NOTE TO USERS

This reproduction is the best copy available.

UMI[®]

Design, Implementation and Performance Evaluation of MPLS Layer 2 Martini VPN

by
Jing Li

A thesis submitted to
the Faculty of Graduate Studies and Research
in partial fulfillment of
the requirements for the degree of

Master of Applied Science
in
Electrical Engineering

Ottawa-Carleton Institute for Electrical and Computer Engineering
Department of Systems and Computer Engineering

Carleton University
Ottawa, Ontario
September, 2004

© copyright
2004, Jing Li

Library and
Archives Canada

Bibliothèque et
Archives Canada

Published Heritage
Branch

Direction du
Patrimoine de l'édition

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 0-612-97419-7

Our file Notre référence

ISBN: 0-612-97419-7

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

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

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

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

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

Acknowledgements

First of all, I would like to take this opportunity to give my deep appreciation and respect to my thesis supervisor, Dr. Dorina C. Petriu, for her trust, patient guidance and inspiration, which were always there when I needed it.

Many thanks also go to EION Inc for providing the funding and environment for my research work. I would like to thank Anand Srinivasan, Pramod Dhakal and Yi Lei for their help throughout my research at EION.

Funding from the National Capital Institute of Telecommunications (NCIT) has made this research work possible. Special thanks are due to Dr. Mohamed Zail and Dr. Robert Cranhall from NCIT for their support for this project.

Thanks to all other friends who have helped my one way or another.

Finally, I would like to thank my wife, my parents and my brother, who have constantly given me encouragement, support and love. To them I dedicate this thesis.

Acronyms

ACE	Adaptive Communication Environment
ATM	Asynchronous Transfer Mode
BGP	Border Gateway Protocol
CE	Customer Edge
CoS	Class of Service
CPE	Customer Premises Equipment
CR-LDP	Constraint-based Routed Label Distribution Protocol
CSMA/CD	Carrier Sense Multiple Access / Collision Detection
DLCI	Data Link Circuit Identifier
EGP	Exterior Gateway Protocol
ER	Explicit Route
FEC	Forward Equivalence Class
IETF	Internet Engineering Task Force
IGP	Interior Gateway Protocol
IP	Internet Protocol
IPLS	IP-only LAN Service
IPSec	Internet Protocol Security
IPv4	Internet Protocol Version 4
IPX	Internet Packet Exchange
L2TP	Layer 2 Tunneling Protocol
LAN	Local Area Network
LDP	Label Distribution Protocol
LER	Label Edge Router
LIB	Label Information Base
LLC	Logical Link Control
LSP	Label Switching Path
LSR	Label Switching Router
MAC	Media Access Control

MPLS	Multi-Protocol Label Switching
MTU	Maximum Transfer Unit
NIC	Network Interface Card
OSI	Open System Interconnect
OSPF	Open Shortest Path First
PDU	Protocol Data Unit
PE	Provider Edge
PPTP	Point-to-point Tunneling Protocol
PPVPN	Provider Provisioned Virtual Private Network
QoS	Quality of Service
RIP	Routing Information Protocol
RSVP	Resource Reservation Protocol
RSVP-TE	Resource Reservation Protocol – Traffic Engineering
SNA	System Network Architecture
TLV	Type-Length-Value
VC	Virtual Circuit
VCI	Virtual Circuit Identifier
VPI	Virtual Path Identifier
VPLS	Virtual Private LAN Service
VPN	Virtual Private Network
VPWS	Virtual Private Wire Service
VR	Virtual Router

Table of Contents

CHAPTER 1 INTRODUCTION	1
1.1 BACKGROUND.....	1
1.2 THESIS OBJECTIVE	3
1.3 THESIS CONTRIBUTION	4
1.4 THESIS OUTLINE	5
CHAPTER 2 BACKGROUND: VPN, MPLS, QOS.....	7
2.1 VPN.....	7
2.1.1 Motivation for using VPNs.....	8
2.1.2 VPN Applications	9
2.1.3 VPN Classification.....	11
2.1.3.1 OSI layer on which the VPN operates	11
2.1.3.2 Customer provisioned or provider provisioned.....	13
2.1.3.3 How the network entities collaborate	14
2.2 MPLS	16
2.2.1 Overview.....	17

TABLE OF CONTENTS

2.2.2	Network devices.....	19
2.2.3	FEC	19
2.2.4	Label binding	20
2.2.5	Label Distribution	23
2.2.6	Label Switched Path	24
2.2.7	Traffic Engineering.....	25
2.3	QoS	27
2.3.1	Traffic Classification	27
2.3.2	Network Element QoS	27
2.3.3	End-to-End QoS.....	29
CHAPTER 3	MARTINI VPN.....	30
3.1	OVERVIEW	30
3.1.1	Motivation for Martini VPN	30
3.1.2	Martini Drafts.....	31
3.1.3	Network Topology	31
3.2	DATA TRANSPORT METHOD	33

TABLE OF CONTENTS

3.2.1	LSP Tunnel Connection	33
3.2.2	Virtual Circuit Connection.....	34
3.2.3	Control Word Negotiation	39
3.3	GENERAL ENCAPSULATION METHOD.....	41
3.3.1	Tunnel Header.....	42
3.3.2	Demultiplexer Field	43
3.3.3	Emulated VC Encapsulation.....	44
3.3.4	Packet Forwarding Procedure	46
CHAPTER 4 DESIGN AND IMPLEMENTATION		48
4.1	SOFTWARE FRAMEWORK	48
4.2	REQUIREMENT ANALYSIS AND HIGH LEVEL DESIGN.....	52
4.2.1	General Requirements.....	52
4.2.2	Concurrency	53
4.3	CLI COMMANDS.....	54
4.4	CONTROL PLANE IMPLEMENTATION	56
4.4.1	Software Structure	56

4.4.2	Label Space Management	57
4.4.3	Configuration Table	59
4.4.4	LSP Tunnel Setup	60
4.4.5	VC Connection Setup	60
4.4.6	Control Word Negotiation	61
4.5	FORWARDING PLANE IMPLEMENTATION	63
4.5.1	Ethernet Emulation	63
4.5.2	Software Structure	67
4.5.3	VC Tables	68
4.5.4	VPN Packet Processing at Ingress PE	69
4.5.5	VPN Packet Processing at Egress	70
CHAPTER 5 PERFORMANCE EVALUATION.....		72
5.1	TEST ENVIRONMENT	72
5.2	TOOLS.....	73
5.3	TEST CASES	75
5.4	ANALYSIS OF TEST RESULTS	77

5.4.1	Data Analysis Methodology	77
5.4.1.1	Confidence Interval	77
5.4.1.2	Confidence Interval with Specified Precision	79
5.4.1.3	Procedure of Measurement Data Evaluation	82
5.4.2	Measurement Data	83
5.4.3	Analysis of Measurement Data	86
CHAPTER 6 PROPOSED QOS ENHANCEMENT		92
6.1	PROBLEM DEFINITION	92
6.2	PROPOSED SOLUTION	94
6.3	IMPLEMENTATION AND EVALUATION	95
6.4	SIMULATION	97
6.4.1	Simulation Model	98
6.4.1.1	Network Topology	98
6.4.1.2	Node Model	99
6.4.1.3	Process Model	100
6.4.1.4	Link Model	102

6.4.1.5	Packet Format.....	103
6.4.2	Simulation Experiments.....	103
6.4.3	Simulation Analysis	106
6.4.3.1	With and Without CoS	106
6.4.3.2	Inter-arrival Time and Throughput	108
6.4.3.3	Queue Size and Throughput.....	112
6.4.3.4	Quantum and Throughput	114
6.4.4	Summary	116
CHAPTER 7	CONCLUSION AND FUTURE WORK.....	118
7.1	SUMMARY OF RESEARCH WORK.....	118
7.2	CONCLUSION OF PRESENT RESEARCH.....	120
7.3	FUTURE WORK	121
REFERENCES		122

List of Figures

Figure 2-1 Remote Access	9
Figure 2-2 Intranet	10
Figure 2-3 Extranet	11
Figure 2-4 MPLS Generic Label Format.....	21
Figure 2-5 ATM as the Data Link Layer	22
Figure 2-6 Frame Relay as the Data Link Layer.....	22
Figure 2-7 PPP, Ethernet as Data Link Layer.....	23
Figure 2-8 MPLS Traffic Engineering.....	26
Figure 3-1 Network Topology	32
Figure 3-2 Format of TLV	35
Figure 3-3 Label Mapping Message Format.....	36
Figure 3-4 VC FEC Element.....	37
Figure 3-5 Interface Parameter	39
Figure 3-6 Control Word Negotiation.....	41
Figure 3-7 Martini VPN Encapsulation	42
	X

Figure 3-8 Packet Forwarding	47
Figure 4-1 Existing Software Architecture	49
Figure 4-2 MPLS Forward Engine	50
Figure 4-3 Software Architectural Changes	53
Figure 4-4 Control Plane Structure	56
Figure 4-5 VC Connection Setup.....	61
Figure 4-6 Ethernet in OSI Model	63
Figure 4-7 Ethernet Data Frame Format.....	65
Figure 4-8 Communication between CE Devices.....	66
Figure 4-9 MPLS Forwarding Structure	67
Figure 4-10 VPN Packet Processing at Ingress PE.....	70
Figure 4-11 VPN Packet Processing at Egress PE.....	71
Figure 5-1 Test Environment.....	73
Figure 5-2 Qcheck Endpoint.....	74
Figure 5-3 Test Case without VPN	76
Figure 5-4 Test Case with VPN.....	76

Figure 5-5 TCP Response Time.....	87
Figure 5-6 TCP Throughput.....	88
Figure 5-7 UDP Response Time	89
Figure 5-8 UDP Throughput.....	90
Figure 6-1 Test Environment.....	96
Figure 6-2 Throughput Comparison	97
Figure 6-3 Network Topology in Simulation.....	98
Figure 6-4 CE Node Model.....	99
Figure 6-5 PE Node Model	100
Figure 6-6 "proc" Process Model.....	101
Figure 6-7 "schdlr" Process Model	102
Figure 6-8 Packet Format.....	103
Figure 6-9 Experiment Case 1: End-to-End Throughput.....	107
Figure 6-10 Experiment Case 2: End-to-End Throughput.....	108
Figure 6-11 Experiment Case 3: End-to-End Throughput.....	110
Figure 6-12 Experiment Case 4: End-to-End Throughput.....	111

Figure 6-13 Experiment Case 5: End-to-End Throughput.....	113
Figure 6-14 Experiment Case 6: End-to-End Throughput.....	114
Figure 6-15 Experiment Case 7: End-to-End Throughput.....	115
Figure 6-16 Experiment Case 8: End-to-End Throughput.....	116

List of Tables

Table 5-1 TCP Response Time Measurement	84
Table 5-2 TCP Throughput Measurement	84
Table 5-3 UDP Response Time Measurement.....	85
Table 5-4 UDP Throughput Measurement	85
Table 6-1 Measurement Data Summary	96
Table 6-2 Key Parameter Setting.....	104
Table 6-3 Simulation Experiments Summary.....	105
Table 6-4 Summary for experiment Case 1	106
Table 6-5 Summary for experiment Case 2	107
Table 6-6Table 6 6 Summary of experiment Case 3	109
Table 6-7 Summary for experiment Case 4	111
Table 6-8 Summary for experiment Case 5	112
Table 6-9 Summary for experiment Case 6	113
Table 6-10 Summary for experiment Case 7	114
Table 6-11 Summary for experiment Case 8	116

Chapter 1 Introduction

1.1 Background

The popularity of Internet applications and the increase in the number of users have generated a considerable growth in Internet communications. Applications like videoconferencing, voice over IP or high-speed subscriber access via DSL raise a high demand for bandwidth and requirements for quality.

With the increasing demands for broadband IP networks, a problem became critical: the traditional IP routing and forwarding was not fast enough! The root cause of this problem is that traditional IP routing and forwarding are overhead intensive. Conventional IP routers look up their routing tables using the IP header of every single packet to decide how to forward the packet through the network. For routers in a large network, the size of the routing table could be considerably large, so the table lookup may take a long time. In addition, the use of the subnet mask requires matching the destination IP address of an incoming packet with a prefix in the routing table based on the longest matching principle; this operation is also computationally intensive. Thus the performance of traditional IP routing is not satisfying for a large size network.

A new approach to packet forwarding found in the Multiple Protocol Label Switching (MPLS) is becoming increasingly popular in IP networking. This protocol aims at increasing the performance of Internet routers for current IP networks. The main idea behind MPLS is the use of the forwarding paradigm based on label swapping. Each

router is responsible for assigning and distributing a set of labels using a variety of criteria; it decouples the forwarding of a packet from the content of the packet's header. This reduces the processing load associated with forwarding operations. The size of a routing table is reduced due to the aggregation of routes into a single pipe in a MPLS cloud.

Some large corporations have multiple sites geographically apart from each other. Each site has a local network. One site may need to communicate to others to send emails, share resources, etc. It would be quite expensive to connect these sites together by private lines. More economical solutions for multi-site connections can be found by using the public network infrastructure. VPN (Virtual Private Network) is the technology that meets this requirement. A VPN is a private multi-site network created by using shared resources within a public network. VPN brings benefits for both customers and service providers. It provides a cost effective method for the customers to connect their geographically separated sites together and let them work just like a private network. Service providers can generate more revenue by providing VPN services to their customers.

Although VPNs can be designed to transport traffic at different OSI layers, the extension of Layer 2 infrastructure, such as Frame Relay, ATM, or Ethernet, seems to be of great value to some customers, especially those in a metropolitan environment. At the same time, service providers with excess capacity in their existing IP core network would want to augment their Layer 2 service offerings.

One of the benefits of Layer 2 VPN is the separation of responsibilities. The service provider is responsible for Layer 2 connectivity while the customer is responsible for Layer 3 connectivity. Another benefit is the ease of configuration. From customer's point of view, configuring a Layer 2 VPN is nothing more than configuring a local Layer 2 connection.

Since MPLS provides an efficient way of forwarding the packets through a pre-configured path, and already has packet-processing modules at the ingress and egress nodes of the path, it can be augmented to forward the VPN traffic across a public network with relative ease. The speed of forwarding is respectable enough for the distant nodes in the VPN network to insure good performance.

1.2 Thesis Objective

The main goal of this research is to study the latest technologies for Layer 2 Ethernet-based point-to-point VPN over MPLS network through actual design, implementation and performance evaluation, and to propose improvements where possible. The following research activities were performed in the thesis:

- Study the available IETF standards or drafts for Ethernet-based point-to-point VPN over MPLS network.
- Design and implement the control plane and forwarding plane of Martini VPN module on EION Open IP software suite.

- Measure the end-to-end performance for the implemented module in terms of delay and throughput.
- Propose a new QoS enhancement approach at the edge nodes of MPLS domain, which provides port-based DiffServ-like service to the VPN traffic flows.

We built a simulation model and conducted simulation experiments to evaluate the effectiveness of this approach in terms of end-to-end throughput. The simulation was conducted on schemes with and without the proposed QoS enhancement, in order to compare them. For the proposed QoS enhancement, simulations were performed with different experimental parameters, such as traffic density, queue size and weight setting.

1.3 Thesis Contribution

The contributions of this research can be divided in two parts:

Practical Contributions:

- Designed and implemented Ethernet-based Martini VPN module on EION Open IP software suite including control plane and forwarding plane. The module is implemented in C++ in a Unix environment.
- Measured the end-to-end performance of the implemented VPN in terms of delay and throughput.

