

DOKUZ EYLÜL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED
SCIENCES

**DATA TRANSMITTING IN MPLS (MULTI
PROTOCOL LABEL SWITCHING) NETWORK
AND QoS (QUALITY of SERVICE) EFFECTS**

by
Burçin ALAN

October, 2011

İZMİR

DATA TRANSMITTING IN MPLS (MULTI PROTOCOL LABEL SWITCHING) NETWORK AND QoS (QUALITY of SERVICE) EFFECTS

**A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for the Degree of Master of Science
in Electrical and Electronics Engineering**

**by
Burçin ALAN**

October, 2011

İZMİR

M.Sc THESIS EXAMINATION RESULT FORM

We have read the thesis entitled “**DATA TRANSMITTING IN MPLS (MULTI PROTOCOL LABEL SWITCHING) NETWORK AND QoS (QUALITY of SERVICE) EFFECTS**” completed by **BURÇİN ALAN** under supervision of **ASST. PROF. DR. ÖZGE ŞAHİN** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.



Asst. Prof. Dr. Özge ŞAHİN

Supervisor



Asst. Prof. Dr. Derya BİRGİN

(Jury Member)



Asst. Prof. Dr. Yavuz ŞENOL

(Jury Member)



Prof. Dr. Mustafa SABUNCU

Director

Graduate School of Natural and Applied Sciences

ACKNOWLEDGMENTS

I would like to thank to my advisor Asst. Prof Dr. Özge Şahin for her encouragements throughout this research. I also would like to thank my family for their endless support.

BURÇİN ALAN

DATA TRANSMITTING IN MPLS (MULTI PROTOCOL LABEL SWITCHING) NETWORK AND QoS (QUALITY of SERVICE) EFFECTS

ABSTRACT

As the amount of traffic on the Internet increases, the network performance decreases; causing network corruption, delay, jitter, and packet loss. Applications such as Web access, e-mail, and file transfer can tolerate network delays while delay-sensitive applications such as voice, video, and other real-time applications cannot.

In a best-effort data transmitting network, increasing bandwidth can be the first step to help with these real-time and delay-sensitive applications but it is not enough. To provide efficient service in the network, some talents must be built into the network. Quality of Service (QoS) protocols is designed to provide the control to a best-effort service and improvement. That's why Quality of Service (QoS) is a very important parameter for data transmitting.

Multi Protocol Label Switching (MPLS) provides traffic engineering and Virtual Private Network (VPN) services and QoS (Quality of Service). In addition, as using MPLS, different services can be provided in same network.

In this thesis, MPLS technology, data flow through the MPLS network, its services, QoS concepts, levels of QoS and QoS effects on the MPLS network are investigated. A network topology with two videophones, two Layer-2 switches and two MPLS service routers is used to generate traffic. At some levels of QoS on the MPLS network, traffic, which is created by videophones calling each other, is analyzed according to the MPLS service router's monitoring results. For applying different levels of QoS, different configurations are verified on the MPLS service router. The effects of QoS are investigated by comparing the differences between the monitoring results at levels of QoS, which are used in the study.

Keywords: multi protocol label switching (MPLS), quality of service (QoS)

MPLS (ÇOKLU PROTOKOL ETİKET ANAHTARLAMA) ŞEBEKESİ ÜZERİNDEN BİLGİ AKTARIMI VE QoS (SERVİS KALİTESİ) ETKİLERİ

ÖZ

İnternet üzerindeki trafik miktarı arttıkça şebeke performansı giderek düşmekte, şebeke üzerinde bozulmalara, gecikmelere, kararsızlığa ve paket kayıplarına neden olmaktadır. Web erişimi, e-posta ve dosya transferi gibi uygulamalar şebeke gecikmelerine dayanabilse de, ses, video ve diğer gerçek zamanlı gecikmelere duyarlı uygulamalar dayanamaz.

“En iyi-gayret” veri gönderimi olan bir şebeke üzerinde, bant genişliğini artırmak, gecikme hassasiyetli ve gerçek zamanlı uygulamaların istenen şekilde iletimi için gerekli ilk adımdır ama yeterli değildir. Verimli bir servis sağlayabilmek için şebeke içerisinde bazı yetenekler kullanılmalıdır. Servis Kalitesi protokolleri en iyi-gayret servislerin kontrolünü sağlamak ve iyileştirme için tasarlanmıştır. Bu nedenle veri iletiminde Servis Kalitesi çok önemli bir parametredir.

MPLS; trafik mühendisliği, sanal özel şebeke servisi ve servis kalitesini sağlar. Ayrıca, MPLS’i kullanarak farklı servisler aynı şebeke içerisinde sağlanabilir.

Bu tezde, MPLS teknolojisi, MPLS şebekesi üzerinden bilgi akışı, MPLS servisleri, QoS kavramı, QoS düzeyleri ve MPLS şebekesi üzerindeki QoS etkileri araştırılmıştır. Trafik yaratabilmek için, iki videofon, iki adet Katman-2 anahtarı ve iki adet MPLS servis yönlendiricisinden oluşan bir ağ topolojisi kullanılmıştır. MPLS şebekesi üzerinde farklı QoS kademelerinde, videofonların birbirini araması ile oluşturulan trafik, MPLS servis yönlendiricinin gösterge sonuçlarına göre analiz edilmiştir. Farklı kademelerde QoS uygulayabilmek için, MPLS servis yönlendiricisi üzerinde farklı konfigürasyonlar gerçekleştirilmiştir. Monitör sonuçları arasındaki farklar karşılaştırılarak QoS etkileri incelenmiştir.

Anahtar sözcükler: çoklu protokol etiket anahtarlama, servis kalitesi

CONTENTS

	Page
M.Sc THESIS EXAMINATION RESULT FORM.....	ii
ACKNOWLEDGMENTS	iii
ABSTRACT	iv
ÖZ	v
CHAPTER ONE – INTRODUCTION	1
1.1 Introduction	1
1.2 Historical Perspective	1
1.3 Literature Overview	5
1.1 Thesis Outline.....	7
CHAPTER TWO – MPLS (MULTI PROTOCOL LABEL SWITCHING).....	8
2.1 What is MPLS?	8
2.2 MPLS and IP	8
2.3 Advantages of MPLS	9
2.4 MPLS Operating Mechanism.....	10
2.4.1 Basic Concepts of MPLS.....	10
2.4.1.1 MPLS Domain	10
2.4.1.2 FEC (Forwarding Equivalent Class)	10
2.4.1.3 Labeled Packet	11
2.4.1.4 Label Stack.....	11
2.4.1.5 LSR (Label Switching Router).....	11
2.4.1.6 Control Component.....	12
2.4.1.7 Forwarding Component	12
2.4.1.8 LER (Label Edge Router)	12

2.4.1.9 LSP (Label Switched Path)	12
2.4.1.10 LDP (Label Distribution Protocol)	13
2.4.2 How Does MPLS Work?	14
2.4.2.1 MPLS Routing	16
2.4.2.1.1 Hop-by-Hop Routing.....	16
2.4.2.1.2 Explicit Routing	16
2.4.2.2 Data Flow in an MPLS Network.....	17
2.5 MPLS Services	18
2.5.1 Traffic Engineering (TE)	18
2.5.1.1 TE Metric	19
2.5.2 Virtual Private Network (VPN)	20
2.5.2.1 VPN Requirements	20
2.5.2.2 VPN Types	21
2.5.2.2.1 Virtual Leased Lines (VLL)	21
2.5.2.2.2 Virtual Private LAN Segments (VPLS)	21
2.5.2.2.3 Virtual Private Routed Networks (VPRNs)	22
2.5.2.2.4 Virtual Private Dial Networks (VPDNs).....	23
2.5.3 Quality of Service (QoS)	23
CHAPTER THREE – QoS (QUALITY of SERVICE)	25
3.1 What is Quality of Service?.....	25
3.2 Why QoS?	26
3.3 IntServ (Integrated Services) Architecture	27
3.3.1 RSVP (Resource Reservation Protocol)	28
3.4 DiffServ (Differentiated Services) Architecture	30
3.4.1 DiffServ Terminology.....	32
3.5 DSCP (Differentiated Services Code Point).....	33
3.6 IP Precedence: Differentiated QoS.....	34

3.7 Per-Hop Behaviors (PHB).....	34
3.7.1 Expedited Forwarding.....	35
3.7.2 Assured Forwarding.....	35
3.8 MPLS Support for DiffServ	36
3.9 End-to-End QoS Levels.....	36
3.9.1 Best-Effort Service	37
3.9.2 Differentiated Service	37
3.9.3 Guaranteed Service	38
3.10 QoS Functions	39
3.10.1 Classification	39
3.10.2 Marking.....	40
3.10.3 Policing	40
3.10.4 Shaping	41
3.10.5 Queuing and Scheduling.....	43
3.10.5.1 FIFO (First-in First-out).....	43
3.10.5.2 WFQ (Weighted Fair Queuing)	44
3.10.5.3 PQ (Priority Queuing).....	45
3.10.5.4 CQ (Custom Queuing)	46
3.11 Queue Management.....	47
3.11.1 RED (Random Early Detection)	48
3.11.2 WRED (Weighted Random Early Detection).....	48

