FATE+. This is achieved by using simulation.

Having proposed the signalling exchanges between the LSRs, the volume of signalling traffic through the core network can be calculated theoretically for any particular traffic situation. However, the nature of networks is such that the estimation of reactionary performance becomes too complex for mathematical analysis. Neither can the results produced by mathematical analysis be guaranteed to be a true representation of the system. Due to statistical variation of the sources it is not possible to analytically determine the number of signalling messages generated nor the reaction upon receipt of them. Therefore simulation models are required to obtain performance data for various scenarios.

The commercial simulator used was OPNET™, a general purpose telecommunications network simulation tool [**MIL3**]. OPNET is a discrete event simulator: the discrete events in the models described in this chapter relate to the signalling packets and the congestion detection mechanism. OPNET uses a graphical interface where simulation models are defined at five levels. The first level is the project level, where the high-level components of a real-world network are identified. At this level network models are created and edited, the simulations are run and the results are analysed. The second level is the network level, where the topology of the simulated network model is defined; the interconnections of nodes (e.g., LSRs, LERs, and sources) are also identified. At the third level (the node level) the elements that make up the network nodes and their interconnections within the networks are defined; these elements include queues, processes, sources, receivers and transmitters. The functionality of each process or queue element is defined in terms of a finite state diagram and the transitions between states; this is the fourth level. The fifth and final level is where the processing in each of the states in the finite state diagram is defined in C code.

This chapter is organised as follows:

Firstly an account of the models developed is given. These include the assumptions made and the techniques used. Then a section on verification and validation is presented.

The results from the FATE and FATE+ simulation models will be compared against an MPLS network without them present. For this comparison, two models of a basic MPLS network is required; one with and one without the FATE / FATE+ schemes.

The performance of the networks will be analysed by comparing the re-mapping times for the renegotiation procedures as described in the previous chapter.

5.2 Simulation Model of the FATE and FATE+ Architectures

The FATE and FATE+ architectures make use of the existing CR-LDP protocol within MPLS. It is upon this that the mechanisms within the proposed schemes are derived. The modelling of the FATE scheme is described first.

5.2.1 Modelling of the FATE network

The simulation model of the FATE network shown in Figure 5-1 is modelled as a single network module as seen at the network level in OPNET. Each of the LSR and LERs are connected via 155Mb/s links.

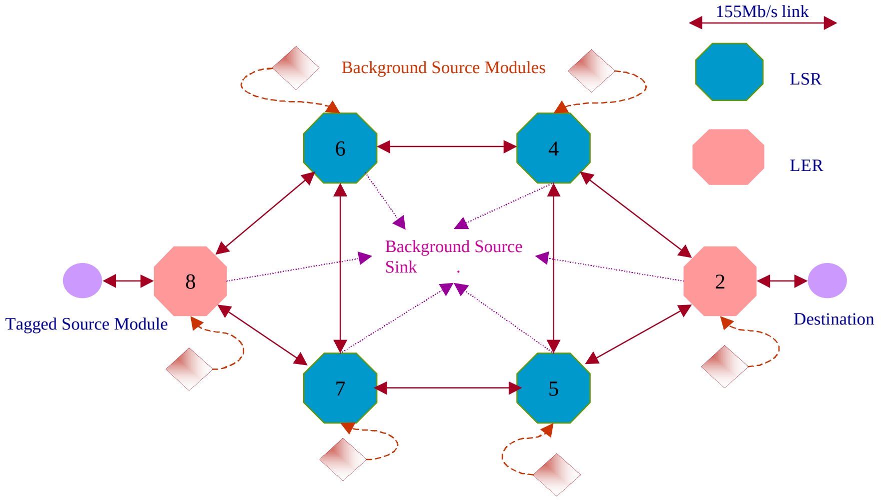


Figure 5-1 Simulation model of the FATE network

5.2.1.1 Tagged Source Module

The tagged source module initiates the establishment of a connection between itself and the destination. In this simulation study, the aggregation of a number of individual flows is modelled as a single connection request. Only the positive outcomes for the initial connection request are modelled. The tagged source module generates a constant bit rate, 33.92Mb/s flow of packets, setting the traffic parameters i.e., bandwidth and loss probability that were requested for within each packet.

Each packet is a fixed size of 424 bits. Within this research, MPLS was assumed to operate over ATM. The cell size of 53 bytes equates to 424 bits. This will usually involve composing a signalling message and transmitting the message to the network with a packetisation delay and a transmission delay. The processing delay is set to 100us and the transmission delay for a 155.52 Mb/s link is 2.726us.

A CBR source was chosen to represent the connection between the source and destination. Due to its deterministic behaviour any variation in its performance will be as a result of the network. It is this predictable behaviour that has made a CBR source the ideal choice as the tagged source. The following describes the OPNET implementation of the tagged source model.

The tagged source module is the parent process. On transmitting an initial request message, it invokes the child process to generate the tagged source. The tagged source module creates an instantaneous tagged source child process for each successful connection. Figure 5-2 and Figure 5-3 illustrates the finite state machines for the two respectively.

Once initialisation of the module is complete at the beginning of the simulation, the module goes to the ‘idle’ state. It remains in the ‘idle’ state until a self-interrupt is received to send an initial packet requesting a certain amount of bandwidth and a ceiling loss probability for a connection set-up.

The transition to the next state is determined by the type of interrupt, e.g., a stream interrupt indicating a Hello message informing it of who its neighbour is or a self-interrupt to generate the tagged source[25].

Once the finite state machine transfers to a particular state, the processing for that state is carried out. In the case of receiving a Hello message, the ID of the sender is recorded in the associated neighbouring database. If the interrupt was a self-interrupt it generates the child

[25] Throughout the remainder of this thesis the constant bit rate source shall be referred to as a tagged source.

process that starts the tagged source. Once the processing is complete the transition is back to the 'idle' state.

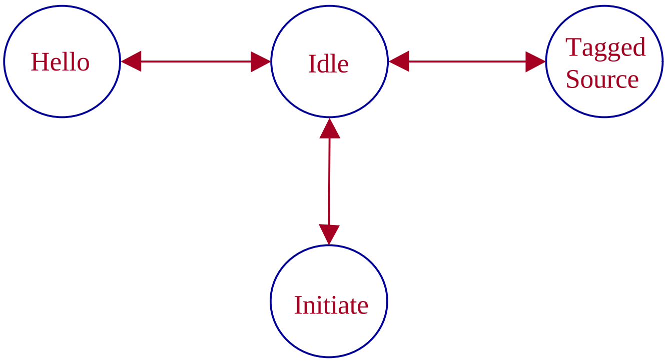


Figure 5-2 Parent Process Source module processing states

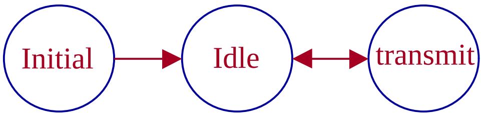


Figure 5-3 Child Process Tagged Source processing states

Figure 5-4 shows the relationship between the parent and child process, a child process exists for each successful connection request. However, in the simulation one child process is spawned for simplicity.

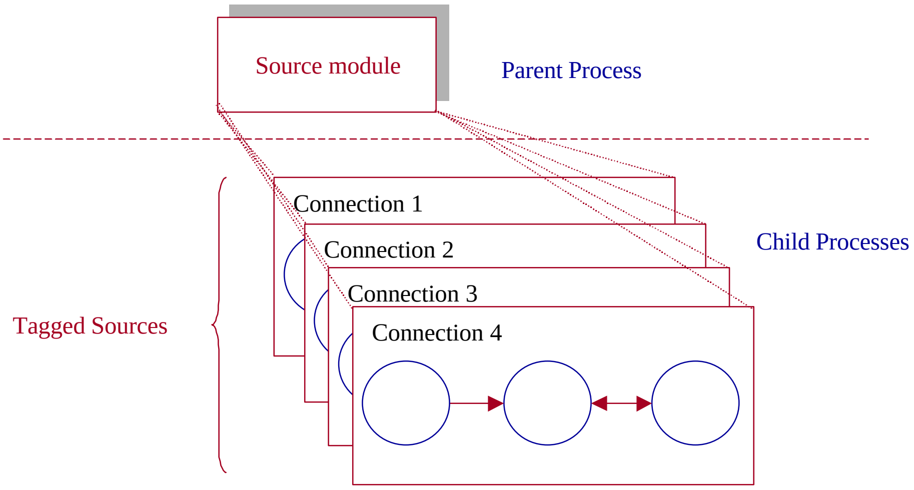


Figure 5-4 Source module process relationship with Tagged sources child processes

5.2.1.2 Background Source Module

The role of the background source module is to cause statistical variation in the traffic traversing the buffers i.e., to cause the loss probability experienced by the individual buffer streams to vary randomly. This is engineered to cause losses to exceed a predetermined threshold as required. Figure 5-5 shows an example of how the buffer stream is configured. The values obtained are explained in Section 5.3.

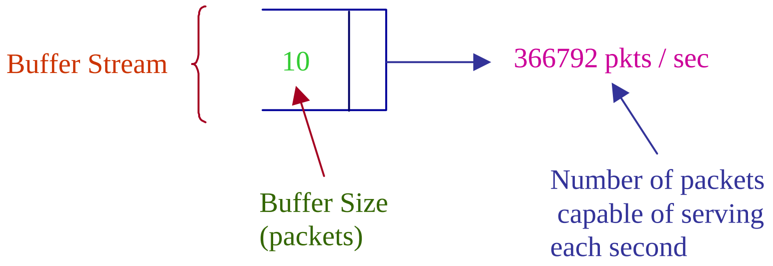


Figure 5-5 Buffer Stream

On off sources were chosen to represent the background traffic in the simulation model as their behaviour exhibits a good 'likeness' to that portrayed by Internet traffic as illustrated in Figure 5-6. This trace represents IP traffic and was obtained from simulations run on a real network with live traffic as part of the European ACTS Expert Project [EXP].

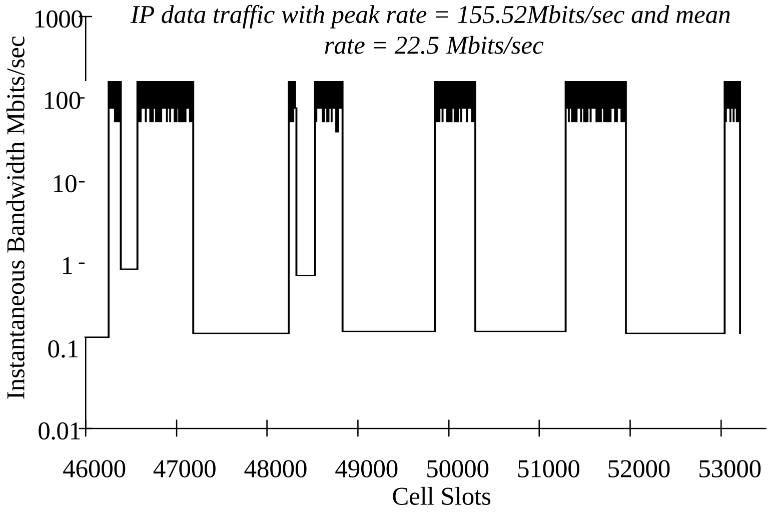


Figure 5-6 IP Traffic Trace

Also on off sources are widely understood and examined [PIT96] and can be described analytically thus making them relatively easy to work with.

A single on off source is fed into the signalling buffer stream i.e., the buffer stream through which the signalling packets traversed, and into the same buffer stream as the tagged source to provide some delay jitter along the signalling path.

The background source module as implemented in OPNET is described below.

The background source module comprises of a parent node process that invokes a child process for each instance of an on off source. The finite state machines that represents them is shown in Figure 5-7 and Figure 5-8 respectively.

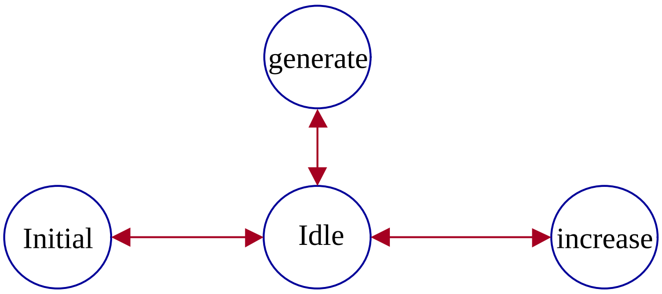


Figure 5-7 Parent Process Background Source State Machine

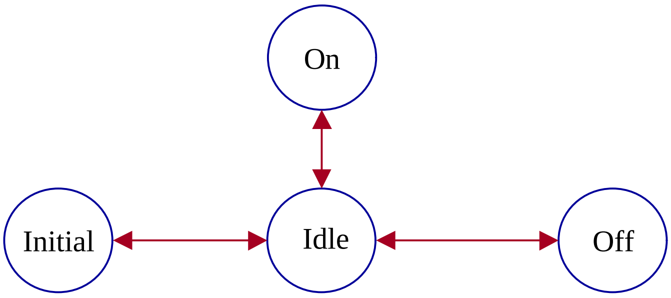


Figure 5-8 Child Process On Off Source State Machine

The on off source chosen for the simulation has the following traffic parameters;

On time = 0.00025 seconds;

Off time = 0.00225 seconds;

Peak Cell Rate = 93500 packets / second;

Distribution = negative exponential for the on and off times.

The dynamic process capabilities in OPNET allow for instances of the on off sources to be initiated from the parent source module as illustrated in Figure 5-9.

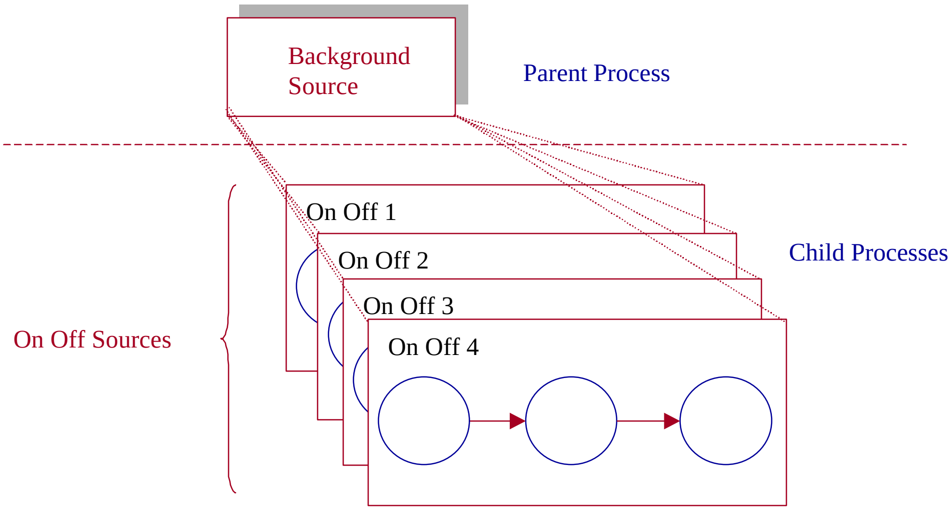


Figure 5-9 Parent Background Source Module relationship with Child On Off Processes

5.2.1.3 Background Source Sink

The background sources feeding the individual buffer streams are destroyed at the output of the Scheduler Module. This prevents an avalanche of packets occurring at the egress LER.

5.2.1.4 LER and LSR Node Models

This simulation model is a signalling protocol based model, so it is necessary to generate the appropriate signalling messages in accordance with the CR-LDP and FATE / FATE+ mechanisms. To achieve this, every request for a connection must be processed on an individual basis and the progress of each connection request must be tracked. The simulation model of the LSR and LER as it appears at the node layer of OPNET is shown below in Figure 5-10.

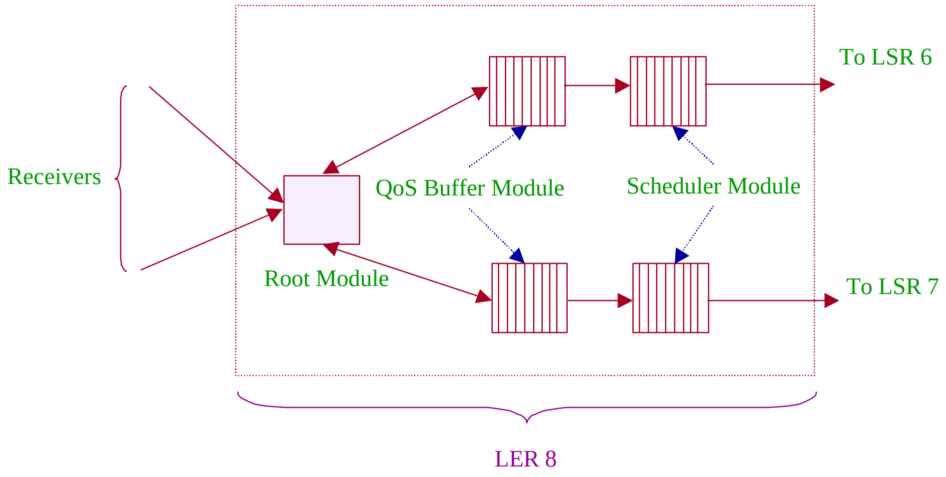


Figure 5-10 Simulation model of LER/LSR

The example depicts the connection that exists between LER 8 and LSRs 6 and 7 at the network layer as shown previously in Figure 5-1.

The node (LSR or LER) is the parent process. This is a static process that will last throughout the simulation. The primary functions of the node parent process are described below.

The behaviour of the LSR and LER are both contained within the node process. The LER has a few additional mechanisms that are not required in the LSRs. In a real network a LSR may act as both a LER/LSR at the same time for different LSPs.

Initialisation of the simulation environment is done by sending Hello messages to all surrounding nodes and gathering the necessary information from received Hello messages to formulate the topology of the network. This information is used to formulate the routing table that is necessary for the duration of the simulation. This operates as a simplified routing process similar to OSPF.

The LER will be in receipt of the initial connection request message generated by the source. As a result of this it will be in transfer of two types of signalling messages: in the role of a ingress LER it can generate the Label Request message to set up the connection and receive a Label Mapping message, in its role as the egress LER for a connection it will receive a Label Request message and generate a Label Mapping.

In situations of congestion the LSR is responsible for generating and receiving CIN, Reneg Success and Reneg Failure notification messages.

Each connection request has a unique LSP ID assigned by the ingress LER. All signalling messages generated by a request will contain this ID; the reply to the signalling messages will also contain this ID. The LER maintains a list of connection requests and their corresponding IDs.

The LER handles unlabelled packets entering and exiting the MPLS network. It assigns the received label from the earlier request set up. LSRs receive a labelled packet and forward the packet based on this label.

The ingress LER is also responsible for transmitting Status Request messages into the network, either periodically or on receipt of a CIN message.

The following subsections describe the individual components of the LSR/LER.

5.2.1.4.1 Root Module

In the case of a LER the Root Module's responsibility is to respond to the request for a connection. It determines an explicit route from itself to the destination. It then generates a Label Request message that is sent to the next hop in the explicit route. On receipt each LSR determines if it can meet the request and if so it continues to forward the Label Request message temporarily reserving the necessary resources. On receiving a Label Request the egress LER allocates a label and sends the value in a Label Mapping message to the source of the Label Request message. On receipt of a Label Mapping message the LSR uses the flow ID to identify the connection this label refers to. The LSR assigns a new label that is forwarded to the next hop along the route and forms a binding with the received label.

In situations of congestion, a congested LSR generates a CIN message containing the loss probability of the relevant buffer and flow and forwards the message to the next hop towards the ingress for the flow. On receipt of a CIN message other LSRs along the path each determine if they are suffering loss. If so they append their details to the existing CIN message and forward it towards the ingress LER. In the case were a LSR receives a CIN message and is not suffering congestion it is simply forwards the CIN message towards the ingress LER.

On receipt of a CIN message the ingress LER takes one of the following decisions:

- Renegotiate along the existing path for a higher buffer;
- Negotiate along an alternative path;
- Accept the current losses and do nothing.

In the case where the ingress LER decides to renegotiate for a higher buffer stream, a Label Request message is sent along the existing LSP. On receipt the LSR either generates a Reneg Success or a Reneg Failure message that is sent back to the ingress LER.

The following section explains how the Root Module was implemented in OPNET.

The finite state machine for the Root Module is shown in Figure 5-11. Once initialisation of the module is complete at the beginning of the simulation, the module goes to the idle state. During initialisation each LSR/LER formulates Hello messages with its address and sends them to all the surrounding LSRs. On receiving a Hello message the ID and the stream connecting this LSR to its neighbour is recorded in a database. Transition to a new state occurs on receipt of an 'initiator' message in the case of the LER and on receiving a Label Request in the case of a LSR. The Root Module handles all the major signalling processing. It receives the initialisation message to establish a LSP and the associated Label Request and Label Mapping messages. On receiving CIN messages it formulates renegotiation procedures and is responsible for generating and maintaining Status Request messages.

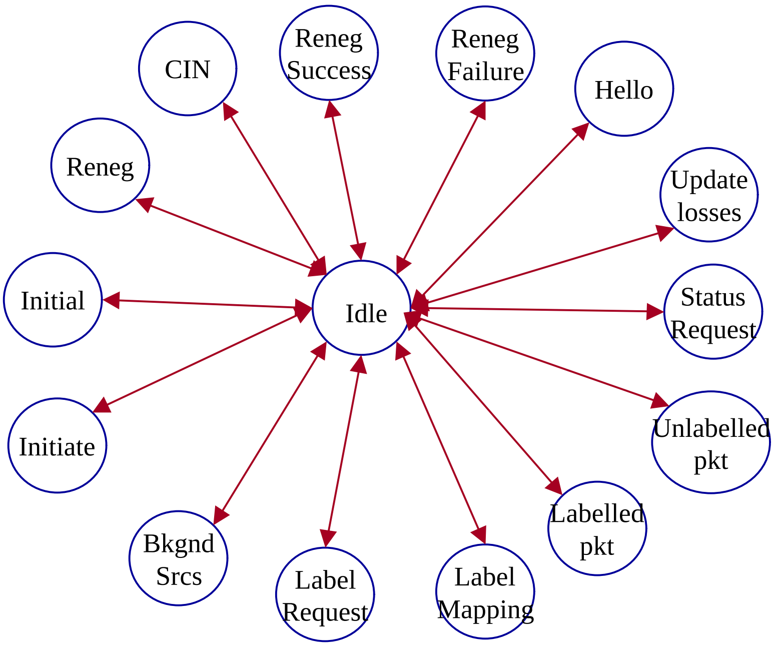


Figure 5-11 Finite State Machine of the Root Module

5.2.1.4.2 QoS Buffer Module

The QoS Buffer Module is dimensioned to have a number of buffer streams catering for different minimum loss probability thresholds. Output buffering has been chosen instead of input buffering as the choice of buffer architecture used in the simulations to avoid Head of Line blocking (HOL) Appendix A.1.

Initially each buffer stream is dimensioned for a given minimum loss probability threshold as shown below in Figure 5-12. Throughout the simulation the value currently being offered by the buffer is calculated periodically and the value passed to the Root Module where it is updated in the Buffer Table maintained by each LSR/LER.

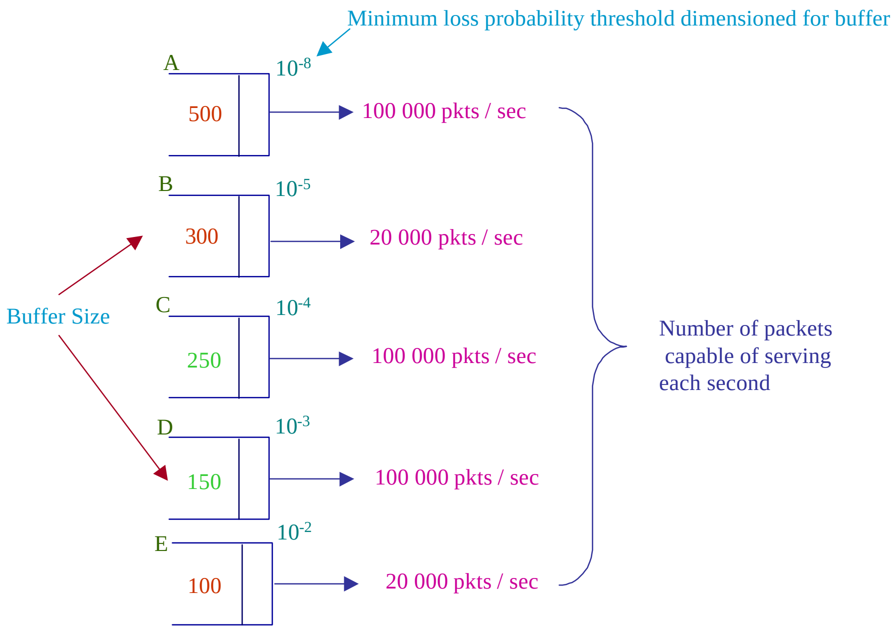


Figure 5-12 Simulation model of QoS Buffer

There are five buffer streams used in the simulation, one for signalling messages dimensioned for a very low loss probability threshold e.g., 10^{-8}, and the others catering each for loss probability thresholds of 10^{-2}, 10^{-3}, 10^{-4} and 10^{-5} respectively. Section 5.3 shows the calculations involved in determining the minimum loss probability thresholds.

The tagged source along with the background sources are passed into one of the buffer streams. The QoS Buffer module calculates the loss probability every X packet loss events (e.g., 50 packets lost at a particular buffer stream). Additional background traffic is fed into the buffer stream containing the tagged source to cause the loss probability in the buffer and flow to exceed a predefined threshold. At this point a CIN message is generated and transmitted towards the ingress LER. Every Y events (i.e., 500 packets arriving at a buffer stream) the long-term loss probability is calculated for that particular buffer stream. This value is sent to the Root module to update the Buffer Table with the current loss probability available within that buffer. This information is included in the periodic Status Request messages sent back to the ingress LER. The following describes how the QoS Buffer Module is implemented in OPNET.

The QoS Buffer is developed as a finite state machine. Once initialisation of the module is complete at the beginning of the simulation, the module either goes to the ‘idle’ state or receives a packet. On receipt of a packet it determines which buffer stream it belongs to and attempts to insert it.

The state diagram for the QoS Buffer is shown below in Figure 5-13.

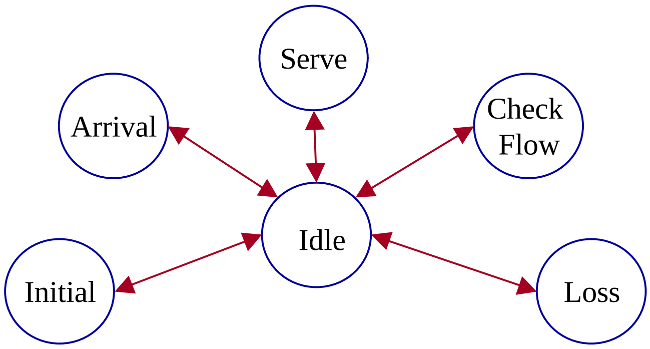


Figure 5-13 Finite State Machine for QoS Buffer Module

5.2.1.4.3 Scheduler Module

The Scheduler performs both scheduling and behaves as a transmission buffer. The Scheduler resides in the LSR/LER and receives packets transmitted from the QoS Buffer Module and buffers them before forwarding them towards the next hop. The scheduler has only one buffer stream and is dimensioned so no loss occurs within it. It serves at a rate of 155.52Mb/s[26] as shown below in Figure 5-14. Every 2.726us it 'visits' the buffer stream, if there is a packet present it will serve it immediately i.e., forward it towards the next hop LSR.

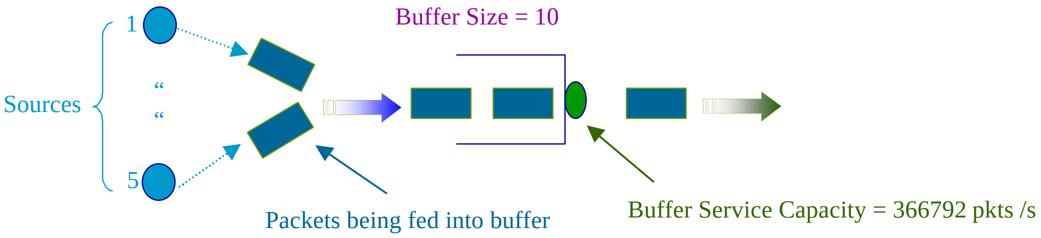

Buffer Size = 10
Sources
1
"
"
5
Packets being fed into buffer
Buffer Service Capacity = 366792 pkts /s

Figure 5-14 Simulation Model of the Scheduler

The following describes how the Scheduler Module was implemented in OPNET.

The scheduler is developed as a finite state machine. Once initialisation has occurred at the beginning of the simulation, the module goes to the 'idle' state or receives a packet. Every 2.726us it serves one if present. The state diagram for the scheduler is shown in Figure 5-15.

[26] The actual rate is 149Mb/s as some of the payload is taken up with overhead packets.

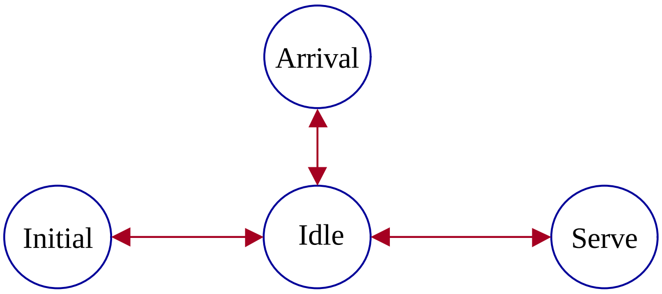


Figure 5-15 Finite State Machine for the Scheduler

5.3 Verification and Validation

Having developed a simulation model, the node needs to be verified and validated. Verification determines whether the model does indeed perform as intended and validation shows whether the model is a true and accurate representation of the system modelled [PIT93][27]. This needs to be carried out at two levels, the first on a fine scale by looking at individual objects that make up the network and then at the whole network.

The simulation model uses some library models supplied with OPNET: the receiver, transmitter and physical link models, in addition to which node processes were developed to model FATE and FATE+ functionality. All library functions were verified and tested using purpose built test models.

Although various queueing models were provided with OPNET, these were found to be inadequate to model multiple server queues with individual buffers.