- Designed and implemented a simulation model based on OPNet and performed simulation experiments to study the proposed QoS enhancement approach.

Contributions to knowledge:

- Proposed a QoS enhancement model for Layer 2 VPN at the edge nodes of the MPLS networks to provide port-based DiffServ-like services.

1.4 Thesis Outline

The rest of the thesis is composed of six chapters, as follows.

Chapter 2 gives a literature overview of the main concepts used in the thesis, such as VPN, MPLS and QoS. Chapter 3 explains the Martini VPN technology in detail. Chapter 4 gives a detailed explanation of the design and implementation of Martini VPN in EION Open IP environment. Chapter 5 presents the performance evaluation of MPLS Martini VPN implementation. The chapter describes the test bed and software tools used for measurements; discusses the test cases and analyzes the results. Chapter 6 proposes an approach for Layer 2 QoS enhancement that can be applied to PE routers. The goal is to provide differentiated service to preferred VPN traffic at the edge of the MPLS network, which contributes to the overall VPN QoS. Real system implementation and simulation are performed to validate the feasibility of the approach. After the presentation of the proposed approach for Layer 2 QoS enhancement, the chapter discusses the implementation and the test results, An OPNet simulation model for analyzing the performance of the proposed solution, the simulation scenarios used and the analysis of

the results are also discussed in this chapter. Chapter 7 gives the summary and conclusions of this research. Future developments of this research are also identified in this chapter.

Chapter 2 Background: VPN, MPLS, QoS

This chapter gives an overview of the main concepts used in this research. The first part of the chapter introduces the motivation for VPN, VPN applications, VPN classification and VPN challenges. The second part describes the MPLS protocol and the third part QoS methodologies.

2.1 VPN

“A VPN is a communications environment in which access is controlled to permit peer connections only within a defined community of interest, and is constructed through some form of partitioning of a common underlying communications medium, where this underlying communications medium provides services to the network on a non-exclusive basis.” [26]

In a simpler and more straightforward form, VPN can also be described as:

“A VPN is private network constructed within a public network infrastructure, such as the global Internet.” [26]

A Virtual Private Network allows an organization to join the Local Area Networks at two or more locations together over the Internet. This allows users to securely access information wherever it may be located on the network.

2.1.1 Motivation for using VPNs

The basic motivation for VPN's is the cost effectiveness of communications. Traditional private networks facilitate connectivity among various network entities through a set of leased lines. The capacity of the lines is not shared with others and is always available. The traffic on the private network belongs only to the company deploying the network, so the performance can be guaranteed. But guaranteed performance comes with a price: traditional private networks are expensive to plan and deploy, especially when international locations are involved.

With the rapidly evolving network technologies and explosive growth of Internet applications, public data networks have already covered large geographical areas worldwide. These public networks provide shared resources to a large number of end users at a very low cost. The maturity of public data networks provides the physical and technical possibility that private networks can be connected through a public data network at a much lower cost. Also, service providers can augment their revenue by providing this type of services.

Another drive for VPNs comes from the market. To increase productivity and reduce costs, many companies encourage their employees to work from home. The development of e-commerce applications makes it possible for the companies to manage their inventory, supply chain and data interchange on line. The companies need reliable connections from their offices to the ever increasing telecommuters, from headquarter to branch offices, from company to suppliers or customers [26].

2.1.2 VPN Applications

Remote Access

VPN provides the possibility for remote users to access corporate resources over the public Internet, while maintaining the privacy of information. Figure 2-1 shows a VPN connection used to connect a remote user to a corporate intranet.

Figure 2-1 Remote Access

Rather than making a long distance (or 1-800) call to a corporate or outsourced network access server (NAS), the user calls a local ISP. Using the connection to the local ISP, the VPN software creates a virtual private network between the dial-up user and the corporate VPN server across the Internet.

Intranet

An intranet is a private network that is contained within an enterprise. It may consist of many interlinked local area networks and also use leased lines in the wide area network. Typically, an intranet includes connections through one or more gateway computers to the outside Internet. The main purpose of an intranet is to share company

information and computing resources among employees. An intranet can also be used to facilitate working in groups and for teleconferences. Figure 2-2 shows a VPN connecting two corporate intranets.

Figure 2-2 Intranet

Extranet

An extranet is a private network that uses public networks to securely share part of the information or operations of a business with suppliers, vendors, partners, customers, or other businesses.

An extranet can be viewed as part of a company's intranet that is extended to users outside the company. It has also been described as a "state of mind" in which the Internet is perceived as a way to do business with other companies as well as to sell products to customers. Figure 2-3 shows a company's extranet connecting with its supplier and customer.

Figure 2-3 Extranet

2.1.3 VPN Classification

There are many different technical solutions to achieve VPN connections. This section describes the classification of VPN technologies. VPN can be classified from different angles. These solutions have their own emphasis and specific applications. Overlapping may exist among the categories.

2.1.3.1 OSI layer on which the VPN operates

VPN connections can be established at different OSI layers.

Layer 1 VPN

Layer 1 VPN basically provides physical connections among customer sites. Those connections are dedicated to a specific customer. Layer 1 VPN is also called "leased line". This approach is quite expensive but the performance is guaranteed since the resource is not shared. TDM, SDH and SONET are examples of this type of VPN connection [50].

Layer 2 VPN

Traditional Layer 2 VPN is based on ATM and Frame Relay virtual circuits, which provides data link connections between customer sites. It requires a dedicated Layer 2 network separated from the IP based network, which brings costs to the service provider in terms of equipment and maintenance. New technologies have been emerging rapidly to extend Layer 2 service across IP based network [9, 28, 30, 31].

Two service models are included in Layer 2 VPN. One is the *point-to-point model*, which is also referred to as *virtual private wire service* (VPWS). The other one is the *multipoint service*, which is also called *virtual private LAN service* (VPLS).

The Point-to-Point Tunnel Protocol (PPTP) [27] and Layer 2 Tunneling Protocol (L2TP) [51] are commonly used protocols that provide encapsulation mechanisms of Layer 2 tunnels. PPTP provides access from Microsoft Windows systems to Microsoft or third party servers with authentication and encryption. L2TP also provides authenticated tunneling over various networks, such as IP, ATM [12].

Layer 3 VPN

Layer 3 VPN provides IP level tunnel connections between customer sites [9, 28]. RFC2547 (MPLS/BGP VPN) [40], Virtual Router (VR) [7] and IP Security (IPSec) [24] fall into this category.

MPLS/BGP VPN transports IPv4 VPN traffic through a provider's network via MPLS tunnels.

There is another L3VPN solution based on the *virtual router* concept. Each PE router runs multiple 'virtual routers', each corresponding to a VPN forwarding table. A tunnel is set up between virtual routers for the same VPN.

IPSec provides a suite of protocols for tunneling and encryption, which are used to establish a CE based VPN. IPsec provides three basic capabilities: authentication and data integrity, confidentiality and secure communication [24, 29].

2.1.3.2 Customer provisioned or provider provisioned

CE-Based VPN

For this type of VPN, all the VPN specific functionality is performed in the CE (Customer Equipment). The provider's network has no knowledge of the VPN traffic, which is treated in the same way as the regular traffic. So the PE devices can be standard IP routers. The customers are fully responsible for the establishment, configuration and management of their VPN connections [12, 19].

There are some potential issues for CE-based VPN.

If a CE-based VPN grows to have a large number of sites, scalability issues will emerge, because for a VPN of N sites, a mesh of $N(N-1)/2$ tunnel connections have to be maintained. This requires a considerable large amount of effort in configuration and management.

Heavy workload would be placed on CE devices to perform encryption/decryption, encapsulation/de-capsulation, which consumes significant amount of processing power and memory.

Since customer provisioned VPN connections are transparent to the provider's network, it is impossible for the provider to identify VPN traffic and provide any quality of service or performance guarantee other than best effort.

PE-Based VPN

For this type of VPN, the VPN specific processing takes place in the edge devices of provider's network. CE devices have no knowledge of VPN connections. So CE devices can be standard switches or routers. There is not need for the customer to upgrade their equipment and usually little work for the customer to manage the VPN. On the other hand, the service provider is responsible of the configuration and management of VPN connections, which means that the edge devices of provider's network have to be VPN enabled. Moreover, the quality of service and performance guarantee to VPN traffic become possible in provider's network [37].

2.1.3.3 How the network entities collaborate

Overlay model

With the overlay model, the CE devices are connected via direct point-to-point links, which are also referred to as end-to-end tunnels. A full mesh of connections is required on top of service provider's network. Tunnels can exist at different protocol levels.

PPTP [27] and L2TP [51], which have been mentioned in 2.1.2.1, are commonly used Layer 2 tunneling protocols. IPsec is a typically example of Layer 3 tunneling protocol.

There are several drawbacks with this approach. The first problem is that VPN customers have to design and maintain their VPNs. The second problem is scalability. For a VPN with N sites, each CE has to maintain $N-1$ links to other CE devices to achieve full mesh connectivity. So the amount of links can be considerably large with the growth of the number of sites. Furthermore, the change of VPN topology involves all of the existing sites, so the effort of maintaining the VPN would be large as well. At last, since the provider's network has no knowledge of VPN traffic, the QoS service cannot be applied to it.

Peer model

Peer model approaches aim to solve the problems characteristic to the overlay model. In contrast to the overlay model, each CE device in the peer model only deals with a single PE device of the provider's network to which it is directly connected. The provider's network provides the connectivity among PE devices. The provider's network acts as a peer of the customer networks. From a customer's standpoint, the scalability

increases significantly, since a CE device just needs to peer with one PE device and the major routing workload is offloaded to the provider's network.

2.2 MPLS

In the last decade, the Internet has grown explosively in terms of number of new users and amount of new applications it provides. In addition to the traditional data services currently provided over the Internet, new voice and multimedia services are being developed and deployed. The Internet has become the network that provides these converged services. These new applications have raised the requirements for increased and guaranteed bandwidth in the backbone of the network. However, due to the deployment of these new applications, the network resources of the existing Internet infrastructure, in terms of speed and bandwidth, have been strained even more. The technology has to advance to meet the ever-increasing demand for bandwidth.

In traditional routing environment, routing information is advertised from one router to another using *interior gateway protocols* (IGPs), such as *routing information protocol* (RIP) and *open shortest path first* (OSPF), or *exterior gateway protocols* (EGPs), such as *border gateway protocol* (BGP). Routes, which contain the information that which IP addresses a route can reach through certain interfaces, are then installed in the routing tables of routers. A packet is forwarded through a network in a hop-by-hop fashion. In each router that the packet traverses, the IP address of the packet must be checked for the longest match against the routing table for the next hop entry, until the packet gets to its destination.

Considering the dimension of today's public network, a router in the core network may have to store thousands of routes in its routing table. Therefore, a 32 bits IP address is defined in IPv4 and IP address length is quadrupled in IPv6. So the longest matching algorithm and table lookup may introduce significant latency in packet forwarding at each router along the way.

Most of the routing protocols deployed today are based on algorithms designed to obtain the shortest path, the acceptable bandwidth or other metrics in the network for packet traversal, but additional metrics such as delay, jitter, and traffic congestion are not taken into account, which can further diminish network performance. By using IGPs, routes can be selected based on predefined metrics, such as hop count for RIP or bandwidth usually for OSPF. But the shortest path does not necessarily mean the best path in terms of bandwidth. Even if a path of preferred bandwidth is selected, all packets will be forwarded in same way along that path, the packets will not be rerouted to other path in case of congestion.

These problems for the traditional IP forwarding mechanism have to be solved to meet the increasing demand of bandwidth. The need for a simpler forwarding method with traffic engineering feature and performance of traditional switches emerges. All these needs can be met with multi-protocol label switching (MPLS).

2.2.1 Overview

MPLS has emerged as a solution to meet the bandwidth-management and service requirements for next-generation IP based backbone networks. It is a versatile solution to

the problems existing in traditional IP networks - speed, scalability, quality-of-service (QoS) management, and traffic engineering.

MPLS is standardized by the Internet Engineering Task Force (IETF) to provide efficient designation, routing, forwarding, and switching of traffic flows through the network.

The following are major aspects defined in the MPLS standard:

- Mechanisms to manage traffic flows of various granularities, such as flows between different hardware, machines, or even flows between different applications
- Independence of Layer-2 and Layer-3 protocols
- Mechanism to map IP addresses to simple, fixed-length labels used by different packet-forwarding and packet-switching technologies
- Interfaces to existing routing protocols such as resource reservation protocol (RSVP) and open shortest path first (OSPF)
- Support for IP, ATM, and frame-relay Layer-2 protocols.

MPLS provides a mechanism for forwarding network traffic patterns that is independent of routing tables. Short labels are assigned to packets that describe how to forward them through the network. MPLS is independent of any routing protocol and supports multiple Layer 2 protocols.

The analysis of the packet IP header is performed just once at the ingress edge of MPLS enabled network. Based on its destination IP address, a packet is then assigned to a stream identified by a 20-bit label. A 32-bit MPLS header that contains the label value and other control information is inserted between Layer 2 and Layer 3 headers. Labels are used as lookup indexes into the forwarding table [16].

2.2.2 Network devices

The network devices that participate in the MPLS protocol mechanisms can be classified into label edge routers (LERs) and label switching routers (LSRs).

LSRs operate in the backbone of an MPLS network that participate in the establishment of LSPs using the appropriate label signaling protocol and switch the data traffic based on incoming label in the packet.

LERs operate at the edge of MPLS network. LERs support multiple ports connected to heterogeneous networks (such as frame relay, ATM, and Ethernet), they participate in the establishment of LSPs, forward the traffic on to the MPLS network at the ingress and distributing the traffic back to the access networks at the egress [42].

2.2.3 FEC

The *forward equivalence class* (FEC) is a logical concept that represents a group of packets with the same requirements for their transport. All packets in the same FEC are forwarded in the same manner in the network. Assigning packets to FECs is done once at the ingress LER, based on the service requirements specified in the IP header of the

packets and the destination IP address. Each FEC is assigned a local significant label. LERs maintain a table, which is called label information base (LIB), to store FEC-to-label bindings [40].

2.2.4 Label binding

A label is bound to an FEC. A label identifies the path a packet should traverse. Depending on different Layer 2 protocols, a label is either carried in the Layer 2 header or encapsulated in a MPLS header along with the packet. The label content of packets is examined at each receiving router to determine the next hop and the outgoing label that will replace the incoming label in the packet before it is sent to its next hop.

Once a packet has been classified into a FEC, the label assigned to that FEC would be inserted in the packet. The packet is then forwarded based on its label value. Label switching takes place in each LSR along the path through the backbone network. The label values are of local significance, which means that this mechanism is only suitable for hopping between LSRs.

Forwarding is the criteria for assigning a label to a FEC.

- Destination IP address for unicast routing
- Traffic engineering requirements
- Multicast Address
- Virtual private network (VPN)

- Quality of Service requirements (QoS)

The generic label format is illustrated in Figure 2-4. The label can be embedded in the header of the data link layer (the ATM VCI/VPI shown in Figure 2-5 and the frame-relay DLCI shown in Figure 2-6) or in the shim (between the Layer-2 data-link header and Layer-3 network layer header, as shown in Figure 2-7).