CHAPTER FOUR – ANALYSIS OF DATA TRANSMITTING IN MPLS NETWORK AND QoS EFFECTS 49

4.1 Topology of The Thesis Work.....	49
4.1.1 General Explanations of The Commands	52
4.1.2 Queue 1 Configuration and Monitoring Results.....	56
4.1.3 Queue 2 Configuration and Monitoring Results.....	65
4.1.4 Queue 3 Configuration and Monitoring Results.....	78
4.1.5 Queue 5 Configuration and Monitoring Results.....	91

4.2 Analysis of The Results.....	104
CHAPTER FIVE – CONCLUSION	107
REFERENCES.....	110
APPENDIX A – Properties of Service Router.....	112
APPENDIX B – Basic Configurations in the SR/ESS Services	119
APPENDIX C – Glossary	132

CHAPTER ONE

INTRODUCTION

1.1 Introduction

Multi Protocol Label Switching (MPLS) is an improved method for transmitting packets through a network by using labels attached to IP packets. MPLS combines Layer-2 switching technologies with Layer-3 routing technologies. The primary aim of MPLS network is to create a flexible networking system, which provides stability and increased performance (Maqousi, 2006).

MPLS is a standard routing and switching platform, which combines the label switching and forwarding technology with routing technology of network layer rather than a service or application. Basic idea of MPLS is routing at edge and switching in core.

MPLS is also a Quality of Service (QoS) enabled technology, which enables traffic engineering and bandwidth guarantees along these paths. Besides, when an MPLS network supports Differentiated Services (DiffServ) architecture, traffic flows can receive class-based admission, differentiated queue servicing in the nodes, preemption priority, and other network behaviors that enables QoS guarantees.

1.2 Historical Perspective

The Internet has developed into an omnipresent network and infused the development of a species of new applications in business and consumer markets. These new applications have driven the need for increased and guaranteed bandwidth requirements in the network's backbone.

Besides to the traditional data services currently provided over the Internet, new video, voice, triple play and multimedia services are being improved. Because of the need for providing these services, the Internet has emerged as the network of choice.

Anyway, the needs such as speed and bandwidth, which placed on the network by these new applications and services, have strained the resources of the existing Internet infrastructure. This conversion of the network toward a packet and cell based infrastructure has introduced uncertainty into what has traditionally been a fairly deterministic network.

Other challenge relates to the forward of bits and bytes over the backbone to provide differentiated classes of services to users. The exponential increment in the user number and the traffic volume causes another dimension to this problem. Class of service (CoS) and Quality of Service (QoS) issues must be addressed in order to support the needs of the wide range of network users. All of these needs urge to term a new technology.

Several label switching initiatives emerged in the mid-1990 to improve the performance of software-based IP routers and provide Quality of Service (QoS). Among these were IP Switching (Ipsilon/Nokia), Tag Switching (Cisco), and ARIS (IBM). In early 1997, an Internet Engineering Task Force (IETF) Working Group was chartered to standardize a label switching technology. MPLS emerged from this effort as another labeling scheme, but one with this distinct advantage: it uses the same routing and host addressing schemes as IP — the protocol of choice in today's networks. Today MPLS is defined by a set of IETF Request for Comments (RFCs) and draft specifications (Miller & Stewart, 2004). This label switching timeline is shown in Figure 1.1.

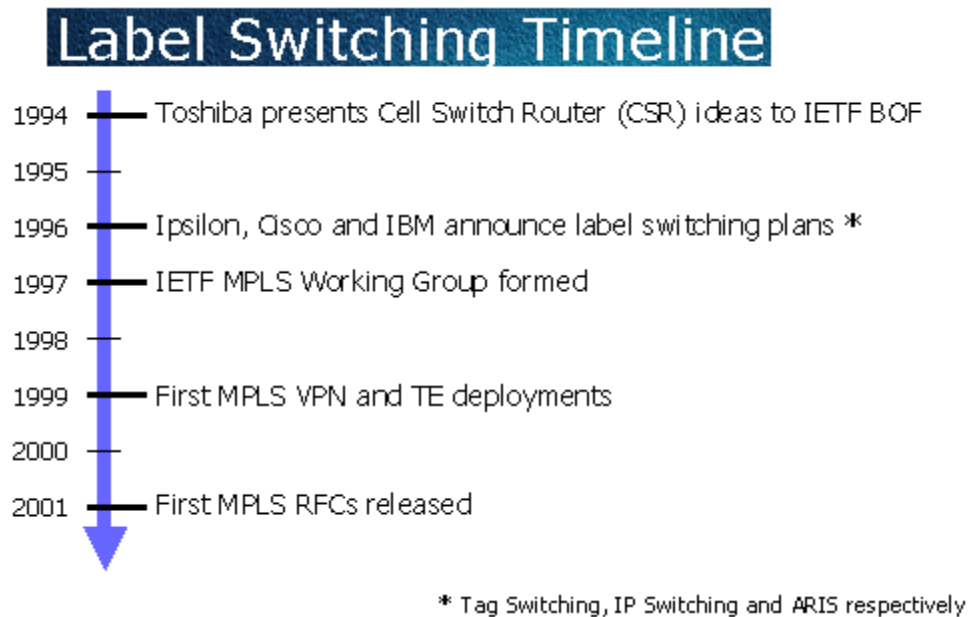


Figure 1.1 Label switching timeline (Evans, 2001)

To provide best solutions for voice, video, triple play, and data MPLS combines the speed and performance of Layer-2 packet-switched networks with the intelligence of Layer-3 circuit-switched networks. Before transferring information, MPLS establishes the end-to-end connection path, which can be selected according to the some needs such as bandwidth and maximum latency, like circuit-switched networks. Additionally, for improving link utilization MPLS provides that multiple applications and customers share a single connection, like packet networks.

Such as Frame Relay and Asynchronous Transfer Mode (ATM), different technologies were previously deployed with actually same aims. MPLS is now substituting these technologies, because it can meet the requirements of current and future technology. Especially, MPLS dispenses with the cell-switching and signaling-protocol baggage of ATM technology. MPLS recognizes that in the core of modern networks, small ATM cells are not demanded. As, modern optical networks are so fast at 10 Gbit/s and well beyond that even full-length 1500 byte packets do not incur real-time queuing delays (to support voice traffic, the need to reduce such delays, having been the motivation for the cell nature of ATM).

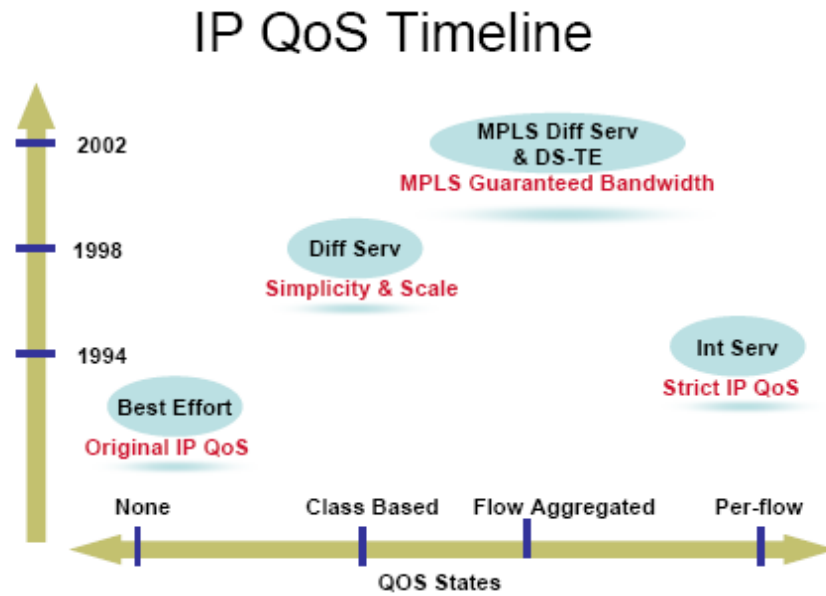


Figure 1.2 IP QoS timeline (Zhang & Ionescu, 2007)

During the past several years, numerous mechanisms have surfaced for providing QoS for communication networks as shown in Figure 1.2. The term of a QoS architecture started in the middle of 1990s. Thenceforth, the IETF has defined two QoS architecture named Integrated Services (IntServ) and Differentiated Services (DiffServ). The IntServ architecture was the initial solution. After, the DiffServ architecture was defined. MPLS later incorporated support for the DiffServ architecture, which the IETF had defined only for IP (Alvarez, 2006).