A modular approach was used. This allowed for each functional component within the queue module to be tested individually and independently. Once it was established that each component operated as intended, the complete model was verified using single stepping techniques and the state of the system was recorded. This showed that the model operated as expected. Furthermore the module was tested against known mathematical models as explained in the following subsections.

[27] The random number generator used in the simulation is explained in Appendix A.3

5.3.1 Sources

A high value service rate was chosen for the buffers to allow variability in the sources feeding them. It was found that low values of service rate resulted in a very rapid change in loss probability.

The service rate of the buffer servicing the tagged source was set at 100 000 packets/second. The PCR of the tagged source fed into the buffer was set to produce

80 000 packets/second. This value was chosen to be less than the service rate of the buffer to prevent loss.

The remaining 20 000 packets/second the buffer is capable of serving is assigned to the on off sources assigned to that buffer stream.

The on off sources were needed to load the individual buffers[28] within the LSRs as shown in Figure 5-16. The objective is to dimension each of the buffer streams to a particular minimum loss probability threshold. To ensure the sources were loading the buffer streams correctly the following procedure was carried out: A number of on off sources were fed into a single buffer, the number of packets served by the buffer over a given time period was recorded and compared against the theoretical value.

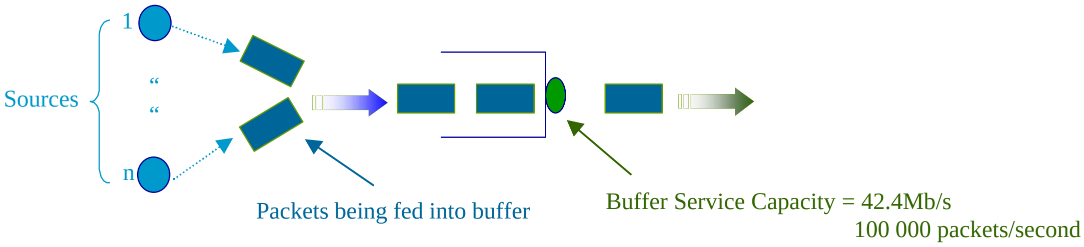

Sources
1
"
"
n
Packets being fed into buffer
Buffer Service Capacity = 42.4Mb/s
100 000 packets/second

Figure 5-16 Source Validation

The simulation parameters of an individual on off source are shown in Table 5-1.

Peak Cell Rate during the on period	**39.64Mb/s (93500 packets/second)**
T_{on} (On Time Duration)	**0.00025 seconds**
T_{off} (Off Time Duration)	**0.00225 seconds**
On Time Distribution	**Negative exponential**
Off Time Distribution	**Negative exponential**

Table 5-1 On Off Source Simulation Parameters

[28] Although the diagram depicts one buffer, all the buffers will be treated in a similar way. The number of sources fed into them being dependent on the minimum loss probability threshold required for that particular buffer stream.

The number of on off sources required to load the buffer to a certain utilisation are estimated as follows:

The average rate at which an on off source operating during its on period is calculated from (1),

$$average\_rate = Peak * \left(\frac{T_{on}}{T_{on} + T_{off}} \right) \qquad (1)$$

The number of sources required for a particular utilisation is found from (2),

Let $r = utilistion,$

$c = buffer\_service\_capacity$,

$n = number\_of\_sources$

$$n = \frac{r * c}{average\_rate} \qquad (2)$$

Table 5-2 shows the number of on off sources required in order to load the buffer to various utilisations[29].

Utilisation	***Number of sources required***
0.2	2
0.4	4
0.6	6
0.8	8
1.0	10

Table 5-2 Number of Sources required at different utilisation

The validation scenario was set up as shown in Figure 5-16, with a number of on off sources feeding into a buffer of infinite[30] size with a service capacity of 42.4Mb/s (100 000 pkts/sec) over a 10-second time period, for different levels of utilisation. The theoretical values for the number of packets the buffer is capable of serving in a 1-second time period is calculated as follows:

[29] The number of sources has been truncated for simplicity.

At 80% utilisation the maximum number of packets the buffer is capable of serving in 1-second time period is: $0.8 * 100000 = 80000\, packets / \sec ond$

Table 5-3 shows the values for all levels of utilisation tested.

[30] An infinite size was chosen to prevent packet losses.

Utilisation	*Number of packets served per second*
0.2	20000
0.4	40000
0.6	60000
0.8	80000
1.0	100000

Table 5-3 Theoretical values for number of packets served by buffer

The simulation was run for 10 different seeds and the results equated for a 1-second time period are shown in Table 5-4.

Utilisation	**Number of Packets Served**				
	Seed_12	***Seed_2***	***Seed_23***	***Seed_34***	***Seed_45***
0.2	19021	19434	19173	18411	19807
0.4	39089	38347	38825	38188	36658
0.6	57235	57052	57322	58319	58667
0.8	76178	77253	76327	77711	79457
1.0	96730	96298	97198	93201	91207

Utilisation	**Number of Packets Served**				
	Seed_55	***Seed_72***	***Seed_78***	***Seed_87***	***Seed_91***
0.2	20129	19575	19718	18744	19821
0.4	39231	38486	38742	37467	37861
0.6	59513	55605	57715	57715	51000
0.8	76981	79244	76793	76793	77724
1.0	97077	95488	97315	97315	96677

Table 5-4 Simulation values for the number of packets served by buffer run for different seeds

90% confidence interval (Appendix A.2) for each value was calculated of utilisation as follows:

The following example shows the calculation at 20% utilisation.

The mean was calculated from (1);

$$Mean, \overline{X}(n) = \frac{\sum_{i=1}^{n} X_i}{n} = 1938.3 \qquad (1)$$

The variance is calculated from (2);

$$s = \frac{\sum_{i=1}^{n} \left(X_i - \overline{X}(n)\right)^2}{n-1} = 289800 \qquad (2)$$

The 90% confidence interval with a *t* distribution is given by (3);

$$\overline{X}(n) \pm t_{n-1,1-\frac{a}{2}} \sqrt{\frac{s\,(n)}{n}} \qquad (3)$$

From the standard normal distribution *t* table for 90% confidence interval and a sample size of 10 the constant value is 1.83. (3) becomes:

$$\overline{X}(10) \pm t_{9,0.95} \sqrt{\frac{s\,(10)}{10}}$$

$$= 19380 \pm 1.83 \sqrt{\frac{289800}{10}}$$

$$= 19380 \pm 312$$

Table 5-5 shows the 90% confidence intervals for the other levels of utilisation obtained from the simulation.

Utilisation	Simulation	Theory
0.2	19380 ± 312	20 000
0.4	38290 ± 456	40 000
0.6	57070 ± 1377	60 000
0.8	77570 ± 633	80 000
1.0	95800 ± 1168	100 000

Table 5-5 90% Confidence Interval for Number of packets served by buffer

Figure 5-17 shows the confidence intervals for the various utilisation loads[31].

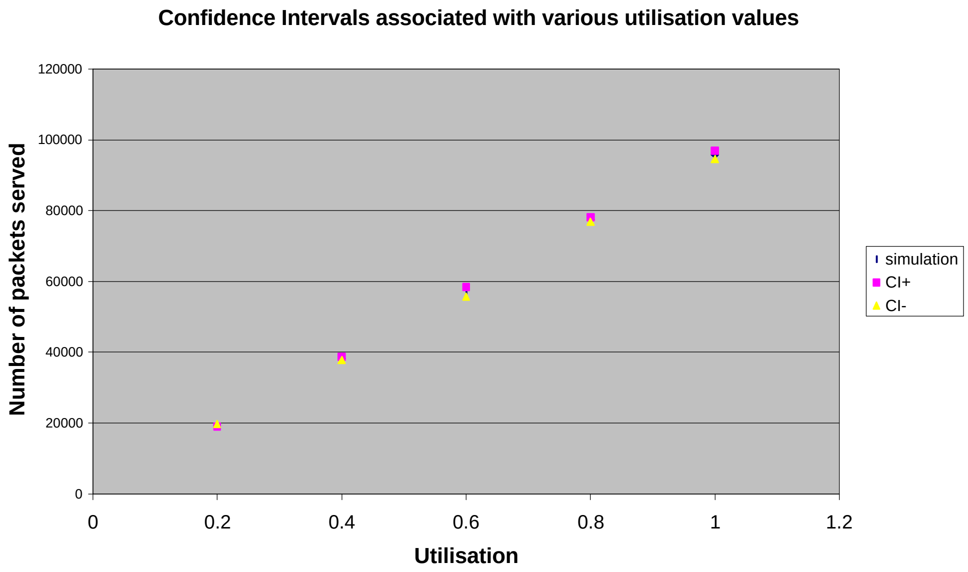


Figure 5-17 90% Confidence Intervals for Utilisation Loads

The graph showing the results of the mean of the simulation runs against the theoretical value is shown in Figure 5-18. The simulation results map very closely to the theoretical value, confirming that the sources are loading the buffer correctly. The slight difference that is observed is due to the approximations when parameterising sources used.

[31] The confidence intervals for all the simulation runs were very small so they are omitted for clarity purposes on subsequent graphs.

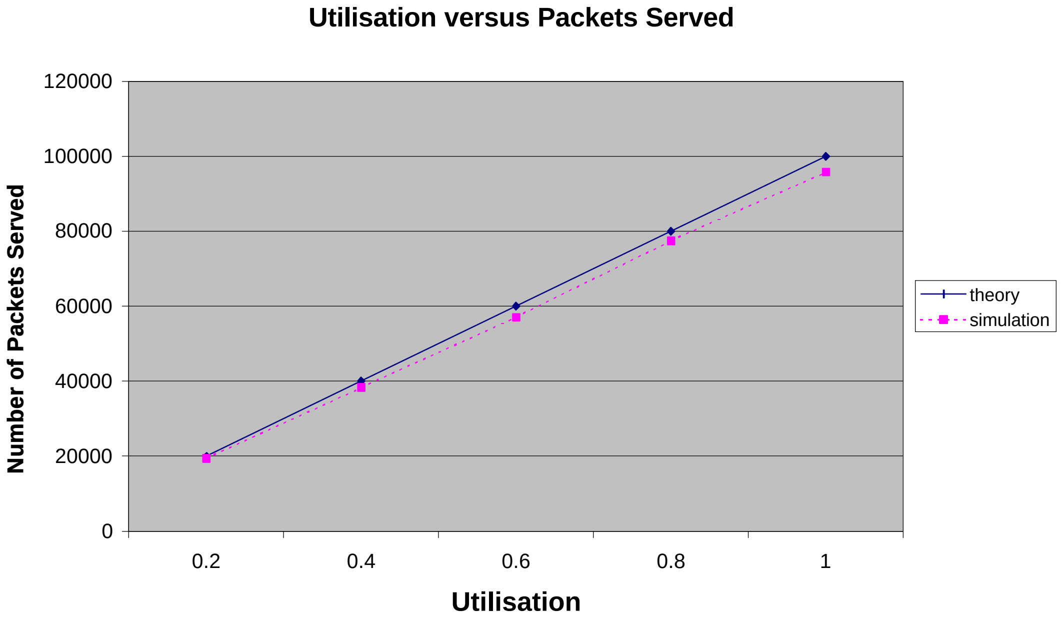


Figure 5-18 Utilisation vs. Packets served

To determine whether the on off sources have an effect on the performance of FATE the traffic profile of the sources is altered. The total cycle time of the source is composed of two parts; the time spent in the on state and the time spent in the off state.

In varying the traffic profile the author has decided to keep the PCR and the total cycle time constant whilst varying the time spent in the on and off states.

This variation in the traffic profile of the source results in a change in the 'burstiness' or number of packets produced while the source is in the on state. With more one than one source behaving in this way the number of packets arriving at the buffer varies and can cause packet losses and the phasing between them to vary.

The values chosen for the sources' on and off times are shown in Table 5-6.

On Time (Seconds)	Off Time (Seconds)	Ratio Off/On
0.00125	0.00125	1
0.001	0.0015	1.5
0.00075	0.00175	2.3
0.0005	0.002	4
0.00025	0.00225	9

Table 5-6 Source Traffic Profile

The duration of the on and off times are calculated as the percentage of the total cycle time whose value is 0.0025 seconds as shown in Table 5-7.

% of the cycle time	Source Profile		Rate (Mb/s)
	On	Off	
0.1	0.00025	0.00225	39.6
0.2	0.0005	0.002	19.8
0.3	0.00075	0.00175	13.21
0.4	0.001	0.0015	9.91
0.5	0.00125	0.00125	7.92

Table 5-7 Source transmission rate

Each of the source profiles transmits at a different rate whilst in the on state.

The peak rate at which the source transmits in the on state is determined by rearranging (1)

$$average\_rate = Peak * \left(\frac{T_{on}}{T_{on} + T_{off}} \right) \qquad (1)$$

In all the simulation scenarios the source profile whose off-on ratio equals 9 is fed into the signalling buffer along with the signalling messages. However all the on off source profiles are fed into the buffer stream along with the tagged source.

5.3.2 Buffer

This section is composed of two subsections one detailing the procedures in determining whether the buffers are serving the packets correctly and the other dimensioned the individual buffer streams for a minimum loss probability threshold.

5.3.2.1 Buffer Servicing

The following scenario was implemented to determine if the buffer was serving at the rate it was dimensioned for.

A single buffer of infinity size (i.e., no packet loss possible) with a service capacity of 42.4Mb/s capable of serving 100 000 packets/second was fed with 8 on off sources load the buffer at 80% utilisation. . This level of utilisation was chosen as a network operator would ideally aim to maintain a high level of utilisation as it would be profitable. Figure 5-19 depicts the scenario.

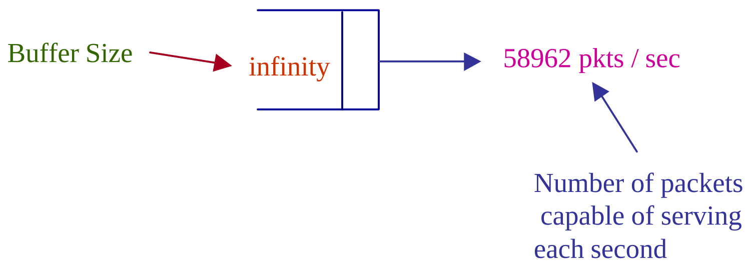


Figure 5-19 Single Buffer configuation

A histogram is set up for the number of packets that could be in the buffer as a packet arrives. On each packet arrival the value of the number of packets present in the buffer is incremented. The number of packets serviced by the buffer was recorded over 10 seconds. The probability of there being packets in the buffer on arrival is calculated and plotted as shown in Figure 5-20.

Values were recorded up to the probability of there being 500 packets in the buffer on arrival, but due to clarity the diagram only shows the range from 0 to 200 packets being in the buffer on arrival.

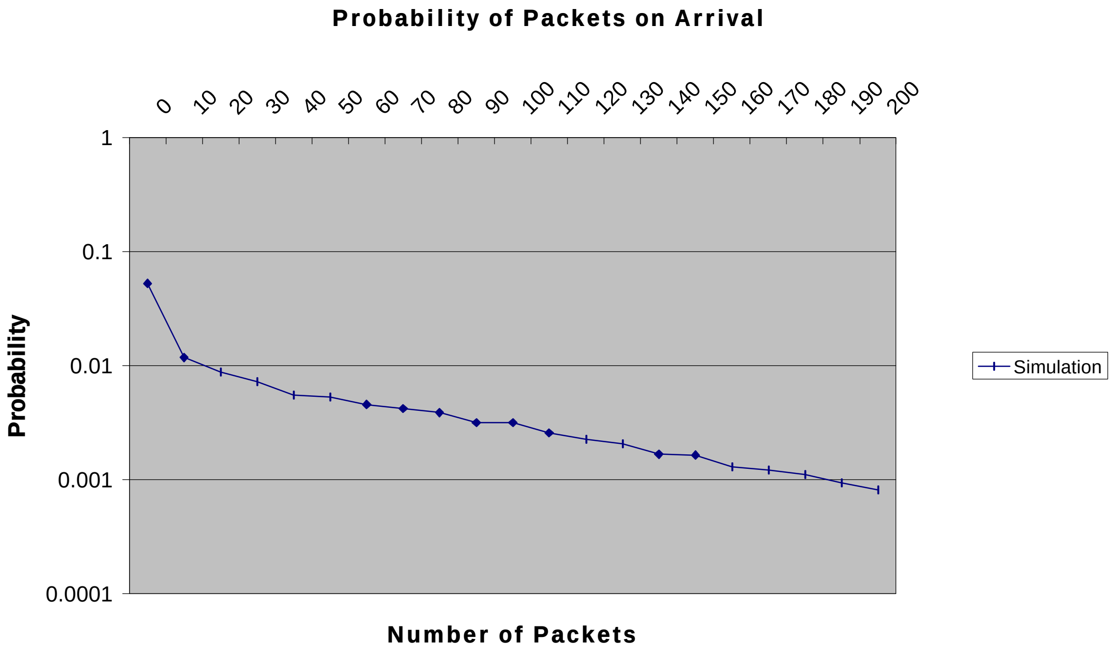


Figure 5-20 Arrival Probabilities

The sum of the probabilities was found from the following equation:

$$p(i) = \sum_{i=0}^{\infty} \frac{n(i)}{x} \qquad (1)$$

where:

$p(i)$ is the probability that there are i packets in the buffer on arrival;

$n(i)$ is the number of times there are i packets in the buffer on arrival;

x is the number of packets received over the simulation period.

The slope of the graph in Figure 5-20 is referred to as the decay rate, h .

Two points are selected on the graph. For this calculation assumes the two points are 10 and 190 along the x-axis. The probabilities at these points are 0.001175 and 0.000935 respectively. As the y-axis is a log scale and the x-axis is a linear scale the slope is calculated as follows:

The difference along the x-axis is calculated from 190-10 = 180.

The decay rate with respect to 180 is:

$$h^{180} = \frac{0.000935}{0.001175} = 0.79574$$

$$180 * \log(h) = \log(0.79574)$$

$$\log h = -\frac{0.099223}{180}$$

$$\therefore \qquad h = 0.998$$

The experimentally derived sum of probabilities was observed at 0.998.

To confirm the buffer is behaving as expected the slope of Figure 5-20 is compared against a theoretical analysis.

The theoretical value of h is found from the following calculation using equations obtained from [SCH2000a][SCH2000b] for multiplexing on off sources and is shown below:

Define the parameters in terms of the equation:

on time in the on state for a single on off source

off time in the off state for a single on off source

h is the peak number of packets / second when the source is in the on state

B is ErlangB loss probability

C is the service capacity of the buffer

N is the number of on off sources

R_{on} is the mean rate in the ON state

T_{on} is the mean duration in the ON state

T_{off} is the mean duration in the OFF state

D is the probability of a call being delayed

r is the utilisation

C' is the equivalent output rate

k is a constant

The parameters for the on off source and buffer are:

T_{on} = 0.00025s

T_{off} = 0.00225s

h = 39.644Mb/s (93 500 packets / second)

C = 42.4Mb/s (100 000 packets / second)

1. ***Calculate Offered Load***.

Determine the applied traffic, i.e., the total arrival rate of bursts from all N sources, times the mean holding time of a burst gives you the applied traffic at the burst level:

Offered Load, $A = on * \frac{N}{on * off}$

At approximately 80% utilisation of the buffer, N = 8.

$$A = 0.8$$

2. Determine the number of servers such that they equal the number of active sources necessary to prevent overloading the output capacity:

$$\text{Number of Servers, } N_0 = \frac{C}{h} = 1.07$$

3. The probability of a call being blocked is found from the following:

$$B = \frac{\frac{A_N}{N!}}{1 + \frac{A^1}{1!} + \frac{A^2}{2!} + \frac{A^3}{3!} + .. + \frac{A^N}{N!}} = 0.167$$

4. Determine the Applied Traffic, A$_p$

$$A_P = \frac{N * h * on}{on + off} = 74800$$

5. The mean rate in the ON state is calculated from:

$$R_{on} = C + \frac{h * A_P}{C - A_P} = 377500$$

The mean duration in the ON state is found from:

$$T_{on} = \frac{h * on}{C - A_P} = 0.000927$$

The probability of a call being delayed is estimated from:

$$D = \frac{N_0 * B}{N_0 - A + A * B} = 0.443$$

6. The mean duration in the OFF state is found from:

$$T_{off} = T_{on} * \frac{1 - D}{D} = 0.001166$$

7. The equivalent output rate C' is found from:

$$C' = \frac{\left(\frac{R_{on}}{r}\right) * T_{on}}{T_{on} + T_{off}} = 11690$$

8. The decay rate is calculated from:

$$h = \left[\frac{1 - \frac{1}{(R_{on} - C) * T_{on}}}{1 - \frac{1}{(C - R_{off}) * T_{off}}}\right] = 0.99$$

The calculation of the decay rate from the simulation results is found from the gradient of the graph and equals 0.998.

This value confirms the buffer is serving the packets as expected and that the packets are arriving at the buffer as expected.

5.3.2.2 Minimum Loss Probability Thresholds

This section explains the process involved in dimensioning each buffer stream for a minimum loss probability threshold.

There are five buffer streams within each LSR/LER node module as shown in Figure 5-21.

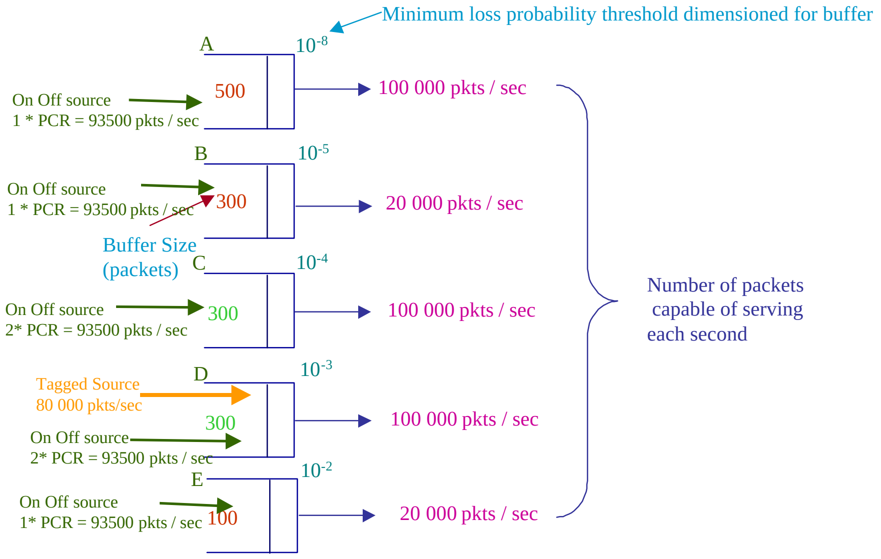


Figure 5-21 Buffer Configuration

Each buffer stream within a LSR/LER is dimensioned for a particular minimum loss probability threshold. To determine how to achieve this the following procedure was carried out for the particular source's on off parameters. The simulation parameters of an individual on off source are shown previously in Table 5-1.

5.3.2.3 One Background source

The minimum loss probability thresholds required for buffer streams B and E are found by using the Exact Fluid Flow Analysis as described fully in [PIT96], as this caters for a buffer stream being fed with a single on off source.

As explained previously in Section 5.3.1 a single on off source is fed into a buffer stream. The on off source's traffic profile is known and the minimum loss probability required for the buffer stream is known, the buffer size in terms of the number of a packets is determined from the following equations.

Consider the scenario where cells arrive at a rate of, *R cells/second* into a buffer servicing at a rate of *C cells/second* from a source whose expected on and off times are *E(ON)* and *E(OFF)* respectively. When the buffer is not full with the source in the *ON* state, the queue will fill at a constant rate of *(R-C)*, but when the buffer is full the cells will be lost at a rate of *(R-C)*. In the *OFF* state where it will remain for at least one time slot on re-entering the *ON* state, it will generate at least one 'excess-rate' arrival.

Define

$CLP =$ cell loss probability

$R =$ ON rate

$T_{on} =$ mean duration in ON state

$T_{off} =$ mean duration in OFF state

$a =$ the probability of generating another excess-rate arrival in the ON state

$s =$ the probability of being silent for another time slot

$X =$ buffer capacity of queue

The mean on duration is found from:

$$a = 1 - \frac{1}{T_{on}(R - C)}$$

The mean off duration is found from:

$$s = 1 - \frac{1}{T_{off} * C}$$

The excess-rate cell loss probability is found from:

$$p(X) = \frac{1}{1 + \left(\left(\frac{s}{a}\right)^X - 1\right) * \left(\frac{1-a}{s-a}\right)}$$

The cell loss probability can then be calculated from:

$$CLP = \frac{R - C}{R} * p(X)$$

The simulation was run for 100 seconds with 10 RNG seeds resulting in the a minimum loss probability threshold of the order of magnitude 10^{-5} and 10^{-2} for buffers B and E respectively as shown in Figure 5-21.

Buffer A is the signalling buffer where all the signalling messages have high priority servicing. A single on off source is fed into the buffer to cause queueing for the signalling messages, but the buffer stream is dimensioned to ensure no loss occurs.

With a buffer size of 500 packets, and a buffer whose service rate is 100 000 packets/second being fed with a single on off source there will be no burst scale queueing thereby ensuring no packet loss. The low loss probability threshold of 10^{-8} is therefore achievable.

The simulation was run for buffer A with a buffer size of 500 packets. No loss was recorded as was expected for the parameters involved.

5.3.2.4 Two Background Sources

Buffer C contains one 39.64Mb/s CBR source being with fed with two on off sources. The assumption made in dimensioning the minimum loss probability for the buffer is as follows:

The 39.64Mb/s CBR tagged source generates 80 000 packets/second. The buffer has the ability to service a maximum of 100 000 packets /second. This leaves the buffer with the ability to service 20 000 packets / second. It is this value that is used along with the two on off sources generating 93 500 packets/second to estimate the loss probability offered by that buffer. Using equations derived by [PIT2000] for the aggregation of less than ten on off sources the minimum loss probability threshold is found.

Buffer D is the higher buffer stream used to transfer the flow into when congestion is detected and is also dimensioned in the same way as Buffer C.

5.3.3 LSR and LER

The node processes for the FATE and FATE+ functions were developed in a similar fashion. These were again verified using single step techniques. Connection dependent processing was verified against MPLS drafts [AND99] [JAM99a] to ascertain that the operation and the resulting signalling messages generated were in accordance with the recommendation and so was the sequence in which they occurred.

The next step was the validation of the complete FATE/FATE+ simulation model. This was carried out using measurements made on the FATE enhanced network.

5.4 Simulation Model for FATE/FATE+

In the previous section simulation models produced as part of this research to investigate the performance of the FATE/FATE+ architecture were described. The focus of the simulation is to investigate the influence of the FATE/FATE+ renegotiation mechanisms and procedures within an MPLS network.

The following scenarios will serve as a comparison between an MPLS network with and without FATE present. The MPLS network without FATE present is used as a point of comparison.

Figure 5-22 illustrates the behaviour of two on off sources and the tagged source. The number of packets generated every 1ms is recorded. The CBR produces 80 packets every 1ms as shown equating to 80 000 packets every second as is expected for a 33.92Mb/s CBR source. Whilst the number produced by the on off sources varies as expected.

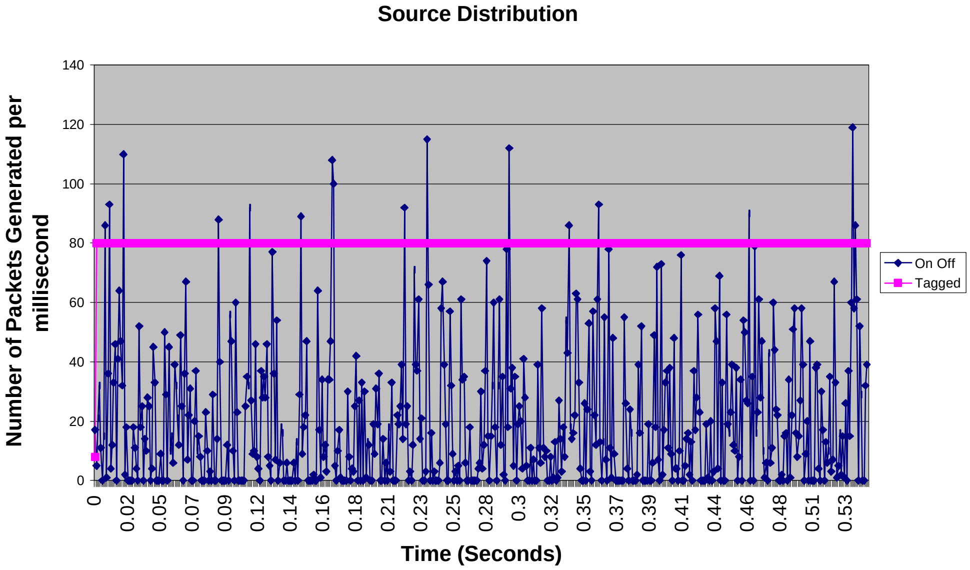


Figure 5-22 Distribution of the Tagged and On Off Sources

The simulation is set up as shown in Figure 5-23. There are various delay components that need to be considered. For the simulations the following values were assigned:

- Propagation delay was considered to be zero;
- Transmission delay for a 150Mb/s link is 2.83us;
- Packetisation delay and processing delay is 100us;
- Queueing delay is determined by simulation experiments.

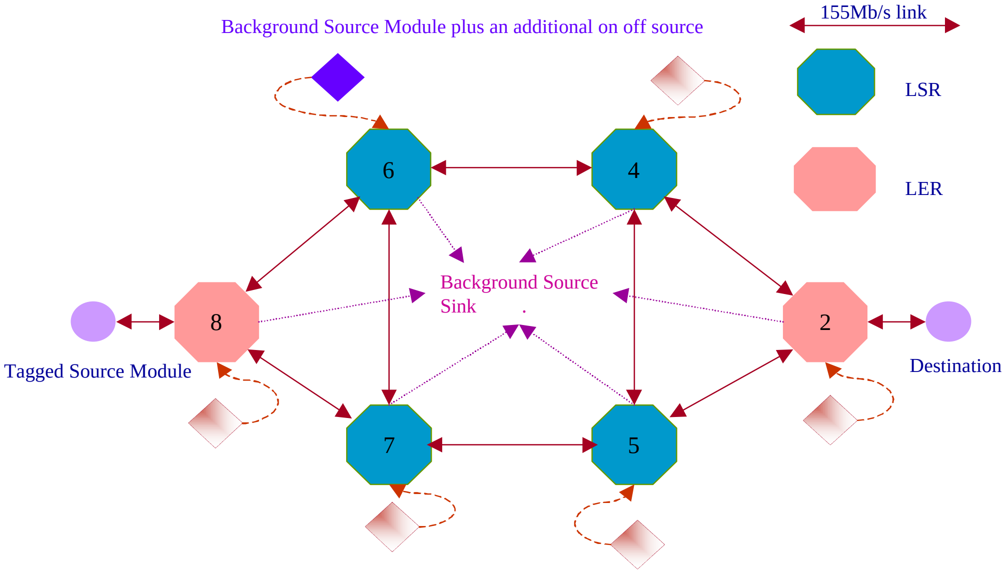


Figure 5-23 Simulation Model - Scenario 1

Figure 5-24 illustrates the simulation model of the individual buffer streams within each LSR/LER.