Figure 2-4 MPLS Generic Label Format

Figure 2-5 ATM as the Data Link Layer

Figure 2-6 Frame Relay as the Data Link Layer

Figure 2-7 PPP, Ethernet as Data Link Layer

2.2.5 Label Distribution

LDP

Label distribution protocol (LDP) is defined by the Internet Engineering Task Force (IETF) to distribute labels in a MPLS network. The Extensions to LDP, namely constraint-based routing CR-LDP, have also been defined to support routing based on QoS and CoS requirements [14].

RSVP

Resource reservation protocol (RSVP), which is initially used to reserve bandwidth based on QoS requirements, has also been extended to support exchange of labels.

MPLS Layer 3 VPN uses extended border gateway protocol (BGP) to exchange the labels between LERs [17].

BGP

The label distribution is performed in the upstream direction, meaning a router advertises labels to the router before itself and receives a label advertised by the router after along the path from the source to the destination [40].

2.2.6 Label Switched Path

In MPLS networks, packets are forwarded along pre-established label-switched paths (LSPs). An LSP consists of incoming and outgoing labels at each router along the path from the source to the destination. A set of packets that are bound to the same FEC traverse along a LSP to its destination through the MPLS network. A LSP is a unidirectional path; the return traffic must use a separate LSP.

A LSP is established through label generation and distribution at each LSR along the path. This action can be triggered by incoming traffic flow, control information or topology change. There are three approaches to initiate the LSP establishment.

Data-driven

In data-driven model, incoming traffic at LER triggers the sending of a label request in the downstream direction. In response to the label request, each LSR along the path generates a label and distributes it in the upstream direction. The label binding is formed and a LSP is then established. LDP is usually used for the signaling.

Control-driven

In control-driven model, a LSP is established before the traffic flow arrives. Label binding and LSP setup can be done either by control information signaling, such as CR-LDP, RSVP-TE, or by command line interface.

Topology-driven

In topology-driven model, the mechanism of path protection or rerouting is implemented at the ingress LER. In case there is a link failure in the LSP, it should be detected and reported to the ingress LER. The ingress LER then reroutes the traffic to an alternative LSP that could be pre-established by control information, or setup on demand by LDP.

2.2.7 Traffic Engineering

Traffic engineering is a process to differentiate traffic and distribute it throughout the network, so as to enhance the utilization of network bandwidth. Traffic engineering enables service providers to offer the best service to their users in terms of throughput and delay. Conventional IGP usually selects the shortest path between two network devices. With traffic engineering, the path selected is not necessarily the shortest one, but the one with acceptable bandwidth. Packet flows may traverse completely different paths even though their originating node and the destination node are the same. This way, all network segments can be used efficiently and differentiated services can be provided as well [21].

In MPLS, LSPs are created independently, specifying different paths that are based on user-defined policies. So traffic engineering can be provided using explicitly routed

paths. RSVP-TE and CR-LDP are two possible approaches to supply dynamic traffic engineering and QoS in MPLS.

As shown in Figure 2-8, in a conventional IP network, the packets from router A and B to C would go along path 1, which is the shortest one selected by IGP based on the IP addresses of the source and destination nodes, even if there congestion on the path. In a MPLS network, by assigning different FEC and labels to different traffic flows, multiple paths can be setup between two network nodes, each for a particular group of traffic flows. So it is possible to route the packets from router B to C through path 2 to avoid congestion.

Figure 2-8 MPLS Traffic Engineering

2.3 QoS

QoS (Quality of Service) is the capability of a network to deliver better service to selected traffic with dedicated bandwidth, controlled jitter and latency, or improved loss characteristics. This can be done either by raising the priority of a traffic flow or by limiting the priority of another traffic flows. There are three aspects involved in QoS implementation [5, 52].

2.3.1 Traffic Classification

Traffic Identification

In order to provide better service to preferred traffic, it first must be first identified.

Traffic Marking

The identified traffic may be marked in order to receive better service throughout the network. If the identified traffic is not marked, then the classification is on a per-hop basis, meaning it only receive preferred service in the device it is on, and its privilege is not passed along to the next router.

2.3.2 Network Element QoS

A network element can provide QoS through congestion management, queue management, link efficiency, shaping and policing.

Congestion Management

Due to the bursty nature of the traffic, congestion could occur in a network element when the amount of traffic exceeds its processing capacity or link speed. The network element can handle this situation by applying different queue schemes to buffer the packets. Possible queue schemes are *priority queueing* (PQ), *customer queueing* (CQ), *weighted fair queueing* (WFQ), and *class-based weighted fair queueing* (CBWFQ).

Queue Management

Because the length of the queues is limited, if the queues overflow, packet would be dropped. Queue management addresses the problem of how to prevent the queues from overflowing, and the criteria of dropping packets when overflow happens.

Link Efficiency

One way to improve the link efficiency is through the segmentation of large packets and then interleaving the fragments with other packets, so that the packets behind the large packet will not be blocked for too long.

Another efficiency improvement is the technique to minimize the overhead bit in the packets to a more manageable size.

Traffic shaping and policing

Shaping is a method to adjust the original bandwidth of a traffic flow to a low level to avoid overflow on the links with low bandwidth. To achieve this goal, excessive traffic is buffered for later transmission to maintain the configured rate.

Policing is similar to shaping, but the excessive traffic is not buffered.

2.3.3 End-to-End QoS

Three end-to-end QoS levels can be provided across a network.

Best-effort Service

This level of service provides basic connectivity with no guarantees. Traffic flows are processed in a first-come-first-serve manner, without differentiation between flows. Regular IP traffic is a typical example that receives this type of service.

Differentiated Service

Also referred to as soft QoS. Some traffic flows, such as streaming video, voice over IP, can receive better service than others with faster handling, more bandwidth and lower loss rate. This is a statistical measurement, not a real time guarantee.

Guaranteed Service

Also referred to as hard QoS. Network resources are reserved for specific traffic [5, 52]. A special protocol, such as RSVP, is needed to reserve the required bandwidth along the path of the traffic.

Chapter 3 Martini VPN

This chapter gives a detailed explanation of Martini VPN. The first part of this chapter introduces the motivation for deploying Martini VPN, and describes the network topology and the Martini drafts. The second part describes the data transport method and the last part covers the encapsulation method.

3.1 Overview

3.1.1 Motivation for Martini VPN

The Martini VPN solution addresses the transport of Layer 2 traffic in emulated, point-to-point connections through an MPLS network backbone. This type of connection is also known as pseudo-wire emulation [18]. This solution is named after Luca Martini who initially proposed it. We use the term "Martini VPN" in the rest of the thesis.

The motivation behind Martini VPN is listed as follows [30].

- To augment the existing Layer 2 service of customer's "legacy" network such as Frame Relay, ATM, or some non-IP service such as SNA, IPX. Thus the existing investment can be protected while building an MPLS packet core.
- To trunk customer's IGP traffic, such as Ethernet across provider's MPLS backbone.

- To enable remote/geographically separate peering points, and to extend the broadcast domain for remote peering

3.1.2 Martini Drafts

The MPLS VPN solution is specified by a set of drafts published by the Internet Engineering Task Force (IETF). The Martini Drafts consist of a set of separate documents. At the time when this thesis was written, the latest available version of the draft documents were as follows.

1. Encapsulation Methods for Transport of Layer 2 Frames Over IP and MPLS Networks [32, 34].

This document specifies the methods for emulated VC encapsulation for a number of Layer 2 protocols, it also specifies the way in which the demultiplexer field is added to the emulated VC encapsulation when an MPLS label is used as the demultiplexer field.

2. Transport of Layer 2 Frames Over MPLS [35]

This document specifies the necessary VC label distribution procedures for the encapsulation methods described in the first document.

3.1.3 Network Topology

Figure 3-1 shows a typical network topology for Martini VPN and the associated network components.

Figure 3-1 Network Topology

CEs are the Customer Edge devices. They are typically owned and operated by the end user. They are also referred to as Customer Premise Equipment or CPE. In Layer 2 VPN context, a CE can be either a Layer 2 device, such as a switch, or a Layer 3 device, such as a router. The routing protocol(s) running on Layer 3 CE devices are of customer's choice. Layer 3 CE devices can have private IP address space. They have no knowledge of the existence of MPLS backbone or VPN.

PEs are the Provider Edge routers. A PE router is directly connected to the CE devices. CEs access provider's MPLS backbone through PE. A PE may be connected to multiple CE devices supporting different customers. PE performs all of the functionality associated with establishing, maintaining and operating Martini VPN.

P routers are Provider routers that reside in provider's MPLS backbone. They are LSRs that switch MPLS packet over the established LSPs. They too are unaware of the existence of VPN.

Martini VPN is a *provider-provisioned peer model* VPN solution. The service provider is responsible for all VPN setup. No extra configuration work is needed to be done at the customer site. The service provider's network provides an emulated, point-to-point connection between CE devices. The Service provider is not involved in the IP routing of customer's traffic. CE devices have no knowledge of being connected across public network. In provider's network, PE devices perform all VPN specific functionality. The MPLS backbone has no knowledge of VPN traffic.

3.2 Data Transport Method

Two level of connections need to be established for Martini VPN operation. A pair of LSP would be used as a bidirectional tunnel between PE routers to guarantee the packet delivery across the MPLS backbone. Second, a virtual circuit in the tunnel needs to be established for a particular customer connection, so the PE router can deliver the received packets from the tunnel to the correct CE device.

3.2.1 LSP Tunnel Connection

MPLS LSPs, or tunnels, between PE routers can be established by LDP or RSVP. The routers in the MPLS network advertise the labels they assign to the routes in their forwarding table via label mapping messages. The LSPs can be established when all routers involved along the path fill-up their forwarding tables with the information carried by the messages. The routers store the received labels along with the interface information in their forwarding table and use them when incoming data packets are to be forwarded. When a packet comes into the MPLS network, a label will be pre-pended to

the Layer 2 header according to its destination address. The packet is then forwarded further in the MPLS domain. The intermediate routers use the label in the packet as the index to its forwarding table entry [14, 17, 39, 41, 42].

3.2.2 Virtual Circuit Connection

The VC is established by distributing the VC label through LDP targeted session between PE routers at both ends of the tunnel.

Purpose of VC Label

The purpose of VC label is to provide necessary information for the egress PE router to forward the packet received from the tunnel. The VC label is mapped to a particular Layer 2 circuit. For ATM AAL5 PDU, the VC label generally corresponds to VPI/VCI value of a particular ATM VC at the egress PE router. If the packet concerned is a Frame Relay PDU, the VC label is associated with the outgoing interface and the DLCI value. If the payload is an Ethernet frame, the VC label is then mapped to the outgoing interface, and perhaps the VLAN identifier.

VC label Distribution

In addition to the advertisement of regular MPLS labels between adjacent LSR for LSP setup, LDP Label Mapping Message is extended to support distributing VC labels and VC ID between PE routers for establishing a VC connection [34].

Downstream unsolicited mode is used for VC label advertisement, and *liberal label retention mode* is used for VC label retention. An LDP directed session is required

between peer PE routers. The LDP session can have a route different from the path along which the Layer 2 PDUs are transported. Liberal label retention mode should be used when distributing VC label [14].

TLV

LDP uses a Type-Length-Value (TLV) encoding scheme to encode the information carried in LDP message [14]. Figure 3-2 shows the generic TLV format.

Figure 3-2 Format of TLV

U bit: Unknown TLV bit.

F bit: Forward unknown TLV bit

Type: Indicates the type of encoded information in the Value field.

Length: Specifies the length of Value field in bytes.

Value: Carries the actual value of the particular information.

Label Mapping Message

Label Mapping Message consists of several TLVs, namely Message ID TLV, FEC TLV, Label TLV and some optional parameters [14]. Figure 3-3 shows the general format of Label Mapping Message.

Figure 3-3 Label Mapping Message Format

Message ID TLV: Indicates the type of the message.

FEC TLV: Encodes the Forward Equivalent Class (FEC) information.

Label TLV: Contains the label information.

VC ID

A unique VC identifier is assigned to each bi-directional VC connection of a particular VC type. The VC ID is distributed together with VC Label in Label Mapping Messages between PE routers. The PE routers maintain the mapping between VC ID and outgoing interface to CE device.

VC ID and VC type information is encoded in FEC Element of FEC TLV when being distributed by Label Mapping Message. Figure 3-4 show the format of VC FEC Element.

Figure 3-4 VC FEC Element

- VC type - Contains a value that represents the type of VC.
- Control Word bit (C) - Indicates control word preference.
- Length - Indicates the length of VC ID and Interface Parameters fields in bytes.
- Group ID - Represents a group of VCs
- VC ID - A 32-bit connection ID that together with the VC type, which identifies a particular VC.
- Interface Parameters - A variable length field used to provide interface specific parameters, such as interface MTU.

VC Label

A local significant 20-bit VC label is assigned to each direction of VC connection. VC label has same format as regular MPLS label. VC labels and VC ID are distributed in Label Mapping Messages between PE routers via LDP directed session. VC label is encoded in the label TLV that is contained by the Label Mapping Message. Figure 3-5 shows the format of the Label field of a label TLV.

The VC label can be assigned manually by the operator or automatically by the PE routers.

The PE routers store the received VC labels in their VPN forwarding tables. When a VPN packet comes to a PE router, a VC label will be pre-pended to the packet before the packet is further encapsulated and sent in the tunnel.

PE routers maintain the mapping between the VC label and VC ID, since VC ID is not carried with the packet.

Interface Parameters

This field may contain interface specific parameters of a particular Layer 2 interface, such as interface MTU, maximum number of concatenated ATM cells etc. Its field structure is shown in figure 3-5.

Figure 3-5 Interface Parameter

Parameter ID defines the type of interface parameter as described in [35].

C Bit

This bit is used to flag the preference of use of control word.

VC type

This field I used to indicate the type of emulated Layer 2 protocols, such as Ethernet, Frame Relay, and ATM etc. The value definition is described in [35].

3.2.3 Control Word Negotiation

As described in 3.3.3, the Layer 2 packet may be encapsulated with a control word, depending on the type of Layer 2 protocol.

For those Layer 2 protocols, such as Frame Relay, for which the control word is required, the C bit must be set to 1 in the label mapping messages. A label mapping

message with $C = 0$ would trigger the label release. A label release message with status code “Illegal C-bit” will be sent.

For those Layer 2 protocols, such as Ethernet, for which the control word is not mandatory, the use of control word can be negotiated via signaling of label mapping messages between peer PE routers. The major factors for the decision of whether to use a control word are the capability and preference to use it for the PE pair. Figure 3-6 shows the process of negotiation by means of a label mapping message [35].

Figure 3-6 Control Word Negotiation

3.3 General Encapsulation Method

As discussed in the previous section, LDP is extended to setup emulated virtual circuits that carry Layer 2 data packet across the network. Multiple virtual circuits can be

aggregated in a single tunnel. Encapsulation of Layer 2 PDUs is required for this purpose. The Layer 2 PDUs are encapsulated at the ingress PE router, transported through the tunnel, decapsulated at the egress PE router and further transmitted to the CE device.

In general, three layers of encapsulation are needed, which are Tunnel Header, Demultiplexer Field and Emulated VC Encapsulation. But for some type of Layer 2 PDUs, such as Ethernet, the first two layers of encapsulation are enough. Figure 3-7 shows the format of each layer. They are discussed in details as follows.