The IETF has defined the DiffServ architecture to provide QoS to the aggregated traffic flow (Blake, 1998; Grossman, 2002; Nichols, 1998). The DiffServ approach is based on a set of enhancements to the IP protocol, which enables scalable service discrimination in an IP network without the need for a per-flow state and signaling at every hop, which are characteristic of IntServ.

1.3 Literature Overview

Multiprotocol Label Switching (MPLS) is a protocol framework used to prioritize Internet traffic and improve bandwidth utilization (Alwayn, 2002). The Multi Protocol Label Switching (Rosen, 1999) architecture, originally presented as a way of improving the forwarding speed of routers, is now emerging as a crucial standard technology that offers new Quality of Service (QoS) capabilities for large-scale IP networks (Rouhana & Horlait, 2002).

There are numerous studies about MPLS network and implementing QoS through the MPLS network.

In one of these studies by Victoria Fineberg, Cheng Chen, XiPeng Xiao, “An end-to-end QoS architecture with the MPLS-based core”, (Fineberg, Chen, Xiao, 2002) a topology, which has two customer locations with Ethernet Local Area Networks (LAN) interconnected with MPLS core network of a service provider (SP) and indicates QoS mechanisms used in various parts, is presented. The article described various network technologies contributing to the end-to-end QoS, including those in the customer premises networks, in the service provider’s core, and the interworking between the LAN and service provider’s core mechanisms. It provided a particular emphasis on the MPLS mechanisms that allow to traffic engineer the core networks and, together with DiffServ, provide QoS guarantees in the network core.

In another study, by B. Kaarthick, N.Nagarajan, S.Rajeev, R.Joanna Angeline, “Improving QoS of Audio and Video packets in MPLS using Network Processors”, (Karthick, Nagarajan, Rajeev, Angeline, 2008), an effective solution for improving QoS of audio and video packets in MPLS networks under real time traffic conditions is presented. In the study, the impact of increased traffic on QoS parameters under heavy loading conditions is investigated and an efficient routing mechanism based on active networking concepts to satisfy QoS requirements of audio and video packets is proposed.

“Decreasing packet loss for QoS sensitive IP traffic in DiffServ enabled network using MPLS TE”, by Muhammad Tanvir, Abas Md Said (Tanvir, Said, 2010), the usefulness of applying Differentiated Services (DiffServ) and MPLS TE in the network to reduce packet drops for drop sensitive applications are demonstrated.

“Priority-Based Congestion Control in MPLS-based Networks”, by Scott Fowler and Sherali Zeadally (Fowler, Zeadally, 2005), a congestion control scheme between the receiving node and the ingress router for MPLS-based networks is proposed. An improvement in the number of packets delivered and better use of network resources is obtained with the simulation results.

“The QoS of the Edge Router Based on Diffserv/MPLS”, by Mao Pengxuan, Zhang Nan, Xiao Yang, Kiseon Kim (Pengxuan, Nan, Yang, Kim, 2009), a scheme based on DiffServ and MPLS that edge routers are responsible for marking and dropping packets is and the core routers are mainly responsible for forwarding the packets proposed. This study also illustrates its effectiveness by performing a simulation using Network Simulator (ns-2). The simulation results show that the proposed scheme can effectively alleviate the congestion of network and improve the QoS.

The QoS technologies offer SPs the means of providing enhanced services that set them apart from competition and make their operations more profitable. Ash (2001), defines QoS as a set of service requirements to be met by the network while transporting a connection or flow. To meet these service requirements, network operators must implement QoS resource management functions including Class of Service (CoS) identification, routing table derivation, connection admission, bandwidth allocation / protection / reservation, priority routing, and priority queuing.

DiffServ emerged as simpler solution to provide QoS as implementing IntServ and Resource Reservation Protocol (RSVP) was difficult (Xiao & Ni, 1999). The main goal of DiffServ (Sundaresan, 1999) was to meet the performance requirements of the user. Differentiated service mechanisms allow network providers to allocate

different levels of service to different users of the Internet. User needs to have Service Level Agreement (SLA) with Internet Service Provider to get DiffServ (Xiao & Ni, 1999). The Diffserv architecture (Blake, 1998) is composed of a number of small functional units implemented in the network nodes. This includes the definition of a set of Per-Hop Behaviors (PHBs), packet classification and traffic conditioning functions like metering, marking, shaping and policing. (Man, Xu, Li & Zhang, 2004)

So far several researchers have proposed many schemes in DiffServ-aware MPLS networks. Rouhana & Horlait (2000), showed how MPLS combined with differentiated services and constraint-based routing forms a simple and efficient Internet model capable of providing applications with differential QoS. No per-flow state information is required leading to increased scalability. They also proposed how this service architecture can interoperate with neighboring regions supporting IntServ and DiffServ QoS mechanisms. Saad, Yang, Makrakis & Groza (2001), combined DiffServ technology with traffic engineering over MPLS to offer an adaptive mechanism that is capable of routing high priority IP traffic over multiple parallel paths to meet delay time constraints. They propose a probe packet method to collect delay measurements along several parallel paths. They use them in an end-to-end delay predictor that outputs a quick current estimate of the end-to-end delay. Chpenst & Curran (2007), proposed network structure and the algorithm offer a solution that dynamically determines QoS-constrained routes with a number of demands and routes traffic within the network so that the demands are carried with the requisite QoS while fully utilizing network resources.

1.4 Thesis Outline

In this thesis, overview of MPLS, data flow in an MPLS network, its concepts, its operating mechanism and services are explained in Chapter two. Chapter three is related with QoS. QoS architectures, levels, functions are explained in the chapter three. Topology of the thesis work and analysis of the results are explained in Chapter four. Last chapter is the conclusion part.

CHAPTER TWO

MPLS (MULTI PROTOCOL LABEL SWITCHING)

2.1 What is MPLS?

MPLS is an Internet Engineering Task Force (IETF) standard and its' architecture is detailed in RFC 3031. MPLS is a Layer-2 switching technology, which enables packet switching at Layer-2 with Layer-3 forwarding information. It combines the high-performance capabilities of Layer-2 switching and the scalability of Layer-3 forwarding. In the MPLS network, routers add labels to packets, and can make forwarding decisions based on these labels. MPLS also reduces usage of CPU size on routers; by making the forwarding decisions based these labels instead of the analyzing the full routing table. At the ingress to the MPLS network, Internet Protocol (IP) precedence bits can be copied as Class of Service (CoS) bits, or can be mapped to set the proper MPLS CoS value in the MPLS Layer-2 label. MPLS CoS information is used to provide differentiated services within the MPLS network. Thence, MPLS CoS enables end-to-end IP Quality of Service (QoS) across an MPLS network.

Packet forwarding in MPLS network enables a service provider network to deploy new services, especially Virtual Private Networks (VPNs) and traffic engineering (TE). These features of MPLS will be mentioned at chapter four.

2.2 MPLS and IP

It is important to understand the differences between MPLS and traditional IP routing forward data across a network. In the traditional IP forwarding, the IP destination address in the packet's header is used to make an independent forwarding decision at each router in the network. These hop-by-hop decisions are basis on network layer routing protocols, like Open Shortest Path First (OSPF) or Border Gateway Protocol (BGP). These protocols provide to find the shortest path through

the network; they do not take into consideration factors, such as latency or congestion. MPLS creates a connection-based model; this connection-oriented architecture provides new possibilities for managing traffic on an IP network. MPLS combines the intelligence of routing with the high performance of switching, which is fundamental to the operation of the Internet and today's IP networks. Beyond its applicability to IP networking, MPLS is being expanded for more applications that are general.

2.3 Advantages of MPLS

Advantages of MPLS can be listed as follows:

- 1- MPLS provides a single integrated network to support both new and existing services and it creates an efficient migration path to an IP-based infrastructure.
- 2- MPLS operates over both existing, such as SONET and new infrastructure, such as 10/100/1000/10G Ethernet and networks, such as IP, ATM, Frame Relay, Ethernet, and TDM.
- 3- MPLS provides traffic engineering. Traffic engineering helps squeeze more data into available bandwidth.
- 4- MPLS supports the delivery of services with Quality of Service (QoS) guarantees. Packets can be marked to transmit as a high quality and provide low end-to-end latency for voice and video.
- 5- MPLS brings the speed and high performance of Layer 2 switching to Layer 3.
- 6- In the MPLS network, routers simply forward packets based on fixed labels, which reduce router processing requirements. MPLS helps carriers for scaling their networks as increasingly large routing tables become more complex to

manage. Transit routers do not need to handle complete routing tables anymore.

- 7- MPLS enables ATM service enhancements and new services. MPLS fixes the problems of IP over Asynchronous Transfer Mode (ATM), like complexity of control and management. MPLS also extends functionality of legacy ATM switches.
- 8- The ultimate benefit is a unified or converged network supporting all classes of service. MPLS provides differentiated performance levels and prioritization of delay-sensitive traffic and non-delay-sensitive traffic on a single network. MPLS addresses traffic management issues by prioritizing time sensitive applications.