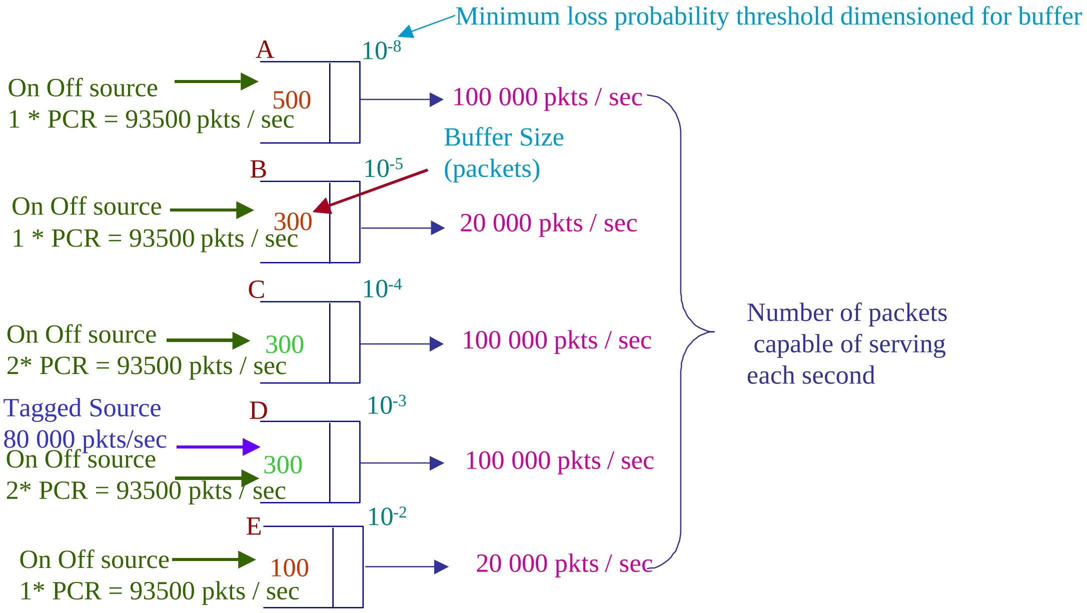


Figure 5-24 Simulation configuration of buffer streams

The following subsections describe the six scenarios the author chose to simulate.

Scenario 1 shows the proposed FATE scheme works. It details congestion occurring, its detection and the FATE mechanism responding by transferring the traffic flows into a higher buffer stream and along an alternative LSP.

Scenarios 2 and 3 repeat the simulations in Scenario 1 were appropriate. The differences between them are the point at which congestion occurs, representing the different number of hops through which the signalling messages have to progress.

Scenario 4 compares the response times of the ingress LER to alleviating congestion on receiving a Status Request and a CIN message.

Scenario 5 evaluates the performance of FATE as the on off source profile is altered. The latency involved in the ingress LER responding to receiving CIN messages is compared with and without a CIN Timer present as the background source's traffic profile is varied. Scenario 5 also monitors the number of CIN messages received by the ingress LER over a fixed time period with and without a CIN Timer present.

Scenario 6 evaluates the performance of FATE+: details congestion occurring, its detection and the FATE+ mechanism responding by transferring the traffic flows into a higher buffer stream and along an alternative LSP.

5.4.1 Scenario 1

Scenario 1 is run to show that the proposed FATE scheme works. When congestion is detected the traffic flows suffering an unacceptably high loss probability are either transferred into a higher buffer stream or into an alternative LSP.

Initially the Tagged Source Module generates the 33.92Mb/s-tagged source producing a packet every 12.5us, which is fed into buffer D of each LSR through LSRs 8,6,4 and 2 as shown in Figure 5-24.

The Background Source Module produces a single on off source that is fed into the signalling buffer of each LSR/LER to represent background signalling traffic.

1ms later two on off sources are fed into the same buffer as the tagged source.

To ensure the simulation has reached stability on losing 50 packets at buffer D the on off sources traversing that buffer are switched off. 2ms when all possible packets belonging to the on off source has been served the on off sources are switched back on. This is repeated twice. On the third iteration when the on off sources are switched on an additional on off source is fed into the same buffer as the tagged source in LSR 6 at a random time. At this point when 50 packets are lost the loss probability experienced by the buffer is calculated.

In this scenario where two on off sources are fed into the buffer neither the loss in the buffer or in the flow exceeds the predetermined threshold. When an additional background source is added to LSR 6 both the buffer and LSP experience a loss that exceeds the predetermined threshold. A CIN message is generated and sent to the ingress LER.

The following subsections show the results obtained when FATE is present in various case studies:

- ♦ Renegotiate then transfer to a higher buffer stream along existing path;
- ♦ Issue Status Request, Renegotiate and then transfer to a higher buffer stream along existing path;
- ♦ Negotiate then transfer to an alternative path;
- ♦ Issue Status Request, Negotiate and then transfer to an alternative path.

In all simulations the results shown are the average of 10 runs with different RNG seeds (Appendix A.3).

5.4.1.1 Transfer to a Higher Buffer stream

This scenario has been set up to prove the FATE scheme works. When a flow traversing a LSR experiences congestion, FATE is activated and the flow is transferred to;

1. A higher buffer stream along the existing path first instance;

2. An alternative path.

Figure 5.25 shows the loss experienced by the LSP before and after renegotation takes place with and without FATE present.

In this scenario on receiving a CIN message the ingress LER decides to renegotiate along the existing path for a higher buffer stream.

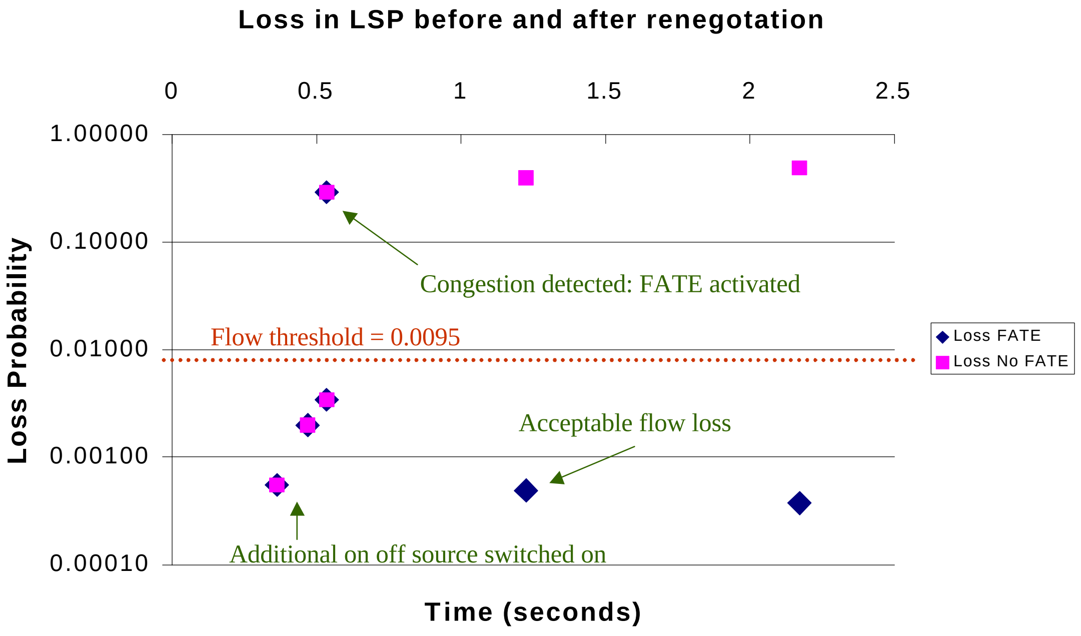


Figure 5-25 Loss in LSP before and after Renegotation

Figure 5.26 shows the times involved from the moment a CIN message is generated through to the ingress LER instigating renegotiation procedures up until when the LSP is transferred to a higher buffer stream.

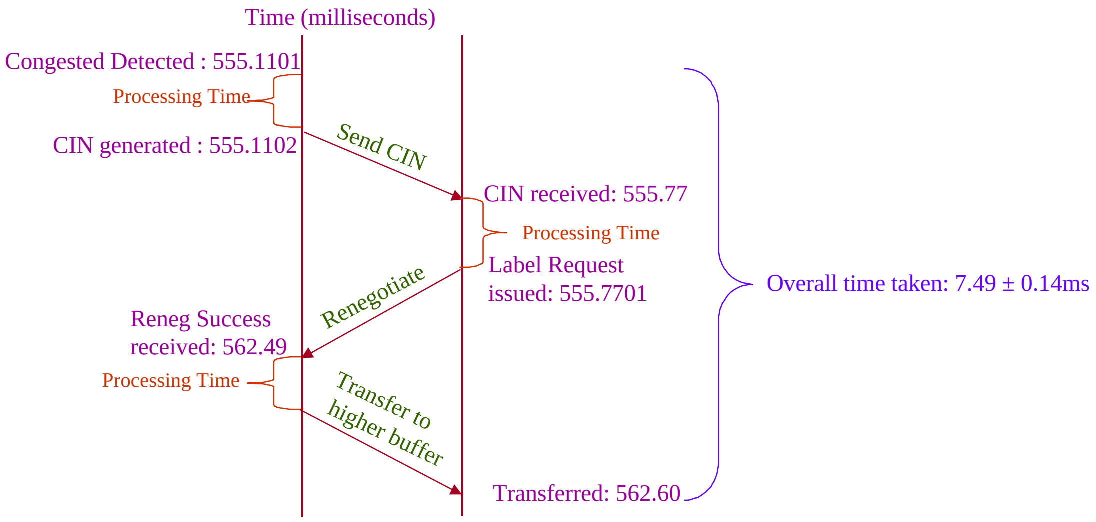


Figure 5-26 Timer between CIN generation, Renegotiation and Transferal to a higher buffer stream

Figure 5.25 shows the FATE scheme works. Congestion is detected, the ingress LER is notified and the flow is transferred to a higher buffer stream resulting in the flow experiencing an acceptably low loss probability.

Figure 5.26 shows that it takes an average of 555ms to lose 50 packets but only 7.49ms to renegotiate for a higher buffer stream. This result illustrates that FATE is *fast acting*.

In the next case on receiving a CIN message the ingress LER formulates and sends a Status Request message before choosing to renegotiate for a higher buffer stream. Figure 5.27 shows the results.

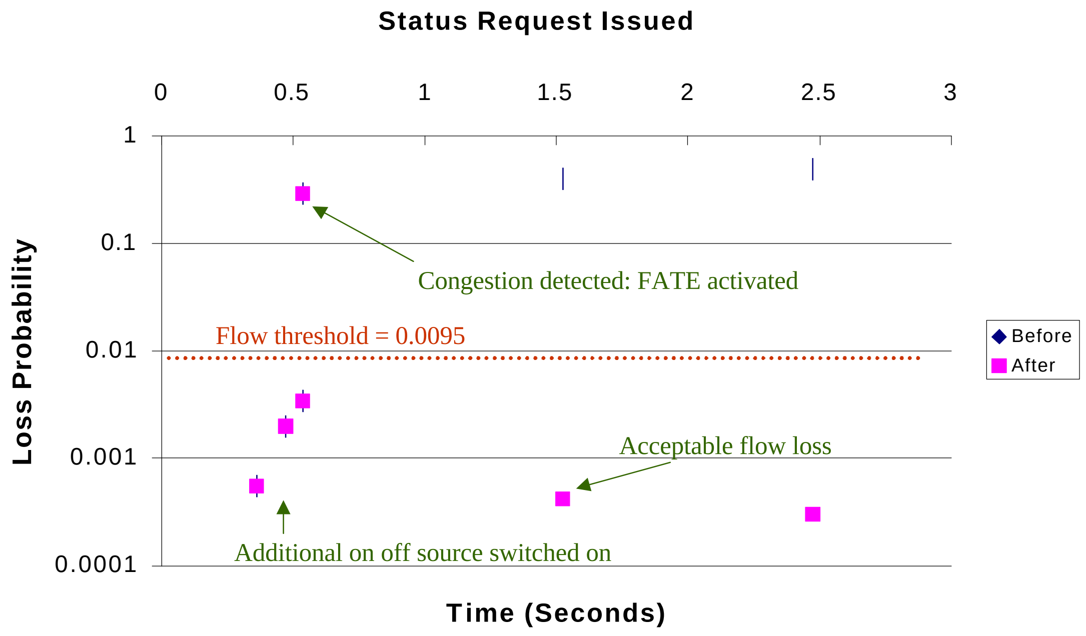


Figure 5-27 Loss before and after a Status Request is issued

Figure 5.28[32] shows the times involved from the moment the CIN message is generated, a Status Request message is generated and received until the flow is transferred into the higher buffer stream.

[32] Figure 5.25 and Figure 5.27 includes the processing times involved in message handling, however subsequent diagrams will omit this for clarity purposes, although the value is included in the calculations.

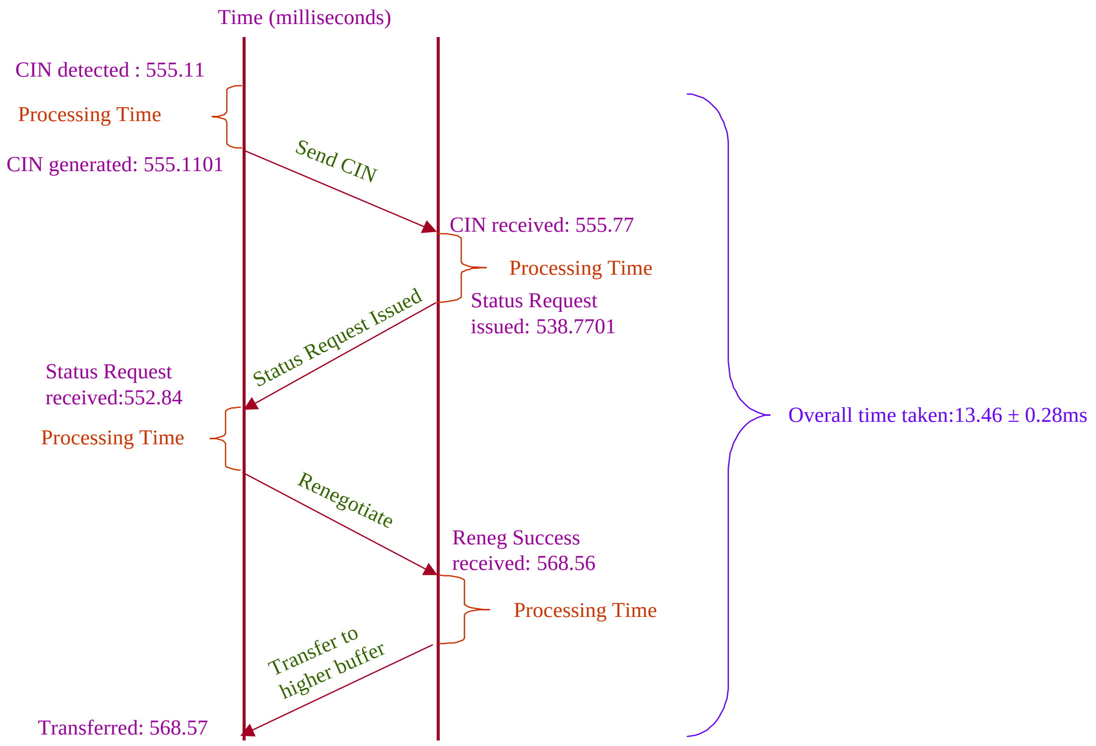


Figure 5-28 Time between CIN generation, Status Request and Transferal to a Higher Buffer Stream

Figure 5.27 shows the FATE scheme works. Congestion is detected, the ingress LER is notified, a Status Request message is issued by the ingress LER and the flow is transferred to a higher buffer stream resulting in the flow experiencing an acceptably lower loss probability.

Figure 5.28 shows that it takes an average of 555ms to lose 50 packets but 13.46ms to issue a Status Request message and renegotiate for a higher buffer stream. The results illustrates how FATE is *fast-acting.*

Figure 5.29 shows a graph of the relationship between the time at which congestion is detected and the overall time to transfer the flows into the higher buffer stream for 10 different seeds. It can be seen that the overall time to transfer the flow is independent of the actual time congestion is detected.

Although the time at which congestion is detected varies from approximately 70ms up to 900 ms the 90% confidence interval for the overall time when renegotiation occurs is small in comparison with a value of $7.49 \pm 0.14ms$ and $13.46 \pm 0.28ms$ if a Status Request message is sent before renegotiation. The difference in congestion detection time is due to whether the bursts from the sources coincide earlier or later.

Summarising and with reference to Figure 5.29 it can be seen that the overall time involved in FATE is insensitive to the time at which congestion occurs. The statistical variation of the on off sources has little impact on the overall times involved in the FATE scheme as confirmed by the horizontal characteristics of the overall time against the congestion detection time.

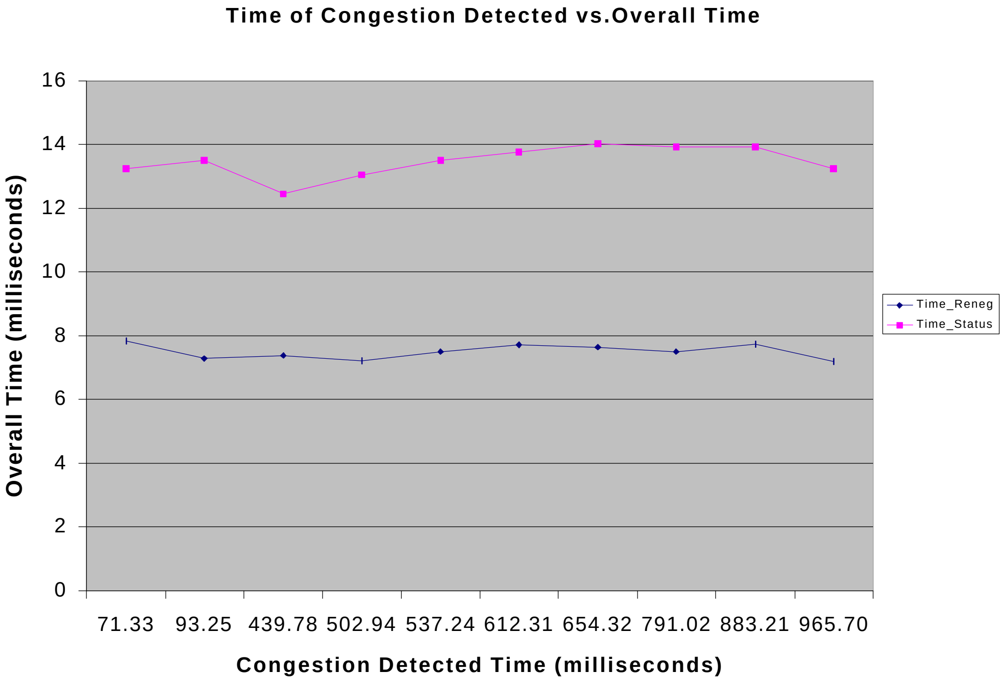


Figure 5-29 Higher Buffer Stream: Congestion Detected vs. Overall Time

5.4.1.2 Transfer to an Alternative Path

In the case where a higher buffer stream cannot be found the ingress LER negotiates for an alternative LSP.

This scenario shows the times involved when the ingress LER decides to negotiate for an alternative path on receipt of a CIN message. The times involved in transferring to an alternative path are shown in Figure 5.30 represents the average of 10 simulations run with different RNG seeds.

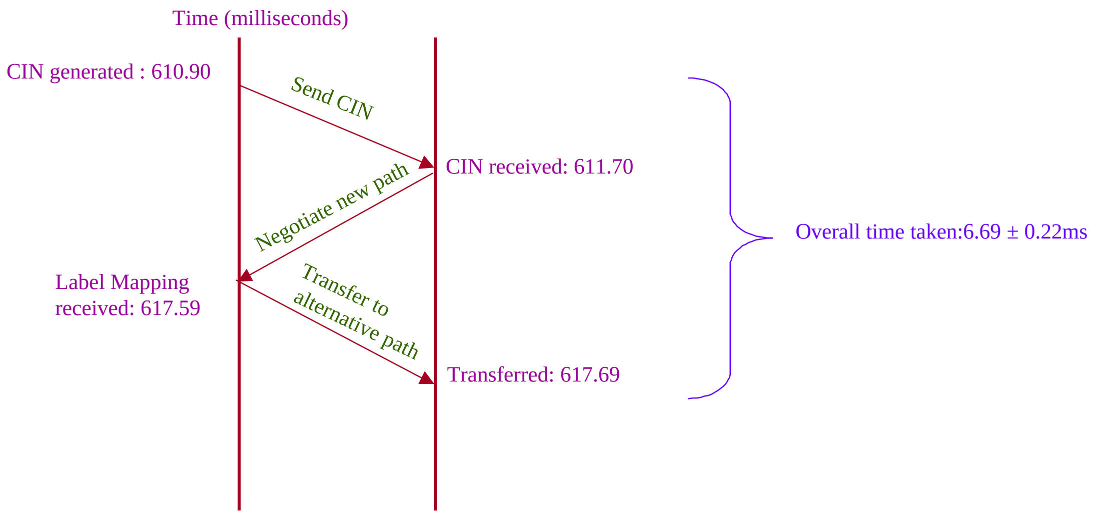


Figure 5-30 Time between CIN generation, Negotiation and Transferal to an Alternative Path

These values are comparable with those obtained when the ingress LER decides to renegotiate along the existing path for a higher buffer stream. As there is no additional processing and the number of hops the signalling messages have to traverse are the same in both cases this is acceptable.

The following scenario shows that on receipt of a CIN message the ingress LER formulates and transmits a Status Request message. On receiving a Status Request message the ingress LER takes the decision to negotiate for an alternative path. The times involved are shown in Figure 5.31.

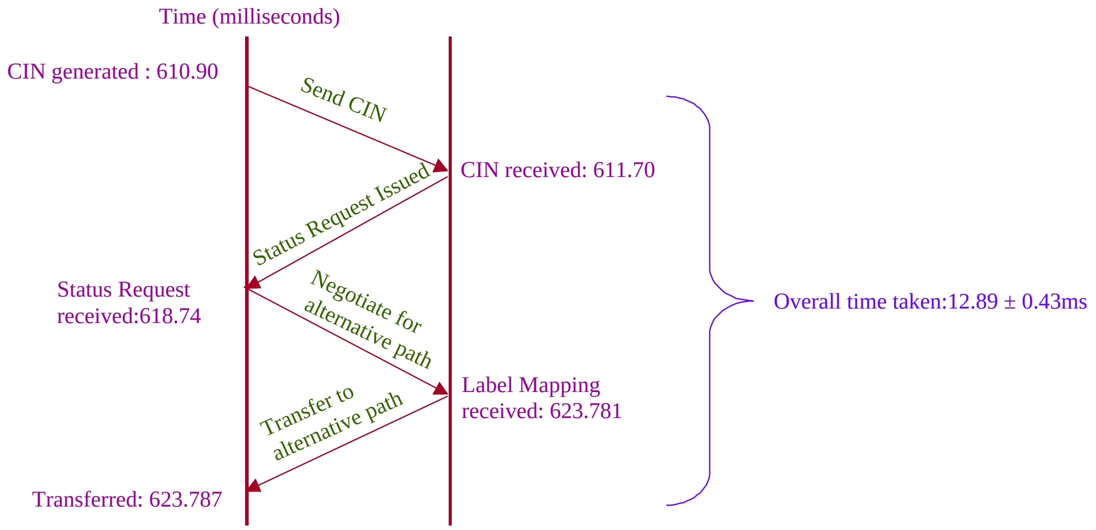

Time (milliseconds)
CIN generated : 610.90
Send CIN
CIN received: 611.70
Status Request Issued
Status Request received:618.74
Negotiate for alternative path
Label Mapping received: 623.781
Transfer to alternative path
Transferred: 623.787
Overall time taken:12.89 ± 0.43ms

Figure 5-31 Time between CIN generation, Status Request, Negotiate and Transfer to an Alternative Path

These values are comparable with those obtained when the ingress LER decides to issue a Status Request message before negotiating for an alternative path. Again as there is no additional processing involved and the number of hops remains the same the values are acceptable.

It has been noticed from the results that the values obtained when negotiating for an alternative path and renegotiating along an existing path for a higher buffer stream are comparable. This has resulted in very little noticeable difference in the overall times for FATE.

Figure 5.30 and Figure 5.31show the times involved in transferring the flows onto an alternative path. The average time taken to lose 50 packets is approximately 610ms while the times involved in transferring the flows are 7ms and 13ms for negotiation and negotiation after a Status Request message is issued respectively.

Figure 5.32 shows a graph of the relationship between the time at which congestion is detected and the overall time to transfer the flows onto an alternative path for 10 seeds. It can be seen that the overall time to transfer the flows is independent of the actual time at which congestion occurs.

Although the range of congestion detection is between 200ms and 900ms the overall time to transfer the flows onto an alternative path is approximately $6.69 \pm 0.22ms$ when the new path is negotiated for as soon as the ingress LER receives a CIN message. In the case where a Status Request message is issued before negotiation occurs the average overall time is $12.89 \pm 0.43ms$. The overall time to transfer the flow is small compared to the increase in latency due to the onset of congestion.

Summarising it can again be said that the overall time involved in FATE is insensitive to the to the time at which congestion occurs. The statistical variation of the on off sources has little impact on the overall times involved in the FATE scheme.

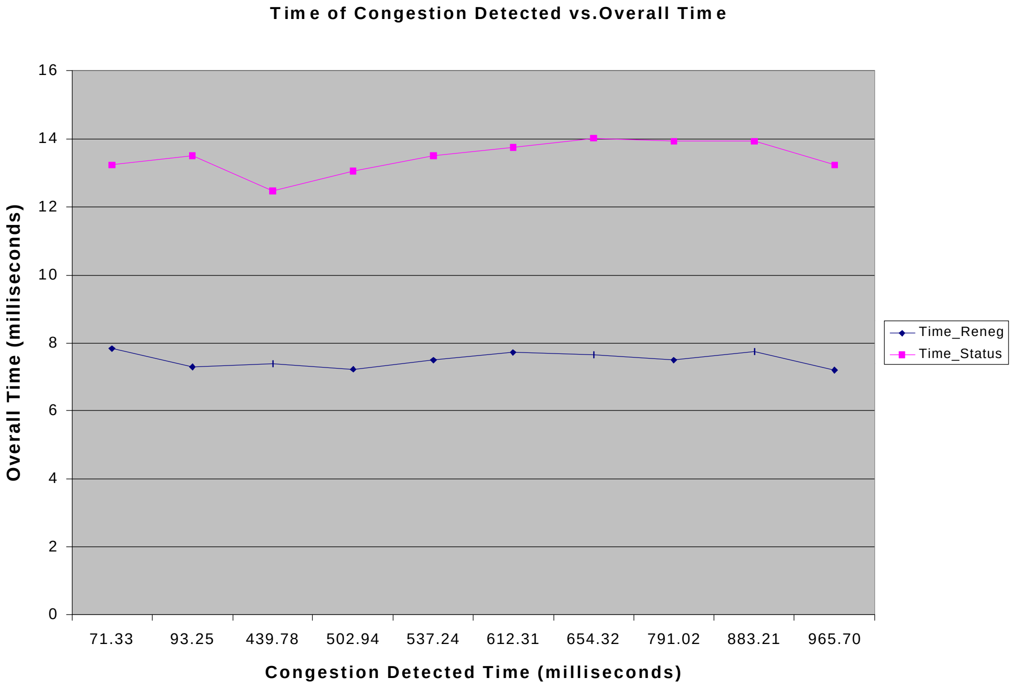


Figure 5-32 Alternative Path: Congestion Detection vs. Overall Time

5.4.1.3 Transfer back to original situation

In promoting traffic flows to a higher buffer stream the network operator is maintaining the SLA made with the customer. However, this is done at a cost to the operator, so it is in its best interest to re-map the traffic flows to their original situation once the ingress LER detects congestion has subsided.

Promotion is carried out at the expense of the network operator.

The following subsections show the results obtained when FATE is present in various case studies:

- Renegotiate then transfer to back to the lower buffer stream along existing path;
- Transfer to the lower buffer stream along existing path;

This scenario shows the times involved in re-mapping traffic flows that have been transferred into a higher buffer stream or along an alternative path back to the original situation.

In this scenario the Status Request messages are transmitted into the network periodically, every 3ms.

The ingress LER detects from the information returned by the Status Request messages that the current loss probability in the buffer the flow is traversing is acceptably low, i.e., lower than the predetermined value assigned to that buffer stream and renegotiates for the original lower buffer stream. The times involved are shown in Figure 5.33.