Figure 3-7 Martini VPN Encapsulation

3.3.1 Tunnel Header

This 32-bit header contains the information necessary to transport the PDU across the MPLS LSP tunnel, or in other words, from the ingress PE router to the egress PE router. As shown in Figure 3-7, this layer has the same format for regular MPLS label encapsulation. The Tunnel Label is the outer label of a Martini VPN label stack. It is assigned by the LSRs along the LSP tunnel and will be swapped at each intermediate P routers, popped at the penultimate router. The information contained in the header is as follows:

- Tunnel Label: This 20-bit field contains the actual value of an assigned label.
- EXP: This was for experimental use, but it carries the CoS information in a MPLS network that supports Differentiated Services.
- S: One bit representing if it is the bottom of the MPLS label stack. The stack bit for this header should be set to 0 since tunnel label would not be the bottom of the label stack.
- TTL: This Time-to-Live field is set to 255 by default or configurable by operator according to RFC 3443 [13].

3.3.2 Demultiplexer Field

Information in this 32-bit space is used to distinguish individual emulated virtual circuit within a single tunnel. The egress PE router uses this information to further forward the packets from the tunnel to their correct CE devices. It is achieved by MPLS label encapsulation as well. Both ingress PE and egress PE router should have the knowledge of the label value for a particular emulated VC. This label is also referred to as VC Label. In the label stack, this layer is below the Tunnel Header, and above Emulated VC Encapsulation if there is one. The demultiplexer field has the same format as that of a Tunnel Header. The information contained in this header is as follows:

- VC Label: This 20-bit field contains the actual value of a VC label.

- EXP: This was for experimental use, but it carries the QoS information in a MPLS network that supports Differentiated Services.
- S: One bit representing if it is the bottom of the MPLS label stack. The stack bit for this header should be set to 1 to indicate the bottom of the label stack.
- TTL: This Time-to-Live field should always be set to 2 initially at ingress PE router, meaning that the label should only be inspected at only two routers (ingress PE and the egress PE).

3.3.3 Emulated VC Encapsulation

Three requirements may need to be satisfied when transporting Layer 2 packets over MPLS backbone:

- For some type of Layer 2 protocols, it is not necessary to transport the entire Layer 2 header with the payload across the network. The Layer 2 header can be stripped at one end of the tunnel (ingress PE router) and reproduced at the other end (egress PE router). But some of the flags in Layer 2 are still necessary for reproducing the Layer 2 header at the egress router [32, 34].
- If the medium on which the encapsulated packets are transported has a requirement for minimum size of the packet, the packets with the size smaller than the requirement may need to be padded. So there should be a way to tell the

length of the real packet after the packet has been padded and encapsulated [32, 34].

- The sequence of the packet flow may need to be preserved [32, 34].

Emulated VC Encapsulation is used to carry some control parameters from the ingress PE router to the egress PE router so as to meet these three requirements. This layer of encapsulation is also referred as Control Word. The fields in the Control Word carry the following meanings:

- Rsvd: The 4-bit Rsvd field is reserved for future use.
- Flags: The 4-bit Flags field is used to carry the flags in Layer 2 header.
- Length: The 6-bit Length field is used to indicate the length of original Layer 2 packet.
- Sequence Number: The 16-bit Sequence number field is used to support packet sequencing in a VC.

For some Layer 2 protocols, such as Frame Relay and ATM AAL5 CPCS-SDU, the control word is required, and for others, like ATM Cell Mode, Ethernet VLAN, Ethernet, HDLC and PPP, the control word is optional. The information in the control word varies for different Layer 2 protocols.

The egress router must be aware of whether the ingress router will send a control word over a specific virtual circuit. This may be achieved by configuration of the routers or by signaling, for example as defined in [33, 35].

3.3.4 Packet Forwarding Procedure

A Martini VPN connection is setup through several steps of router configuration and signaling.

The first step is to setup a LSP tunnel from the ingress PE router to the egress PE router, as described in 3.2.1.

The second step is to setup a VC connection in a pre-established LSP tunnel, as described in 3.2.2.

The third step, if applicable, is to signal the usage of a Control Word for certain type of Layer 2 protocols, as described in 3.2.3

Once the three steps are successfully performed, VPN traffic can be forwarded accordingly.

Figure 3-8 depicts the generic packet forwarding scenarios. As shown in the diagram, when a Layer 2 packet comes to the ingress PE router, for some type of Layer 2 protocols such as Frame Relay and ATM AAL5 CPCS-SDU, the Layer 2 header is stripped. The flags in the Layer 2 header are encapsulated in the Control Word that is prepended to the Layer 2 payload. Then the packet is further encapsulated with VC label and tunnel label before is sent to MPLS backbone. But for other types of Layer 2 protocols such as Ethernet, the Layer 2 header is encapsulated with payload, and the Control Word is optional.

Figure 3-8 Packet Forwarding

In MPLS backbone, the P routers perform label-swapping operation on packet's label stack. Only the tunnel label is involved since it is the outer label in the stack. The P routers strip the tunnel label and map it to an NHLFE. Using the information in NHLFE, it can get an outgoing label, encode the new label stack into the packet and then forward it. The last P router along the tunnel, which is the penultimate hop, may pop the tunnel label and forward the packet to the egress PE router.

At the egress PE router, the VC label is popped and inspected against the internal mapping to find the outgoing interface. If the peer PE router uses a control word, it is popped as well. The Egress gets the knowledge of control word preference of peer by signaling during the VPN setup stage. The flags contained in the control word are used to reproduce the Layer 2 header. The packet is then forwarded to the CE device.

Chapter 4 Design and Implementation

This chapter gives a detailed explanation of the design and implementation of Martini VPN in the EION Open IP environment. The first part introduces the software framework of the EION Open IP suite. The second part explains the requirement analysis and high-level design. The third part describes the control plane implementation and the last part covers the data plane implementation.

4.1 Software Framework

The software design of the Martini VPN is based on the software architecture of EION Open IP Environment. Layer 2 Martini VPN is added into the existing software framework [1, 2, 43].

EION Open IP Environment is a real-time software suite that enhances the functionality of network devices to provide high performance interoperability. It provides IP communications functional blocks that reside between the operating system and the application/service layer of various Internet-enabled devices. It is designed to be portable across multiple platforms, such as Unix, Linux, VxWorks. Figure 4-1 depicts the software architecture of the Open IP Environment.

Figure 4-1 Existing Software Architecture

The entire software structure consists of four subsystems, namely Common Control Plane, Common Data Plane, Common Management Plane and Common System Plane.

Common Control Plane

Common Control Plane is used to perform the control information signaling between various protocol modules, routing table managers and routing policy server. It uses separate APIs to interface with other subsystems.

Common Forwarding Plane

The Common Forwarding Plan is responsible for forwarding received packets correctly and efficiently to the egress interfaces according to the type of packets and internal policy, which may involve traffic identification, routing table lookup, packet encapsulation etc. Its function is mainly performed by the Forwarding Engine module.

Forwarding Engine maintains the routing tables for different type of traffic. Forwarding Engine supports unicast and multicast routing.

The following diagram shows the internal modules in MPLS Forward Engine and their functions in the packet forwarding process.

Figure 4-2 MPLS Forward Engine

The box with a thick outline represents a control object. "VpnPktHandler" is a newly added object that handles VPN specific functionality.

Common Management Plane

Common Management Plane subsystem consists of a set of external interfaces to manage the modules in Open IP Environment. It provides open standard APIs for command line interface, network configuration and management.

Common System Plane

Common System Plane subsystem is built on top of the ACE object-oriented framework [44-47]. The ACE framework provides a rich set of reusable components that perform common communication software tasks across a range of OS platforms, such as Unix, Linux, VxWorks, and Windows etc. It helps developers to develop high-performance and real-time communication services and applications by hiding the details and complexity of inter-process communication, event demultiplexing, explicit dynamic linking, and concurrency for a wide range of operating systems. The application level can just use the standard APIs it provides, thus the development effort at application level can be alleviated [45].

From Layer 2 Martini VPN's perspective, LDP module would be responsible for setup, delete and release LSP tunnel and VC connectivity. Forwarding Engine would do the packet encapsulation/decapsulation and packet routing. CLI commands would be used to configure a VPN and manage the status of VPN operation. Therefore, major software modification would occur in these three modules to support Layer 2 Martini VPN functionality.

4.2 Requirement Analysis and High Level Design

4.2.1 General Requirements

As described in chapter 3, Martini VPN requires implementing two major functionalities at PE routers. Since P routers in MPLS backbone have no knowledge of VPN traffic, no additional functionality needs to be added to them for this purpose.

First, a PE router should be able to distribute information to its peer, which is necessary to setup a connection at VC level. The information contains VC label, VCID, group ID and control word if necessary. LDP is selected to exchange this information in the Martini draft [35]. LDP needs to be extended to carry VC Label Request Message and VC Label Mapping Message. Unlike regular LDP messages that are sent between directly connected LSRs, VC Label Request Message and VC Label Mapping Message are sent between PE routers that are usually not directly connected. LDP needs to be extended to send messages in directed sessions between PE routers. This message signaling functionality was implemented and integrated in the Common Control Plane of EION Open IP Environment, as described in section 4.1.

Second, an ingress PE router should be able to identify the VPN interface and its Layer 2 PDU type, such as Ethernet, Frame Relay, ATM etc. The ingress PE router should encapsulate the incoming Layer 2 VPN packet with tunnel label, VC label and control word if necessary, and forward it to a pre-established LSP tunnel connecting both PE routers. Egress PE router should be able to identify the VPN packet coming from the tunnel, demultiplex the packet, and forward it to the correct outgoing interface based on

the VC label carried with the packet. This part of functionality is implemented and integrated in the Common Forwarding Plane of EION Open IP Environment as described in section 4.1.

In addition to the functionality mentioned above, which is specified in the Martini VPN drafts, CLI commands also needed to be implemented, in order to configure, operate and maintain the VPN connections. They were added to the Common Management Plane of EION Open IP Environment, as described in 4.1.

Figure 4-3 visually explains the software changes in each subsystem. The boxes with bold outline are the modules to be extended or created.

Figure 4-3 Software Architectural Changes

4.2.2 Concurrency

Multi-processing

EION Open IP Environment is based on real-time, concurrent platforms, which support multitasking computing. Multiple processes can exist and be scheduled by the processor. Each subsystem in EION Open IP Environment has one or more concurrent processes, on which different modules run. Each process has its own thread of control and data. All processes are derived from the ACE process model [44].

All processes are created sequentially at initialization time according to their priority and dependency.

Inter-process Communication

In EION Open IP Environment, message passing is the main form of communication between processes. Messages are sent and received in a mailbox style. No synchronization is required between sender and receiver. Each process is allocated with a message buffer of a fixed length. All the messages to a particular process are queued in its message buffer. The process dequeues and processes the messages when it gets its turn to execute.

4.3 CLI Commands

As described in 4.1, Common Management Plane of EION Open IP Environment provides open standard APIs for Command Line Interface. CLI commands are developed to enable the network operator to configure, maintain and delete the VPN connection. Three sets of CLI commands were developed in the thesis. The first set is used to turn on and off the Martini VPN functionality, the second set is used to configure the individual

VC connections and the third set is used to show the detailed data of VPN connections. All the CLI commands for Martini VPN developed in the thesis are listed as follows.

```
route mpls l2vpn
```

This command enables the Martini VPN functionality.

```
no route mpls l2vpn
```

This command disables the Martini VPN functionality

```
mpls l2vpn session <ipa>
```

This command triggers the setup of directed TCP session between for LDP messages. <ipa> is the IP address of peer PE router.

```
mpls l2vpn route <vcid> from <cid> to <ipa> [controlword  
<cw>] [vclabel <vclabel>] [ttl <ttl>]
```

This command configures the individual VC connection. <vcid> is the VC ID assigned to the VC connection. <cid> is the circuit ID of VPN interface. <ipa> is the IP address of the other end of tunnel. <cw> is the control word preference. <vclabel> is the VC label value if the operator wants to assign it manually. <ttl> is the Time-to-Live value in demultiplexing field of encapsulation.

```
no mpls l2vpn route <vcid>
```

This command deletes a VC connection.

```
show mpls l2vpn [vcid <vcid>]
```

This command displays detailed data of VC connection. If VC ID is input, only the data of the specific VC is displayed, otherwise the data of all VCs is displayed.

4.4 Control Plane Implementation

4.4.1 Software Structure

Figure 4-4 shows the class diagram of the control plane. The boxes with gray filling are the new classes created for Martini VPN purposes. The boxes with bold outline are active classes, which have their thread of control at run time. The boxes with a black upper right corner are the existing classes that needed internal changes to accommodate the Martini VPN functionality.

Figure 4-4 Control Plane Structure

VpnMgr maintains the VpnConfigureTbl that contains the configuration data for VC connections. VpnMgr takes CLI commands to fill out the VpnConfigureTbl. VpnMgr interacts with MplsLdp to receive and send VC related LDP messages. Four more LDP messages classes, which are VcLabelRequest, VcLabelMapping, VcLabelRelease,

VcLabelWithdraw, are added into the collection of MplsLdp. VpnMgr updates VC tables in MPLS Forwarding Engine by send messages to MplsPacketProcessor.

4.4.2 Label Space Management

Label Allocation

As described in 3.3, in order to transport a Martini VPN packet over an MPLS network to a customer site, the packet must be processed with at least two layers of encapsulation, if the control word encapsulation is not required. The outer layer is the tunnel header, which contains the tunnel label. This layer of encapsulation ensures the delivery of the packet from ingress PE to egress PE through the LSP tunnel. The label is checked and replaced with a new one at each LSPs along the tunnel in the MPLS backbone. Below the tunnel header is the demultiplexer layer. This layer of encapsulation ensures the delivery of the packet at a PE router to a correct outgoing interface, which is connected to the CE device. This label is only checked at the egress PE.

Both labels are generated and distributed by the egress PE router [34, 42]. After the outer label is stripped, the egress PE router needs to know what kind the next label is, so it can process it properly. Because the second layer label could be another tunnel label, a VC label for Martini VPN, or a VC label for BGP VPN etc.

In this research, the entire label space of egress PE router is divided into several ranges, each for regular LSP, Martini VPN VC, BGP VPN VC respectively. According to the Martini draft [34], the label space range for Martini VPN is further divided into two parts, one for manual allocation and another one for automatic allocation.

Label Generation

Label resources at the egress PE router are managed by the Label Manager module. It is responsible for label allocation, label release, label reuse for all applications. The algorithms for label allocation and release employed by the Label Manager are listed as follows.

Algorithm: Label Allocation

```

1:  label allocation function call by other module; get the passed parameter of label type
2:  switch (label type)
3:  case LSP:
4:    if released label pool for LSP is not empty, then
5:      get a label from the pool
6:      return label
7:    else
8:      if lastLspLabel + 1 is within the range then
9:        lastLspLabel ++
10:       return lastLspLabel
11:     else
12:       return -1
13:     end if
14:   end if
15: case Martini VC:
16:   if released label pool for Martini VC is not empty, then
17:     get a label from the pool
18:     return label
19:   else
20:     if lastMartiniVcLabel + 1 is within the range then
21:       lastMartiniVcLabel ++
22:       return lastMartiniVcLabel
23:     else
24:       return -1
25:     end if
26:   end if
27: default: return -1

```

Algorithm: Label Release

```

1:  label release function call by other module; get the passed label value

```

```

2: switch (label value)
3:   case LSP label:
4:     insert the label into released label pool for LSP, then
5:     return true
6:   case Martini VC label:
7:     insert the label into released label pool for Martini VC, then
8:     return true
9:   default: return false

```

4.4.3 Configuration Table

MplsVpnMgr maintains the configuration table, which contains the information about VC connections. Part of the information in the table is entered through local CLI commands, while other parts are extracted from the VC label mapping messages from the peer PE. To minimize the number of messages, MplsVpnMgr only sends a message to update the VC tables in MplsVpnPkthandler when the information of an entry is complete.