2.4 MPLS Operating Mechanism

2.4.1 Basic Concepts of MPLS

2.4.1.1 MPLS Domain

It is an adjacent set of nodes, which operate MPLS routing and forwarding.

2.4.1.2 FEC (Forwarding Equivalent Class)

It is a group of IP packets, which are forwarded in the same manner; such as same destination, same forwarding path, and same class of service. A Forwarding Equivalent Class (FEC) is a collection of common actions conjoined with a class of packets.

The FEC associated with a Label Switched Path (LSP) denotes which packets are mapped to that LSP. LSPs are expanded through a network as each LSR appends incoming labels for a FEC to the outgoing label assigned to the next hop for the given FEC.

2.4.1.3 Labeled Packet

Labeled packet is a packet into that a label has been encoded. It is labeled by routers that are in the network.

2.4.1.4 Label Stack

Label stack is a group of labels that are carried by one labeled packet and organized as a Last-in, First-out (LIFO) stack.

2.4.1.5 LSR (Label Switching Router)

Label Switching Router (LSR) is an MPLS node, which is capable of forwarding Layer-3 packets. LSRs perform the label switching function. LSRs make different functions based on its position in a Label Switched Path (LSP). Routers in a LSP do one of the following below:

The name of the router at the beginning of an LSP is the ingress label edge router (ILER). The ingress router can encapsulate packets with an MPLS header and then transmit it to the next node along the path. A Label Switched Path can only have one ingress router.

A Label Switching Router (LSR) can be any medium router in the Label Switched Path (LSP) between the ingress and egress routers. An LSR swaps the incoming label with the outgoing MPLS label and forwards the MPLS packets to the next node in the LSP.

When an LSR assigns a label to a FEC, it must let other LSRs in the path know about the label. LDP helps to establish the LSP by enabling a set of procedures that LSRs can use to distribute labels.

2.4.1.6 Control Component

It is used to distribute label, to choose routing path, to compose forwarding table, to establish and release the LSP.

2.4.1.7 Forwarding Component

It is to forward-labeled packets based on the forwarding table.

2.4.1.8 LER (Label Edge Router)

Label Edge Router (LER) is an MPLS node that connects a MPLS domain that operates at the edge of an MPLS network. It uses routing information to determine appropriate labels to be added, labels the packet, and then forwards packets that labeled into the MPLS domain.

Likewise, upon receiving a labeled packet, which is destined to exit the MPLS domain, the LER removes the label and forwards the rest IP packet using normal IP forwarding rules.

2.4.1.9 LSP (Label Switched Path)

Label Switched Path (LSP) is a path, which through one or more Label Edge Router's at one level of the hierarchy followed by packets in a particular FEC. An LSP can have 0-253 transit routers.

The name of the router at the end of an LSP is the egress label edge router (ELER). The egress router removes the MPLS encapsulation info that changes from an MPLS packet to a data packet, and then it forwards the packet to its last destination with using information in the forwarding table. Each LSP can have only one egress router and the ingress and egress routers cannot be the same router in the

LSP. Nevertheless, the router can act as an ingress, egress, or transit router for one or more LSPs according to the network design.

There are some kinds of LSP types. One of them is the static LSPs. A static LSP specifies a static path. All routers that the LSP traverses must be configured manually with labels. No signaling like LDP or RSVP is required.

The other one is the signaled LSPs. A Signaled LSP is built with using a signaling protocol like Resource Reservation Protocol-Traffic Engineering (RSVP-TE) or Label Distribution Protocol (LDP). The signaling protocol lets labels to be assigned from an ingress router to the egress router. Signaling is initiated by the ingress routers. It is enough to configure only on the ingress router and is not required on intermediate routers. There are two signaled LSP types. The first one is explicit-path LSPs. RSVP-TE is used to set up explicit path LSPs by MPLS. Configuration is made manually in the hops within the LSP. Configuration must be as either strict or loose meaning which the LSP either must take a direct path from the previous hop router to this router or can traverse through other routers in the intermediate hops. The second is Constrained-path LSPs. In this type, the intermediate hops of the LSP are assigned as dynamically. A constrained path LSP based on the Constrained Shortest Path First (CSPF) routing algorithm to find a path, which satisfies the constraints for the LSP. Successively, CSPF based on the topology database provided by Open Shortest Path First (OSPF) or Intermediate System to Intermediate System Protocol (IS-IS). Once the CSPF found the path, and then RSVP uses the path to request the LSP set up. CSPF calculates the shortest path based on the limitations such as bandwidth, class of service, and specified hops.

2.4.1.10 LDP (Label Distribution Protocol)

Label Distribution Protocol (LDP) is a protocol, which delivers labels in non-traffic-engineered applications. LDP lets routers to establish LSPs through a network by mapping network-layer routing information directly to data link layer-switched paths. LDP lets an LSR to request a label from a downstream LSR so it can bind the

label to a specific FEC. The downstream LSR answers to the request from the upstream LSR by sending the requested label.

LDP signaling and MPLS label manager work together to manage the relationships between labels and the corresponding FEC. For service-based FECs, LDP works together with the Service Manager to identify the Virtual Leased Lines (VLLs) and Virtual Private LAN Services (VPLSs) to signal.

2.4.2 How MPLS Works?

MPLS is a technology that used to optimize traffic forwarding through a network. MPLS assigns labels to packets for transmitting across a network. The labels are contained in an MPLS header and added into the data packet (Figure 2.1).

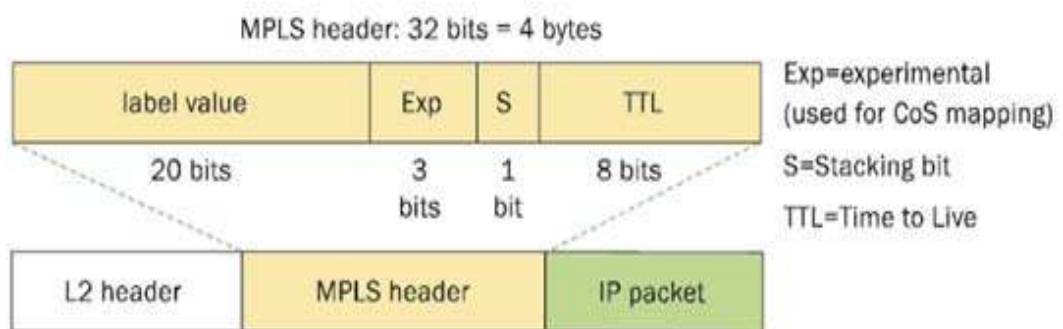


Figure 2.1 MPLS header format on an MPLS packet

MPLS Label Stack Encoding, the label stack is represented as a sequence of label stack entries as described in RFC 3032. Each label stack entry is represented by four octets. Figure 2.1 also shows the label placement in a packet.

Table 2.1 Packet/Label field description

Field	Description
Label	This 20-bit field carries the actual value (unstructured) of the label.
Exp	This 3-bit field is reserved for experimental use. It is currently used for Class of Service (CoS).
S	This bit is set to 1 for the last entry (bottom) in the label stack, and 0 for all other label stack entries.
TTL	This 8-bit field is used to encode a TTL value.

Table 2.1 shows the description of field of the MPLS header format. The 32-bit MPLS header contains the 20 bits label, which carries the actual value of the MPLS label. The Exp bit field, which is called also three bit CoS field, can affect the queuing and discard algorithms applied to the packet as it is forwarded through the network. The S field is a single bit stack field, which supports a hierarchical label stack. TTL field is an eight bits time-to-live (TTL) field, which provides conventional IP TTL functionality (see Figure 2.1) (Jolly & Latifi, IEEE, 2005).

These short, fixed-length labels, which added to data packet carry the information that tells each switching router how to process and forward the packets, from source to destination. They have meaning only on a local node-to-node connection. As each router forwards the packet, it swaps the current label for the appropriate label to transmit the packet to the next router. This system enables very-high-speed switching of the packets through the core MPLS network. MPLS defragments the best of both Layer-3 IP routing and Layer-2 switching. In actually, it is sometimes called a “Layer 2½” protocol. While routers need network-level intelligence to make a decision where to send traffic, switches only send data to the next hop, and so are inherently simpler, faster, and less costly. MPLS is based on traditional IP routing protocols to advertise and establish the network topology. MPLS is then overlaid on top of this topology. MPLS preconcert the path data takes across a network and encodes that information into a label that the network’s routers understand. This is the connection-oriented approach formerly mentioned. Since route planning occurs ahead of time and at the edge of the network; where the customer and service

provider network meet; MPLS-labeled data needs less router horsepower to traverse the core of the service provider's network.

2.4.2.1 MPLS Routing

MPLS networks build Label-Switched Paths (LSPs) for data crossing the network.