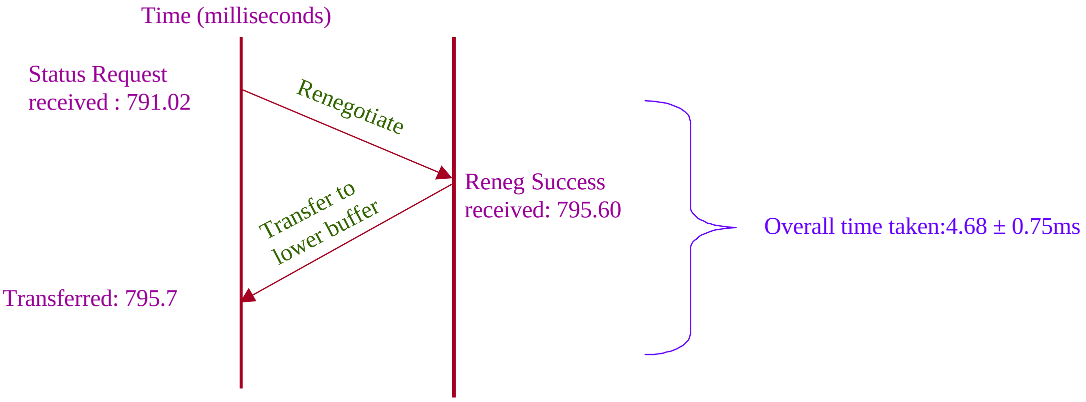

Time (milliseconds)
Status Request received : 791.02
Renegotiate
Reneg Success received: 795.60
Transfer to lower buffer
Transferred: 795.7
Overall time taken:4.68 ± 0.75ms

Figure 5-33 Transferred to the original lower buffer stream

In the next scenario on receipt of a Status Request message the ingress LER determines the loss probability experienced by the buffer through which its flow traverses is acceptable, and transfers the traffic flows back unto the original path. The times involved are as shown in Figure 5.34.

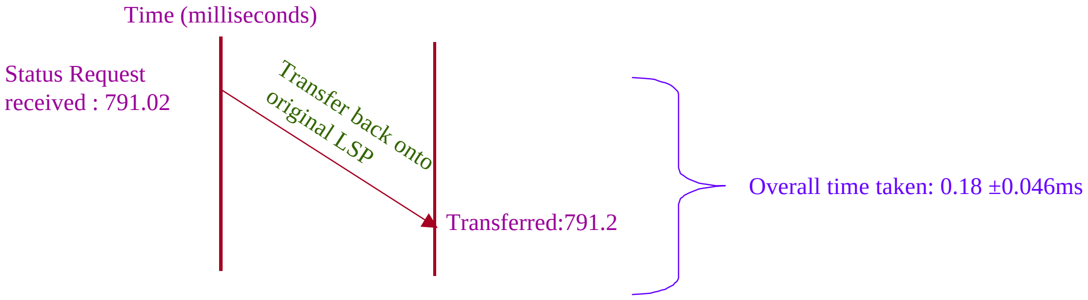

Time (milliseconds)
Status Request received : 791.02
Transfer back onto original LSP
Transferred:791.2
Overall time taken: 0.18 ±0.046ms

Figure 5-34 Transferred to original path on receipt of a Status Request message

Figure 5.33 and Figure 5.34 show the times involved in returning the traffic flows back to the original situation in response to Status Request messages that have been sent periodically into the network.

By transmitting Status Request messages into the network periodically or once a renegotation has taken place enables the ingress LER to detect when congestion has subsided and to react to it. This allows the traffic flows to be re-mapped to the first choice or optimum route. It is noted that the times involved in this process are relatively low when compared to the times needed to transfer the flows to avoid congestion.

The next subsections repeat the appropriate simulations in Scenario 1 for when congestion occurs at the ingress and the egress LERs.

5.4.2 Scenario 2

This scenario repeats the above simulations with the loss occurring within a buffer stream in the egress LER 2 for the LSP as depicted by Figure 5.35. In this scenario the path length represents the maximum number of hops the signalling messages have to traverse.

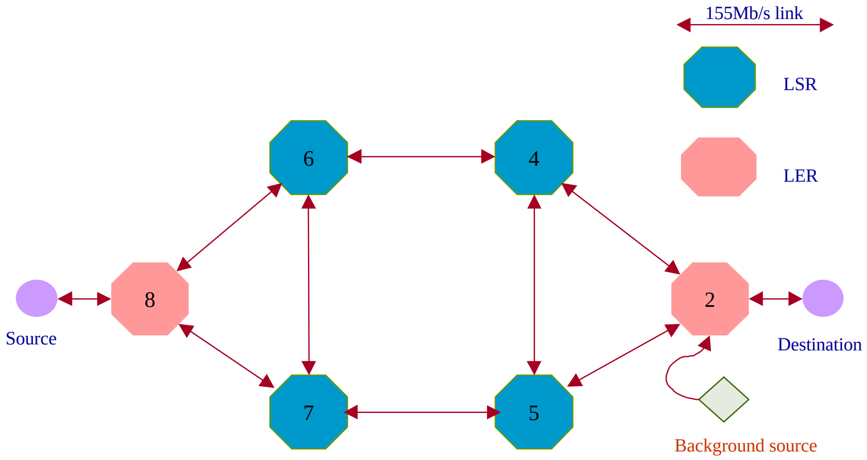


Figure 5-35 Simulation Model - Scenario 2

The following subsections depict the case studies listed:

- Renegotiate and then transfer to a higher buffer stream;
- Issue Status Request and then transfer to a higher buffer stream;
- Negotiate and then transfer to an alternative path;
- Issue Status Request and then transfer to an alternative buffer stream.

5.4.2.1 Transfer to a Higher Buffer

On detecting that a buffer is congested the egress LER formulates and sends a CIN message to the ingress LER. On receipt of a CIN message the ingress LER renegotiates along the existing path for a higher buffer stream. Figure 5.36 shows the times involved in the process.

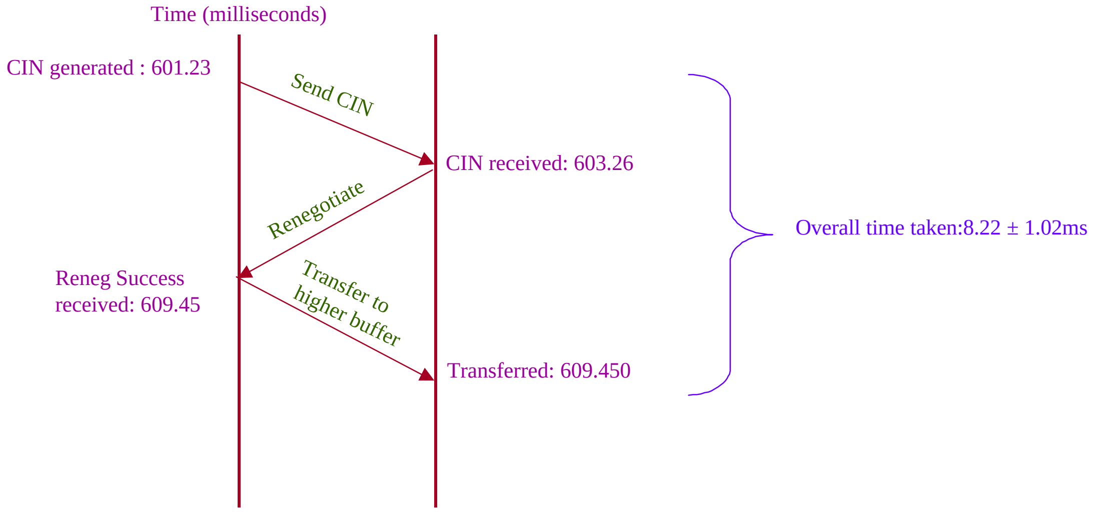


Figure 5-36 Egress LER: Time between CIN generation, Renegotation and transferral to a higher buffer stream

In the next case study, on receipt of a CIN message the ingress LER issues a Status Request message and on its return renegotiates along the existing path for a higher buffer stream. The times involved are shown in Figure 5.37.

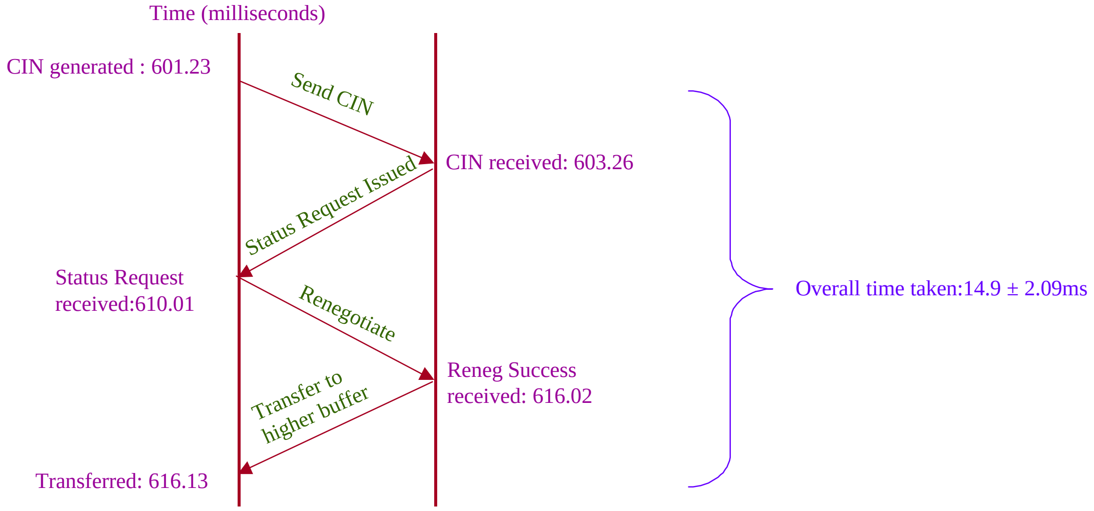


Figure 5-37 Egress LER: CIN generation, Status Request issued, Renegotiation and transferral to a higher buffer stream

5.4.2.2 Transfer to an alternative path

In this scenario on receipt of a CIN message the ingress LER negotiates for an alternative path. The times involved are shown in Figure 5.38.

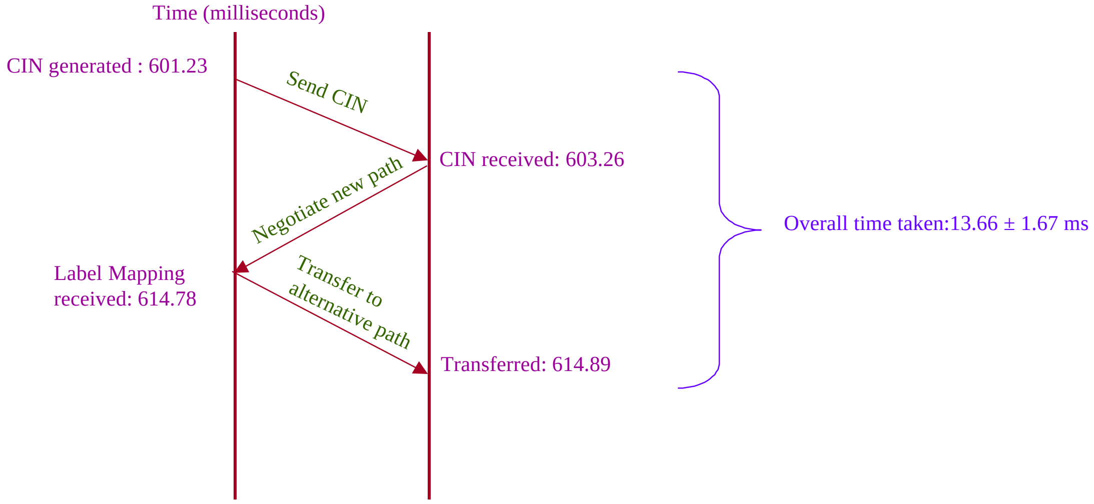


Figure 5-38 Egress LER: CIN generation, Negotiate, Transfer to an alternative path

In this case study a CIN message, the ingress LER formulates and transmits a Status Request message. On its return the ingress LER negotiates for an alternative path. The times involved in this process are shown in Figure 5.39.

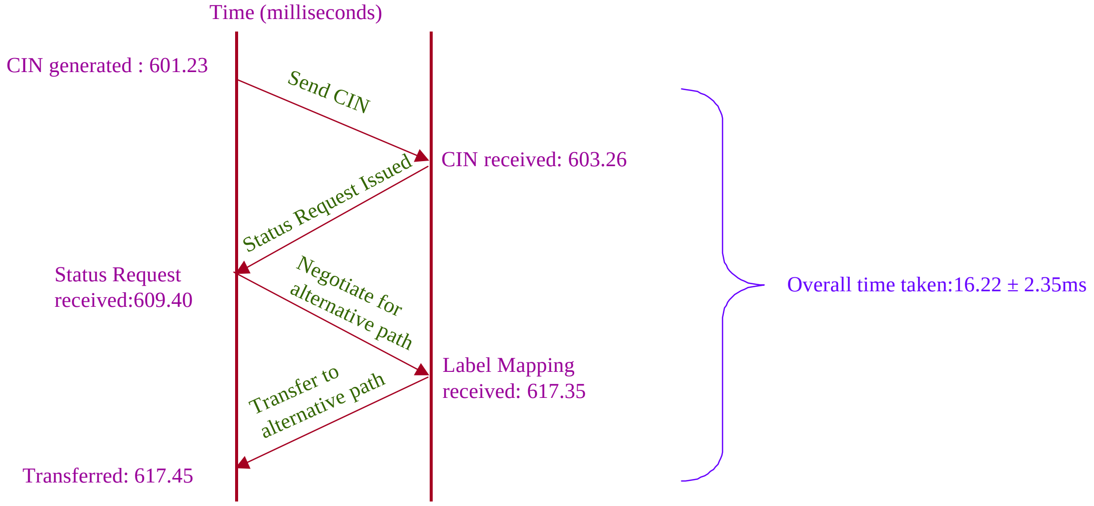

Time (milliseconds)
CIN generated : 601.23
Send CIN
CIN received: 603.26
Status Request Issued
Status Request received:609.40
Negotiate for alternative path
Label Mapping received: 617.35
Transfer to alternative path
Transferred: 617.45
Overall time taken:16.22 ± 2.35ms

Figure 5-39 Egress LER: CIN generation, Status Request, Negotiate, Transfer to an Alternative Path

Again it can be seen that the trends exhibited in the previous scenario exists here. The overall time to transfer the flow is small compared to the increase in latency due to the onset of congestion. The times involved in negotiation and renegotiation are comparable., as was expected due to the absence of additional processing. Also appearing in this scenario was the additional increase in the overall values as was expected due to the increase in the number of hops the signalling messages had to traverse between the ingress LER and the egress LER.

5.4.3 Scenario 3

This scenario repeats the simulations in Scenarios 1 and 2 with the loss occurring within a buffer stream in the ingress LER 8 for the LSP as depicted by Figure 5.40. In this case the distance between the ingress LER and the congested LSR is zero.

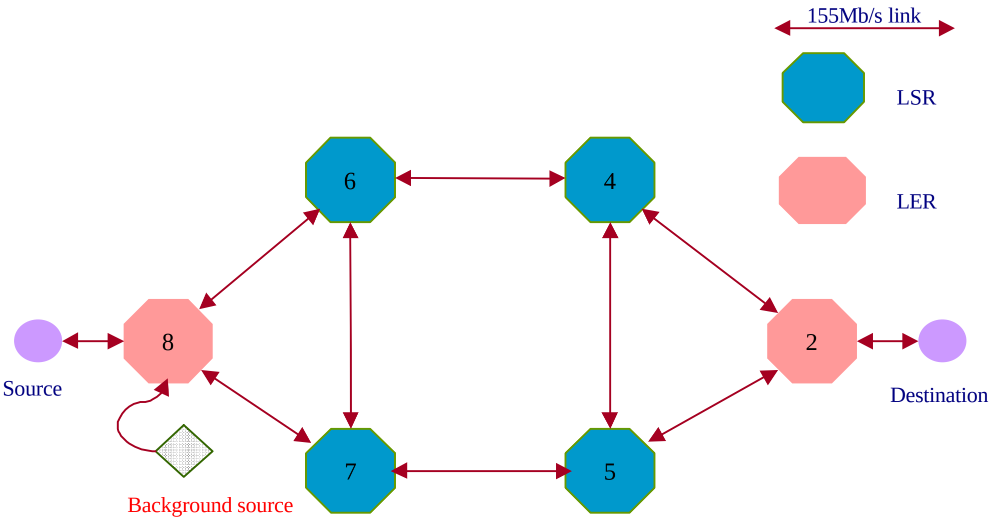


Figure 5-40 Simulation Model – Scenario 3

The following subsections depict the case studies listed:

- Renegotiate and then transfer to a higher buffer stream;
- Issue Status Request and then transfer to a higher buffer stream;
- Negotiate and then transfer to an alternative path;
- Issue Status Request and then transfer to an alternative buffer stream.

5.4.3.1 Transfer to a Higher Buffer

In this scenario on determining that it the ingress LER is congested a CIN is not generated. Instead the ingress LER renegotiates along the existing path for a higher buffer stream. Figure 5.41 displays the times involved.

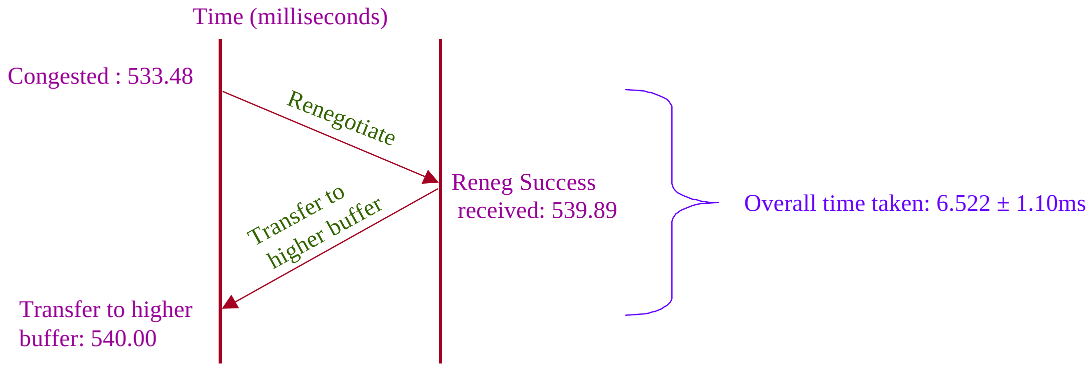

Time (milliseconds)
Congested : 533.48
Renegotiate
Reneg Success received: 539.89
Transfer to higher buffer
Transfer to higher buffer: 540.00
Overall time taken: 6.522 ± 1.10ms

Figure 5-41 Ingress LER: Congestion detected, Renegotiate and Transfer to a Higher Buffer Stream

When the ingress LER determines it is congested, it negotiates along an alternative path. Figure 5.42 shows the times involved in this process.

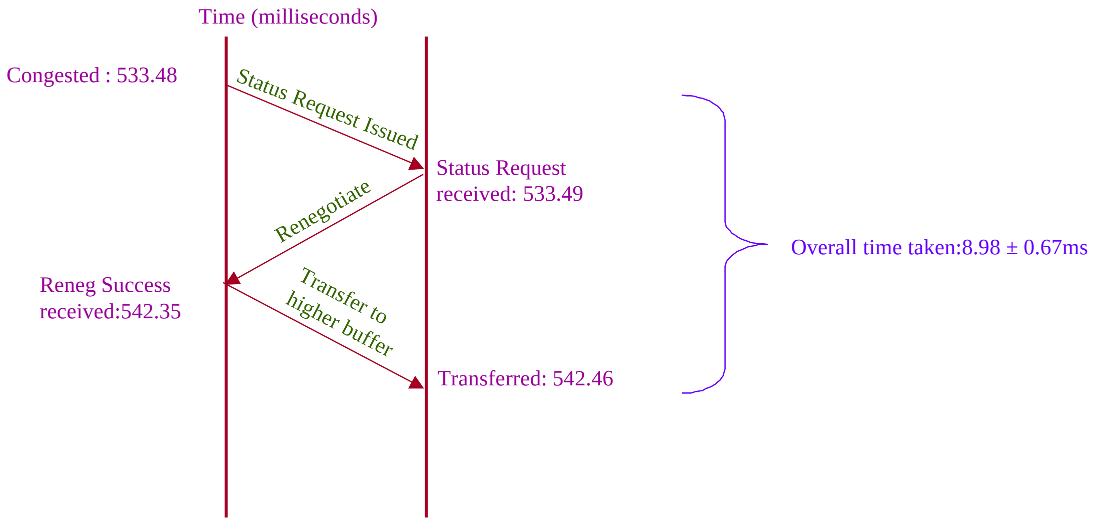

Time (milliseconds)
Congested : 533.48
Status Request Issued
Status Request received: 533.49
Renegotiate
Reneg Success received:542.35
Transfer to higher buffer
Transferred: 542.46
Overall time taken:8.98 ± 0.67ms

Figure 5-42 Ingress LER: Congested detected, Status Request and transfer to a higher buffer stream

5.4.3.2 Transfer to an Alternative Path

On detecting it is congested the ingress LER negotiates for an alternative path. The times are shown below in Figure 5.43.

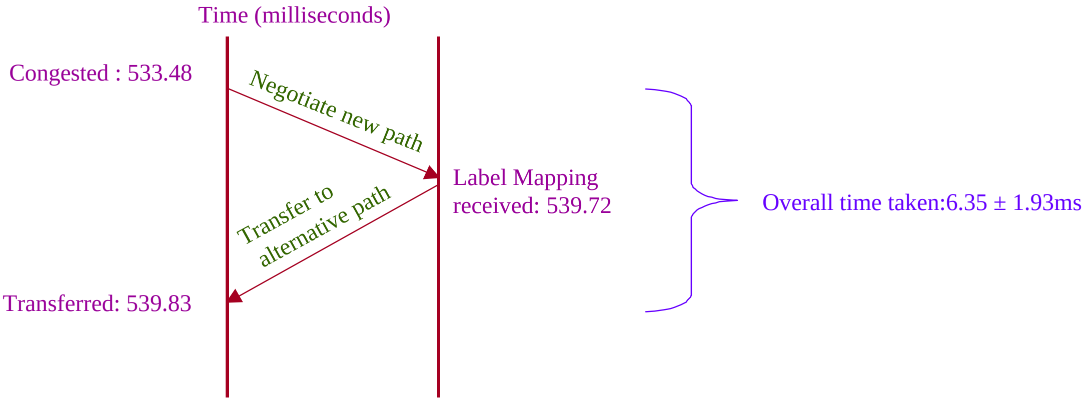

Time (milliseconds)
Congested : 533.48
Negotiate new path
Label Mapping received: 539.72
Transfer to alternative path
Transferred: 539.83
Overall time taken:6.35 ± 1.93ms

Figure 5-43 Ingress LER: Congestion detected, Negotiate and Transfer to an Alternative path

On detecting congestion the ingress LER formulates and transmits a Status Request message and decides to negotiate for an alternative path on its return. The times involved are shown in Figure 5.44.

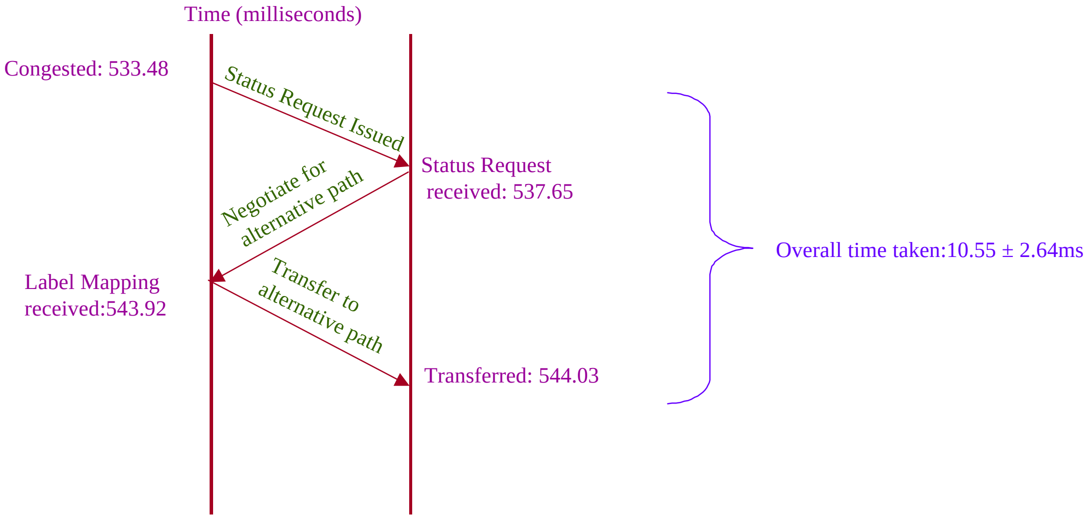

Time (milliseconds)
Congested: 533.48
Status Request Issued
Status Request received: 537.65
Negotiate for alternative path
Label Mapping received:543.92
Transfer to alternative path
Transferred: 544.03
Overall time taken:10.55 ± 2.64ms

Figure 5-44 Ingress LER: Congested detected, Negotiate and transfer to an alternative path

Again it can be seen that the trend exhibited in the previous scenario exists here. The overall time to transfer the flow is small compared to the average time for congestion to occur.

In all three scenarios for simulation purposes the propagation delay was considered to be zero. However, considering a real terrestrial system where the for each 1000 kilometres the delay is 5ns. For the maximum number of hops considered above the maximum would e approxiametly 30ns. This value is very small and shows that FATE adds little delay to a realistic network.

5.4.4 Scenario 4

In this scenario a comparison is made in the time it takes the ingress LER to respond to receiving Status Request messages it has sent into the network, and in receiving CIN messages from a congested LSR.

5.4.4.1 Status Respond versus CIN Respond

The information returned by the Status Request message contains the loss probability currently being experienced in all the buffer streams within each LSR along the LSP.

If the value contained within the Status Request message exceeds a value assigned to a buffer stream i.e., minimum dimensioned loss probability threshold, this is taken by the ingress LER as a possible sign of congestion in the buffer stream of that LSR.

This simulation compares the response times of the ingress LER to receiving a Status Request messages indicating congestion is occurring in a flow initiated by the ingress LER and on receiving a CIN message from the congested LSR.

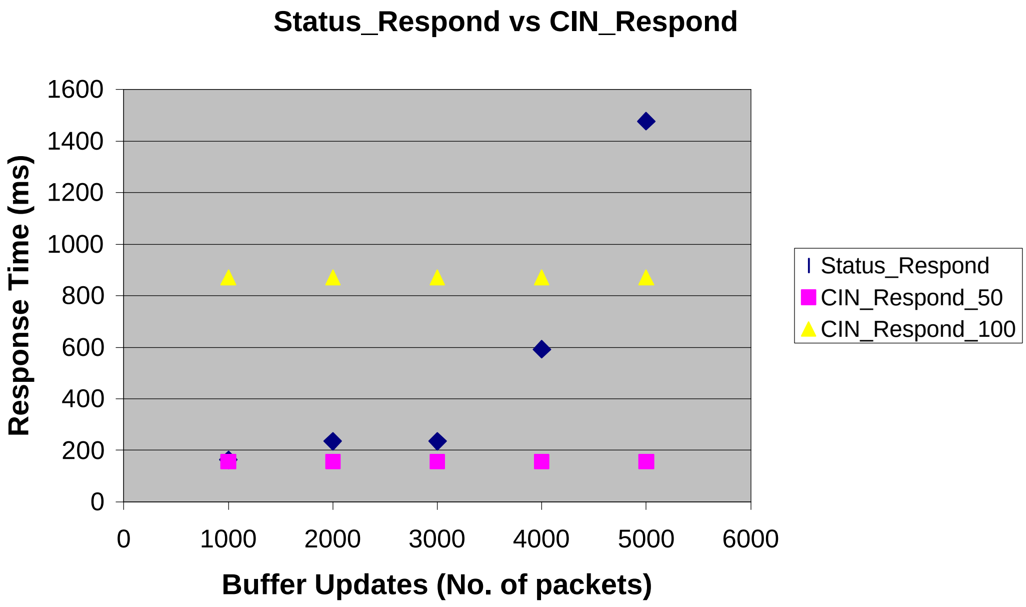


Figure 5-45 Status Respond vs CIN Respond

Figure 5.45 shows the notification times involved in comparing the ingress LER's response to receiving Status Request and CIN messages for different packet losses of 50 and 100.

The time at which the ingress LER receives indication of congestion is comparable at lower packet losses and low buffer table updates with both the Status Request and CIN message receipts. With low buffer table updates congestion detection will occur sooner than for larger buffer table updates, and almost equals the time at which the actual congested LSR generates a CIN message.

For low buffer table updates and high packet losses the Status Request message outperforms the CIN message by as much as 4 times. The Status Request messages outperform the CIN message because at lower buffer table updates congestion detection occurs quicker than the timer taken for the LSR to accrue high packet losses. However, for high buffer updates the CIN message indicates probable congestion by as much as 7.5 times over the Status Request message mechanism. This trend results in the LSR loses the required number of packet losses at a quicker rate than it takes the buffer stream to detect loss.

It can also be seen that the CIN notification is independent of the value of the buffer table updates, and the Status Request messages are independent of the number of packet losses.

Summarising, the simulation results have shown that there is more than one suitable means of detecting congestion, both of which are independent of each other. One way is to allow the ingress LER to issue Status Request messages periodically or for it to be informed by the congested LSR via CIN messages.

Both Status Request and CIN messages were employed within the FATE simulations for different purposes.

5.4.5 Scenario 5

In this scenario the effects of changing the on off source’s traffic profile is investigated to see if it effects the performance of FATE. In this simulation the ‘burstiness’ of the source is defined by the ratio of the on to off times as explained previously in Section 5.3. Table 5-8 shows how the ‘burstiness’ of the source is represented.

On time	***Off Time***	***Burstiness = Off / On***
0.00125	0.00125	1
0.001	0.0015	1.5
0.00075	0.00175	2.3
0.0005	0.002	4
0.00025	0.00225	9

Table 5-8 Source’s Burstiness Representation

5.4.5.1 Number of CIN receipts before ingress LER responds

This simulation is configured as explained previously in Scenario 1 with an additional on off source being fed into LSR 6 to cause the loss probability in the flow to exceed the predetermined threshold.

To dampen the response of FATE i.e., to prevent the ingress LER from responding to each and every CIN message it receives and thus cause path selection flapping, the author has proposed two possibilities.

- ♦ Varying the number of CIN messages the ingress LER responds to;
- ♦ The inclusion of a CIN Timer.

This simulation investigates the effects of both of those proposals.

The number of CIN messages the ingress LER has to receive before responding is predetermined and varied throughout the simulations. The time before the ingress LER responds to the indication of congestion is recorded with and without a CIN timer being present in the network.

In the scenario where the CIN Timer is present, it is set to 1ms with the number of packets being lost before the buffer loss probability is calculated, being set to 50.