Each entry in the configuration table corresponds to a VC connection. The configuration table is implemented as AVL tree with vcid playing the role of search key for each entry. The following is the data structure of a table entry.

```

struct VpnContrigureEntry{
    int vcid;                // VC ID
    struct VpnConfigureData data; // configuration data
};

struct VpnConfigureData{
    int flag;                // flag to indicate data existence
    int cid;                 // VPN interface circuit ID
    int vcTypeLocal;         // Layer 2 type of local interface
    int mtuLocal;            // MTU setting at local PE
    int vcLabelIn;           // incoming VC label

```

```

    bool cwLocal;           // local control word preference
    int vcTypeRemote;       // Layer 2 type of remote interface
    int ipAddrRemote;       // the IP address of remote PE
    int mtuRemote           // MTU setting at remote PE
    int vcLabelOut;         // outgoing VC label
    bool cwRemote;          // remote control word preference
};

```

4.4.4 LSP Tunnel Setup

The process of LSP tunnel establishment is beyond the scope of this thesis. A LSP tunnel is assumed to be pre-established between PE routers. This could be done through CLI commands.

4.4.5 VC Connection Setup

As described in 4.4.3, VpnMgr maintains a VPN configuration table. Each entry of the table is associated with a VC connection. Part of the information in an entry is extracted from the CLI commands entered by the operator, such as VC ID, incoming VC label, VPN interface circuit ID and local control word preference. Another part of the information is extracted from the VC Label Mapping Messages sent by the peer PE router, such as outgoing VC label and remote control word preference. Once VpnMgr receives the complete information, it sends a message to the VPN part in Forwarding Engine to update the mapping between VC label and VPN interface in the VC tables. A VC connection is established when the mapping is done at both PE routers. Figure 4-5 shows the sequence diagram of the procedure for VC connection setup.

Figure 4-5 VC Connection Setup

4.4.6 Control Word Negotiation

Control word negotiation procedure is implemented in this research, which is described in 3.2.3, although it is optional for Ethernet. The following is the algorithms for the control word negotiation at VC configuration and Label Mapping Message processing.

Algorithm: VC configuration with control word

- 1: receive CLI command
- 2: store data in configuration table
- 3: if received Label Mapping Message from peer
- 4: if remote peer does not prefer control word, then
- 5: send Label Mapping Message with C = 0
- 6: send message to Forwarding Engine to update VC tables
- 7: return
- 8: else

```

9:      if control word is preferred locally, then
10:         send Label Mapping Message with C = 1
11:         send message to Forwarding Engine to update VC tables
12:         return
13:      else
14:         send Label Mapping Message with C = 0
15:         return
16:      end if
17: end if
18: else
19:      if control word is preferred locally, then
20:         send Label Mapping Message with C = 1
21:         return
22:      else
23:         send Label Mapping Message with C = 1
24:         return
25:      end if
26: end if

```

Algorithm: Processing Label Mapping Message

```

1:  receive Label Mapping Message
2:  store the data in configuration table
3:  if status code = Wrong C-bit, then
4:     print error message
5:     return
6:  else
7:     if CLI configuration command entered, then
8:        if control word preference is same for both sides (remote C = local C), then
9:           send message to Forwarding Engine to update VC tables
10:          return
11:        else
12:           if remote C = 1, then
13:              return
14:           else
15:              send Label Mapping Message with status code = Wrong C-bit
16:              end Label Mapping Message with C = 0
17:           end if
18:        end if
19:     else
20:        return
21:     end if
22: end if

```

4.5 Forwarding Plane Implementation

4.5.1 Ethernet Emulation

As described in 3.1, the goal of Martini VPN is to emulate point-to-point Layer 2 connections through MPLS network backbone between CE devices. This type of connection is also known as pseudo-wire emulation. This research focuses on Ethernet/IEEE 802.3 emulation.

Ethernet

Ethernet is a LAN technology that is used to connect multiple network devices in a relatively small geographic area to form a high-speed data network. Ethernet functions at the lowest two layers of the OSI reference model. As show in Figure 4-6, IEEE 802.3 defines the physical layer and MAC layer, while LLC layer is specified separately in IEEE 802.2.

OSI Reference Model

Figure 4-6 Ethernet in OSI Model

IEEE 802.3 defines three types of physical layers, namely 10 Mbps Ethernet, 100 Mbps Fast Ethernet and 1000 Mbps Gigabit Ethernet. The Ethernet connection between network nodes in this research is Fast Ethernet.

IEEE 802.3 also specifies CSMA/CD as the mechanism for media access control. Each station is identified by a unique 48-bit MAC address. Each station listens for traffic on the wire to determine the gaps data transmissions. A station may transmit data any time when it finds the medium free. The transmitting station stops transmitting as soon as it has detected the collision and then must wait a certain length of time before attempting to retransmit the frame [3].

Each NIC card has a unique 6-byte MAC address, which is divided into a 3-byte vendor ID and a 3-byte vendor-defined field. Ethernet manufacturers are assigned a unique vendor ID, and are then responsible for insuring that all of their devices have unique addresses in the last 3 bytes. An all-ones address is interpreted to mean broadcast. All stations listen to the data frames on the medium, but only accept the frames that have the same destination MAC address as its own.

Figure 4-7 shows the basic data frame format of Ethernet. The 7-byte PRE indicates a coming frame. The 1-byte SOF indicates next six bytes are destination MAC address. The 6-byte DA and SA are the MAC addresses for receiving and sending station respectively. The 4-byte Length/Type field indicates either the number of data bytes that are contained in the data field, or the frame type ID if the frame is assembled using an optional format. The maximum length of data field is 1500 bytes. If the user data is less

than 46 bytes, it has to be padded. The 4-byte FCS contains the cyclic redundancy check (CRC) value.

Figure 4-7 Ethernet Data Frame Format

ARP

The *address resolution protocol* (ARP) is a protocol to map IP addresses to the MAC addresses. The protocol operates below the network layer as a part of the interface between the OSI network and OSI link layer. A station on Ethernet LAN that needs to send data to another station, it first broadcasts ARP request containing the IP address of the other station, and only the station with the correct IP address accepts the message and sends its MAC address in the reply message [38].

Implementation Issues

The goal of Martini VPN is to utilize MPLS network to emulate a “direct connection” between CE devices. So customer sites have the knowledge that they are

connected through a public network. The customer sites can have their private IP address space, and the provider carries out the routing of customer traffic.

To achieve this goal, the provider's network should deliver whatever it receives from one customer site to the other. The entire Layer 2 packet is encapsulated and sent through the LSP tunnel, except the preamble. CE devices do not have to be Layer 3 devices. The communication between stations on Ethernet starts with an ARP message sequence. As shown in Figure 4-8, ARP request uses a broadcast address as its destination MAC address. It can be accepted, encapsulated and sent through the tunnel by PE1. The ARP request is recovered to its original form at PE2 and sent to the CE device. The ARP reply uses MA1 as destination MAC address, it may not be accepted by PE2, because its MAC address is MA3. So the VPN ports of PE need to be set in promiscuous mode in order to receive packets with any destination MAC address.

Figure 4-8 Communication between CE Devices

A problem may be caused by the promiscuous mode – the packet sent by a PE to a CE may be received by the PE itself, and then send it over to a peer PE. The packet would be bouncing back and forth between PE routers, causing a degradation of network

performance in the MPLS network. The solution is to implement a mechanism in PE to recognize the packets it sends to CE based on its source MAC address.

4.5.2 Software Structure

Figure 4-9 shows the software structure of the MPLS Forwarding Engine. The boxes with gray filling are the new classes created for Martini VPN purposes. The boxes with bold outline are active classes, which have their own thread of control at run time. The boxes with a black upper right corner are the existing classes that needed internal changes to accommodate Martini VPN functionality.

Figure 4-9 MPLS Forwarding Structure

VpnPktHdlr maintains two routing tables, namely **IngressVcTbl** and **EgressVcTbl**, for VC connections. **VpnPktHdlr** takes the messages for **VpnMgr** in the Common

Management Plane to add, update and delete the entries in these two tables. VpnPktHdlr uses Encoder to perform VC layer encapsulation.

PacketProcessor receives various packets from all incoming interfaces. It passes data packets along to MplsComForwarder. MplsComForwarder further passes VPN packets to VpnPktHdlr for processing. At the ingress PE router, VpnPktHdlr does VC layer encapsulation for a VPN packet; MplsComForwarder then encapsulates the packet again with tunnel header and it sends it into the LSP tunnel. At the egress PE router, VpnPktHdlr decapsulates the VPN packet, find the mapping between its VC label and the outgoing interface, and sends the packet to the respective CE device.

4.5.3 VC Tables

Ingress VC Table

At the ingress PE router, the VPN packets from CE device are passed over to VpnPktHdlr for processing. VpnPktHdlr uses IngressVcTbl to find the mapping between the incoming interface and the outgoing VC label, and does VC layer encapsulation. Ingress VC table is implemented as an AVL tree with the incoming circuit ID as search key for each entry. Following is the data structure of the ingress VC table.

```

struct IngressVcEntry {
    int inCid;                // incoming circuit ID
    struct IngressVcData;    // ingress VC data
};

struct IngressVcData {
    int vcId;                // VC ID
    int outLabel;           // outgoing VC label
    int peerPeIpAddr         // peer PE IP address
    bool cwLocal;           // local control word preference
    int remoteMtu;          // remote MTU
}

```

```

    set<MacAddr> remoteMac;    // remote site MAC addresses
};

```

Egress VC Table

At the egress PE router, VPN packets from the LSP tunnel are passed over to VpnPktHdlr for processing. VpnPktHdlr uses EgressVcTbl to find the mapping between the incoming VC label and the outgoing interface, and sends the packets to the CE device. Egress VC table is implemented as an AVL tree with incoming VC label as the search key for each entry. Following is the data structure for the ingress VC table.

```

struct EgressVcEntry {
    int inLabel;                // incoming VC label
    struct EgressVcData;        // egress VC data
};

struct EgressVcData {
    int vcid;                   // VC ID
    int outCid;                 // outgoing circuit ID
    int cwRemote                // peer control word preference
};

```

4.5.4 VPN Packet Processing at Ingress PE

VPN packets from CE devices are identified at ingress PE router by its incoming port, encapsulated and transported along the LSP tunnel. Figure 4-10 shows the sequence diagram of VPN packet processing at an ingress PE router.

Figure 4-10 VPN Packet Processing at Ingress PE

4.5.5 VPN Packet Processing at Egress

VPN packets from CE devices are identified by its incoming VC label at the egress PE router, decapsulated and transported to the CE device. Figure 4-11 shows the sequence diagram of VPN packet processing at an egress PE router.

Figure 4-11 VPN Packet Processing at Egress PE

Chapter 5 Performance Evaluation

This chapter presents the performance evaluation of the MPLS Martini VPN implementation. The goal of the evaluation is to assess the network performance impact introduced by Martini VPN implemented in this research, in term of end-to-end response time and throughput. The first section explains the adopted test environment. The second section introduces the software tools employed to measure the performance data. The third section describes the test cases. The analysis of experimental results is presented in the last section.

5.1 Test Environment

As described in chapter 3, Martini VPN functionality is mainly performed at the MPLS edge. The MPLS backbone has no knowledge of VPN traffic; it just forwards it in the same way as it does for regular MPLS packets. Their response time is the same for Martini VPN traffic or regular MPLS traffic. Since the purpose of the test is to evaluate the overhead introduced by Martini VPN, the test is focused on PE routers; the backbone routers between PEs are omitted in the test environment setting. Figure 5-1 shows the test environment. Two SUN ULTRA 10 workstations with Solaris 7.0 operating system are used as PE routers. They are running EION Open IP software suite. Two PCs with Windows 2000 are used as CEs or hosts, which have performance evaluation software running. These nodes are connected with Fast Ethernet.

Figure 5-1 Test Environment

5.2 Tools

Qcheck [4] is used as the software tool to evaluate the end-to-end performance. Qcheck is a free software utility designed to tests network performance and connectivity. Besides testing the network response time and connectivity which Ping command offers, Qcheck can also measure throughput, and streaming performance. Qcheck generates traffic that emulates actual application traffic, giving a realistic measurement of how multi-protocol networks handle different types of applications, including large data transfer and streaming applications. With the installation of Qcheck Performance Endpoints software, the network performance between any two computers in a network can be measured. Figure 5-2 shows the graph interface of Qcheck endpoint software.

Figure 5-2 Qcheck Endpoint

Response Time Test

Qcheck measures the timing of application-level data flows to and from an Endpoint. Following is the transaction at the sockets level: Connect, Loop, Start stopwatch, Send bytes, Receive bytes, Stop stopwatch, End loop, Disconnect. Connects and disconnects are not included in the timing.

Throughput Test

Qcheck does simple, application-level throughput measurements. Throughput is calculated using this formula:

$$throughput = (bytes_sent + bytes_received) / measured_time$$

5.3 Test Cases***Test with VPN off***

Response time and throughput are measured in this test case. IP addresses for each interface of each node are shown in Figure 5-3. LSPs are setup between PE1 and PE2. Qcheck is used to do the measurement. The CEs are connected to PEs with Fast Ethernet. CE has to use publicly unique IP address to be able to communicate with PE.

Figure 5-3 Test Case without VPN

Test with VPN on

Response time and throughput are measured in this test case. IP addresses for each interface of each node are shown in Figure 5-4. LSPs are setup between PE1 and PE2. VPN is turned on at PE1 and PE2. Qcheck is used to do the measurement. The CEs are connected to PEs with Fast Ethernet. CEs can use own private IP address space.

Figure 5-4 Test Case with VPN

5.4 Analysis of Test Results

5.4.1 Data Analysis Methodology

The output data from the measurement experiments has a stochastic nature. Although the input to the PE router is predefined, the unpredictable internal multi-processing, resource contention and queueing at PE have an impact on the output. That means the processing time and waiting time of the packets, arriving at the PE router at different times may be different, which leads to the randomness of the output data.

Usually the mean value, variance and standard deviation are of main interest for output data analysis. To estimate the mean value of a population, we can take random samples from that population and compute its sample mean value. We need a method to know how well the sample statistic estimates the underlying population values. Confidence intervals are normally used for this purpose [15, 23]. A confidence interval provides a range of values, which is likely to contain the expected mean value with a certain level of confidence.

5.4.1.1 Confidence Interval

A confidence interval for the mean is an estimated range of values that is likely to include the mean value. It is calculated from a given set of sample data [15]. The range of a confidence interval is defined by the confidence limits, which are the lower and upper boundaries / values of a confidence interval.

Interval estimates are often desirable because the estimate of the mean varies from sample to sample. Instead of a single estimate for the mean, a confidence interval generates a lower and upper limit for the mean. The interval estimate gives an indication of how much uncertainty there is in our estimate of the true mean. The narrower the interval, the more precise is our estimate.

Confidence intervals are constructed at a *confidence level* selected by the user, e.g. 95%. It means that if the same population is sampled on numerous occasions and interval estimates are made on each occasion, the resulting intervals would bracket the true population parameter in approximately 95% of the cases. A confidence stated at $1 - \alpha$ level can be thought of as the inverse of a significance level α [15, 23]. Although the choice of the confidence level is somewhat arbitrary, in practice 90%, 95%, and 99% intervals are often used, with 95% being the most commonly used.