An LSP is described by a sequence of labels assigned to nodes on the packet's path from source to destination. LSPs can direct the packets in two ways. The way is hop-by-hop routing or explicit routing.

2.4.2.1.1 Hop-by-Hop Routing. The next hop is selected for a given Forwarding Equivalency Class (FEC) by each MPLS router independently in hop-by-hop routing. A FEC defines a group of packets, which are forwarded at the same way; all packets assigned to a FEC receive the same routing manner. FECs can be composed according to an IP address route or the service requirements for a packet, such as low latency.

MPLS uses the network topology information distributed by traditional Interior Gateway Protocols (IGPs), routing protocols like IS-IS or OSPF in the case of hop-by-hop routing. This is such like to traditional routing in IP networks, and the LSPs follow the routes the IGPs decision.

2.4.2.1.2 Explicit Routing. The whole list of nodes traversed by the LSP is designated in advance in explicit routing. The path designated could be optimal or not, but is based on the general view of the network topology and, potentially, on additional constraints. This is called Constraint-Based Routing. Resources may be reserved to ensure QoS along the path. This allows traffic engineering to be deployed in the network to optimize use of bandwidth.

2.4.2.2 Data Flow in an MPLS Network

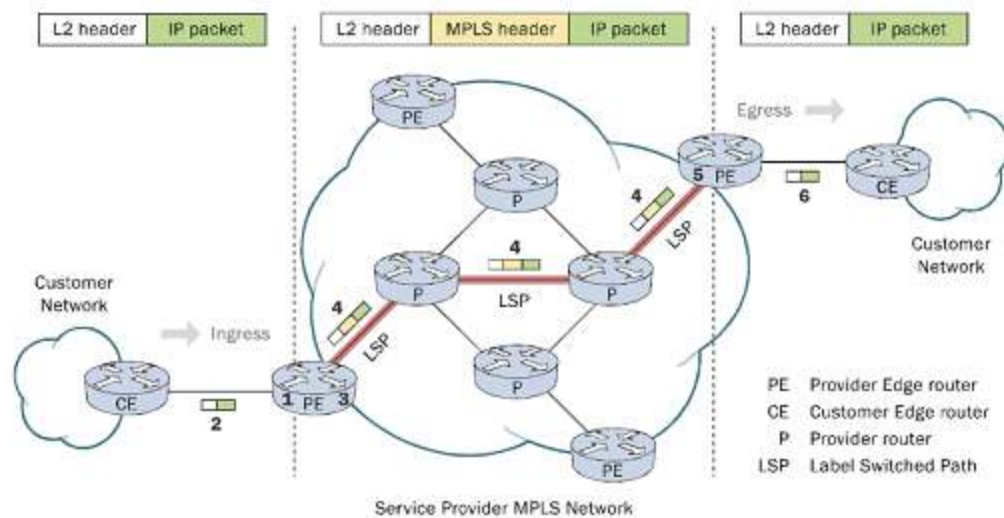


Figure 2.2 MPLS network

A typical MPLS network and its associated elements are shown in Figure 2.2. The cloud, which is in the central, represents the MPLS network itself. All data packets within this cloud are MPLS labeled. All traffic between the cloud and the customer side is not MPLS labeled (IP for example). The customer side router; which is named Customer Edge (CE) routers; interface with the service provider side; which is named as Service Provider Edge (PE) routers (also called Label Edge Routers, or LERs). At the ingress side that means incoming side of the MPLS network, PE routers add MPLS labels to packets. At the egress side that means outgoing side of the MPLS network, the PE routers remove the labels. Inside the MPLS cloud, Provider (P) routers; also named Label Switching Routers (LSR); switch traffic hop-by-hop based on the MPLS labels. In Figure 2.2, the flow of data through the MPLS network can be seen.

1. First of all, the PE routers establish LSPs through the MPLS network to remote PE routers before traffic is forwarded on the MPLS network.

2. From the Customer network non-MPLS traffic; such as Frame Relay, ATM, Ethernet is sent, through its CE router, to the ingress PE router, which operates at the edge of the provider's MPLS network.
3. The PE router looks up the information in the packet to associate it with a FEC, and then adds the suitable MPLS label(s) to the packet.
4. All intermediary P routers swap the labels as specified by the information in its Label Information Base (LIB) to forward the packet to the next hop along the LSP.
5. The final MPLS label is removed and the packet is forwarded by traditional routing mechanisms at the egress PE.
6. The packet is forwarded to the destination CE and into the customer's network.

2.5 MPLS Services

One of the primary aims of MPLS, boosting the performance of software-based IP routers, has been substituted as advances in silicon technology have enabled line-rate routing performance implemented in router hardware. Meanwhile, additional advantages of MPLS have been realized especially VPN services, traffic engineering and QoS.

2.5.1 Traffic Engineering (TE)

Traditional routing chooses the shortest path. That is why all traffic between the ingress and egress routers passes through the same links and this causes traffic congestion. LDP signaled paths only follow the IGP routing path. Traffic engineering lets a high degree of control over the path, which packets take, and it provides more efficient usage of the network resources. Traffic redirection is done through BGP or

IGP shortcut. Load balancing can be verified, resource utilization and network redundancy can be improved by TE.

Traffic engineering provides managing the flow of traffic through the network while optimizing use of network resources. At the same time, it also supports the network's customers and their QoS needs. In MPLS network, Traffic Engineering focuses on two aspects: traffic oriented objectives and resource-oriented objectives.

Traffic oriented objectives try for minimizing traffic loss, minimizing delay and jitter, maximizing throughput and providing to obey Service Level Agreements (SLA). Resource oriented objectives deal with the network resources such as link capacity, routers, available bandwidth etc.

To cope with the traffic volume increases, MPLS traffic engineering prefers to use existing bandwidth more efficiently by allowing packets to be routed along explicit routes and with specific bandwidth guarantees than adding bandwidth. This is known as Constraint-Based Routing. Constraint-Based Routing manages traffic paths within an MPLS network, allowing traffic to be directed to desired paths.

MPLS traffic engineering is typically located in the core of the MPLS network, while QoS is used at the edge. QoS at the edge ensures that high priority packets get preferred manner, while traffic engineering avoids traffic congestion and aptly utilizes available bandwidth resources. QoS and TE enable organizations to move away from multiple, specialized networks together for data, video, and voice to a single converged IP/MPLS network, significantly reducing overhead and cost.

2.5.1.1 TE Metric

The TE metric is a parameter that can be used to construct a TE topology, which is different from the IP topology. When the use of the TE metric is chosen for an LSP, the shortest path calculation after the TE constraints are applied will select an LSP path based on the TE metric instead of the IGP metric.

The TE metric is configured under the MPLS interface. Both the TE and IGP metrics are advertised by OSPF and IS-IS for each link in the network. The TE metric is an important part of the traffic engineering extensions of both IGP protocols.

An LSP consigned for real-time and delay sensitive user and control traffic has its path computed by CSPF using the TE metric. The TE metric is configured to represent the delay figure, or a combined delay/jitter figure, of the link. In this instance, the shortest path satisfying the constraints of the LSP path will effectively represent the shortest delay path.

2.5.2 Virtual Private Network (VPN)

A Virtual Private Network (VPN) is a private network service delivered over a public network. VPNs service provide to customers to allow remote locations securely connected over a public network, without the expense of buying or leasing dedicated network lines. MPLS enables VPNs by providing a circuit-like, connection-oriented framework, allowing carriers to deploy VPNs over the traditionally connectionless IP network infrastructure.

2.5.2.1 VPN Requirements

Opaque transport of data between VPN sites, because the customer may be using non-IP protocols or locally administered IP addresses that are not unique across the SP network. QoS guarantees to meet the business requirements of the customer in terms of bandwidth, availability and latency.

In addition, the management model for IP-based VPNs must be sufficiently flexible to allow either the customer or the SP to manage a VPN. In the case where an SP allows one or more customers to manage their own VPNs, the SP must ensure that the management tools provide security against the actions of one customer adversely affecting the level of service provided to other customers.

2.5.2.2 VPN Types

Brittain & Farrel (2004), define the VPN types as below.

2.5.2.2.1 Virtual Leased Lines (VLL). Conceptually, this is the easiest application of MPLS to VPNs. Each point-to-point VLL is provisioned as an LSP tunnel between the appropriate customer sites. VLL provide connection-oriented point-to-point links between customer sites. The customer perceives each VLL as a dedicated private (physical) link, although it is, in fact, provided by an IP tunnel across the backbone network. The IP tunneling protocol used over a VLL must be capable of carrying any protocol that the customer uses between the sites connected by that VLL.

2.5.2.2.2 Virtual Private LAN Segments (VPLS). VPLS provide an emulated LAN between the VPLS sites. As with VLLs, a VPLS VPN requires use of IP tunnels that are transparent to the protocols carried on the emulated LAN. The LAN may be emulated using a mesh of tunnels between the customer sites or by mapping each VPLS to a separate multicast IP address.