Every 50-packet losses the buffer loss probability is calculated, if it exceeds a predetermined threshold the flow loss probability is calculated. If the loss in the flow has exceeded a predetermined threshold a CIN message is generated and the CIN Timer is set. When the CIN Timer expires the loss probability in the flow is recorded over the last time period, if it has exceeded the predetermined threshold a CIN message is generated and the CIN Timer is set.

In the case where the CIN Timer is not present, every 50 packet losses the loss probability in the buffer is calculated, if it exceeds the predetermined threshold the loss in the flow is also calculated, if it too exceeds a predetermined threshold a CIN message is generated. Every 50 packet losses the procedure is repeated.

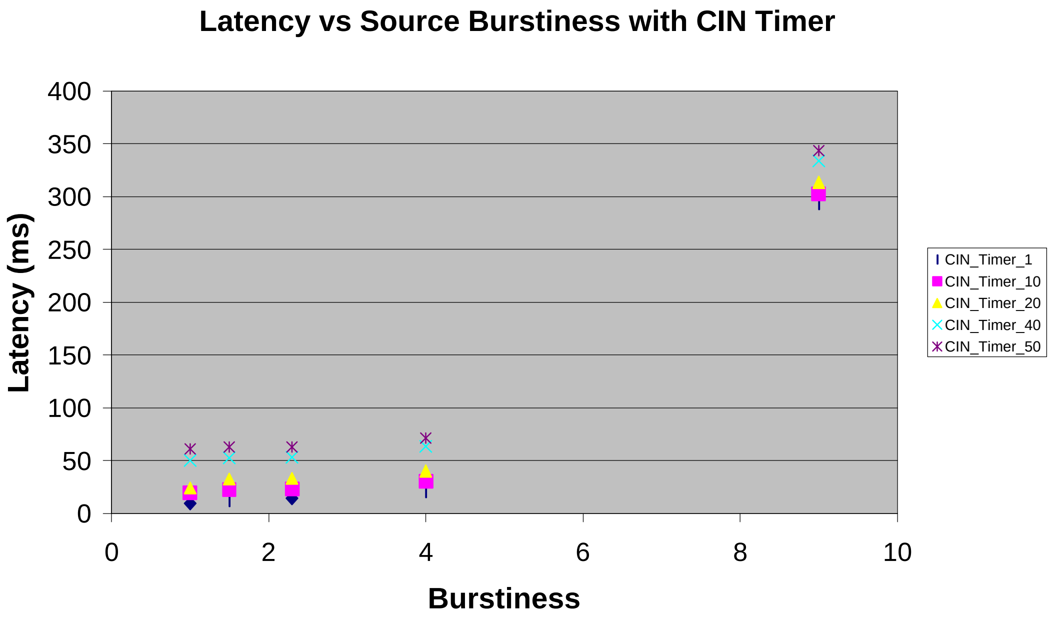


Figure 5-46 Latency vs. Burstiness with CIN Timer

Figure 5.46 shows the latency involved in the ingress LER responding to congestion detected for different values of CIN receipts e.g., 1,10,20..50 with the CIN Timer present.

It can be seen that when the sources are less bursty the required number of CIN messages are quicker than the more bursty sources. Without the simulation results the author would expect the more bursty sources to lose a larger number of packets and therefore produce the required number of CIN messages in a shorter period of time. However the simulation results prove otherwise. This may be due to the burst periods in the less bursty sources overlapping in the sources resulting in burst scale queueing occurring i.e., a large number of packets arrive at the buffer at a rate that exceeds the service rate resulting in the packets being lost. Whereas in the more bursty sources the probability of the bursts overlapping in the sources is smaller.

At low values of CIN receipt the times involved are comparable for all types of traffic profiles. This may be attributed to the fact that when a burst occurs the number of packets required to generate a CIN message are lost. However at higher values of CIN receipt and for the more bursty sources the times involved in generating the CIN messages increases. This may be due to the point made earlier about there being a lower probability of burst scale queueing occurring resulting in a longer time before the required number of losses occur.

The same trend is noted when the CIN Timer is not present as shown in Figure 5.47. However at lower values of CIN receipts and for the more bursty source the times are higher than without the CIN Timer present. This is due to the CIN Timer presence detecting that congestion still exists every 1ms whereas without the CIN Timer the required number of packets needs to be lost before the loss probability is calculated.

Error bars are omitted in the figures for clarity.

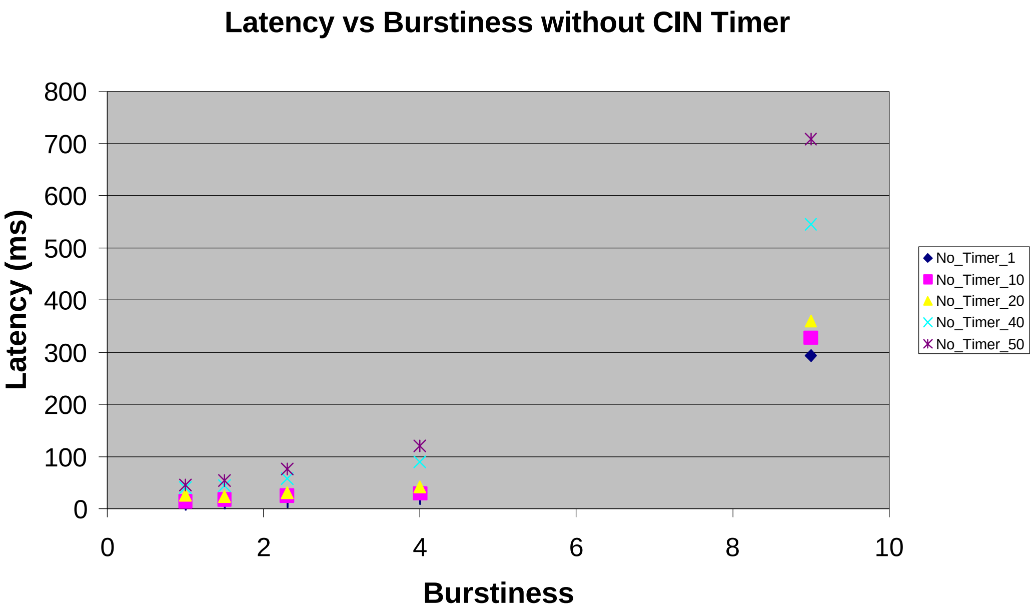


Figure 5-47 Latency vs. Burstiness without CIN Timer

The latency involved in the ingress LER receiving the required number of CIN messages is dependent on two components as shown in Figure 5.48.

- The delay for the number of losses to occur which in turn is dependent on the 'burstiness' of the source;
- The delay in transmitting the CIN message from the congested LSR back to the ingress LER. This component of delay is relatively constant as expected for all the traffic profiles of the on off source as the default on off source is multiplexed with the signalling messages.

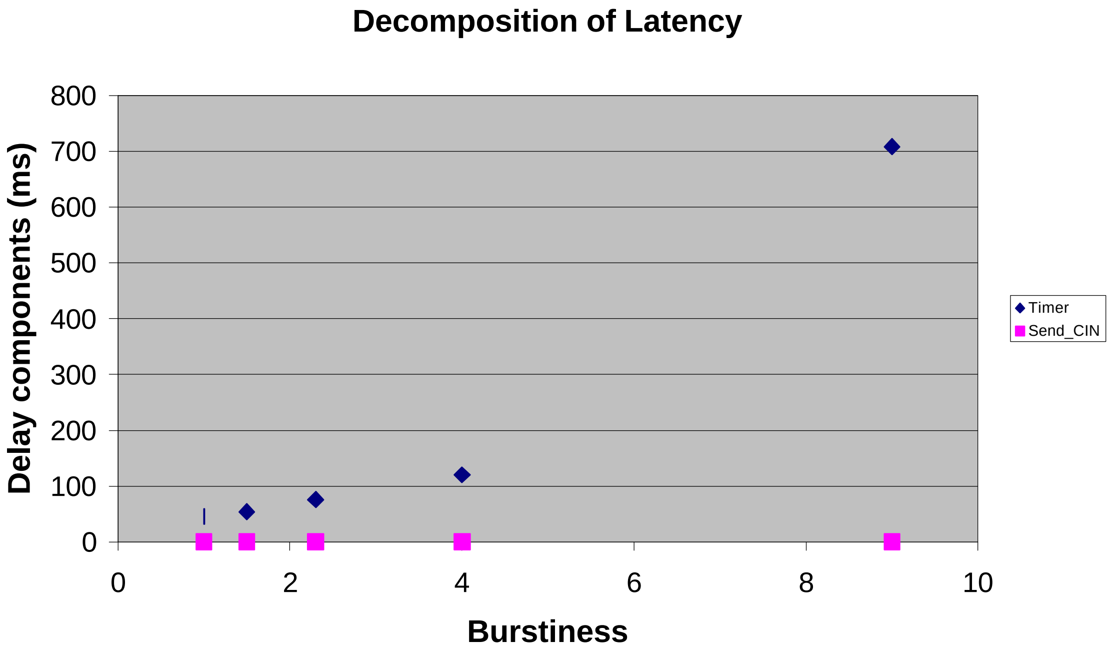


Figure 5-48 Delay components of Latency

Figure 5.48 shows the two delay components involved in the time it takes the ingress LER to respond. There is a difference in the time it takes to accumulate the 50 losses as would be expected for the traffic profiles of the sources.

As the burtiness of the source increases so does the latency involved in accumulating the required number of packets lost before a CIN message is generated. This again is due to the bursts overlapping in the less bursty sources and not in the more bursty sources.

The times involved in issuing and receiving a CIN message are almost equal at approximately $0.56 \pm 0.051ms$. This almost constant value is expected, as the source profile of the background signalling messages remains the same for all the sources. The delay in transmitting the CIN messages is quite small as the signalling traffic is given high priority in terms of scheduling, therefore they don't anticipate encountering much queuing delay. However, this value is dependent on how loaded the signalling buffers are.

The following individual graphs show the response times for different values of CIN receipts with and without the CIN Timer present.

Figure 5.49 shows that the ingress receives the first CIN message at the same time with and without the CIN Timer present.

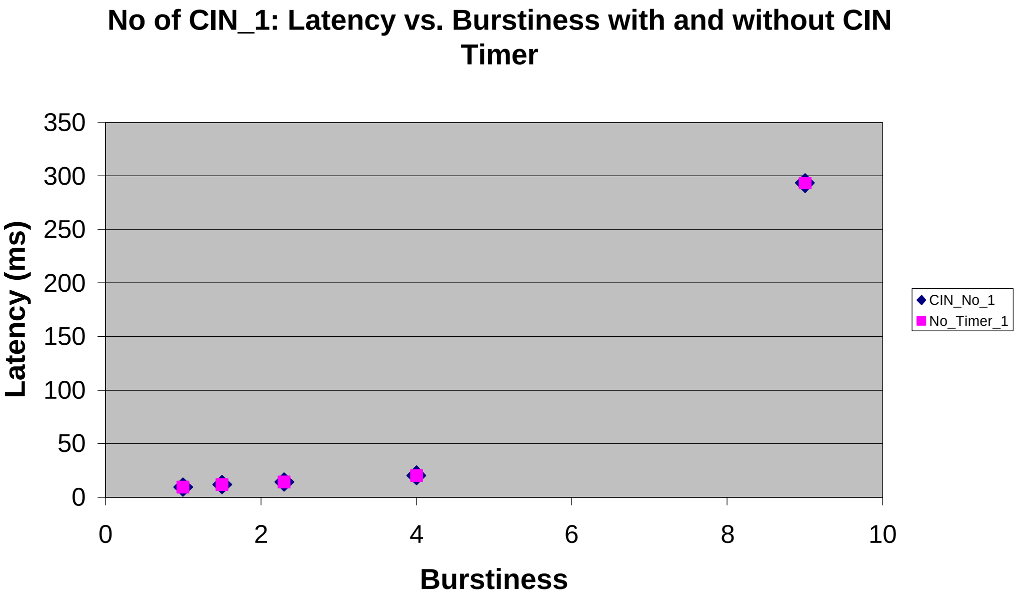


Figure 5-49 No_of_CIN_1

The times involved in the ingress LER receiving one CIN message is identical with and without the CIN Timer present. This is expected as the first CIN message has to be generated before the CIN Timer is set.

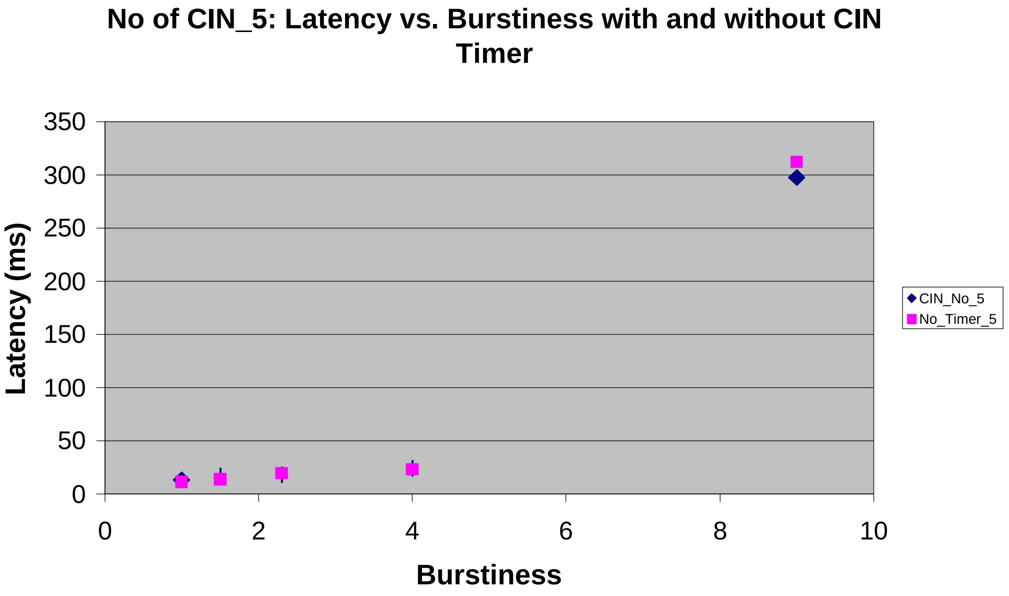


Figure 5-50 No_of_CIN_5

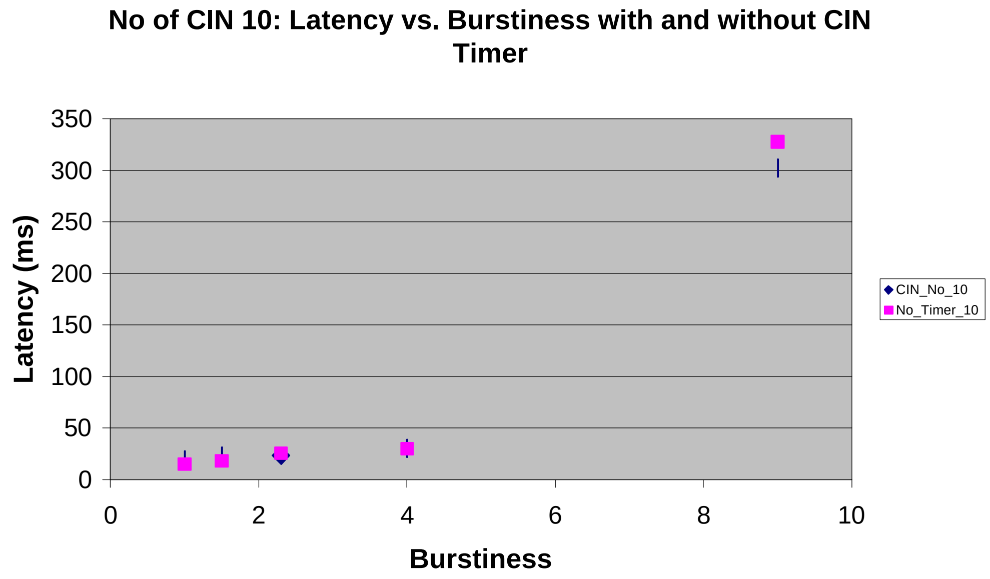


Figure 5-51 No_of_CIN_10

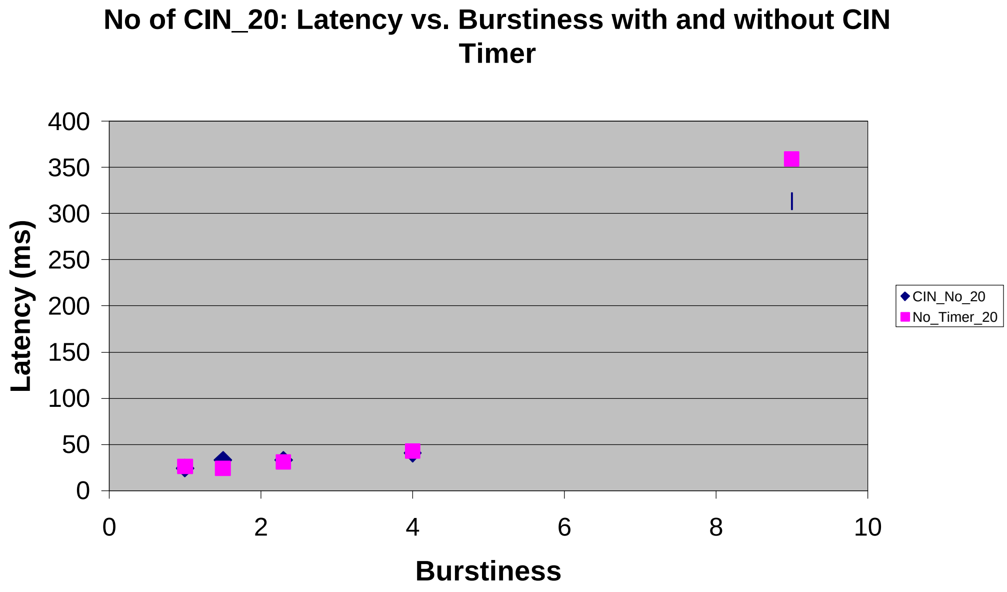


Figure 5-52 No_of_CIN_20

With CIN receipt values less than 20 the response times with and without the CIN Timer are comparable.

With higher values of CIN receipts and for burstier sources there is a noticeable difference in the response times with and without the CIN Timer as shown in the following graphs.

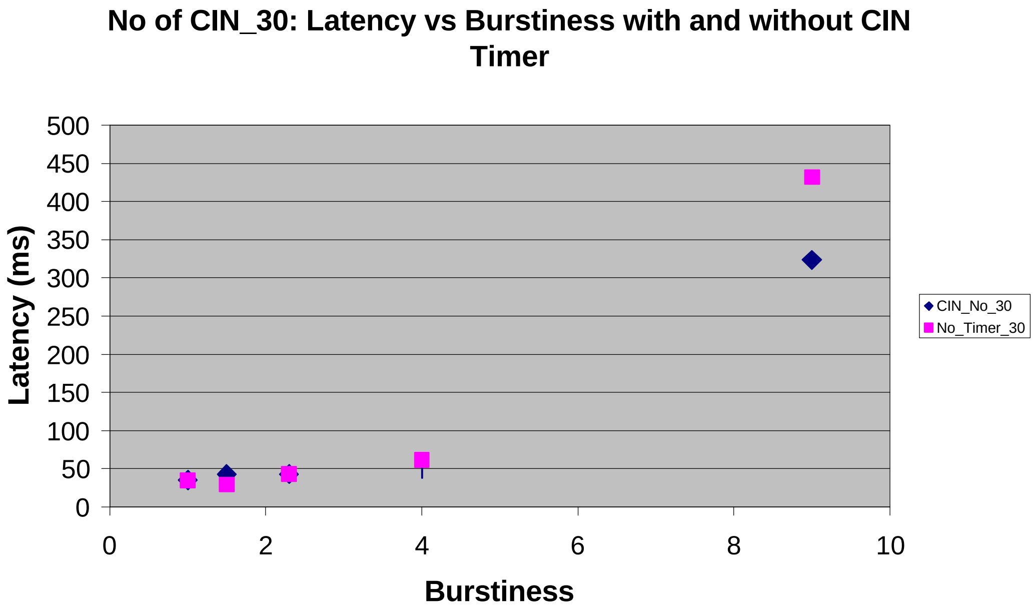


Figure 5-53 No_of_CIN_30

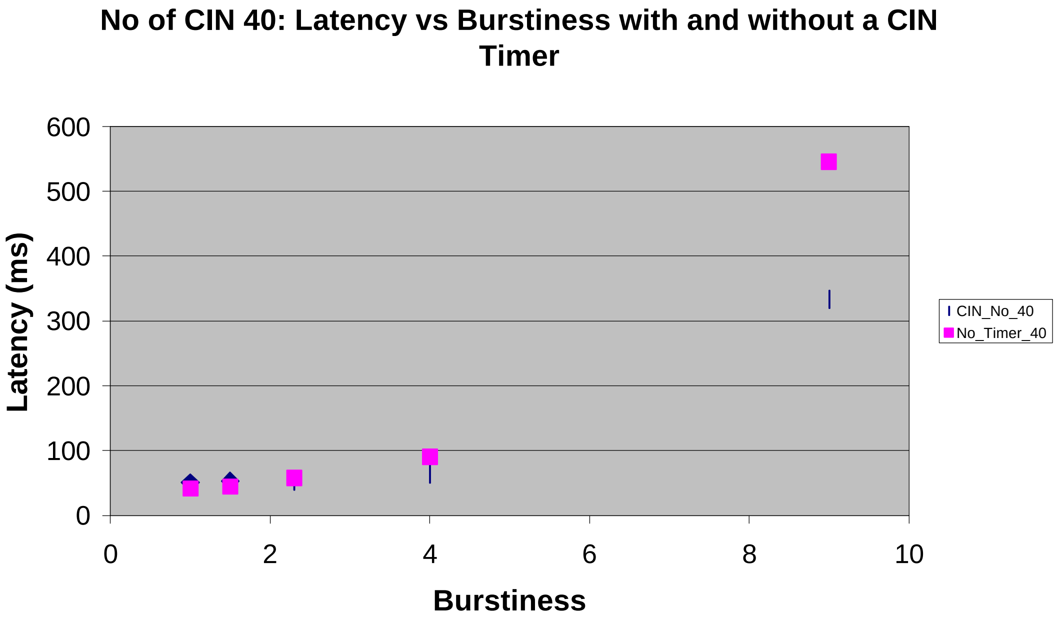


Figure 5-54 No_of_CIN_40

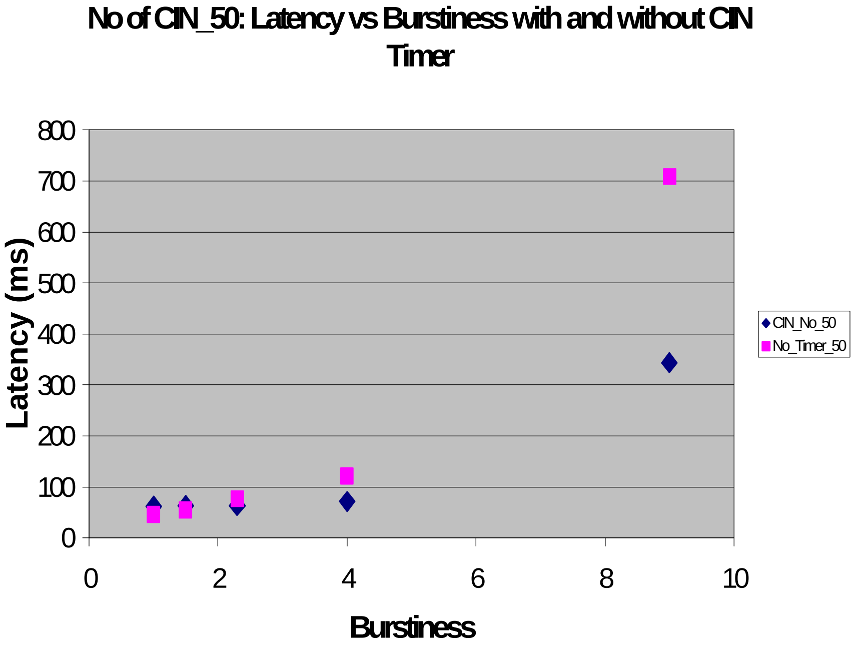


Figure 5-55 No_of_CIN_50

There is a general trend in the results obtained in the relationship with and without the CIN Timer present. The more bursty sources show a marked difference between the time it takes to receive the same number of CIN messages with and without the CIN Timer. With the CIN Timer present the congestion can be detected quickly. Without the CIN Timer the ingress LER has to wait a longer period of time before it is informed that there is congestion in the network.

However the less bursty sources show little difference with the CIN Timer's presence. A probable reason could be that the timer expires at approximately the same time as the source loses the required 50 packets.

As the burstiness of the sources increases the time required for the ingress LER to respond to the appropriate number of CIN messages increases. This trend is replicated in all values of CIN receipts. Initially the expectation would be for burstier sources to respond quicker to packet losses and to generate CIN messages at a faster rate than the less bursty sources. However the less bursty sources respond at least six times as fast.

This is attributed to the following reason:

The less bursty sources are likely to spend more time in the on state than the more bursty sources. When there are a number of these sources multiplexed together there is a higher probability that there on times will coincide resulting in their total transmission rate exceeded the service rate of the buffer thereby resulting in greater packet losses. With the burstier sources as their on times last for a shorter period of time they are less likely to coincide resulting in very few packet losses occurring during their on time.

Another difference is noted in performance with and without the CIN Timer with the CIN Timer set at 1ms. For the less bursty sources the time for the ingress LER to respond to receiving a CIN message is comparable with and without a CIN Timer. The CIN Timer's value of 1ms coincides with the time it takes the network to lose the appropriate number of packets.

With the more bursty sources and at higher numbers of CINs generated by the congested LSR there is a noticeable difference in the times to receive the CIN messages with and without the CIN Timer present. The time before the ingress LER reacts is less without a CIN Timer than with one. This may be because the number of packets required to be lost before generating a CIN occurs before the CIN Timer expires and the loss is noted.

With an increase CIN Timer value to 20ms there is a noticeable difference in the ingress LER response times. The increase in delay is almost 16 fold with an increase in the CIN Timer value. Increasing the CIN Timer value results in a noticeable delay in notifying the ingress LER of congestion as can be seen in Figure 5.56.

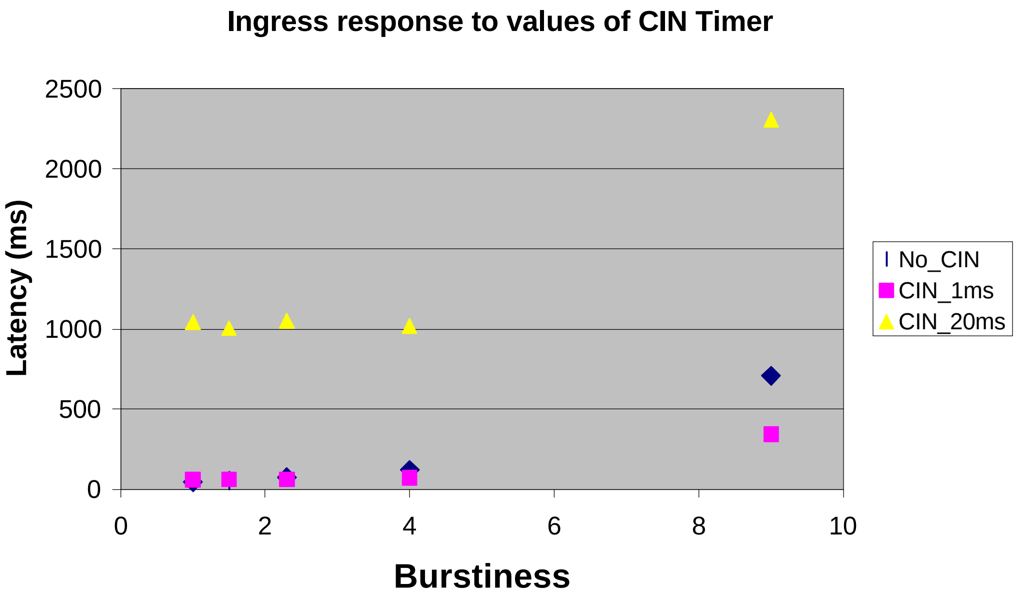


Figure 5-56 Ingress Response with change in CIN Timer value

The action taken upon the expiration of the CIN Timer is configurable. As seen in in the results one approach upon the timer expiring is that the loss probability is calculated immediately and if it exceeds the predetemined loss threshold a CIN message is generated. Another approach once the timer has expired is to start counting the number of losses afresh and when a predetermined number of losses have arisen then perform the loss probability calculation. The later acts as a dampening effect making it less responsive, whereas the former makes the scheme can make the scheme more responsive.

5.4.5.2 Number of CINs received by the ingress LER

This simulation is configured as explained previously in Scenario 1 with an additional on off source being fed into LSR 6 to cause the loss probability in the flow to exceed the predetermined threshold.

A further proposal to dampen the response of the network is to vary the number of CIN messages the ingress LER responds to for different numbers of packet losses.

This simulation is run for a 100ms-time period. In that time the number of CIN messages the ingress LER receives with and without a CIN Timer present are recorded for differing values of packet losses.

For the scenario where the CIN Timer is present its value is set to 1ms. Every X-packet losses the buffer loss probability is recorded, if it exceeds a predetermined threshold the flow loss probability is recorded. If the loss in the flow has exceeded a predetermined threshold a CIN message is generated and the CIN Timer is set. On expiring the loss probability in the flow is recorded over the last time period, if it has exceeded the threshold a CIN message is generated and the CIN Timer is set.

In the case where the CIN Timer is not present, every X packet losses the loss probability in the buffer is recorded, if it exceeds the threshold the loss in the flow is recorded, if it too exceeds a threshold a CIN message is generated. Every X packet losses the procedure is repeated.