The confidence interval shouldn't be interpreted as a probability. A 95% confidence interval does not mean that there is a 95% probability that the interval contains the true mean. The interval computed from a given sample either contains the true mean or it does not. Instead, the level of confidence is associated with the method of calculating the interval. The confidence coefficient is simply the proportion of samples of a given size that may be expected to contain the true mean. That is, for a 95% confidence interval, if many samples are collected and the confidence interval computed, in the long run about 95% of these intervals would contain the true mean.

Confidence limits are defined as [23]:

$$\bar{Y} \pm t_{(\alpha/2, N-1)} S / \sqrt{N} \quad (5.1)$$

Where $\bar{Y}$ is the sample mean, S is the sample standard deviation, N is the sample size, α is the desired significance level, and $t_{(\alpha/2, N-1)}$ is the upper critical value of the t distribution with $N - 1$ degrees of freedom. Note that the confidence coefficient is $1 - \alpha$.

From the formula 5.1, it is clear that the width of the interval is controlled by two factors:

- As N increases, the interval gets narrower from the $\sqrt{N}$ term. That is, one way to obtain more precise estimates for the mean is to increase the sample size.
- The larger the sample standard deviation, the larger the confidence interval. This simply means that data with a large standard deviation are going to generate wider intervals than data with a smaller standard deviation.

5.4.1.2 Confidence Interval with Specified Precision

The precision of the confidence interval is given by its width – the difference between the upper and lower limits. Short intervals provide us with very precise information about the location of the population mean, while the wide intervals do not. For a given sample size, increasing the confidence level of an interval decreases the precision and vice versa.

The previous section discusses how to calculate the confidence interval for samples of a given size. Sometimes we need to do the inverse, calculating the sufficient sample size to meet a given level of confidence and precision.

Suppose the user specifies a confidence level $1 - \alpha$ and an error criterion ε , the sample mean $\hat{\theta}$ is used to estimate the population mean θ , which is desired to be within the range $\hat{\theta} \pm \varepsilon$ with a high probability of at least $1 - \alpha$. Assume that an initial sample size N_0 has been observed, which are considered as N_0 independent replications. The N_0 samples will be used to obtain an initial estimate S_0^2 of the population variance σ^2 . To meet the half-length (h.l.) criterion, a sample size N must be chosen such that $N \geq N_0$ which gives:

$$h.l. = \frac{t_{(\alpha/2, N-1)} S_0}{\sqrt{N}} \leq \varepsilon \quad (5.2)$$

where:

$t_{(\alpha/2, N-1)}$: the 100(1 - $\alpha/2$) percentage point of a t-distribution

N : the sample size

S_0 : the variance of initial estimation

ε : the half-length error

By solving for N in Equation 5.2 we obtain the smallest value of N satisfying the condition $N \geq N_0$. N can be determined by:

$$N \geq \left(\frac{t_{(\alpha/2, N-1)} S_0}{\varepsilon} \right)^2 \quad (5.3)$$

Since $t_{\alpha/2, R-1} \geq Z_{\alpha/2}$, an initial estimate for N is given by:

$$N \geq \left(\frac{Z_{\alpha/2} S_0}{\varepsilon} \right)^2 \quad (5.4)$$

Where:

$Z_{\alpha/2}$ is the $100(1-\alpha/2)$ percentage point of the standard normal distribution.

After determining the final N , collect the additional $N - N_0$ number of samples or start over to make N replications. The half-length of the confidence interval is then given by:

$$h.l. = \frac{t_{(\alpha/2, N-1)} S}{\sqrt{N}} \quad (5.5)$$

This value has to be smaller than the precision ε specified by user. However, with the additional $N - N_0$ observations, the variance estimator S^2 may differ from the initial estimate S_0^2 , which would cause the half-length to be greater than user specified precision. If the difference is too large, the procedure should be repeated to determine an even larger sample size N .

5.4.1.3 Procedure of Measurement Data Evaluation

Step 1: Setup the test environment and run the specific test case with Qcheck endpoint software, as described in 5.3.

Step 2: Estimate the initial sample size

According to [15], the recommended initial sample size is at least 5, but 10 or more are recommended depending on the accuracy of the sample data and the final size required. In this research, for each test case, twenty statistically independent samples are collected initially (i.e. $N_0 = 20$).

Step 3: Calculate the mean of the collected sample values.

$$\hat{\theta} = \frac{1}{r} \sum_{i=1}^r Y_r, r = 1, \dots, N_0 \quad (5.6)$$

where:

$\hat{\theta}$ is the sample mean of the entire experiment;

Y_r the value of r th sample

Step 4: Calculate the variance estimate S_0 with N_0 initial samples.

$$S_0^2 = \sum_{r=1}^{N_0} \frac{(Y_r - \hat{\theta})^2}{N_0 - 1} \quad (5.7)$$

Step 5: Estimate the sample size N

Given a half-length for a confidence interval, the sample size can be estimated to meet this criterion. In this research, a confidence interval at the 95% level for a half-length error is set at 10% of sample mean. The minimum sample size can be calculated with inequality (5.3) and (5.4).

Step 6: Adjust the estimated value of the sample size

If we get $N \geq N_0$ in step 5, additional $N - N_0$ samples are collected and the mean value of the sample is recalculated. Otherwise, the estimate value of sample size is good enough to give results with the desired precision.

5.4.2 Measurement Data

Tests for throughput measurements are executed with packet data size of 100 bytes, 500 bytes and 1000 bytes, respectively,; whereas tests for response time measurements are executed with 100 bytes, 500 bytes, up to 5000 bytes.. Table 5-1 and 5-2 show the summary of test results for TCP packets, while table 5-3 and 5-4 shows the results for UDP packets.

Table 5-1 TCP Response Time Measurement

Data Size (byte)	VPN	Sample Mean (ms)	Standard Deviation	Variance	95% Confidence Interval
100	ON	4.9333	0.5833	0.3289	4.9333 ± 0.2087
	OFF	4.9000	0.6074	0.3567	4.9000 ± 0.2174
500	ON	4.9667	0.6687	0.4322	4.9667 ± 0.2393
	OFF	4.9000	0.6074	0.3567	4.9000 ± 0.2174
1000	ON	4.9667	0.9643	0.8989	3.9667 ± 0.3451
	OFF	4.9333	0.8683	0.7289	3.8667 ± 0.3107
2000	ON	11.2333	0.9643	0.8989	11.2333 ± 0.4274
	OFF	10.0000	1.1943	1.3789	10.0000 ± 0.4096
3000	ON	13.7000	0.7944	0.6100	13.7000 ± 0.2843
	OFF	13.0667	0.9072	0.7956	13.0667 ± 0.3246
4000	ON	16.8333	0.9129	0.8056	3.8667 ± 0.3267
	OFF	16.0333	0.8899	0.7656	3.8667 ± 0.3184
5000	ON	17.2333	1.5687	2.3789	17.2333 ± 0.5614
	OFF	16.2000	1.1265	1.2267	16.2000 ± 0.4031

Table 5-2 TCP Throughput Measurement

Measurement (byte)	VPN	Sample Mean (Mbps)	Standard Deviation	Variance	95% Confidence Interval
100	ON	1.6888	0.3331	0.1073	1.6888 ± 0.1192
	OFF	1.7602	0.4171	0.1682	1.7602 ± 0.1492
500	ON	1.6865	0.3433	0.1139	1.6865 ± 0.1229
	OFF	1.8563	0.3939	0.1500	1.8563 ± 0.1410
1000	ON	1.6644	0.3757	0.1365	1.6644 ± 0.1344
	OFF	1.7843	0.4107	0.1630	1.7843 ± 0.1470

Table 5-3 UDP Response Time Measurement

Measurement (byte)	VPN	Sample Mean (ms)	Standard Deviation	Variance	95% Confidence Interval
100	ON	3.9667	0.8899	0.7556	3.9667 ± 0.3184
	OFF	3.8667	0.8996	0.7822	3.8667 ± 0.3219
500	ON	3.9333	0.6397	0.3956	3.9333 ± 0.2289
	OFF	4.0000	0.8710	0.7333	4.000 ± 0.3117
1000	ON	4.0000	0.9469	0.8667	4.0000 ± 0.3388
	OFF	3.9667	0.8899	0.7656	3.9667 ± 0.3184
2000	ON	7.1667	1.1667	1.2056	7.1667 ± 0.3996
	OFF	6.9333	0.9444	0.8622	6.9333 ± 0.3380
3000	ON	8.8667	1.1059	1.1822	8.8667 ± 0.3957
	OFF	8.6333	0.9994	0.9656	8.6333 ± 0.3576
4000	ON	9.1000	1.0169	1.0900	9.1000 ± 0.3800
	OFF	8.9333	0.9072	0.7956	8.9333 ± 0.3246
5000	ON	10.1667	1.1472	1.2772	3.9667 ± 0.3184
	OFF	9.9333	1.4840	2.1289	3.9667 ± 0.3184

Table 5-4 UDP Throughput Measurement

Measurement (byte)	VPN	Sample Mean (Mbps)	Standard Deviation	Variance	95% Confidence Interval
100	ON	1.2816	0.3013	0.0878	1.2816 ± 0.1078
	OFF	1.3059	0.4565	0.2014	1.3059 ± 0.1633
500	ON	1.3107	0.4566	0.2015	1.3107 ± 0.1634
	OFF	1.2882	0.4198	0.1707	1.2882 ± 0.1502
1000	ON	1.2604	0.3229	0.1008	1.2604 ± 0.1155
	OFF	1.2883	0.4142	0.1659	1.2883 ± 0.1482

5.4.3 Analysis of Measurement Data

The graphic presentation of experiment output is shown in Figure 5-5, 5-6, 5-7 and 5-8, which are derived from table 5-1 to table 5-4.

Figure 5-5 and 5-7 show the response time per byte for TCP and UDP packets respectively. There are two lines in each diagram representing the cases when VPN is on and off.

Figure 5-6 and 5-8 show the throughput per byte for TCP and UDP packets respectively.

Figure 5-5 TCP Response Time per Byte

Figure 5-6 TCP Throughput per Byte

Figure 5-7 UDP Response Time per Byte

Figure 5-8 UDP Throughput per Byte

The following can be concluded:

- In our experiments, the performance was measured for a single VPN, in which case the VPN processing does not introduce a significant overhead in terms of response time and throughput. Due to the limitation of the size of

the test bed used for measurements, we were not able to measure the processing overhead with a considerably large number of VPN provisioned.

- For packet size less than 1000 bytes there is no significant impact on either response time or throughput. But for packet size greater than or equal to 2000 bytes there is a significant difference in response time, because of the fragmentation and de-fragmentation needed to be applied to large packets.
- Although we studied the real system by doing the measurements on the test bed explained in 5.1, there were a number of limitations. First, the accuracy of the measurements was limited since a lot of other processes were running on the workstations that competed with the process being measured for resources. Second, it is difficult to provision multiple VPN connections on a test bed. Third, it is not convenient to manipulate all the parameters. Last, the measurement is tightly bound to the specific test bed, which may not fit with other systems. In order to overcome these limitations, simulation was adopted for other performance studies presented in the next chapter.

Chapter 6 Proposed QoS Enhancement

This chapter proposes an approach for Layer 2 VPN QoS enhancement that can be applied to PE routers. The goal of this approach is to provide differentiated services to preferred VPN traffic at the edge of the MPLS network, which contributes to the overall VPN QoS. A real system implementation and simulation are performed to validate the feasibility of the approach. The first section defines the problem being studied. The second section proposes the new approach for the Layer 2 QoS enhancement. The third section discusses the implementation and test results. The fourth section presents the design of the OPNet simulation model. Simulation scenarios and data analysis are also discussed in this section.

6.1 Problem Definition

With the growth of the IP traffic and bandwidth in the backbone network, the edge devices are becoming the network bottleneck. More complexity and performance burden have been placed on edge devices due to the increasing application demands [6].

Unlike the backbone routers, whose primary function is to forward packets, edge devices must provide more services concurrently.

Edge devices are the access points for customer traffic to the provider's network. Customer traffic may come in through various interfaces and technologies, such as DSL, dial-up, wireless, leased line, IP, ATM, Frame Relay, and Ethernet etc. The ingress edge devices should be able to process different types of incoming traffic and send them over

to the core network, while the egress edge device should send the traffic from the core to the different customers in the data format that is supported by the customer.

For packet processing, edge devices must perform multiple complex and processor-intensive tasks, such as routing table look up, CAC, traffic classification, traffic shaping, traffic policing, MPLS label mapping, multicast tunneling, IPv6 encapsulation, VPN encapsulation etc.

Edge devices must implement a wide range of routing protocols, such as OSPF, BGP, RIP, LDP, RSVP-TE, L2TP etc, which means that edge devices have to manage a large number of concurrent routing sessions, forwarding table entries.

The concurrency of the above-mentioned complex tasks would bring a heavy burden to the limited resources of the edge devices, and performance degradation may occur in the form of congestion, long response time, and low throughput. Providing better service to preferred traffic within the edge devices may be an approach for achieving end-to-end performance [22, 25, 37].

Intensive studies have been done to provide QoS to preferred traffic in the circumstance of limited network resources. There are three aspects of QoS implementation, namely traffic classification, network element QoS and end-to-end QoS, which are explained in 2.3.

For traffic classification in particular, the current approach to traffic marking is to use the Layer 3 header or Layer 2 header to carry the flag of the service class. IEEE 802.1D (superseding IEEE 802.1p, which is derived from IEEE 802.1Q) defines an

elementary QoS scheme at Layer 2 that lets end-stations request priority and allows network devices to enforce it by using a tag in the packet header. 802.1D introduces QoS at Layer 2 by enabling "Prioritized Ethernet" with up to 8 priorities in Ethernet and Token Ring networks. The eight discrete priority levels range from the default of best effort, through excellent effort (a business-critical application, but tolerant of some delay), interactive multimedia (sensitive to delay or jitter), and reserved (highest priority) [8, 10, 11, 20, 36].

For Martini VPN, the provider's network just provides point-to-point Layer 2 emulated connectivity between customer sites without getting involved in the routing of customer's traffic. The Layer 3 header of the packet is not processed. If 802.1D is chosen to implement the QoS at Layer 2, then all the routers along the tunnel have to support this protocol, which may not be the case. If customer devices and provider's routers do not implement 802.1D, then QoS at Ethernet level cannot be done unless the customers and the provider upgrade their equipment [10, 36].

In the test environment of this research, the device drivers for the NIC cards on host machines and routers do not support 802.1D.

6.2 Proposed Solution

According to [48], a service provider's network should support access QoS service or edge-to-edge QoS service. To provide QoS enhancement to Martini VPN traffic within the PE routers, a new approach for traffic identification and marking scheme is proposed in this section.

Traffic Identification

Martini VPN traffic is bound to the incoming interfaces at ingress PE, which is the incoming port for Ethernet, VPI/VCI for ATM, or DLCI for Frame Relay etc. So traffic of different Martini VPNs can be identified and classified based on the interface from which it comes. Within the ingress PE, a prioritized service can be applied to the VPN traffic based on its classification through a proper queueing scheme, and congestion control mechanisms.

Traffic Marking

The identified and classified VPN traffic should be recognized at the egress PE, as well, in order to get prioritized service there. This requires the classification information to be carried with the packets over the MPLS backbone to the egress PE router. We propose to use the three EXP bits in the demultiplexer field of the Martini VPN encapsulation to carry this information, which can support up to eight classes of service.