VPLS (Virtual Private LAN Services) is a multi-point L2 VPN model that has generated significant interest of late.

In Layer-2 VPNs, the PE and CE routers need not be routing peers as required in Layer-3 VPNs. Instead, only a Layer-2 connection needs to exist between PE and CE, with the PE routers simply switching incoming traffic into tunnels configured to one or more other PE routers. A Layer-2 MPLS VPN determines reachability through the data plane by using address learning, in contrast with Layer-3 VPNs, which determine reachability through the control plane by exchanging BGP routes. (Figure 2.3)

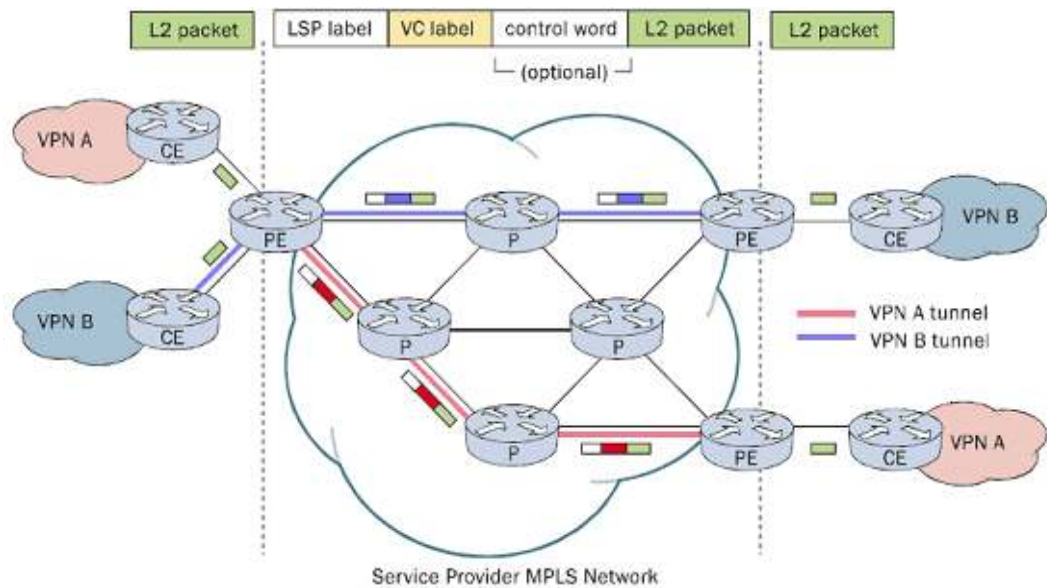


Figure 2.3 Layer 2 VPN MPLS network

2.5.2.2.3 Virtual Private Routed Networks (VPRNs). VPRNs emulate a dedicated IP-based routed network between the customer sites. Although a VPRN carries IP traffic, it must be treated as a separate routing domain from the underlying SP network, as the VPRN is likely to make use of non-unique customer assigned IP addresses. Each customer network perceives itself as operating in isolation and disjoint from the Internet. It is; therefore, free to assign IP addresses in whatever manner it likes. These addresses must not be advertised outside the VPRN since they cannot be guaranteed to be unique more widely than the VPN itself.

L3 VPNs use a two-level MPLS label stack (see Figure 2.4). The inner label carries VPN specific information from PE to PE. The outer label carries the hop-by-hop MPLS forwarding information. The P routers in the MPLS network only read and swap the outer label as the packet passes through the network. They do not read or act upon the inner VPN label — that information is tunneled across the network.

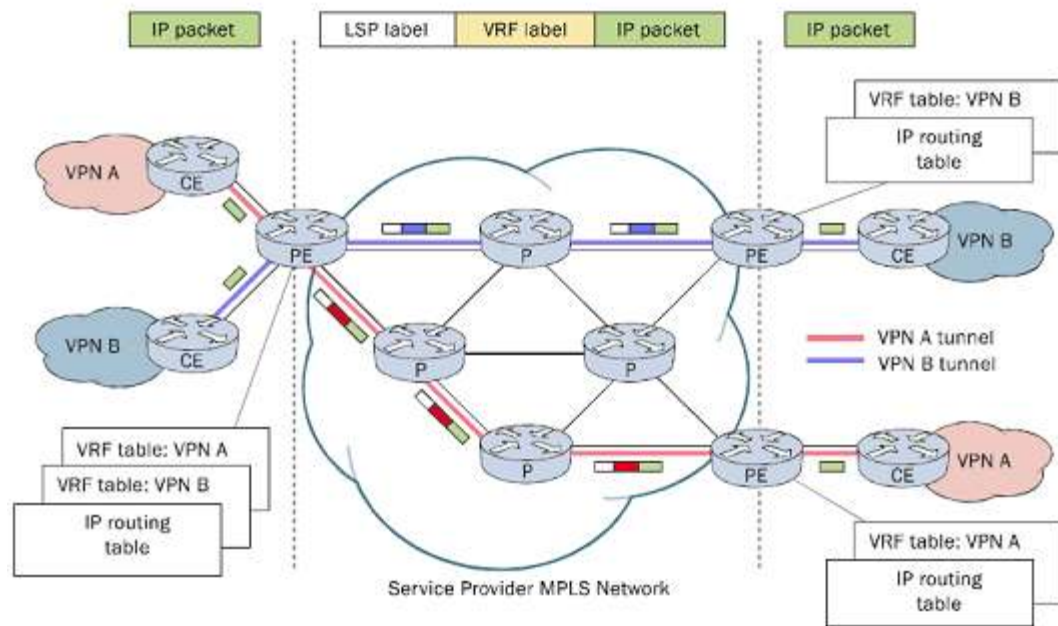


Figure 2.4 Layer 3 VPN MPLS network

2.5.2.2.4 Virtual Private Dial Networks (VPDNs). VPDNs allow customers to outsource to the SP the provisioning and management of dial-in access to their networks. Instead of each customer setting up their own access servers and using PPP sessions between a central location and remote users, the SP provides a shared, or very many shared access servers. PPP sessions for each VPDN are tunneled from the SP access server to an access point into each customer's network, known as the access concentrator. The last of these VPN types is providing a specialized form of access to a customer network. The IETF has specified the Layer 2 Tunneling Protocol (L2TP), which is explicitly designed to provide the authentication and multiplexing capabilities required for extending PPP sessions from a customer's L2TP.

2.5.3 QoS (Quality of Service)

QoS (Quality of Service) refers to resource-reservation-control mechanisms rather than the achieved service quality. QoS (Quality of Service) is the capability to enable different priority to different applications, or to ensure a definite level of performance to a data flow. For instance, a required jitter, packet dropping, bit rate,

delay, and/or bit error rate can be guaranteed. QoS guarantees are very consequential if the capacity of network is insufficient, notably for real-time streaming multimedia applications such as voice over IP, triple play applications, IP TV and online games. These applications require fixed bit rate and they are delay sensitive applications, so if networks that the capacity is a limited resource, QoS become critical for efficient transmitting.

QoS is described as the ability of a network to recognize different service demands of different application traffic flowing through it. It provides to comply with Service Level Agreements (SLA) negotiated for each of the application services, while trying to maximize the network resource utilization. QoS is certainly necessary in a multi-service network, in order to meet SLAs of different services and to maximize the network utilization. Without QoS, data is transmitted in a network on a first-in first-out (FIFO) basis, also can be said as best-effort service. In such a case, data is not assigned priority, based on the type of application that they support. As a result, different behaviors for different types of application traffic are not possible. Thence, Service Level Agreements (SLA) for any service cannot be met.

QoS will be explained deeply in Chapter three.

CHAPTER THREE

QoS (QUALITY of SERVICE)

QoS (Quality of Service) is a set of features in a service router that can help service providers provide service level agreements (SLA) for the different application transmitted over a multi-service network.

QoS (Quality of Service) is used for classifying and prioritizing the chosen traffic through a network. QoS enables establishing an end-to-end traffic priority policy to enhance control and throughput of important data. QoS provides to cope with available bandwidth so that the most important traffic is forwarded first. To supply this, when specifying the QoS, some factors such as latency, jitter, packet loss and throughput are taken into consideration.

This chapter introduces how the MPLS network can provide QoS and how the QoS information is propagated in MPLS networks. In succeeding sections, the two service models IntServ and DiffServ are described individually.

3.1 What Is Quality of Service?

Quality of Service (QoS) has become popular in the past few years because limited number of networks have unlimited bandwidth, this situation causes that congestion is always a possibility in the network. The increasing convergence of network services leads directly to the requirement for QoS, which means to give priority to important traffic over less important traffic and make sure it is delivered.