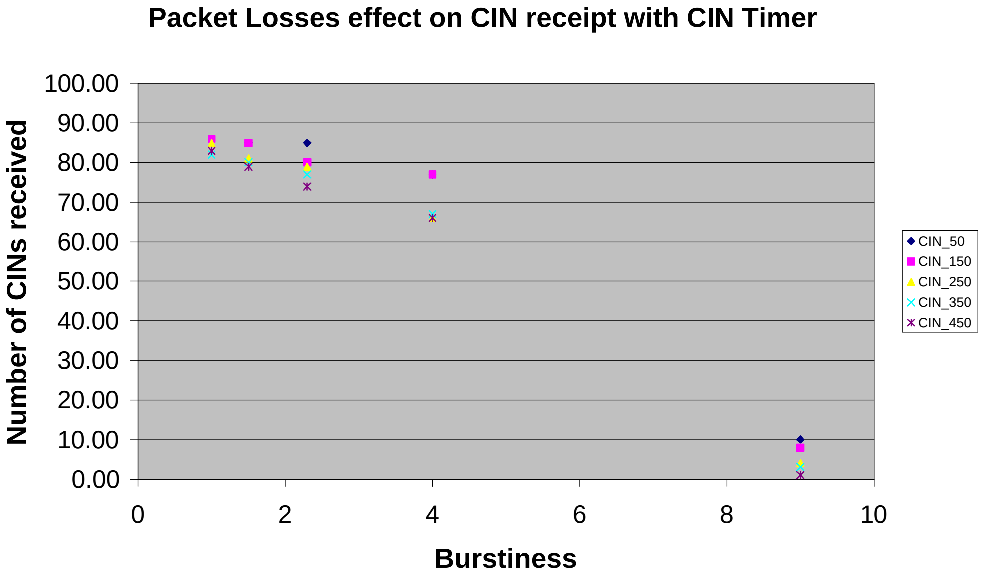


Figure 5-57 The general trend of different packet losses on the number of CINs received with a CIN Timer

Figure 5.57 and Figure 5.58 shows the results obtained with and without a CIN Timer present for different values of packet losses.

There is a general trend observed in both graphs. As the burstiness of the sources increases the number of CIN messages received by the ingress LER decreases. This is expected when the graph is compared with the response times experienced by the ingress LER in the previous set of simulations. The more burstier the source the smaller the probability of the bursts overlapping therefore the longer the time to accumulate the necessary packet losses resulting in fewer CIN messages being generated.

With the CIN Timer present the number of CIN messages received by the ingress LER is comparable for all the values of packet losses selected. This is due to a large number of packets being lost at the same time equalling all the selected values of packet losses.

In general the lower packet losses generate a larger number of CIN messages than the higher packet losses. This trend is more noticeable in the simulation runs without the CIN Timer present for the lower bursty sources. The number of CIN messages received by the ingress LER for the more bursty sources are almost equal. The CIN Timer's presence results in a higher number of CIN messages being generated, this is because every 1ms the CIN Timer detects congestion still exists and notifies the ingress LER. Without the CIN Timer the a certain number of packets has to be lost before the ingress is informed. This trend can be seen for all values of packet losses as the following graphs illustrate.

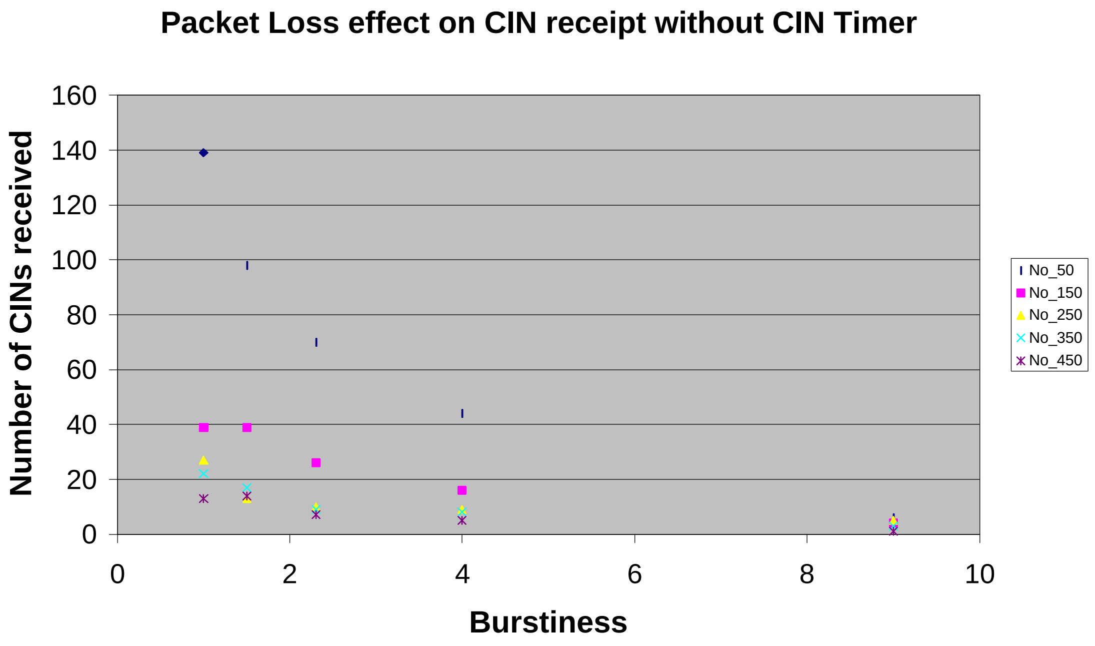


Figure 5-58 The trend of different number of packet losses on the number of CINs received without CIN Timer

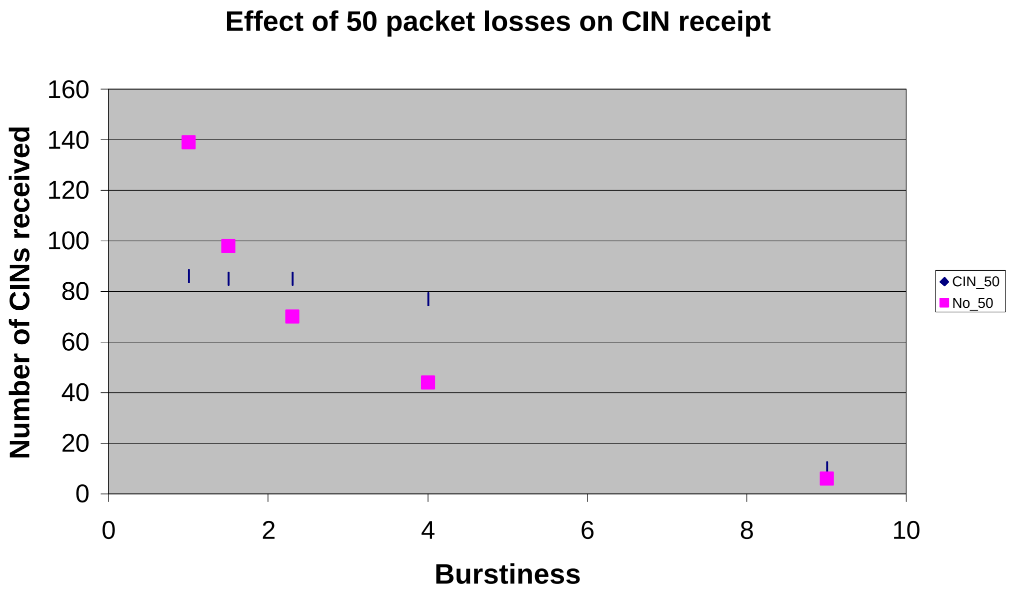


Figure 5-59 50 packet losses

Figure 5.59 shows that for low packet losses the number of CIN messages received is fairly constant when a CIN Timer is present for the lower bursty sources. The CIN Timer has a normalising effect on the number of CINs generated. In the situation where the required number of packet losses occur, once the CIN Timer has been set no action can be taken i.e., no CIN messages are generated until it expires. However, without the CIN Timer present CIN messages are generated each time the required number of losses occurs resulting in the higher number of CINs received by the ingress LER. The time it takes the source to lose the required number of packet losses is significantly smaller than the value of 1ms set for the CIN Timer.

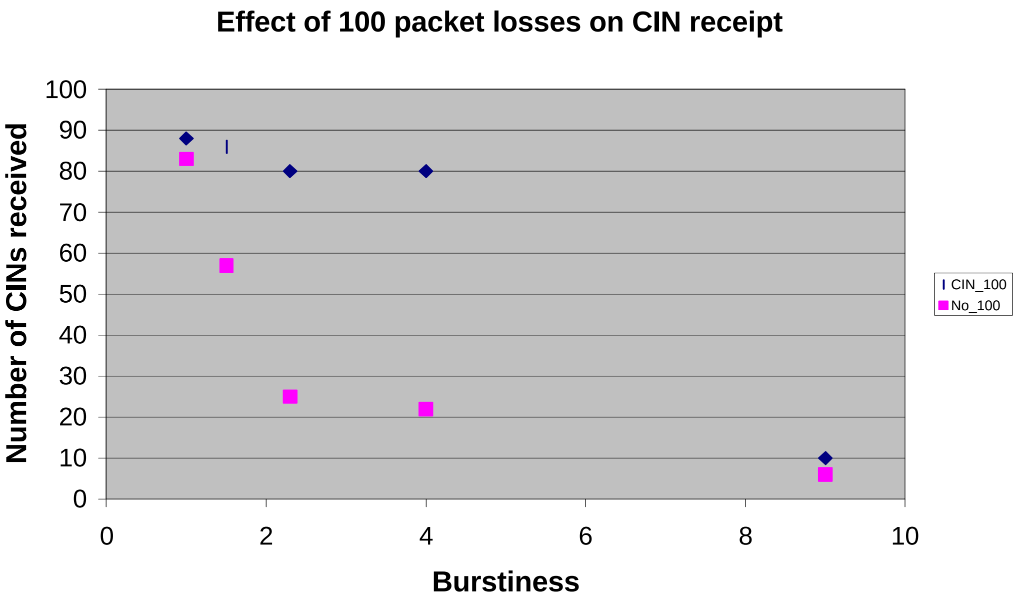


Figure 5-60 100 packet losses

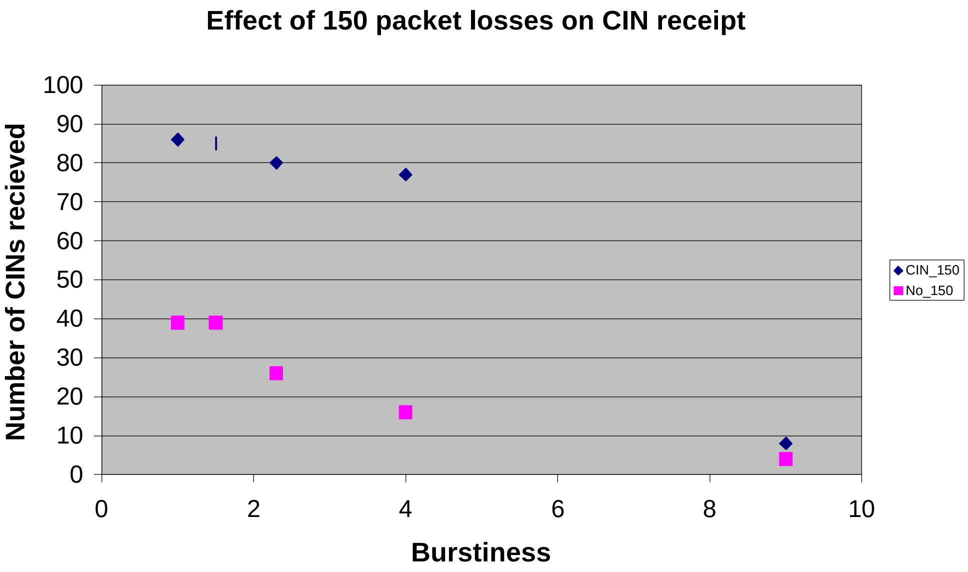


Figure 5-61 150 packet losses

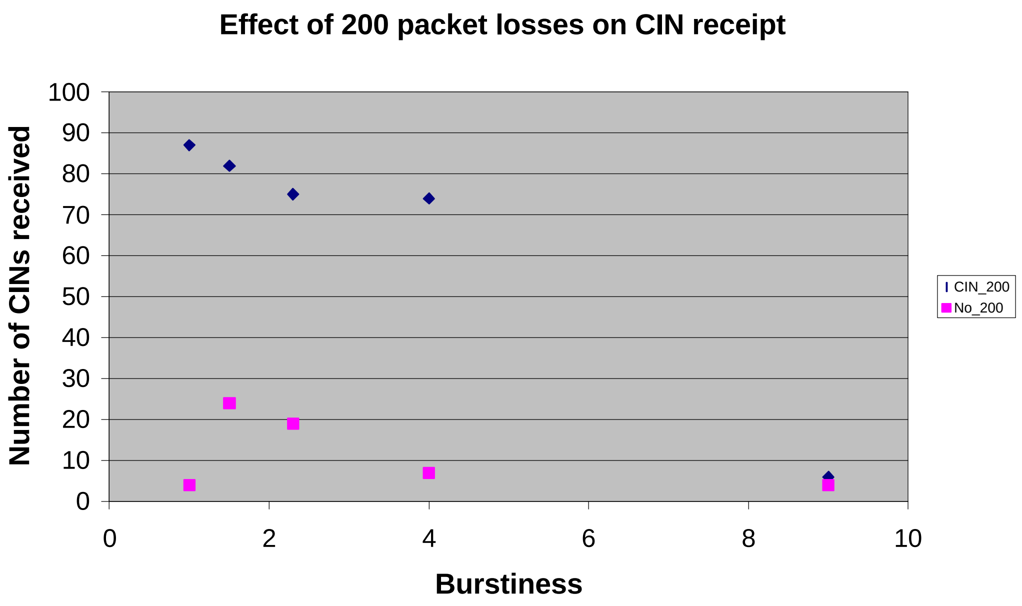


Figure 5-62 200 packet losses

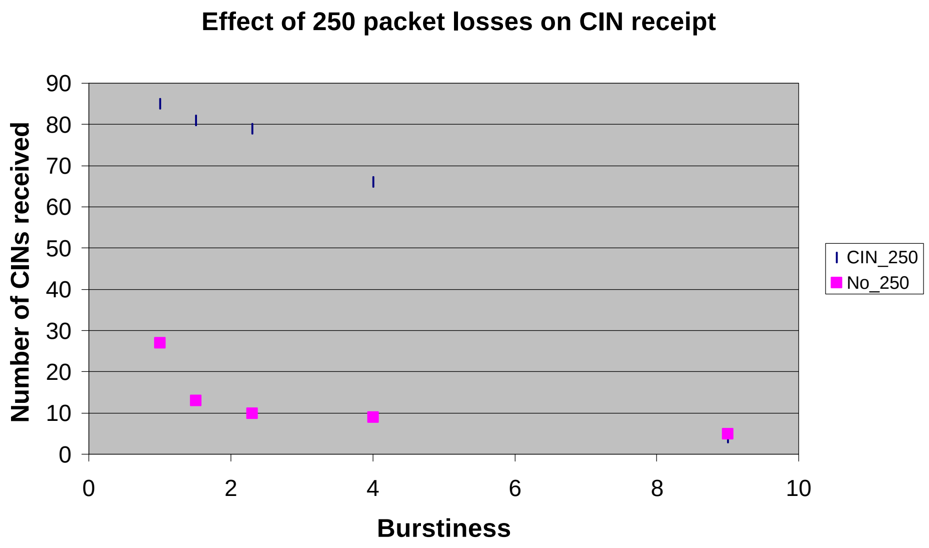


Figure 5-63 250 packet losses

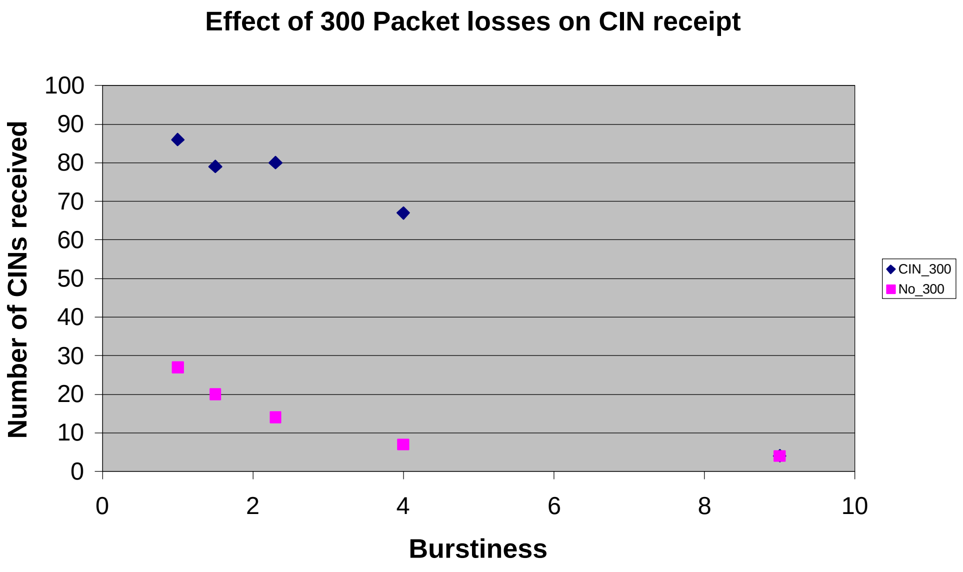


Figure 5-64 300 packet losses

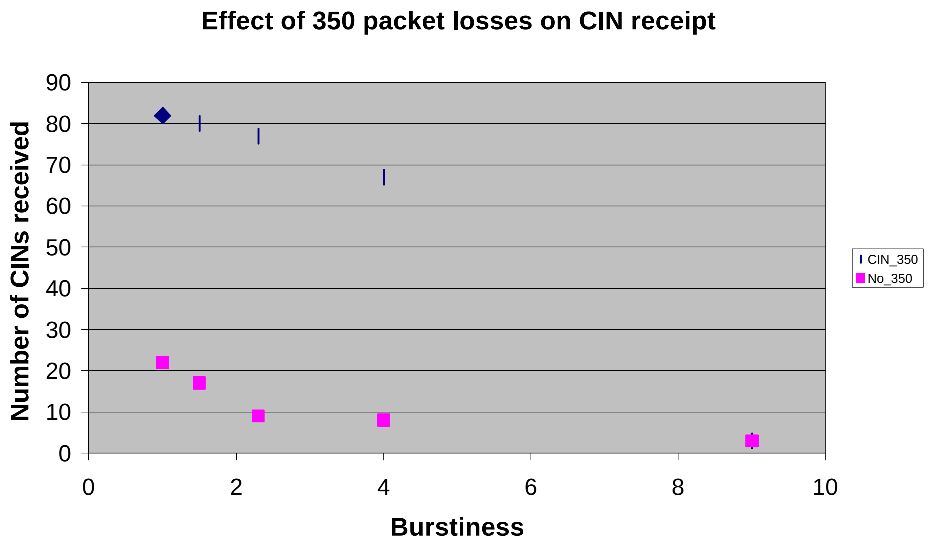


Figure 5-65 350 packet losses

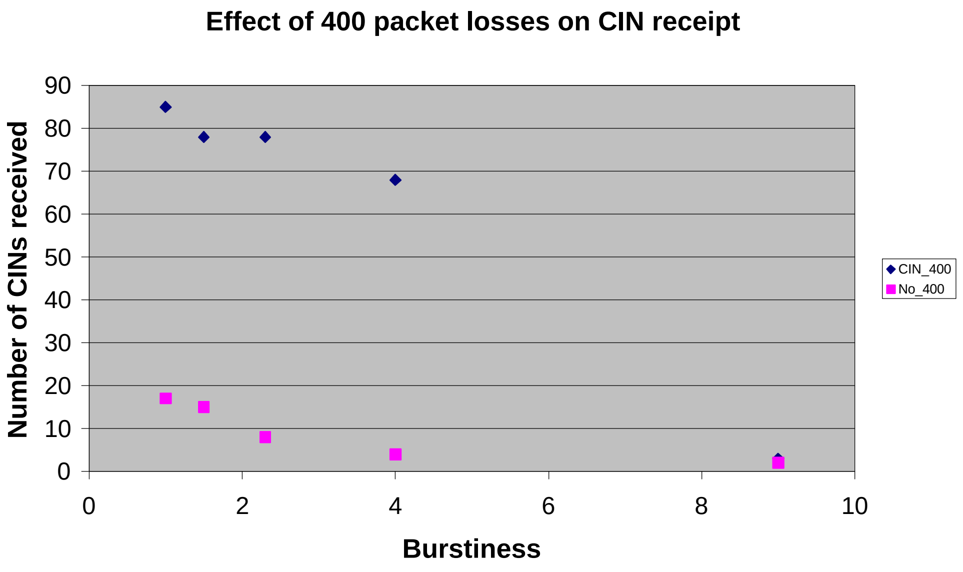


Figure 5-66 400 packet losses

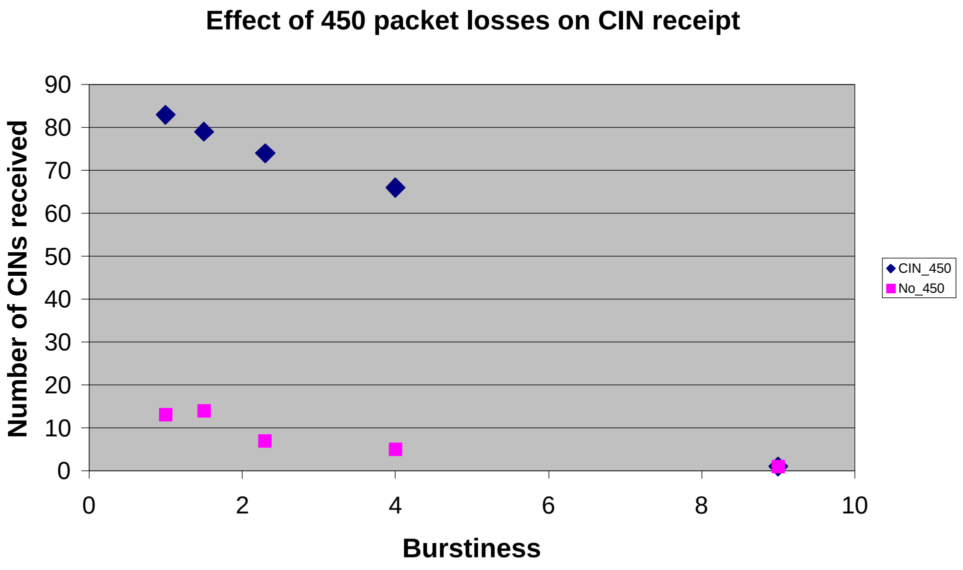


Figure 5-67 450 packet losses

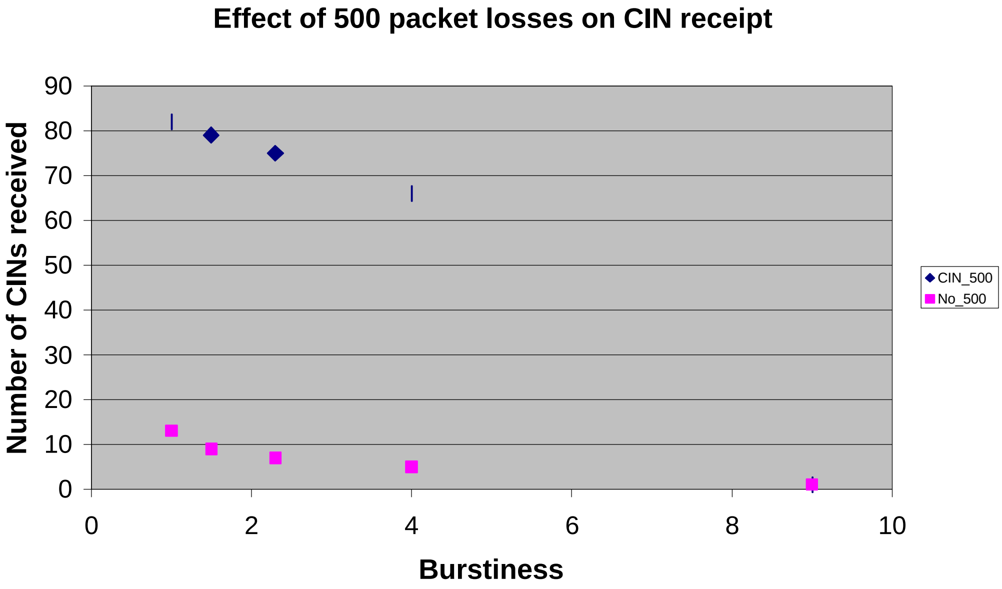


Figure 5-68 500 packet losses

As the burstiness of the sources increases, the number of CIN messages received by the ingress LER decreases. As explained for the last set of results the more bursty sources are less likely to be in their on state at the same time resulting in less numbers of packets being lost, whereas the less bursty sources are more likely to lose packets when their on times coincide.

A noticeable trend in the difference in the number of CIN messages received with and without a CIN Timer present occurs with the higher values of packet losses. Upon the CIN Timer expiring the loss probability is calculated, if it exceeds the loss threshold a CIN message is immediately generated regardless of whether the required number of packet losses has been lost, thus resulting in a large number of CIN receipts by the ingress LER. Without the CIN Timer present the correct number of packet losses needs to be accumulated before a CIN message is generated resulting in smaller numbers of CINs received by the ingress LER.

A separate simulation is run changing the CIN Timer value to 20ms. The results obtained for the number of CIN messages received by the ingress LER is shown below in Figure 5.69.

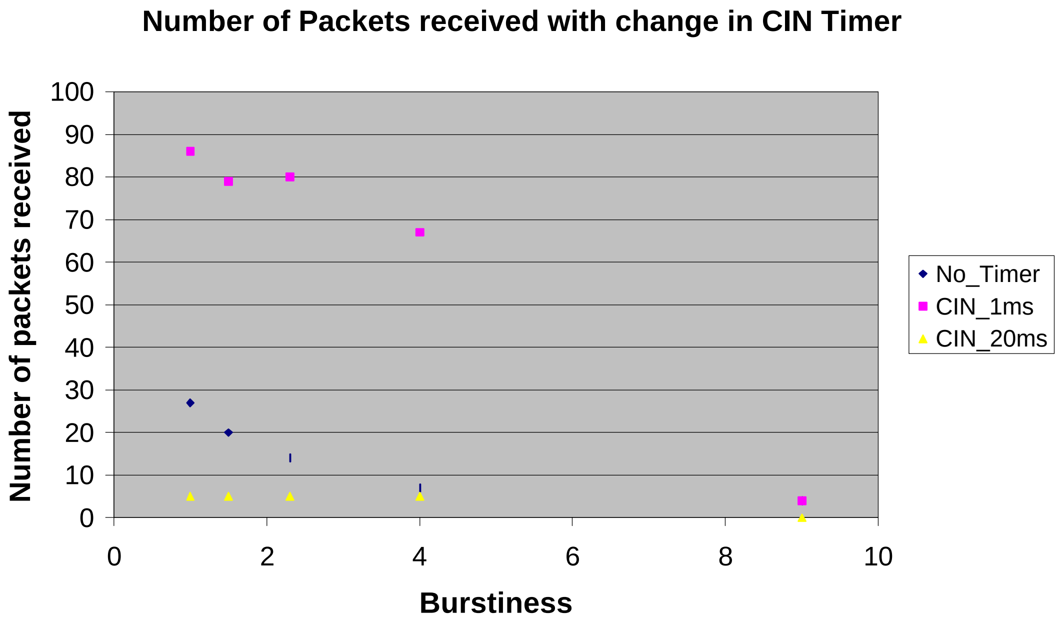


Figure 5-69 Variability in CIN Timer on Number of CINs received

The results show that the number of CIN messages received by the ingress LER decreases as the burstiness of the source increases. The cause for this has been explained previously.

As the CIN Timer value increases the number of CIN messages received by the ingress LER decreases as expected due to the greater value in between monitoring the loss probability. The high value assigned to the CIN Timer result in a slower response to congestion by the ingress LER as it receives a smaller number of CIN messages with a larger interarrival spacing. Whereas the smaller the CIN Timer interval the more CIN messages the ingress LER receives but the quicker it can response to congestion as the interarrival times between the CIN receipts will be an indication to the ingress LER of trouble.

5.4.6 Scenario 6

The simulation run in this scenario evaluates the performance of FATE+. FATE+ differs from FATE in that the congested LSR is responsible for the decision making at the time of congestion.

The simulation model is configured as explained previously in Scenario 1 with LSR 6 being fed with an additional on off source.

The following subsections depict the case studies listed:

- Transfer to a higher buffer stream;

- Re-map via an alternative downstream LSR;
- Transfer back to original path;
- Re-map via alternative upstream LSR.

5.4.6.1 Transfer to a higher buffer stream

On detecting congestion the congested LSR remaps the traffic flows into a higher buffer stream. The times associated with this process are shown in Figure 5.70.

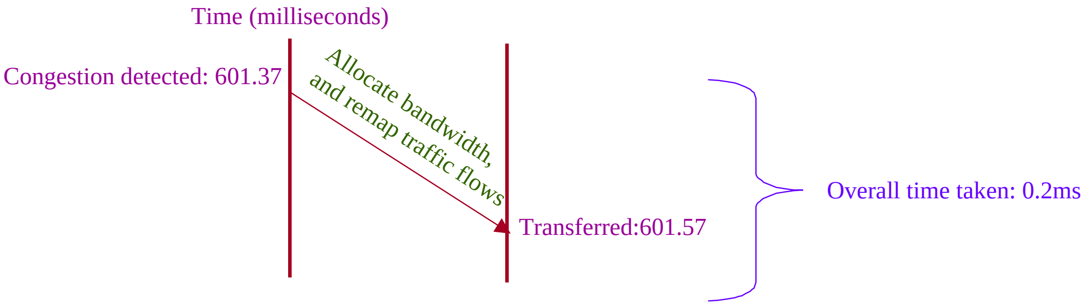

Time (milliseconds)
Congestion detected: 601.37
Allocate bandwidth, and remap traffic flows
Transferred:601.57
Overall time taken: 0.2ms

Figure 5-70 Remap flows into higher buffer stream

Once congestion has subsided the time to remap the flows back onto the lower buffer stream is equal to the remapping time into the higher buffer stream.

5.4.6.2 Re-map flows via an alternative downstream LSR

When the congested LSR determines it cannot cater for the flow in the current output buffer stream it will negotiate with an alternative next hop and re-route the traffic flows via a new next hop. The times involved with this process are shown in Figure 5.71.