6.3 Implementation and Evaluation

The proposed approach is implemented and tested in the real system for feasibility study. The well-known Weighted Round-Robin queueing scheme is used at PE routers to provide better throughput for the preferred VPN traffic [49]. Figure 6-1 shows the test environment. Instead of serving one CE, as explained in 5.1, each PE router is connected to two CEs. Two VPN connections are configured in PEs, one between CE1 and CE2, and the other between CE3 and CE4. Qcheck, which was presented in 5.2, is used to measure the throughput of the two VPNs.

Figure 6-1 Test Environment

VPN 1 represents the preferred traffic. The quantum set for queue 1 and 2 are 100 and 10 respectively. Table 6-1 presents the measurement data of the measurement.

Table 6-1 Measurement Data Summary

Measurement	CoS	VPN	Sample Mean	Standard Deviation	Variance	95% Confidence Interval
Throughput (Mbps)	ON	VPN1	0.7110	0.1737	0.0291	0.7110 ± 0.0621
		VPN2	0.7792	0.1768	0.0302	0.7792 ± 0.0633
	OFF	VPN1	1.1194	0.1710	0.0283	1.1194 ± 0.0612
		VPN2	0.1959	0.0454	0.0020	0.1959 ± 0.0162

Figure 6-2 gives a visual comparison of the end-to-end throughput for both VPNs. In the diagram, the group of bars on the left hand side represents the throughput without proposed QoS enhancement, while the group on the right hand side represents the throughput when the proposed approach is applied.

Figure 6-2 Throughput Comparison

The result shows that the throughput of the preferred traffic can be improved significantly by applying the proposed approach.

6.4 Simulation

There are some limitations in assessing the proposed approach by measurements in a real system, as described in the previous section. First, it is not convenient to set up a network with multiple nodes, since the required resources are not always available. Second, there might be other applications running in a real system, which may interfere with the module being tested. So the accuracy of a specific test can be difficult to control. Third, sometime it is time consuming to setup the test environment and make parameter changes. And last, the available test tool may not meet all of the needs.

In order to study the proposed approach with more control and depth, simulation with OPNet is performed. This section describes the simulation model, simulation experiments and the analysis of the simulation results.

6.4.1 Simulation Model

6.4.1.1 Network Topology

The proposed Martini VPN QoS approach only concerns CE devices and PE routers; P routers in the provider's network behave in the same way as when forwarding regular MPLS packets, so they are omitted in the simulation.

As shown in Figure 6-3, the network topology employed in the simulation consists of two PE nodes, which are connected to each other via bidirectional links. Each PE node is connected with five CE nodes.

Figure 6-3 Network Topology in Simulation

CE_0 and CE_5 are the pair of CE devices that belong to same VPN customer, so as CE_1 and CE_6, CE_2 and CE_7, CE_3 and CE_8, CE_4 and CE_9. The bidirectional traffic of different VPN is aggregated on to the links between PE_1 and PE_2.

6.4.1.2 Node Model

Two node models are developed according to the network topology shown in Figure 6-3.

CE Node Model

The first one is a CE node model. A CE node generates packets and sends them to a PE node. A CE node also absorbs packets addressed to it from a PE node. Figure 6-4 shows the components in CE node models.

Figure 6-4 CE Node Model

Source component “src” is the one that generates packets of various lengths at various inter-arrival times. Processor component “proc” sends the packets from “src” out through the transmit component “xmt”. “proc” also absorbs the packets from the receiver component “rcv”.

PE Node Model

The second is the PE node model. A PE node receives the packets from the CE nodes, buffers them if necessary, and sends them over to the peer PE node. It does a similar job to the packets from the peer PE node and dispatches them to the correct CE nodes. Figure 6-5 shows the components in the PE node model.

Figure 6-5 PE Node Model

The scheduler component “schdlr” has six receive/transmit component pairs. Pair 1 to 5 is connected to individual CE nodes, while pair 6 is connected to the peer PE node.

“schdlr” buffers the received packets and schedules the sending of the buffered packets according to internal queueing scheme and policy.

6.4.1.3 Process Model

proc

The process model employed by “proc” of a CE node has four states as shown in figure 6-6.

Figure 6-6 "proc" Process Model

The process starts from “init” state and goes into “idle” state. At a packet arrival event from “src” (as shown in figure 6-4), the process enters the “xmt” state, in which “proc” sends the packet out, and then returns back to the “idle” state. At the event of packet from “rcv”, the process enters “rcv” state, in which “proc” destroys the received packet.

PE

The process model employed by “schdlr” of a PE node has four states as well, as shown in figure 6-7.

Figure 6-7 "schdlr" Process Model

Similar to the previous process model, “schdlr” process starts from the “init” state and goes into the “idle” state. At the event of a packet arrival (as shown in figure 6-5), the process enters “enqueue” state, in which “schdlr” queues the packet according to its configured queueing scheme and policy, and then returns back to the “idle” state. At the scheduled event of extracting packet, the process enters the “extract” state, in which “proc” extracts packets from its internal queues, according to the queueing scheme and policy, and sends them away.

6.4.1.4 Link Model

As shown in figure 6-3, the links connecting the nodes are set with a data rate of 100Mb/s, in order to simulate fast Ethernet connections.

6.4.1.5 Packet Format

The format of packet header used in simulation is defined as in figure 6-8.

source_address	dest_address
arrival_time	CoS

Figure 6-8 Packet Format

- source_address: the ID number of a CE node from which a packet originates.
- dest_address: the ID number of a CE node to which a packet terminates.
- arrival_time: the time when a packet is received by a node.
- CoS: Class of Service

This generic packet header contains necessary information to suit the need of correct packet delivery and the queueing schemes employed in the simulation.

The payload of a packet has a fixed length of 1024 bytes.

6.4.2 Simulation Experiments

Goal

The main goal of the simulation is to study the effectiveness of proposed approach described in 6.2. The performance factor of interest in the simulation is the throughput of VPN traffic of different priorities. As shown in Figure 6-3, VPN 1 is given the highest priority; VPN 2 next, and VPN 3 is of the lowest priority.

Scenario

The packets from CE nodes are identified and classified at the ingress PE based on their source address. The individual traffic gets different level of service at ingress PE according to its classification. The classification information is carried over to the egress PE in the “COS” field of the packet header, so the packet can have the same level of service at egress PE as at ingress PE. Bidirectional traffic is adopted in the simulation.

Key Parameters

The following table gives a summary of the key parameter settings adopted in the simulation.

Table 6-2 Key Parameter Setting

Node	Parameter
CE	<ul style="list-style-type: none"> • Packet inter-arrival time: exponential • Packet size: constant 1024 bytes
PE	<ul style="list-style-type: none"> • Service rate: 1Mbps • Queueing algorithm: depending on experiment case • Buffer size: depending on experiment case • Quantum for each queue: varies depending on experiment case
Link	<ul style="list-style-type: none"> • Data rate: 100Mbps

Experiment Cases

This section gives a summary of the simulation experiments performed to study the effectiveness of proposed QoS enhancement. The simulation results are analyzed in the next section.

As shown in Table 6-3, case 1 studies the scenario with a single FIFO queue in PE routers. Case 2 studies the scenario with Weight Round Robin queueing at PE routers. The rest of the cases are derived from case 2, but with different settings of the inter-arrival time, queue size and quantum, which are indicated by bold fonts.

Table 6-3 Simulation Experiments Summary

Experiments	Model
Case 1	<ul style="list-style-type: none"> • Queueing: FIFO • Inter-arrival time: 0.005 second • Queue size: 100,000 • Quantum: NA
Case 2	<ul style="list-style-type: none"> • Queueing: Weighted Round Robin • Inter-arrival time: 0.005 second • Queue size: 20,000 • Quantum 0: 3000 • Quantum 1: 2000 • Quantum 2: 1000
Case 3	<ul style="list-style-type: none"> • Queueing: Weighted Round Robin • Inter-arrival time: 0.01 second • Queue size: 20,000 • Quantum 0: 3000 • Quantum 1: 2000 • Quantum 2: 1000
Case 4	<ul style="list-style-type: none"> • Queueing: Weighted Round Robin • Inter-arrival time: 0.0025 second • Queue size: 20,000 • Quantum 0: 3000 • Quantum 1: 2000 • Quantum 2: 1000
Case 5	<ul style="list-style-type: none"> • Queueing: Weighted Round Robin • Inter-arrival time: 0.005 second • Queue size: 10,000 • Quantum 0: 3000 • Quantum 1: 2000 • Quantum 2: 1000
Case 6	<ul style="list-style-type: none"> • Queueing: Weighted Round Robin • Inter-arrival time: 0.005 second • Queue size: 40,000 • Quantum 0: 3000

	<ul style="list-style-type: none"> • Quantum 1: 2000 • Quantum 2: 1000
Case 7	<ul style="list-style-type: none"> • Queueing: Weight Round Robin • Inter-arrival time: 0.005 second • Queue size: 20,000 • Quantum 0: 1500 • Quantum 1: 1000 • Quantum 2: 500
Case 8	<ul style="list-style-type: none"> • Queueing: Weight Round Robin • Inter-arrival time: 0.005 second • Queue size: 20,000 • Quantum 0: 6,000 • Quantum 1: 4,000 • Quantum 2: 2,000

6.4.3 Simulation Analysis

6.4.3.1 With and Without CoS

From the output of case 1, shown in Table 6-4 and Figure 6-9, the throughput is evenly distributed among the three traffic flows. By applying differentiated treatment to the traffic flows, the output of case 2 presents a significant change in throughput, which is depicted in Figure 6-10. With the parameter setting listed in Table 6-3, the throughput of VPN 1 is 71.93% higher than the one of VPN 2, which is also 143.20% higher than the one of VPN3.

Table 6-4 Summary for experiment Case 1

Scenario	Node	Sample Mean (bps)	Variance	Standard Deviation (bps)	95% Confidence Interval (bps)
	CE_0	99.03	2,213.35	47.04	[96.11, 101.94]

	CE_0	99.03	2,213.35	47.04	[96.11, 101.94]
	CE_2	98.40	2,109.99	48.06	[96.44, 102.40]

Figure 6-9 Experiment Case 1: End-to-End Throughput (Single FIFO)

Table 6-5 Summary for experiment Case 2

Scenario	Node	Sample Mean (bps)	Variance	Standard Deviation (bps)	95% Confidence Interval (bps)
End-to-End Throughput	CE_0	181.18	4,106.50	64.08	[177.21, 185.15]
	CE_1	105.38	1,895.70	43.53	[102.68, 108.08]
	CE_2	43.33	1,103.05	33.21	[41.27, 45.39]

Figure 6-10 Experiment Case 2: End-to-End Throughput (WRR)

6.4.3.2 Inter-arrival Time and Throughput

Experiment case 3 and case 4 are based on case 2, but with double and half the mean value of inter-arrival time, respectively.

The output of case 3, which is shown in Table 6-6 and Figure 6-11, indicates that with decreased inter-arrival, the difference between the throughputs decreases as well. With doubled inter-arrival, the throughput of VPN 1 drops by 43.53%, and the difference between individual throughputs is less than 1 packets/s.

Table 6-6 Table 6 6 Summary of experiment Case 3

Scenario	Node	Sample Mean (bps)	Variance	Standard Deviation (bps)	95% Confidence Interval (bps)
End-to-End Throughput	CE_0	102.30	391.80	19.79	[100.57, 104.04]
	CE_1	101.57	362.08	19.02	[99.90, 103.23]
	CE_2	101.01	363.66	19.06	[99.33, 102.68]

Figure 6-11 Experiment Case 3: End-to-End Throughput (WRR, increased inter-arrival time)

The output of case 4 demonstrates the opposite trend to the one in case 3. The difference between the individual throughputs increases with the inter-arrival time. With the half mean value of inter-arrival time, the throughput of VPN 1 increases by 15.19%, while the throughputs of VPN 2 and VPN 3 decrease by 57.23% and 85.02% respectively.

Table 6-7 Summary for experiment Case 4

Scenario	Node	Sample Mean (bps)	Variance	Standard Deviation (bps)	95% Confidence Interval (bps)
End-to-End Throughput	CE_0	208.71	3,583.53	59.86	[203.45, 213.96]
	CE_1	45.07	703.60	26.52	[42.75, 47.40]
	CE_2	6.49	49.62	7.04	[5.87, 7.11]

Figure 6-12 Experiment Case 4: End-to-End Throughput (WRR, decreased inter-arrival time)

6.4.3.3 Queue Size and Throughput

The output of case 5, in which the queue size of PE nodes is reduced to half, shown in Figure 6-13. Comparing to case 2, the throughput of VPN 1 decreases by 23.46%, which the throughput of VPN 2 and VPN 3 increase by 28.14% and 93.53%.

Table 6-8 Summary for experiment Case 5

Scenario	Node	Sample Mean (bps)	Variance	Standard Deviation (bps)	95% Confidence Interval (bps)
End-to-End Throughput	CE_0	138.67	1,037.79	32.21	[136.68, 140.67]
	CE_1	135.04	1,019.12	31.92	[133.06, 137.02]
	CE_2	83.86	597.20	24.43	[82.34, 85.37]

Figure 6-13 Experiment Case 5: End-to-End Throughput (WRR, decreased Q size)

By doubling the queue size in PE nodes, the output of case 6 shows the overall decline in throughput, which can be seen in Figure 6-14.

Table 6-9 Summary for experiment Case 6

Scenario	Node	Sample Mean (bps)	Variance	Standard Deviation (bps)	95% Confidence Interval (bps)
End-to-End Throughput	CE_0	138.67	1,037.79	32.21	[136.68, 140.67]
	CE_1	135.04	1,019.12	31.92	[133.06, 137.02]
	CE_2	83.86	597.20	24.43	[82.34, 85.37]

Figure 6-14 Experiment Case 6: End-to-End Throughput (WRR, increased Q size)

6.4.3.4 Quantum and Throughput

In case 7, the quantum for each queue is reduced to half of the setting in case 2. The output is shown in Table 6-10 and Figure 6-15. The throughput of VPN 1 reduces by 2.33%, while the throughput of VPN 2 and VPN 3 increase by 14.62% and 46.61%.

Table 6-10 Summary for experiment Case 7

Scenario	Node	Sample Mean (bps)	Variance	Standard Deviation (bps)	95% Confidence Interval (bps)
End-to-End Throughput	CE_0	176.97	1,759.50	41.94	[174.37, 179.57]
	CE_1	120.79	1,368.30	36.99	[118.49, 123.08]
	CE_2	63.53	1,139.08	33.75	[61.44, 65.62]

Figure 6-15 Experiment Case 7: End-to-End Throughput (WRR, decreased weight difference)

Case 8 does the opposite of case 7. The quantum of each queue in case 2 is doubled. Table 6-11 and Figure 6-16 give the summary and graphic presentation of the output data. The throughput of VPN 1 and VPN 2 increases by 15.42% and 18.93%, but the throughput of VPN 3 drops by 29.88%.