Quality of Service (QoS) refers to the ability of a network to provide better service to chosen network traffic over different technologies, including Asynchronous Transfer Mode (ATM), Ethernet, Frame Relay and 802.1 networks. The primary aim of QoS is to give priority including allocated bandwidth, controlled jitter and latency and improved loss characteristics. QoS provides the fabric building

blocks that can be used for future applications in campus, Wide Area Network (WAN), and service provider (SP) networks.

QoS provides to maximize network resource utilization by giving priority access to network bandwidth for chosen high priority traffic, and in the absence of high priority traffic, by enabling to gain the bandwidth committed to high priority traffic for low priority traffic.

Congestion avoidance and traffic prioritization are elemental considerations for QoS. Congestion is not desirable. Congestion avoidance provides enough capacity between source and destination. Traffic prioritization can be made by selecting certain traffic flows and then prioritize them, by using queuing to perform a gearbox on traffic, by using scheduling to empty the queues with taking into account the priority and congestion state.

A general QoS operational model is shown in Figure 3.1

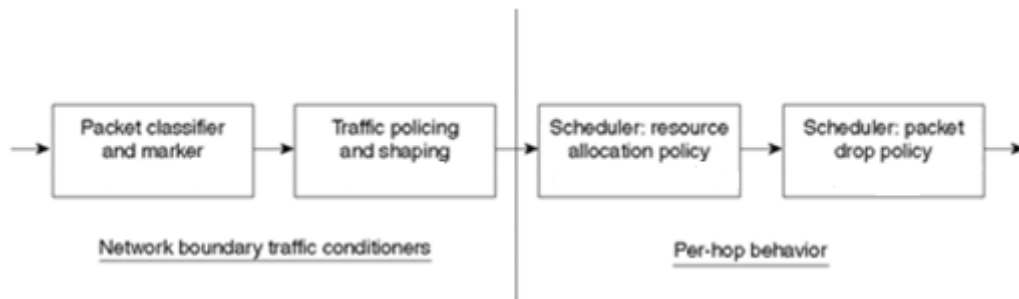


Figure 3.1 General QoS operational model (Álvarez, 2006)

3.2 Why QoS?

The difference between telephone companies, cable companies, and Internet service providers (ISP) is decreasing fast. All service providers are faced with raising needs to offer special services involving voice, video, and data application traffic. A service provider offering voice, video, and data application services using a single network infrastructure is commonly referred to as a triple-play service provider.

Congestion in the network is the main cause for Triple play to break down and this is one of the main area where QoS will make a difference. If there is congestion in the network, QoS is needed to transmit certain traffic as high priority traffic without latency, jitter or packet drop. Congestion state is shown in Figure 3.2.

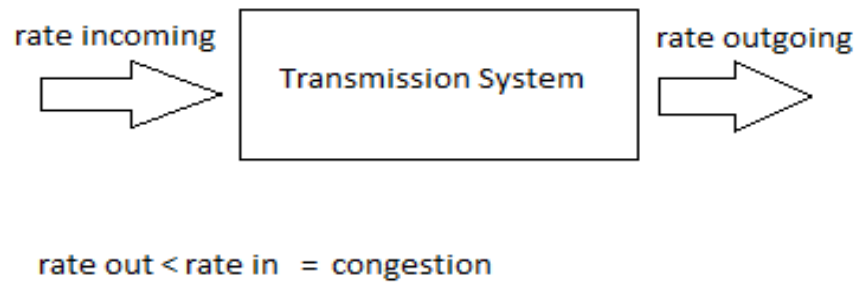


Figure 3.2 Congestion state

The Internet Engineering Task Force (IETF) has defined two architectures to implement QoS in an IP network: Integrated Services (IntServ) Architecture and Differentiated Services (DiffServ) Architecture. IntServ uses the signaling protocol, which name is Resource Reservation Protocol (RSVP). For the flows of traffic that hosts send, they signal to the network by way of RSVP what the QoS needs are. DiffServ uses the DiffServ bits in the IP header to qualify the IP packet. The routers look at these bits for marking, queuing, shaping, and setting the drop precedence of the packet. DiffServ model do not need any signaling protocol, thus DiffServ has a big advantage against to IntServ. The IntServ model uses a signaling protocol, which must run on the hosts and routers. If the network has too many flows, the routers must keep state information for each flow passing through it. This is an important scalability problem, which is why IntServ has not proven to be popular.

3.3 IntServ Architecture

The goal of IP Quality of Service (QoS) is to deliver guaranteed and differentiated services on the any IP based network. Guaranteed and differentiated services provide different levels of QoS, and each describes an architectural model for delivering QoS.

The Internet Engineering Task Force (IETF) creates the IntServ Working Group in 1994 to expand the Internet's service model to meet the requirements voice and video applications. Its' goal is to obviously describe the new improved Internet service model, and likewise to provide the average for applications to express end-to-end resource requirements with support mechanisms in routers and subnet technologies. It follows the aim of managing those flows individually, which requested specific QoS. Two services that guaranteed and controlled load are defined for this aim. Guaranteed service enables deterministic delay guarantees, in as much as; controlled load service enables a network service close to that provided by a best-effort network under lightly loaded conditions. (Postel, 1981)

The Intserv model needs per-flow guaranteed QoS on the Internet. The quantity of state information needed in the routers can be huge, with the thousands of flows, existing on the Internet today. This situation can cause scaling problems, as the state information increases as the number of flows increases. This makes Intserv hard to deploy on the Internet.

In Intserv, a Quality of Service (QoS) signaling protocol, Resource Reservation Protocol (RSVP) is used. RSVP is a QoS signaling protocol that enables end applications requiring definite guaranteed services to signal their end-to-end QoS demands to acquire service guarantees from the network.

3.3.1 RSVP (Resource Reservation Protocol)

Resource Reservation Protocol (RSVP) is an IETF Internet standard (RFC 2205) protocol for permitting an application to reserve network bandwidth dynamically. RSVP provides applications to ask for a certain QoS for a data flow, as shown in Figure 3.3.

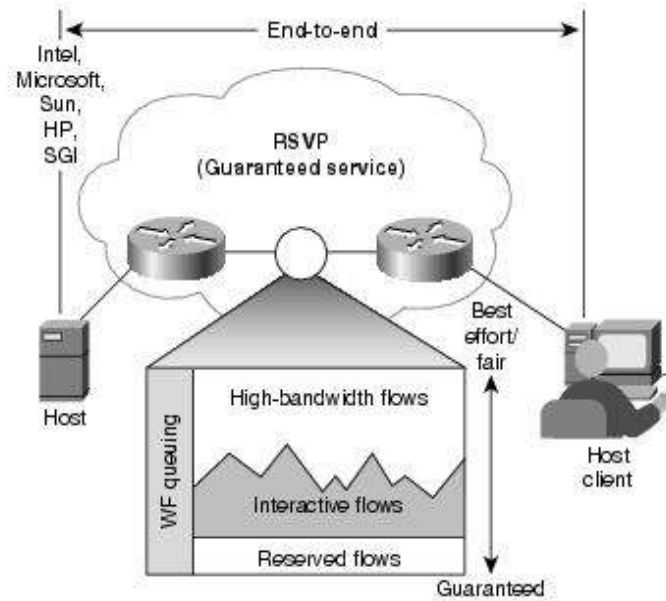


Figure 3.3 RSVP implemented in a network

IntServ can use RSVP as the reservation setup protocol. One of the principles of this architecture is that applications communicate QoS needs for individual flows to the network. These needs are used for resource reservation and admission control. RSVP can perform this. Anyway, RSVP is often but incorrectly equated to IntServ. RSVP and IntServ share a common history, but they are eventually independent. Two different working groups at the IETF developed their specifications. RSVP has suitability as a signaling protocol outside IntServ. Correspondingly, IntServ could use other signaling mechanisms.

RSVP enables applications to signal per-flow QoS requirements to the network. Service parameters are used to determine quantity particularly for admission control.

RSVP is used in multicast applications such as audio, video conferencing and broadcasting. Despite the fact that, the initial objective for RSVP is multimedia traffic, there is an obvious interest in reserving bandwidth for unicast traffic such as Network File System (NFS), and for Virtual Private Network (VPN) management.

3.4 DiffServ Architecture

In 1998, the DiffServ Working Group was designed under IETF. The working group was formed to present architecture with a simple QoS approach, which could be applied to both IPv4 and IPv6. DiffServ is a bridge between IntServ guaranteed QoS requirements and the best effort service presented by the Internet today. By classifying traffic into classes, DiffServ enables traffic differentiation with relative service priority among the traffic classes.

The DiffServ architecture depends on the interpretation of classes of traffic with different service demands. The traffic classification is captured by marking in the packet header. Further network nodes examine this marking to determine the packet class and allocate network resources in accord with locally defined service policies. The service characteristics are unidirectional with a qualitative definition in terms of latency, jitter, and loss. DiffServ nodes have no knowledge of individual flows and they are stateless from a QoS point of view. Relatively few packet markings are possible regarding to the number of micro flows, which a node may be switching at a given point in time. Anyway, the concept of aggregating traffic into a small number of classes is intrinsic to DiffServ. The architecture deliberately makes a trade off between granularity and scalability. RFC 2475 introduces the architecture.