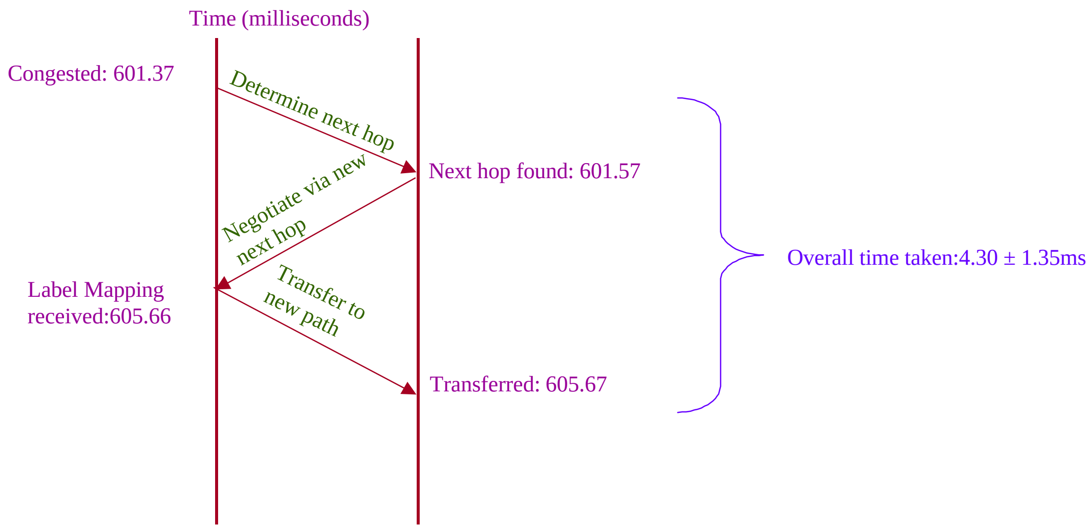


Figure 5-71 Re-route flows via new next hop

5.4.6.3 Re-map traffic flows back to original situation

In all cases to re-map flows back onto their original situation, an acceptable buffer loss probability is noted on two successive occasions. To re-map traffic flows back onto original path the following times are involved.

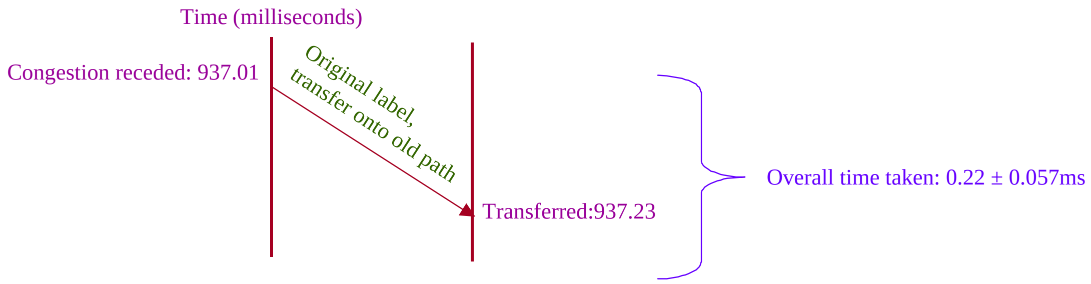

Time (milliseconds)
Congestion receded: 937.01
Original label, transfer onto old path
Transferred:937.23
Overall time taken: 0.22 ± 0.057ms

Figure 5-72 Re-map traffic flows into original path

5.4.6.4 Re-map flows via an alternative upstream LSR

In the next scenario the congested LSR determines it is unable to handle the traffic flow through any of its buffer streams and decides to have the upstream LSR re-route the traffic.

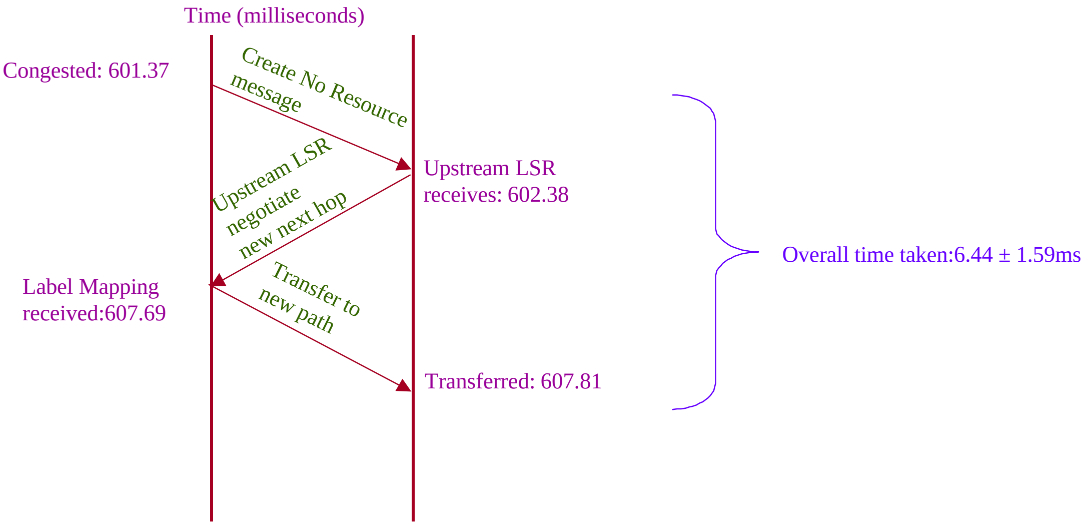

Time (milliseconds)
Congested: 601.37
Create No Resource message
Upstream LSR receives: 602.38
Upstream LSR negotiate new next hop
Label Mapping received:607.69
Transfer to new path
Transferred: 607.81
Overall time taken:6.44 ± 1.59ms

Figure 5-73 Re-map flows via the upstream LSR

When congestion has subsided the congested LSR informs the upstream LSR that the flows can be re-mapped via itself. The times involved are are shown below.

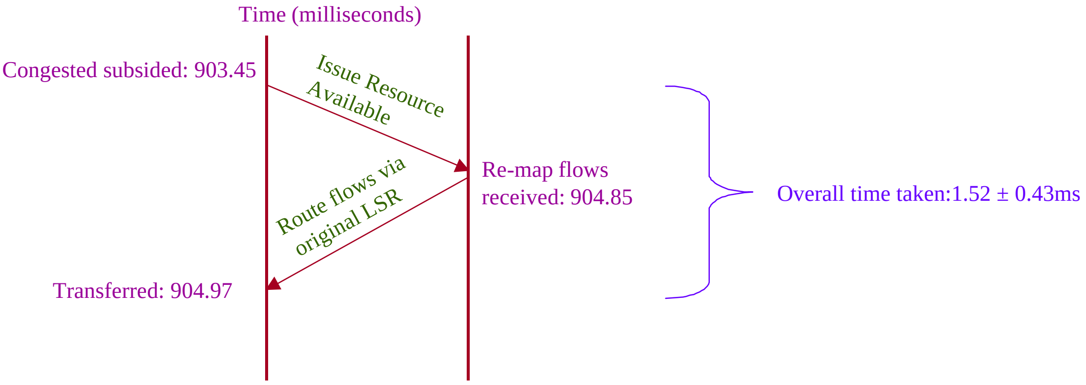

Time (milliseconds)
Congested subsided: 903.45
Issue Resource Available
Re-map flows received: 904.85
Route flows via original LSR
Transferred: 904.97
Overall time taken:1.52 ± 0.43ms

Figure 5-74 Re-route via original LSR

These results show that FATE+ works. Traffic flows experiencing congestion are re-mapped to avoid this. The times involved are much smaller than those found in FATE but were expected as the congested LSR takes control of the situation without the involvement of the ingress LER eliminating the processing needed to create and transmit CIN messages. FATE+ would ideally be used in ‘loosely’ routed LSPs where the network has control over the route the flows may take.

5.5 Summary

In this chapter, the performance of all the FATE/FATE+ schemes discussed in the thesis was determined through experiments carried out using simulation tools. It was found that the proposed scheme works, it detects potential congestion and reacts to avoid further packet losses. Different values of CIN receipts and responses were investigated for a variety of source profiles.

Chapter 6: Discussion

A central challenge facing network operators is maintaining customer guarantees whilst sustaining profitability. A fundamental approach to this problem adopted by many providers' has been to map traffic flows onto a physical topology using least cost metrics calculated by Interior Gateway Protocols. The limitations of this mapping were often resolved by over provisioning bandwidth along links that were anticipated being heavily loaded or to employ limited forms of load-sharing. However, as networks grow larger and the demands of the customer becomes greater (in terms of required bandwidth and QoS), the mapping of traffic flows onto physical topologies needs to be approached in a fundamentally different way so that the offered load can be supported in a controlled and efficient manner.

The aim of this research has been to develop a rapid-acting congestion control scheme for MPLS networks requiring minimal changes to the existing LDP/CR-LDP protocol. In achieving this, the author has developed a novel mechanism called FATE/FATE+, which provides a new approach to the problem of maintaining customer guarantees during transient periods within MPLS networks.

FATE/FATE+ is a set of mechanisms and procedures that detect congestion within traffic flows and reacts to circumvent further packet losses. Within MPLS there are two types of LSPs that can be established: explicitly routed and loosely routed. FATE addresses the former whilst FATE+ adheres to the latter.

In FATE, upon detecting congestion, the congested LSR formulates a CIN message and transmits it to the ingress LER that the LSP initiates from. This method places the responsibility of what appropriate action should be taken onto the ingress LER. As the initiator of the LSP, the ingress LER may have a specific route it wishes its traffic flows to traverse and by informing the ingress LER of potential problems, allows the ingress LER to decide whether to: renegotiate along the existing path for a higher QoS stream, negotiate for an alternative LSP or to accept the current loss being experienced by the traffic flows.

FATE+ addresses the issue of a loosely routed LSP where the ingress LER has no knowledge of the actual links/LSRs that the traffic flows are traversing. In this situation the congested LSR takes the appropriate action once it has detected a problem that exists. The congested LSR can transfer the LSP traffic into a higher QoS stream via a different downstream LSR or via an alternative upstream LSR.

Both FATE/FATE+ have been evaluated through simulation and the results have shown a reduction in the loss probability experienced by congested traffic flows from the order of 10^{-1} to 10^{-4} in the scenarios presented. Another feature of using FATE/FATE+ is that it can be implemented in conjunction with a MPLS network running “vanilla” LDP/CR-LDP.

Simulation results also show that the overall time to transfer the flow either into a higher buffer stream or along an alternative path is small compared to the increase in latency due to the onset of congestion as seen in Scenarios 1 for loss experienced in an intermediate LSR, scenario 2 for loss in an egress LER and scenario 3 for loss in an ingress LER. With FATE/FATE+ taking no more than 14ms and 5ms respectively to perform the switch over once the loss detection threshold is reached.

During periods of congestion, traffic flows are re-mapped either into higher buffer streams or along alternative paths. It is beneficial to both the operator and the customer to re-map these flows back into their original condition. From the operator’s perspective mapping flows into a higher buffer stream to ensure the flows are meeting the guarantees agreed upon equates to offering those flows a more expensive service than they were experiencing prior to the onset of congestion. It is therefore desirable for the network operator to re-map the flows back into the lower priced service level, and to re-use the available resources profitably.

From the customer’s perspective, their traffic flows being promoted at no expense to itself is a bonus, however providing the service agreement is maintained, the difference in performance should be negligible. However, the original LSP when chosen (before the congestion situation) was the optimal route (in terms of cost and or latency) and is therefore desirable to re-map the traffic flows onto them, thus relieving the higher quality resource so that the network operator can make use of the resources.

Simulation results have shown that in both FATE/FATE+ the times involved in re-mapping traffic flows back along the original path or into the lower buffer stream is quite low, typically no more than about 7ms.

Simulations were repeated at different LSRs within the network, where the sole intention was to vary the number of hops the signalling messages had to traverse. The minimum number of hops were zero when the congested LSR was the ingress LER, and the maximum number of hops was six when the congested LSR was the egress LER. Although, for simulation purposes the propagation delay was assumed to be zero this has little impact relative to the queueing delay and transmission delays[33].

In any congestion control scheme there are three important components that are necessary: the ability to detect congestion, the ability to maintain information about the link state in the network and the ability to take appropriate congestion resolution measures.

In terms of detecting congestion, the author initially considered three possibilities:

- Using a Congestion Detection Timer mechanism;
- Detecting and measuring the actual number of packet losses as an indicator of congestion;
- Using Status Request messages as an explicit detection mechanism – by monitoring the Buffer Table Updates.

The Congestion Detection Timer was implemented first and the initial simulation results showed that the reaction times were not as dynamic as was initially anticipated. This therefore indicated a fundamental weakness in the scheme.

The timer was set at the beginning of the simulation, the number of packets passing through a particular buffer were recorded along with the number of lost or destroyed packets. Once the timer had expired the packet loss was calculated.

The main concern with this approach was the possibility that whilst the detection scheme was in capture mode no action would ever be taken until this phase was completed, irrespective of the packet losses. For example it was shown that this could result in large numbers of packet losses before the network was able to react and would also be capture time-dependent, leading to the problem of deciding what the appropriate timer values should be. Using a sliding capture window is also flawed, as it is difficult to select an appropriate window size according to different network conditions.

Counting a specified number of packet losses relative to the number of packets through was found to be an alternative and more appropriate method and was therefore implemented in the FATE/FATE+ simulations to test for its suitability. The mechanism worked as follows: From the moment a packet is lost the number of packets arriving into that particular buffer stream and the lost ones are recorded. When a predetermined number of packets have been recorded as being lost, the loss probability currently being experienced by that buffer is calculated. The number of packets entering and being lost from the buffer is reset to zero and

[33] However, considering a terrestrial system it is noted that there is a 5ns delay per metre of network transmission medium. For the maximum number of hops considered above, the maximum propagation delay would be approximately 3ms. Assuming each optical fibre WAN link is 100km. This value is very small and shows that FATE adds little delay to a realistic network.

the process repeats continuously. This method was found to be fast acting and self-scaling when compared to the former.

The third option is for the ingress LER to periodically transmit Status Request messages into the network. Each LSR calculates the current loss probability being experienced by its buffer streams. For example, for every 1000 packets arriving at a particular buffer stream the loss probability is calculated. The number of packets arriving is referred to as the buffer table updates. This approach was also found to meet the requirements for providing supplementary information within FATE – to enable the ingress LER to make more informed congestion resolution action and was implemented in the FATE simulations. On receiving the Status Request message on its return, the ingress LER can determine whether a particular LSR is experiencing congestion in any of its buffer streams. The simulation results conducted as part of this research compared the times involved in notifying the ingress LER of congestion, either by Status Request messages or by receipt of CIN messages. It was found that at low values of packet losses and for low buffer table updates, both approaches are comparable. With low buffer table updates congestion detection will occur sooner than for larger buffer table updates, and almost equals the time at which the actual congested LSR generates a CIN message. For low buffer table updates and high packet losses the Status Request message outperforms the CIN message by as much as 4 times. The Status Request messages outperform the CIN message because at lower buffer table updates congestion detection occurs quicker than the timer taken for the LSR to accrue high packet losses. However, for high buffer updates the CIN message indicates probable congestion by as much as 7.5 times over the Status Request message mechanism. This trend results in the LSR loses the required number of packet losses at a quicker rate than it takes the buffer stream to detect loss.

Summarising, it can be seen that the ingress LER should transmit Status Request messages periodically into the network for low values of buffer table updates and low values of packet losses, but in the case of high buffer table updates, the CIN messages mechanism would be more appropriate. Both Status Request and CIN messages were employed within the FATE simulations.

The second necessary component in congestion control is maintaining state information about the utilisation of the links/LSRs within the MPLS network. There are a number of reasons why this maintenance is important:

- To make sure that routing protocol updates are maintained consistently;
- Renegotation / Negotiation procedures;
- Restoring traffic flows to original LSP.

In the first case the routing protocol needs to return the most up to date information about the most appropriate LSP for connection establishment. What was an ideal choice a period of time ago is not necessarily going to remain so due to variation in the traffic loading within the network. On receiving notification that a LSR is congested the ingress LER needs to either renegotiate for a higher buffer stream or negotiate for an alternative path. In this situation the ingress LER needs to have the most recent information about the network conditions. Finally, once the transient bursts have subsided, the ingress LER needs to be informed when the situation is such that the traffic can be re-mapped to the original LSP path.

Re-mapping traffic flows back to the original situation is the sole responsibility of the ingress LER. To enable the ingress LER to know when the time is suitable to initiate this process there is a need to know how utilised the links/LSRs along a given LSP are.

Status Request messages are transmitted periodically into the network by the ingress LER once FATE has been activated. On receipt of a Status Request message indicating the loss probability the buffer streams along a given LSP are acceptable, the ingress LER can then instigate proceedings to re-map the LSPs back to the optimum situation. Calculation of the optimum situation is beyond the scope of this thesis as there is a huge amount of research currently being carried out in the area of QoS Routing [MA97][NAH98].

In the quantitative analysis of FATE the role of the Status Request messages was to indicate to the ingress LER the status of the buffer streams along a LSP of particular interest to the ingress LER, i.e., a LSP along which the traffic flows initiating from the ingress LER traverse. However, from the simulation results, it can be seen that the role of the Status Request messages are two fold; they can be used to detect possible congestion in place of CIN messages *and* to detect congestion dissipating. In both cases the ingress LER utilises this to either react to or to recover from congestion as the scenarios described in Scenarios 1 and 4.

To dampen the occurrence of signalling storms and route flapping, the author considered a number of mechanisms:

To prevent a large generation of CIN messages being created when congestion is detected, a predetermined number of packets needs to be lost before the loss probability within that particular buffer stream is calculated. However, a limitation with this scheme is having knowledge of what the appropriate number of packet losses should be set to, for a particular source's traffic profile.

The author also considered employing an Optional Response Timer to allow the ingress LER to determine when to respond to congestion notification. On receipt of a CIN message the ingress LER sets the timer and if it receives another CIN message before the timer expires it takes appropriate action such as renegotiating for a higher buffer stream or negotiating for an alternative path. However, this appeared redundant in the face of the ingress LER responding to a predetermined number of CIN messages, which is the method implemented in the simulations.

As seen by the results obtained in scenario 5, the CIN Timer has absolute priority irrespective of the number of losses that occur in the interval. Once the CIN Timer has expired, the loss probability is calculated, if it exceeds the loss threshold a CIN message is immediately generated irrespective of whether a specified number of packet losses have occurred. This approach could be amended to prevent the automatic generation of a CIN message by ensuring that the specified number of losses have occurred before calculating the loss probability each time the CIN Timer expires. This approach is less responsive than the former. However, in some cases the CIN Timer has a normalising effect on the number of CIN messages generated irrespective of traffic conditions. Without a CIN Timer present and depending upon the traffic conditions it is possible for large numbers of packets to be lost resulting in a higher number of CIN messages being generated, which would not have occurred had a timer been present.

Throughout this thesis the author kept the inclusion of timers to a minimum. The CIN Timer was employed to monitor buffer streams and traffic flows that had been flagged as suffering particularly high losses. However, the fundamental problem with timers as seen by the simulation results is determining an appropriate expiry value. This is seen as a potential weakness of FATE.

The traffic profile of the source was varied for a number of simulations. The overall cycle time of the source's on and off periods were kept constant, but the duration for which it was on was varied. By doing this the 'burstiness'of the source was altered whilst keeping the mean packet generation rate constant, allowing comparisons between the different traffic profiles to be performed.

In the case where the ingress LER responded to a different number of CIN messages, it was found that the less burstier the source the quicker the CIN messages were generated, and the faster the ingress LER was able to respond to receiving the correct number of CIN messages. However, with the more bursty sources the response times were lower, again this trend is due to the probability of the bursts overlapping, being much lower in the case of the more bursty sources, whose on durations are shorter and less likely to overlap.

The response delay consists of two components: the delay to accumulate the correct number of packet losses at the congested LSR and the delay in transmitting the CIN message to the ingress LER. The latter was found to be constant as was expected as the source profile of the background signalling messages were kept constant. The delay and delay variation in transmitting the CIN messages is quite small as the signalling traffic is given a high priority buffer traverse with the only other traffic being background signalling messages.

To address the issue of scalability the author proposed assigning a Virtual Source/Virtual Destination (VS/VD) pair for the aggregation of LSPs entering the domain at one point and exiting at another point, using label stacking or tunnelling within the autonomous MPLS domain of interest. By employing label stacking within the domain and assigning VS/VD pairs, the issue of scalability is removed whilst allowing the operator control of the LSPs traversing its network. It allows for efficient utilisation of the limited network resources and the additional capability of controlling congestion. With the VS/VD paradigm, the congestion control messages need only propagate along as far as the virtual source for the ingress LER and to the virtual destination for the egress LER. With the virtual endpoints of the LSP defined, aggregation of many LSPs can be treated as an individual LSP.

In promoting traffic flows to a higher buffer stream the network operator is maintaining the SLA made with the customer. However, this is done at a cost to the operator, so it is in its best interest to re-map the traffic flows to their original situation once the ingress LER detects congestion has subsided. The author implemented two threshold levels to act as triggers. The higher loss threshold triggered the path change to the higher buffer stream or alternative path, while the lower threshold causes the return to the original path. This implies that network conditions have to get improve significantly before the moved LSPs are disturbed again. This reduces the probability of path selection flapping.

To assess the feasibility of the FATE scheme proposed here, it is useful to compare the approach with other proposals addressing similar issues. The aim of this comparison is to establish how effectively the congestion approach taken by the author meets the requirements of such congestion control schemes. However, publications in this area are few; the one paper that has addressed the relevant areas is a recent proposal developed at Nortel Networks and AT&T called the QoS Resource Management Method (QRMM) as explained fully in Appendix A.4.

Although the QRMM addresses the issue of renegotiation once a connection has been established it does not address the issue of dynamic re-distribution of traffic flows over a shorter time period but tends more to long term provisioning. As a result of that it is impossible to make a one to one comparison between this thesis and the paper. It was also only published in March 2000 which did not leave the author with time to fully investigate the appropriateness of their scheme.

The main significance of this work is the proposition of a congestion control scheme for MPLS networks. The FATE scheme has been shown to work. FATE is *fast acting* and alleviates congestion occurring in a LSP by either re-mapping the relevant traffic flows into a higher buffer stream or along an alternative path. The first part of the work reported in this thesis establishes the validity of the protocol. The use in the proposed protocol mechanisms and procedures currently being standardised by the IETF's MPLS working group ensures compatibility and conformance with existing and future MPLS platforms.

The result is a reliable and flexible protocol that adheres to the standards and solves the current problem of detecting and alleviating congestion for WAN architectures. To maintain compatibility with the existing LDP/CR-LDP protocols the author has proposed a number of signalling message Type Length Value (TLV) whose format is consistent with the TLVs used in LDP/CR-LDP along with new data structures to maintain additional state information

The FATE scheme proposed by the author meets the requirements and has resulted in a protocol that can be considered as a practical congestion control scheme for MPLS networks.

Two limitations of the FATE scheme presented in this thesis are the increased levels of signalling in the core network and the need to maintain additional state information about CR-LSPs and buffer streams. For any physical architecture different to the traditional MPLS architecture this is the price to be paid for a versatile and adaptable platform. However, the author believes the increase in signalling load is within acceptable bounds as aggregation of CIN messages, and thresholds varying the number of packet losses ensures signalling load is kept low.

However further work could investigate the impact of this on the network.

Chapter 7: Conclusions and Further Work

The objectives set out for this research have been achieved. A novel congestion control scheme capable of reacting rapidly to transient bursts within a MPLS network has been developed.

The key findings of this research are as follows:

- ♦ A significant reduction in traffic flows experiencing congestion is observed when using FATE/FATE+;
- ♦ A self scaling congestion detection mechanism is employed that offers superior performance when compared to the sliding window mechanism;
- ♦ FATE/FATE+ is relatively insensitive to different source traffic profiles;
- ♦ Methods to dampen the responsiveness of the scheme deployed have been successfully implemented. These operate at both the congested LSR i.e., varying the number of packet losses in the detection phase and the ingress LER i.e., responding to different numbers of CIN receipts;
- ♦ Obtaining recent link state information in terms of Status Request messages provide additional information that facilitate appropriate congestion alleviation action to be taken;

However, further work could evaluate the impact various types of background source traffic profiles will have on FATE. Although this simulation study investigated using homogeneous sources whose mean transmitting rate were kept constant, the peak transmitting rate was varied to represent Internet traffic. It would be of interest to investigate the performance evaluation of heterogeneous sources, as WANs will not typically be composed purely of sources with similar characteristics.

A number of parameters to reduce the sensitivity of FATE were introduced in the scheme, such as varying the number of packets lost before calculating the loss probability. The number of CIN messages the ingress LER should receive before responding to the notification of congestion was also varied. Further simulations could determine what the optimum values should be, associated with each source traffic profile. This would enable sensible provisioning of a network catering for different types of applications.

Simulation results depicting the performance of FATE and FATE+ as individual mechanisms have been seen. However the effect of FATE and FATE+ operating simultaneously would be of importance as it is highly probable that network operators will be expected to maintain both explicitly and loosely routed LSPs concurrently.

Another area of further work involving intelligent agents or artificial intelligence could be applied together with traffic engineering metrics to predict the onset of congestion prior to it actually arising and to take proactive action accordingly.

Appendix A

A.1 Head of Line Blocking

Head of Line Blocking occurs when a packet needs to be transmitted to an output stream that is empty but is held in a queue because the packet in front of it cannot be forwarded as the output stream to which it is destined is currently busy. In output buffering this phenomenon does not occur as the only contention results from packets being buffered in front of a packet destined for the same output stream as illustrated below in Figure A. 1.

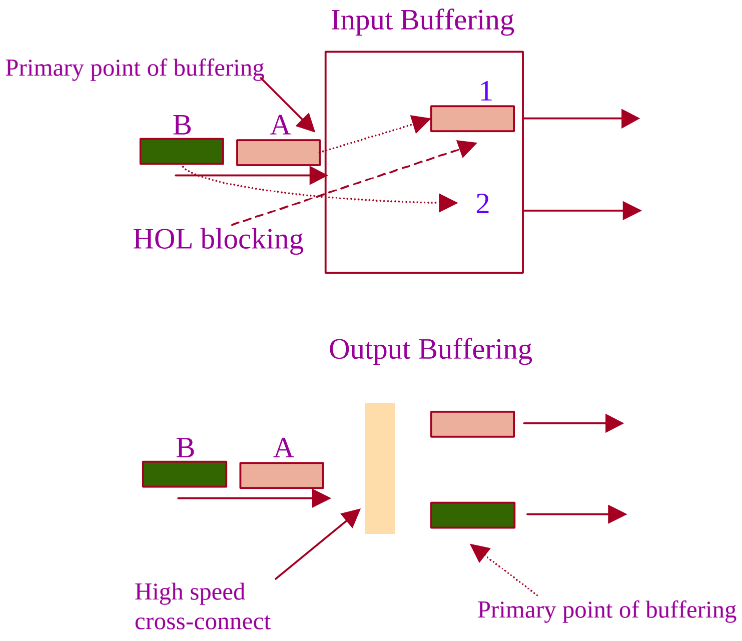


Figure A. 1 Buffer Architectures

Packet A is destined for egress port 1 – but can't be transmitted there due to temporary resource conflict. Meanwhile packet B is blocked behind packet A due to FIFO buffered ingress queue – even though port 2 is free.

A.2 Confidence Intervals

The output from one run of a simulation is a sequence of measurements, which depends on the particular sequence of random numbers used. These measurements need to be evaluated statistically in order to provide reliable results for the steady state performance. A confidence interval quantifies the confidence that can be ascribed to the results from a simulation experiment [PIT93].

For example, a 90% confidence interval means that for 90% of simulation runs for which an interval is calculated, the actual value for the measure of interest falls within the calculated interval Figure A. 2. On the other 10% of occasions, the actual value falls outside the calculated interval.

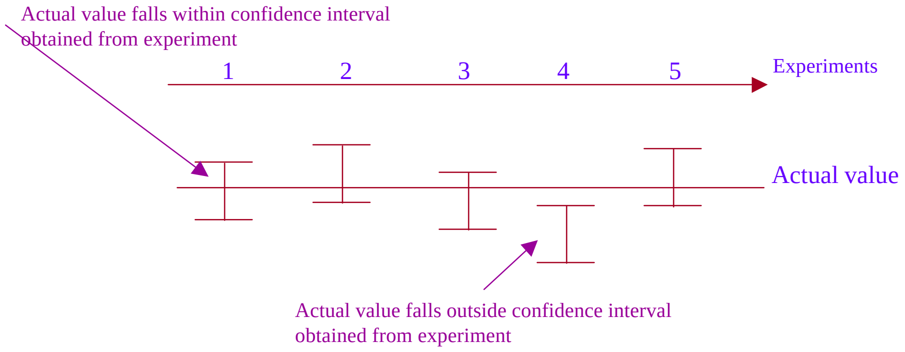

Actual value falls within confidence interval
obtained from experiment
1
2
3
4
5
Experiments
Actual value
Actual value falls outside confidence interval
obtained from experiment

Figure A. 2 Confidence Intervals

This is not equivalent to 90% of the performance estimates from independent simulation runs falling within a 90% confidence interval about the actual value. The actual percentage of times that a confidence interval does span the correct values is called the coverage.

A.3 Random Number Generation

The purpose of random number generation is to produce a sequence of number, drawn from the uniform distribution over the range 0 to 1, which appears to be independent. A good random number generator (RNG) should appear to be uniformly distributed on [0,1] and should not exhibit any correlation between generated numbers. It must be fast and avoid the need for much storage. A random number sequence must be reproducible; this aids debugging, and can be used to increase the precision of results.

OPNET generates a sequence of random numbers based on a *random number seed*. Different seeds produce different sequences of random numbers.

Once a simulation model has been constructed, it is typically exercised under a number of different conditions in order to characterise the system it represents. While the model itself remains the same, aspects of its environment, or parameters that it offers are varied in order to establish patterns of behaviour or relationships between certain inputs of the system and selected outputs. The inputs and outputs vary with each system that is modelled and depend on goals of the simulation study.

Many simulations rely on stochastic [LAW91] modelling of certain elements in order to represent behaviour that is not know in a precise fashion but that can be characterised by associating probabilities with a set of outcomes.