Table 6-11 Summary for experiment Case 8

Scenario	Node	Sample Mean (bps)	Variance	Standard Deviation (bps)	95% Confidence Interval (bps)
End-to-End Throughput	CE_0	176.97	1,759.50	41.94	[174.37, 179.57]
	CE_1	120.79	1,368.30	36.99	[118.49, 123.08]
	CE_2	63.53	1,139.08	33.75	[61.44, 65.62]

Figure 6-16 Experiment Case 8: End-to-End Throughput (WRR, increased weight difference)

6.4.4 Summary

The following conclusions can be drawn from the simulation results:

- By applying the proposed approach, the higher service level traffic will have a better treatment at PE routers, and relatively better end-to-end throughput can be achieved in case of congestion.
- The proposed scheme kicks in at high traffic intensity. For large inter-arrival time (i.e., low intensity) as in Figure 6-11, there is enough network capacity to serve all traffic classes at the same throughput. However, when the inter-arrival time goes down (i.e., the intensity goes up) as in Figure 6-12, the throughput difference becomes apparent between classes.
- Increasing the queue size can result in a greater difference in end-to-end throughput between traffic of first and second priority.
- Decreasing the difference in quantum setting among the queues leads to closer throughput values of the traffic flows. The throughput of the traffic with highest priority declines and the ones of lower priorities increases. Increasing the difference in quantum setting can cause greater end-to-end throughput of the traffic with highest priority.
- There is no single factor that can decide solely the end-to-end throughput being observed. The actual value is affected by the combination of inter-arrival time, queue size and quantum setting.

Chapter 7 Conclusion and Future Work

7.1 Summary of Research Work

The primary objective of the present research is to study the latest technologies for Ethernet-based point-to-point VPN over MPLS network through actual design, implementation and performance evaluation, and to propose improvements where possible. A new QoS enhancement approach at the edge nodes of the MPLS domain is proposed, which provides port-based DiffServ-like service to the VPN traffic flows. A simulation model based on OPNet for the proposed QoS enhancement approach has been designed, evaluated and analyzed. Overall the research activities can be summarized as follows:

- Reviewed the MPLS Layer 2 VPN related technologies:
 - MPLS Standards
 - IETF RFC and drafts for Ethernet-based point-to-point over MPLS network (Martini VPN)
 - QoS technology, with a special focus on QoS at the Ethernet level, including IEEE 802.1Q, 802.1P and 802.1D
 - EION Open IP environment

- Designed, implemented and tested the control plane and forwarding plane of Martini VPN module integrated with EION's Open IP software suite. Measured the end-to-end performance in terms of delay and throughput.
- Studied the OPNet Modeler. OPNet Modeler is a simulation tool for network modeling and simulation, allowing to design and study communication networks, devices, protocols, and applications with a good flexibility and scalability. OPNet Modeler is widely used by the technology organizations to accelerate the R&D process.
- Developed the simulation model for the proposed QoS enhancement approach
- Analyzed potential scalability issues with Martini VPNs. Proposed a QoS enhancement approach at the edge nodes of an MPLS domain, which provides port based DiffServ-like service.
- Designed the simulation scenarios to evaluate the effectiveness of this new approach. Examined the impact of various factors that influence performance of individual VPN traffic flows, such as inter-arrival time, queue size, and quantum for each service class.
- Performed simulation experiments and analyzed the simulation results, which will result in the conclusion of this research.

7.2 Conclusion of Present Research

The main conclusions drawn from the studies undertaken in the present research are summarized as follows:

- Based on the actual implementation on EION Open IP suite and the test data from the specific test environment described in chapter 4 and 5, the Ethernet-based Martini VPN introduce negligible performance overhead in terms of end-to-end delay and throughput.
- By applying the proposed approach, the VPN traffic flow with a higher service level can have better treatment at PE routers, which will lead to relatively better end-to-end throughput in case of congestion.
- Based on the queueing scheme employed in the simulation model, which is Weighted Round Robin, the difference of end-to-end throughput among traffic flows is affected by several main factors: the inter-arrival time, queue size and the weight setting for each traffic flow.
- Increasing the inter-arrival time generally leads to greater end-to-end throughput of the most preferred traffic, and greater throughput difference among traffic flows. Decreasing the value shows the opposite trend. This implies the benefit is more noticeable at high traffic density.
- Greater difference in end-to-end throughput between traffic flows can also be resulted by increasing the queue size or the weight setting. So in order to

achieve a desired throughput for the preferred traffic flow, the queue size and the weigh setting have to be managed carefully.

- The study is limited to end-to-end performance for point-to-point connections, involving edge routers with weighted round robin internal queueing scheme. The extension of current study to a broader area would be interesting, which is mentioned in the next section.

7.3 Future Work

The following are some recommendations for future work:

- The scalability issues and performance impact of the edge nodes for a large number of VPN provisioned needs to be studied in the real system.
- The simulation scenarios used the well-known Weighted Round Robin queueing scheme; other queueing schemes may also be considered.
- Martini VPN is for point-to-point pseudo-wire emulation service. The study of point-to-multipoint model, which is also referred to as Virtual Private LAN Service (VPLS), may be more interesting for practical applications.

References

1. "CSP-Ace_OpenIP2_1_Dev_Doc", Eion Inc Document,
2. "CSP-Posix_OpenIP2_1_Dev_Doc", Eion Inc Document,
3. "Ethernet Technology", Cisco System, Inc.,
http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/ethernet.pdf
4. "Product: Performance Application - Qcheck", IXIA,
http://www.ixiacom.com/products/performance_applications/pa_display.php?skey=pa_q_check
5. "Quality of Service Networking", Cisco Systems, Inc.,
http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/qos.pdf
6. "Scalability at the Edge: When is enough enough?" Agilent Technologies, Inc., 2002, <http://advanced.comms.agilent.com/insight/2002-06/Scalability.pdf>
7. "Virtual Routing for Provider Edge Applications", IP Infusion Inc, 2002,
http://www.ipinfusion.com/pdf/VirtualRouting_app-note_3rev0302.pdf
8. "Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Common specifications - Part 3: Media Access Control (MAC) Bridges", ISO/IEC 15802-3: 1998, ANSI/IEEE Std 802.1D, 1998 Edition, LAN MAN Standards Committee of the IEEE Computer Society, Dec 1998,
9. "White Paper - MPLS Virtual Private Network MPLS Virtual Private Network", Spirent Communication, Inc., Mar 2003,
<http://www.spirentcom.com/documents/760.pdf>
10. "Testing 802.1P QoS with Chariot", Application Note, NetIQ Corporation, Mar 2004,
http://download.netiq.com/Library/Application_Notes/NetIQ_Testing_802.1P_QoS_with_Chariot.pdf
11. "802.1Q IEEE Standards for Local and metropolitan area networks - Virtual Bridged Local Area Networks", IEEE Std 802.1Q, 2003 Edition, IEEE, May 2003,
12. "Private Use of Public Networks for Service Provider", 3Com communication, Nov 1998, <http://www.adimpleo.com/library/3com/500649.pdf>

13. P. Agarwal and B. Akyol, "*Time-To-Live (TTL) Processing in Multi-Protocol Label Switching (MPLS) Networks*", RFC 3443, Internet Engineering Task Force, Jan 2003, <http://www.ietf.org/rfc/rfc3443.txt>
14. L. Andersson, P. Doolan, N. Feldman, A. Fredette, and B. Thomas, "*LDP Specification*", RFC 3036, Internet Engineering Task Force, Jan 2001, <http://www.ietf.org/rfc/rfc3036.txt>
15. J. Banks, J.S. Carson, B.L. Nelson, and D.M. Nicol, "*Discrete-Event System Simulation*", Prentice Hall, 2001,
16. U. Black, "*MPLS and label switching networks*", Prentice Hall, 2001,
17. R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, "*Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification*", RFC 2205, Internet Engineering Task Force, Sep 1997, <http://www.ietf.org/rfc/rfc2205.txt>
18. S. Bryant and P. Pate, "*PWE3 Architecture*", Internet Engineering Task Force, Mar 2003, <http://www.ietf.org/internet-drafts/draft-ietf-pwe3-arch-07.txt>
19. M. Carugi and J.D. Clercq, "*Virtual Private Network Services: Scenarios, Requirements and Architectural Constructs from a Standardization Perspective*", IEEE Communication Magazine, Jun 2004,
20. Q. Chiruvolu, A. Ge, D. Elie-Dit-Cosaque, M. Ali, and J. Rouyer, "*Issues and Approaches on Extending Ethernet Beyond LANs*", IEEE Communications Magazine, Mar 2004, <http://www.utdallas.edu/~mx033100/CommMagMarch04.pdf>
21. J.-M. Chung, "*Analysis of MPLS Traffic Engineering*", Proceedings of 43rd IEEE Midwest Symposium on Circuits and Systems, Aug 2000,
22. J. Clercq and O. Paridaens, "*Scalability Implication of Virtual Private Networks*", IEEE Communication Magazine, May 2002,
23. J.L. Devore, "*Probability and statistics for engineering and the sciences*", Duxbury Press, 2000,
24. N. Doraswamy and D. Harkins, "*IPsec, The New Security Standard for the Internet, Intranets, and Virtual Private Networks*", Prentice Hall, 1999,
25. F. Falmieri, "*VPN scalability over high performance backbones - Evaluating MPLS VPN against traditional approaches*", Proceedings of IEEE International Symposium on Computers and Communication, 2003,
26. P. Ferguson and G. Huston, "*What is a VPN?*" 1998, <http://www.clark.net/pub/timw/vpn/Tech/vpn.pdf>

27. K. Hamzeh, G. Pall, W. Verthein, J. Taarud, W. Little, and G. Zorn, "*Point-to-Point Tunneling Protocol*", RFC 2637, Internet Engineering Task Force, Jul 1999, <http://www.ietf.org/rfc/rfc2637.txt>
28. J. Harrison, M. Finlayson, and R. Sugarman, "*VPN TECHNOLOGIES - A COMPARISON*", Data Connection Ltd., Feb 2003, <http://www.dataconnection.com/network/download/whitepapers/vpntechwp.pdf>
29. J.D. Harwood, "*IPsec: Technology and Application Overview*", Vesta Corporation, Jan 2001, <http://www.vesta-corp.com/IpsecOverview.pdf>
30. P. Knight and C. Lewis, "*Layer 2 and 3 Virtual Private Network: Taxonomy, Technology, and Standardization Efforts*", IEEE Communication Magazine, Jun 2004,
31. K. Kompella and M. Leelanivas, "*Layer 2 VPNs Over Tunnels*", draft-kompella-l2vpn-l2vpn-00.txt, Internet Engineering Task Force, Jan 2004, <http://www.ietf.org/internet-drafts/draft-kompella-l2vpn-l2vpn-00.txt>
32. L. Martini and E.C. Rosen, "*Encapsulation Methods for Transport of Ethernet Frames Over IP/MPLS Networks*", draft-ietf-pwe3-ethernet-encap-06.txt, Internet Engineering Task Force, Apr 2004, <http://www.ietf.org/internet-drafts/draft-ietf-pwe3-ethernet-encap-06.txt>
33. L. Martini and E.C. Rosen, "*Pseudowire Setup and Maintenance using LDP*", draft-ietf-pwe3-control-protocol-07.txt, Internet Engineering Task Force, Jun 2004, <http://www.ietf.org/internet-drafts/draft-ietf-pwe3-control-protocol-08.txt>
34. L. Martini and D. Tap, "*Encapsulation Methods for Transport of Layer 2 Frames Over IP and MPLS Networks*", draft-martini-l2circuit-encap-mpls-07.txt, Internet Engineering Task Force, Jun 2004, <http://www.ietf.org/internet-drafts/draft-martini-l2circuit-encap-mpls-07.txt>
35. L. Martini and D. Tappan, "*Transport of Layer 2 Frames over MPLS*", draft-martini-l2circuit-trans-mpls-14.txt, Internet Engineering Task Force, Jun 2004, <http://www.ietf.org/internet-drafts/draft-martini-l2circuit-trans-mpls-14.txt>
36. P.L. Montessoro and D. Pierattoni, "*Advanced research issues for tomorrow's multimedia networks*", Proceedings of ITCC IEEE, 2001, http://citeseer.ist.psu.edu/cache/papers/cs/24811/http:zSzzSzweb.diegm.uniud.itzSzUtentizSzpierlucaszSzpublic_htmlzSzresearchzSzITCC2001_main.pdf/advance-d-research-issues-for.pdf
37. A. Nagarajan, "*Generic Requirements for Provider Provisioned Virtual Private Networks (PPVPN)*", Internet Engineering Task Force, Jun 2004, <ftp://ftp.rfc-editor.org/in-notes/rfc3809.txt>

38. D.C. Plummer, "*An Ethernet Address Resolution Protocol*", RFC 826, Internet Engineering Task Force, Nov 1982, <http://www.ietf.org/rfc/rfc826.txt>
39. E. Rosen and L. Andersson, "*Framework for Layer 2 Virtual Private Networks (L2VPNs)*", *draft-ietf-l2vpn-l2-framework-04.txt*, Internet Engineering Task Force, Mar 2004, <http://www.ietf.org/internet-drafts/draft-ietf-l2vpn-l2-framework-04.txt>
40. E. Rosen and Y. Rekhter, "*BGP/MPLS VPNs*", RFC 2547, Internet Engineering Task Force, Mar 1999, <http://www.ietf.org/rfc/rfc2547.txt>
41. E. Rosen, D. Tappen, G. Fedorkow, Y. Rekhter, D. Farinacci, T. Li, and A. Conta, "*MPLS Label Stack Encoding*", RFC 3032, Internet Engineering Task Force, Jan 2001, <http://www.ietf.org/rfc/rfc3032.txt>
42. E. Rosen, A. Viswanathan, and R. Callon, "*Multi-protocol Label Switching Architecture*", RFC 3031, Internet Engineering Task Force, Jan 2001, <http://www.ietf.org/rfc/rfc3031.txt>
43. T. Rybczynski, "*New World Routing - Open IP*", Nortel Networks, Sep 2000, http://www.nortelnetworks.com/solutions/financial/collateral/csm_openip_v1.pdf
44. D.C. Schmidt, "*ACE Class Category*", http://www.cs.wustl.edu/~schmidt/ACE_wrappers/docs/ACE-categories.html
45. D.C. Schmidt, "*ACE Overview*", <http://www.cs.wustl.edu/~schmidt/ACE-overview.html>
46. D.C. Schmidt and S.D. Huston, "*C++ Network Programming, Volume 1, Mastering Complexity with ACE and Patterns*", Addison-Wesley, 2002,
47. D.C. Schmidt and S.D. Huston, "*C++ Network Programming, Volume 2, Systematic Reuse with ACE and Frameworks*", Addison-Wesley, 2003,
48. Y. Serbest and W. Augustyn, "*Service Requirements for Layer-2 Provider Provisioned Virtual Private Networks*", *draft-ietf-l2vpn-requirements-01.txt*, Aug 2004, <http://www.ietf.org/internet-drafts/draft-ietf-l2vpn-requirements-01.txt>
49. D. Stiliadis, "*Traffic Scheduling in Packet-Switched Networks: Analysis, Design, and Implementation*", *Computer Engineering*, University of California, 1996, <http://citeseer.ist.psu.edu/stiliadis96traffic.html>
50. T. Takeda, I. Inoue, R. Aubin, and M. Carugi, "*Layer 1 Virtual Private Networks: Service Concepts, Architecture Requirements, and Related Advances in Standardization*", IEEE Communication Magazine, Jun 2004,

REFERENCES

51. W. Townsley, A. Valencia, A. Rubens, G. Pall, G. Zorn, and B. Palter, "*Layer Two Tunneling Protocol "L2TP"*", RFC 2661, Internet Engineering Task Force, Aug 1999, <http://www.ietf.org/rfc/rfc2661.txt>
52. Z. Wang, "*Internet QoS - Architechures and Mechanisms for Quality of Service*", Morgan Kaufmann Publishers, 2001,