Stochastically modelled elements depend on a random number source on which to base their behaviour. By 'drawing' from the source, these elements can incorporate variability into appropriate actions or decisions as they are taken. For example, a packet generator module in OPNET is a stochastic element that issues a random stream of packets over time. In order to accomplish this, at each packet generation, the generator uses a random number to set the time for its next packet generation.

By its very nature, it is impossible for a computer program to exhibit genuinely unpredictable behaviour. If a simulation program remains the same for multiple simulation runs, then any change in its behaviour must come from a change in its operating environment (i.e., its input). In particular even the random number stream used to implement stochastic behaviour, must be forced into a different mode in order to yield different results from simulation to simulation.

The mechanism used to select new random number sequences relies on starting the random number generator in a different state. This initial state is known as the *random number seed* because it determines all future output of the random number generator.

For a simulation that incorporates stochastic element, each distinct random number seed value will produce different behaviour and yield new results. Each particular simulation can be thought of as representing one possible scenario of events for the modelled system. However, no single simulation can necessarily be used as an accurate gauge of "typical" system behaviour, since it is conceivable that even atypical behaviours, provided they are possible, may be achieved for some random number seed. Therefore, a technique that is frequently used is to exercise the simulation model multiple times while varying the random number sequence. The results obtained from the separate simulations can be combined (usually simply by averaging) to estimate typical behaviour.

In OPNET all of the random numbers draw from a single random number sequence initialised with the value of the seed environment attribute. The random number generator used to create this sequence is provided by the host computer's operating system and may vary on certain platforms. OPNET random number generation sources used on the operating system used to run the simulations is the function random(). Using this function guarantees identical results for simulations run with the same seed, regardless of which platform is used.

A.4 QoS Resource Management Method

The scheme developed at Nortel Networks and AT&T is called the QoS Resource Management Method (QRMM) [ASH2000a]. It implements techniques used in Public Switched Telephone Networks (PSTNs) to standardise service classification, bandwidth allocation, bandwidth protection, and priority routing treatment for all network services. In this scheme bandwidth is allocated in discrete changes to three *virtual networks* that provide *high priority*, *normal priority* and *low priority* services. Examples of services that fall within these three VNs are, a) defence voice communication, b) constant rate, interactive, delay-sensitive, voice and c) best effort services such as variable rate, non-interactive, non-delay sensitive voice mail.

The edge routers make periodic discrete changes in bandwidth allocation along a Constraint-based Routed Label Switched Path (CRLSP) based on the overall aggregated bandwidth demand. Three optional type/length/value (TLV) parameters are proposed in this scheme.

A *Depth of Search (DoS) TLV* parameter in the CRLDP label request message to control the amount of bandwidth allocated on individual links along the CRLSP.

A *modify TLV* parameter in the CRLDP label request message is used to allow modification of the assigned traffic parameters.

A *crankback-TLV* parameter in the CRLDP notification message to allow an edge router the option of searching for an alternative CRLSP that can accommodate the bandwidth request.

The bandwidth allocation control is based on the estimated bandwidth needs, the bandwidth currently being used and the utilisation of the links in the CRLSP. The decision on whether to increase or decrease bandwidth is the sole responsibility of the ingress LER that periodically monitors the bandwidth in use along the CRLSP. In allocating bandwidth, the CRLDP is used to specify appropriate parameters in the label request message to request bandwidth allocation changes on each link on the CRLSP, and to determine if the link bandwidth can be allocated on each link on the CRLSP.

In order for the ingress LER to modify the amount of bandwidth allocated to a request, it needs to determine the QoS resource management parameters. These include the VN priority (*key*, *normal*, or *low*), the *VN bandwidth* currently being used, the *VN bandwidth thresholds* and whether the CRLSP is a first choice CRLSP or an alternative CRLSP. Using these parameters a VN depth-of-search (DoS) table is accessed to determine a DoS load state threshold, or the “depth” to which the network capacity can be allocated to the request. The ingress LER then selects a first choice CRLSP according to a routing table selection criterion. Having selected the CRLSP, the ingress sends a label request message, enclosing the explicit route, the requested traffic parameters, the *DoS TLV* parameter and the *modify TLV* parameter, along the selected CRLSP. On receiving the label request message each LSR applies the QoS resource management rules to determine whether or not it can allocate the bandwidth request. The QoS resource management rules entail that the LSR determine the *link load state*, i.e., whether they are reserved, busy, heavily loaded or lightly loaded, based on the bandwidth in use and the available bandwidth. The link load state is then compared against the received DoS threshold in the label request message. If the available bandwidth on the link and the link load state is lower than the requested DoS threshold then the LSR sends a notification message with the crankback TLV parameter to the ingress LER, which can then try an alternative CRLSP. If the available bandwidth on the link and link load state is higher than the requested DoS threshold, the bandwidth modification request is then deemed successful.

The QRMM scheme monitors bandwidth usage on a path and reactively request resources as appropriate.

Author’s Publications

[HOL1] Holness Felicia, Phillips Chris, *“Dynamic Congestion Control Mechanism for MPLS Networks”*, Internet, Performance and Control Network Systems, part of SPIE’s Voice, Video and Data Communications, Boston, MA, 5-8 November 2000.

[HOL2] Holness Felicia, Phillips Chris, *“Congestion Control Mechanism for Traffic Engineering within MPLS Networks”*, 5th International Symposium on Interworking (IFIP / TC6) Bergen, Norway, October 3 – 6th 2000.

[HOL3] Holness Felicia, Phillips Chris, *“Effective Traffic Management for MPLS Networks*”, accepted at 4th World Multiconference on Systemics, Cybernetics and Informatics (SCI2000), Sherton World Resort, Orlando, Florida, July 23-26th, 2000.

[HOL4] Holness Felicia, Phillips Chris, “ *Dynamic QoS for MPLS Networks*”, 16th UK Teletraffic Symposium, (UKTS), Nortel Networks, Harlow, England, May 22-24, 2000.

[HOL5] Holness Felicia, Phillips Chris, *“Dynamic Traffic Engineering within MPLS Networks”*, 17th World Telecommunications Congress,(WTC/ISS2000) Birmingham, England, May 7-12, 2000.

[HOL6] Holness Felicia, Griffiths John, “*Multiprotocol Label Switching within the Core Network*”, accepted FITCE Congress 99, Netherlands, August 24th – 28th 1999. Published in British Telecommunications. Eng. British Telecommunications Engineering, vol.18, pt.2, Aug. 1999, pp.97-100. UK.

[HOL7] Holness Felicia, Griffiths John, “*Interoperability between MPLS and ATM*” 7th International Conference on Telecommunication Systems, (ICT99), Modeling and Analysis, Nashville, Tennessee, March 18- 21 1999.

[HOL8] Holness Felicia, Griffiths John, “*Cell Switched Routers- An Overview*” Proceedings of the 4th Communications Network Symposium, Manchester, England July 7- 8th 1997.

References

[AHM97] Ahmed H, Callon R, Malis A, Roy J, ‘*IP Switching for Scalable IP Services’*, Proceedings of the IEEE, Vol. 85, No.12, December 1997.

[AND99] Andersson.L, Doolan.P, Feldman.N, et al., *LDP Specification*, http://search/ietf/org/internet-drafts/draft-ietf-mpls-ldp-06.txt, October 1999,02/03/00.

[ASH2000a] Ash.J, Lee.Y, Ashwood-Smith.P, et al., *LSP Modification Using CR-LDP*, http://search.ietf.org/internet-drafts/draft-ietf-mpls-crlsp-modify-01.txt,February 2000, 03/03/00.

[ASH2000b] Ash,J, ‘*Traffic Engineering and QoS Methods for IP-,ATM-, & TDM-Based Multiservice Networks’*, http://www.ietf.org/internet-drafts/draft-ash-te-qos-routing-00.txt.

[AWD99] Awduche D, ‘*MPLS and Traffic Engineering in IP Networks’*, IEEE Communications Magazine, Vol.37, No.12, December 1999, pp42-7.

[AWD2000] Awduche.D, Berger.L, Gan.D-H, et al., *RSVP-TE: Extensions to RSVP for LSP Tunnels*, http://search.ietf.org/internet-drafts/draft-ietf-mpls-rsvp-lsp-tunnel-05.txt, February 2000.

[BAK] Baker F, ‘*Real-Time Services for Router Nets’*, Data Communications, May 21 1996.

[BLA] Black D, ‘*Differentiated Services and Tunnels’*, July 2000, http://search.ietf.org/internet-drafts/draft-ietf-diffserv-tunnels-02.txt.

[BON99] Bonica.R, Tappan.D, Gan. *D-H, ICMP Extensions for Multiprotocol Label Switching*,http://search.ietf.org/internet-drafts/draft-bonica-icmp-mpls-01.txt, May 1999,17/03/00.

[BOR98] Borthick S, ‘*Router Startups Betting on MPLS’*, Business Communications Review, November 1998 Vol.28, No.11, http://www.bcr.com/bcrmag/11/nov98p14.htm

[BRA97] Braden.R, Zhang.L, Berson.S, et al., *Resource Reservation Protocol (RSVP) Version 1 Functional Specification*, http://info.internet.isi.edu:80/in-notes/rfc/files/rfc2205.txt.

[BRI2000] Brittain.P, Farrel.*A MPLS Traffic Engineering: A Choice of Signalling Protocols*, http://www.datcon.co.uk/mpls/mplswpdl.htm, January 17, 2000.

[CAL99a] Callon.R, Doolan.P, Feldman.N, et al., *A Framework for Multiprotocol Label Switching*, http://search.ietf.org/internet-drafts/draft-ietf-mpls-framework-05.txt, September 1999, 24/03/00.

[CHE99] Chen.T, Oh.T, *Reliable Services in MPLS*, IEEE Communications Magazine, Vol.37, No.12, December 1999, pp58 –62.

[CIS97] White Paper, '*Advanced QoS Services for the Intelligent Internet'*, http://www.cisco.com

[CRO96] Crowcroft J, Wang Zheng, *'A Rough Comparison of the IETF and ATM Service Models'*, IEE Colloquium on Towards Gigabit Networking (Digest No.1996/118). IEE. 1996, pp.6/1-8. London, UK.

[CRO99] Crowcroft J, *'IP over Photons: How not to waste the waist of the hourglass'*, 7th International Workshop on Quality of Service, IWQoS99, IEEE 1999, pp 9-11.

[DAV] Davie B, Doolan P, Rekhter Y, '*Switching in IP Networks IP Switching, Tag Switching and Related Technologies'*, Morgan Kaufmann Publishers Inc, ISBN 1-55860-505-3

[DeM] De Marco M, Trabucchi S '*An efficient congestion control scheme for providing QoS to I-VPN'*, Part of the SPIE Conference on Quality-of-Service Issues Related to the Internet,

[DIFF] Differentiated Services Web Page, http://diffserv.lcs.mit.edu.

[DRI97] Driscoll D, Mehravari N, Olson M, '*Performance comparison between ATM LAN Emulation, Classical IP over ATM, and Native ATM in a Multi-Platform Multi-Operating System Environment'*, MILCOM 97. MILCOM 97 Proceedings (Cat. No.97CH36134). IEEE. Part vol.1, 1997, pp.434-8 vol.1. New York, NY, USA.

[DUM98] Dumortier P, '*Toward a New IP over ATM Routing Paradigm'*, IEEE Communications Magazine, January 1998.

[ESA95] Esaki H, '*High Speed IP Packet forwarding over internet using ATM technology*' SPIE-Int. Soc. Opt. Eng. Proceedings of Spie - the International Society for Optical Engineering, vol.2608, 1995, pp.44-52. USA.

[EXP] Definition of Optimum Traffic Control Parameters and Results of Trials; EXPERT Deliverable 15; AC094/EXPERT/WP41/DS/R/P/015/B1.

[FEL2000] Feldman A Greenberg A, Lund C, Reingold N Rexford J, '*Netscope: Traffic Engineering for IP networks*', IEEE Network, Vol.14 No.2, March-April 2000, pp.11-19.

[FER98] Ferguson P, Huston G, '*Quality of Service Delivering QoS on the Internet and in Corporate Networks*', 1998, ISBN 0-471-24358-2.

[FLO93] Floyd S, Jacobson V, '*Random Early Detection Gateways for Congestion Avoidance*', IEEE/ACM Transactions on Networking, Vol. 1, No. 4, August 1993.

[FLO95] Floyd S, Jacobson V, '*Link-sharing and Resource Management Models for Packet Networks*', IEEE/ACM Transactions on Networking, Vol. 3 No. 4, pp 365-386, August 1995.

[FRY2000] Fryer J, '*IP+ATM=MPLS*', Telecommunications Online, February 2000, http://www.telecoms-mag.com/issues/200002/tcs.html.

[FUM98] Fumy Walter, Haas Ingbert, '*Security techniques for the global information infrastructure*', Proceedings of the IEEE Globecom 1998, p 3141-3136.

[GAN] Gandluru, M,'*Optical Networking and Dense Wavelength Division Multiplexing (DWDM)*', July 2000, http://www.cis.ohio-state.edu/~jain/cis788-99/dwdm/index.html

[GUA98] Guarene E, Fasano P, Vercellone V, ' *IP and ATM Integration Perspectives*', IEEE Communications Magazine January 1998.

[IETF] Internet Engineering Task Force Web Site

http://www.ietf.cnni.reston.va.us/html.charters/wg-dir.html.

[ITU] International Telecommunciations Union, http://www.itu.int/

[JAC98] Jacobson V, '*Congestion Avoidance and Control*', Computer Communication Review, Vol.18 No. 4, pp 314-329, August 1988.

[JAIN] Jain R, '*IP over DWDM*', April 2000, http://www.cis.ohio-state.edu/~jain/talks/h_aipwd.html

[JAM99a] Jamoussi.B, *Constraint-Based LSP Setup using LDP*, http://search.ietf.org/internet-drafts/draft-ietf-mpls-cr-ldp-03.txt, September 1999, 12/02/00.

[JOH97] Johnson B, Jotwani K, '*Cells VS. Frames which wins on the backbone*', Data Communications, December 1997.

[JON98] Jon C , Bennett R, Zhang H, *'Hierarchical Packet Fair Queueing Algorithms'*, Proceedings of SIGCOMM'96, August 1996.
Boston, Massachusetts, November 1998, SPIE Vol. 3529 0277-786X/98/.

[KAU99] Kaufman D, '*Delivering Quality of Service on the Internet'*, Telecommunications (Americas Edition) Vol.33, No. 2,February 1999.

[KAT97] Katsube Y, Nagami K, Matsuzawa S, Esaki H, *'Interworking Based on Cell Switch Router – Architecture and Protocol Overview'* Proceedings of the IEEE, Vol. 85, No. 12, December 1997.

[KEM97] Kempainen S, '*Gigabit Ethernet and ATM go Neck and Neck in the Communications Race'*, EDN, Vol.42, No. 1, 2nd January 1997.

[KES98] Keshav S, Sharma R, '*Issues and Trends in Router Design'*, IEEE Communications Magazine, May 1998.

[KLE76] Kleinrock L, '*Queuing Systems, Vol.2: Computer Applications'*, John Wiley and Sons, 1976.

[KLE96] Kleinrock L, Gail R, *'Queueing Systems: Problems and Solutions'*, John Wiley and Sons, 1996.

[KUM98] Kumar V P, Lakshman, Stiliadis D, '*Beyond Best Effort: Router Architectures for the Differentiated Services of Tomorrow's Internet'*, IEEE Communications Magazine, May 1998.

[LAN21] '*LAN Emulation over ATM'*, 1.0 af-lane-0021.000, ftp://ftp.atmforum.com/pub/approved-specs/af-lane-0021.000.pdf

[LAN93] '*LAN Emulation Client Management Specification*', af-lane-0093.000, ftp://ftp.atmforum.com/pub/approved-specs/af-lane-0093.000.pdf

[LAW91] Law A, Kelton W, '*Simulation Modeling and Analysis'*¸ Second Edition, McGraw Hill International Editions, 1991, ISBN 0-07-100803-9

[LI2000] Li.L, Feldman.N, Andersson.L et al, *'IP Traffic Engineering Using MPLS Explicit Routing in Carrier Networks'*, http://www.nortelnetworks.com/products/announcements/mpls/source/collateral/whitepaper.html, 28/02/00.

[LU98] Lu Hui-Lan, Faynberg Igor, et al, *'Network Evolution in the context of the global information infrastructure'*, IEEE Communications Magazine, Vol. 36, No.8, p 98-102, August 1998.

[LUK97] Lukowsky J, Polit S, '*IP Packet Switching on the GIGAswitch/FDDI System*', http://www.networks.europe.digital.com/dr/techart/gsfip-mm.html January 1997.

[LUO] Luoma M, Peuhkuri M, Yletyinen, '*Quality of service for IP voice services – is it necessary?*', Part of the SPIE Conference on Quality-of-Service Issues Related to the Internet, Boston, Massachusetts, November 1998, SPIE Vol. 3529 0277-786X/98/.

[MA97] Ma Q, Steenkiste P, '*Quality of Service Routing for Traffic with Performance Guarantees*', Published by Chapman and Hall, IFIP

[MAN97] Mankin.A, Baker.F, Braden.B, et al., *Resource Reservation Protocol (RSVP) Version 1 Applicability Statement*, http://info.internet.isi.edu:80/in-notes/rfc2208.txt.

[MAN98] Manchester J, Anderson J, Doshi B, Dravida S, '*IP Over SONET*',

IEEE Communications Magazine, May 1998.

[McD] McDermott, '*Optical Networks in the Real World*', IEEE/LEOS Summer Topical Meeting Digest Broadband Optical Networks and Technologies: An Emerging Reality, 1998 pp1137-8, New York, NY,USA.

[MER000] Mercankosk G, '*The Virtual Wire Per Domain behaviour – Analysis and Extensions*', July 2000, http://search.ietf.org/internet-drafts/draft-mercankosk-diffserv-pdb-ww-00.txt.

[MET99a] Metz C, '*RSVP: General-Purpose Signaling for IP*', May-June 1999, http://computer.org/internet/

[MET99b] Metz C, '*IP Switching Protocols and Architectures*', McGraw-Hill Publishers ISBN 0-07-041953-1

[MIA99] Miah B, '*A new charging scheme for ATM based on QoS*', PhD Thesis, University of London, June 1999.

[MIL3] Mil 3 Web Page, OPNET, http://www.mil3.com

[MON99] Monday M, '*IP QoS: At the Edge and in the Core*', Telecommunications (Americas Edition) Vol. 33 No. 4 Part 2, April 1999, http://www.telecommagazine.com.

[MPO87] '*Multi-Protocol Over ATM Specification v1.0*', af-mpoa-0087.000, ftp://ftp.atmforum.com/pub/approved-specs/af-mpoa-0087.000.pdf.

[MPO114] *'Multi-Protocol Over ATM Specification v1.0'*, af-mpoa-0114.000, ftp://ftp.atmforum.com/pub/approved-specs/af-mpoa-0114.000.pdf

[NAH98] Nahrstedt K, Chen S, *'An Overview of Quality of Service Routing for the Next Generation High-Speed Networks: Problems and Solutions'*, IEEE Network, Special Issue on Transmission and Distribution of Digital Video, November 1998.

[NIC2000] Nichols K, Carpenter B, '*Definition of Differentiated Services Behaviour Aggregate and Rules for their Specificiation'*, February 2000, http://search.ietf.org/internet-drafts/draft-ietf-diffserv-ba-def-01.txt.

[NOR99] White Paper, '*IP QoS – A Bold New Network'*, http://www.nortelnetworks.com/

[NOR2000] White Paper, '*Quality of Service in Frame-Switched Networks'*, http://www.nortelnetworks.com/products/02/papers/3510.html

[PAG] Page M, *' Architectural Options for Future IP Networks'*, http://www.nortelnetworks.com

[PAR94] Partridge G, '*Gigabit Networking'*, Addison Wesley Publishing, 1994, ISBN 0-201-56333-9.

[PAR97] Park C, Choi H, Kim J, Lee J, *'Next Hop Resolution using Classical IP over ATM*', Proceedings. 22nd Annual Conference on Local Computer Networks. LCN'97 (Cat. No.97TB100179). IEEE Comput. Soc. 1997, pp.106-10. Los Alamitos, CA, USA

[PHI] Phillips C, Kirkby P, *'ATM and the Future of Telecommunciations Networking', Nortel Networks.*

[PIT93] Pitts J M, '*Cell-Rate simulation modelling of asynchronous transfer mode telecommunications networks'*, PhD Thesis, University of London, 1993.

[PIT96] Pitts J M, Schormans J A, *'Introduction to ATM Design and Performance'*, Published by John Wiley and Sons, 1996, ISBN 0-471-96340-2.

[PIT2000] Pitts J M, Schormans J A, *'Introduction to IP and ATM Design and Performance'*, 2nd Edition, Published by John Wiley and Sons, Autumn 2000.

[REK99] Rekhter.Y, Rosen.E, *Carrying Label Information in BGP-4,* http://search.ietf.org/internet-drafts/draft-ietf-mpls-bgp4-mpl-04.txt, January 2000.

[RFC1247] Moy J, '*OSPF Version 2'*, ftp://ftp.isi.edu/in-notes/rfc1247.txt

[RFC1349] Almquist P, '*Type of Service in the Internet Protocol Suite'*, ftp://ftp.isi.edu/in-notes/rfc1349.txt.

[RFC1577] Laubach M, '*Classical IP and ARP over ATM'*, January 1994, ftp://ftp.isi.edu/in-notes.rfc1577.txt.

[RFC1633] Braden R, Clark D, Shenker S, '*Integrated Services in the Internet Architecture: an Overview'*, June 1994, ftp://ftp.isi.edu/in-notes/rfc1633.txt.

[RFC1661] Simpson W, '*The Point –to-Point Protocol (PPP)*, July 1994, http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1661.txt

[RFC1771] Rekhter y, Li T, '*A Border Gateway Protocol 4 (BGP-4)'*, ftp://ftp.isi.edu/in-notes/rfc1771.txt.

[RFC1987] Newman P, Edwards W, Hinden R, et al, '*Ipsilon's General Switch Management Protocol Specification Version 1.1'* , August 1996, ftp://ftp.isi.edu/in-notes/rfc1987.

[RFC2205] Braden R, Zhang L, Berson S, Herzog S, '*Resource ReSerVation Protocol – Version 1 Functional Specification'*, September 1997, ftp://ftp.isi.edu/in-notes/rfc2205.txt.

[RFC2210] Wroclawski J, '*The Use of RSVP with IETF Integrated Services'*, September 1997, ftp://ftp.isi.edu/in-notes/rfc2210.txt.

[RFC2225] Laubach M, Halpern J, '*Classical IP and ARP over ATM'*, April 1998, ftp://ftp.isi.edu/in-notes/rfc2225.txt.

[RFC2297] Newman P, Edwards W, Hinden E, et al, '*Ipsilon's General Switch Management Protocol Specification Version 2.0'*, March 1998, ftp://ftp.isi.edu/in-notes/rfc2297.

[RFC2332] Luciani J, Katz D, Piscitello D,Cole B,Doraswamy N, '*NBMA Next Hop Resolution Protocol (NHRP)*, April 1998, ftp://ftp.isi.edu/in-notes/rfc2332

[RFC2336] Luciani J, '*Classical IP to NHRP Transition'*, July 1998, ftp://ftp.isi.edu/in-notes/rfc2336.txt.

[RFC2386] Crawley E, Nair R, Rajagopalan B, Sandick H, '*A Framework for QoS-based Routing in the Internet'*, August 1998, ftp://ftp.isi.edu/in-notes/rfc2386.txt.

[RFC2430] Li T, Rekhter Y, '*A Provider Architecture for Differentiated Services and Traffic'*, October 1998, ftp://ftp.isi.edu/in-notes/rfc2430.txt.

[RFC2474] Nichols K, Blake S, Baker F, Black D, *'Definition of the Differentiated Services Field (DS Field) in the Ipv4 and Ipv6 Headers'*, December 1998, ftp://ftp.isi.edu/in-notes/rfc2474.

[RFC2638] Nichols K, Jacobson V, Zhang L, '*A Two-bit Differentiated Services Architecture for the Internet'*, July 1999, ftp://ftp.isi.edu/in-notes/rfc2638.

[RFC2676] Apostolopoulos G, Kama S, Williams D, Guerin R, Orda A, Przygienda T, 'QoS Routing Mechanisms and OSPF Extensions', August 1999, ftp://ftp.isi.edu/in-notes/rfc2676.

[RFC2750] Herzog S, *'RSVP Extensions for Policy Control'*, January 2000, ftp://ftp.isi.edu/in-notes/rfc2750.txt.

[RFC2836] Brim S, Carpenter B, Le Faucheur F, *'Per Hop Behaviour Identification Codes'*, May 2000, ftp://ftp.isi.edu/in-notes/rfc2836.

[ROB97a] Roberts E, *'IP on Speed'*, Data Communications Magazine, March 1997.

[ROB97b] Roberts E, *'IP Routing with ATM Speed'*, Data Communications Magazine, January 1997.

[ROS98] Rosenberg J, Schulzrinne H, *'Internet Telephony Gateway Location'*, Proceedings IEEE Infocom'98, Conference on Computer Communications, Seventeenth Annual Joint Conference of the IEEE Computer and Communciations Societies, IEEE Part Vol2,1998,pp488-96.

[ROS99a] Rosen.E, Rekhter.Y, Tappan.D, et al., '*MPLS Label Stack Encoding*', http://search.ietf.org/internet-drafts/draft-ietf-mpls-label-encaps-07.txt, September 1999, 17/03/00.

[ROS99b] Rosen.E, Viswananthan.A, Callon.R, '*Multiprotocol Label Switching Architecture*', http://search.ietf.org/internet-drafts/draft-ietf-mpls-arch-06.txt, August 1999, 03/03/00.

[ROS2000a] Rosen E, Rekhter Y, Fedorkov G et al, '*MPLS Stack Encoding'*, http://search.ietf.org/internet-drafts/draft-ietf-mpls-label-encaps-08.txt, July 2000.

[SAL97] Salgarelli L, Marco M, Meroni G, Trecorde V, *'Efficient Transport of IP Flows Across ATM Networks*', IEEE ***ATM*** '97 Workshop Proceedings (Cat. No.97TH8316). IEEE. 1997, pp.43-50. New York, NY, USA.

[SCH96] Schill A, Kuhn S, Breiter F, '*Internetworking over ATM: Experiences with IP/IPng and RSVP*', Proceedings of JENC7. 7th Joint European Networking

Conference (JENC7). Networking in the Information Society. TERENA. 1996, pp.161/1-11.

[SCH2000a] Schormans J A, Pitts J M, et al '*Buffer Overflow Probability for Multiplexed On-Off VoIP Sources'*, Electronics Letters, 16th March 2000, Vol.36, No.6, pp523-524.

[SCH2000b] Schormans J A, Pitts J M et al, '*Analysis of Local Internet / Enterprise Networking Multiplexer for on-off VOIP source'*, Submitted to Electronics Letters.

[SEE2000] Seetharaman S, '*IP over DWDM'*, July 2000, http://www.cis.ohio-state.edu/~jain/cis788-99/ip_dwdm/index.html

[SEM2000a] Semeria.C, '*Multiprotocol Label Switching: Enhancing Routing in the New Public Network'*, http://www.juniper.net/techcenter/techpapers/mpls/mpls.htm, 22/02/00, 25/02/00.

[SEM2000b] Semeria.C, '*Traffic Engineering for the New Public Network'*, http://www.juniper.net/techcenter/techpapers/TE_NPN.html, 22/02/00, 25/02/00.

[SHA98a] Shankar B, '*Breaking the IP Gridlock'*, Telecommunications (International Edition) Vol.32, No.2, p59-60,62, February 1998.

[SHA98b] Shaikh A, Rexford J, Shin K, *'Evaluating the Overheads of Source-Directed Quality of Service Routing'*, Proceedings 6th International Conference on Network Protocols, IEEE Computer Society, 1998, pp.42-51.

[SHA99] Shaikh A, Rexford J, Shin A, *'Load-Sensitive routing of long-lived flows'*, ACM Computer Communication Review, Vol.29, No.4, pp. 215-216 October 1999.

[SIY97] Siyan S Karanjit '*Inside TCP/IP'*, Third Edition, ISBN 1-56205-714-6

[SMI99] Smith.P A, Jamoussi.B, '*MPLS Tutorial and Operational Experiences'*, http://www.nanog.org/mtg-9910/mpls.html, October 1999.

[SPO98] Spohn D, McDysan D, '*ATM Theory and Applications'*, Mc Graw Hill Series on Computer Communications, 1998, ISBN 0-07-045346-2.

[TES99] Testa B, '*Try Walking in my Shoes'*, August 1999, http://www.internettelephony.com/archive/8.23.99/Cover/cover.html

[TIN96] Tinsley R, Lang L, '*ATM MPOA vs. IP Switching'*, September 1996, http://www.newbridgenetworks.com/

[TRI98] White Paper, *‘Comparison of IP-over-SONET and IP-over-ATM Technologies',* March 1998, http://www.trillium.com/whats-new/wp_ip.html

[VAU97] Vaughan-Nichols S, ’*Switching to a Faster Internet’,* Computer, Vol.30 No. 1, p31-32, January 1997.

[WAL97] Walker D, Kelly F, Solomon J, *‘Tariffing in the New IP/ATM Environment’,* Telecommunications Policy, Vol.21, pp. 283-295, 1997.

[WHI98] White P, ‘*ATM Switching and IP Routing Integration: The Next Stage in Internet Evolution’,* IEEE Communications Magazine, April 1998.

[WOR2000] Worster T, Katsube Y et al, *‘MPLS Label Stack Encapsulation in IP’,* August 2000, http://search.ietf.org/internet-drafts/draft-worster-mpls-in-ip-02.txt.

[XIA97] Xiao Xipeng, Ni Lionel, Vuppala Vibhavasu, *‘An Overview of IP Switching and Tag Switching*’, Proceedings. 1997 International Conference on Parallel and Distributed Systems (Cat. No.97TB100215). IEEE Comput. Soc. 1997, pp.669-75. Los Alamitos, CA, USA.

[XIA99] Xiao Xipeng, ‘*Traffic Engineering with MPLS in the Internet’*, IEEE Network Magazine, Mar 2000.