The DiffServ architecture provides a structure within, which enables to suggest a range of network services based on performance. A desirable performance level can be chosen by marking the packet's Differentiated Services Code Point (DSCP) field to a specific value. This specific value clearly described the Per-Hop Behaviors (PHB) given to the packet within the service provider network. Typically, the service provider and customer compromise a profile defining the rate at which traffic can be submitted at each service level. Packets submitted in overabundance of the concerted profile might not be allotted the requested service level.

The DiffServ architecture specifies the basic mechanisms. By using these mechanisms as building blocks, a sort of services can be build. A service defines

some important characteristic of transmission in a network, like packet loss, jitter, throughput and delay. In addition, a service can be characterized in terms of the relative priority of access to resources in a network. PHB is specified on all the network nodes of the network offering this service, after a service is defined, also DSCP is assigned to the PHB. A PHB is a forwarding behavior given by a network node to all packets carrying a specific DSCP value. The associated DSCP field in its packets is carried by the traffic demanding a specific service level.

The PHB, based on the DSCP field in the packet, is observed by the all nodes in the DiffServ domain. Additionally, the network nodes, which are on the DiffServ domain's boundary, carry the significant function of conditioning the traffic entering the domain. Traffic conditioning includes functions like packet classification and traffic policing. Traffic conditioning is very important in engineering traffic carried within a DiffServ domain, such that the network can observe the PHB for all its traffic entering the domain.

The DiffServ architecture is illustrated in Figure 3.4.

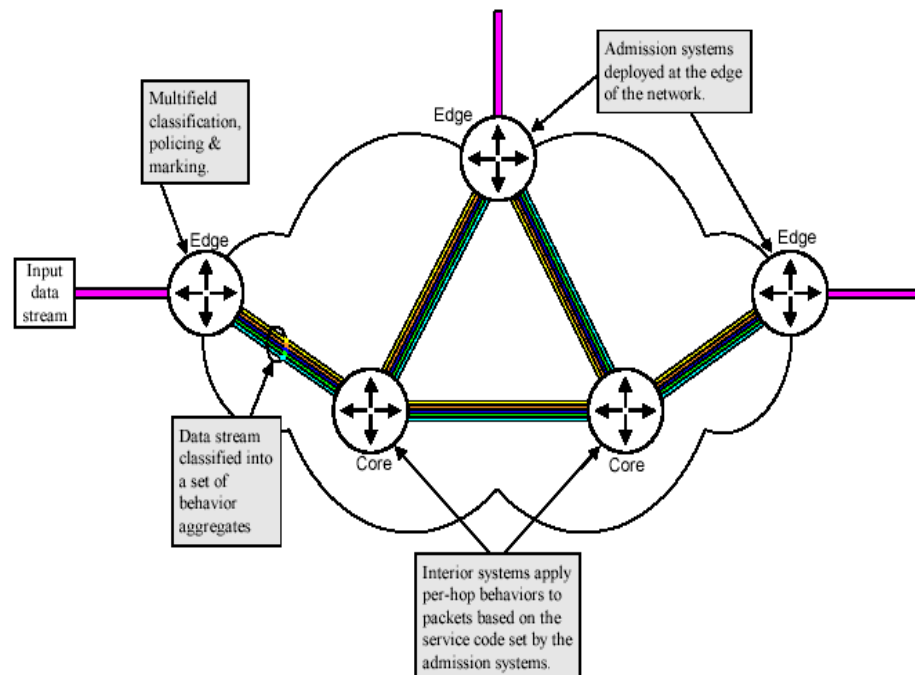


Figure 3.4 DiffServ architecture (Bingöl, 2005)

3.4.1 DiffServ Terminology

The DiffServ architecture introduces many new terms. RFC 2475 and RFC 3260 introduce the list of these terms.

Domain: It is a network with a common DiffServ implementation.

Region: It is a group of contiguous DiffServ domains.

Egress node: It is the last node traversed by a packet before leaving a DiffServ domain.

Ingress node: It is the first node traversed by a packet before entering a DiffServ domain.

Interior node: It is a node in a DiffServ domain, which is not an egress or ingress node.

DiffServ field: It is the header field where packets carry their DiffServ marking. This field corresponds to the six most significant bits of the second byte in the IP header.

DSCP: It is a specific value assigned to the DiffServ field.

Behavior aggregate (BA): It is a collection of packets traversing a DiffServ node with the same DSCP.

Per-hop behavior (PHB): It is a forwarding behavior or service that a BA receives at a node.

Traffic profile: It is description of a traffic pattern over time, generally, in terms of a token bucket.

Marking: It means that setting the DSCP in a packet.

Metering: It means that measuring of a traffic profile over time.

Policing: It means that discarding of packet to enforce conformance to a traffic profile.

Shaping: It means that buffering of packets to enforce conformance to a traffic profile.

Traffic conditioning: It is the process of enforcing a traffic conditioning specification through control functions such as marking, metering, policing, and shaping.

3.5 Differentiated Services Code Point (DSCP)

Differentiated Services Code Point (DSCP) is a value that is assigned to each packet entering a DiffServ domain. The assigned value is written into the DS field in the packet header. For IPv4, the DSCP field is the six most significant bits of the Type of Service (ToS) field in the IP header.

To distinguish classes from each other, some way is needed to understand different characteristics of the packets and generalize them into a set of classes. Inside a DS domain, many individual application-to-application flows share a certain DSCP. The core routers interest only Behavior Aggregate (BAs) instead of particular flows, because the collection of packets sharing a DSCP is referred to as one BA.

The DSCP can be used to identify 64 different BAs, and IETF defines a small set of standard DSCPs for ability to work together among different DS domains. However, a DS domain is free to use non-standard DSCPs inside the domain as long as packets are remarked when they leave the DS domain.

3.6 IP Precedence: Differentiated QoS

The three precedence bits in the IPv4 header's Type of Service (ToS) field are utilized to specify class of service for each packet. The IP Precedence ToS Field in an IP Packet Header is shown in Figure 3.5. IP precedence provides partition traffic to be up six classes of service, the other two are reserved for internal network use. This signal can be used to provide the suitable expedited handling throughout the network by the queuing technologies.

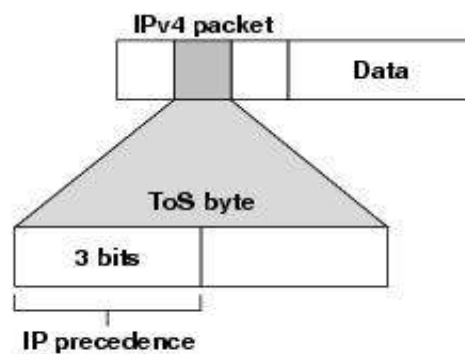


Figure 3.5 IP precedence ToS field

By setting the IP precedence bits, traffic that is identified can be marked. Therefore, it requires to be classified just once.

3.7 Per-Hop Behaviors (PHB)

The DiffServ architecture describes a PHB as the forwarding behavior. It represents a qualitative description of the latency, jitter, or loss characteristics. The PHB explanation does not quantify these characteristics. A PHB group contains one or more related PHBs that are performed concurrently. Packets are mapped to PHBs in respect to their DSCP by DiffServ nodes. These DSCP-to-PHB mappings are not mandated. It provides the flexibility to configure arbitrary mappings if desired. The architecture defines the class selectors for backward compatibility with the use of the Precedence field in the IPv4 TOS octet, so they are the only exception. DiffServ

domains, which are not using the recommended mappings, are more likely to have to remark traffic when interfacing with other DiffServ domains. PHB groups are part of the current DiffServ specifications: Expedited Forwarding (EF), Assured Forwarding (AF1, AF2, AF3, and AF4), Class Selector (CS), and Default. A node may support multiple PHB groups concurrently.

3.7.1 Expedited Forwarding

The Expedited Forwarding (EF) describes a low-latency, low-jitter, low-loss behavior, which a DiffServ node may implement. This PHB performs like a building block for the transport of real-time traffic over a DiffServ domain. Free from the amount of non-EF traffic, a DiffServ node must serve EF traffic at a higher rate than its arrival rate to support this behavior. This difference between the EF arrival and service rate helps guarantee that EF traffic encounters empty or near empty queues, which reduces the queuing latency and jitter during normal node operation. Reducing of queuing latency provides low latency, low jitter and low loss by preventing exhaustion of packets buffers. RFC 3246 and RFC 3247 describe and discuss this PHB in detail.

3.7.2 Assured Forwarding

The Assured Forwarding (AF) describes four different levels of forwarding guarantee, which a DiffServ node may support. Simply, it describes how a DiffServ node may support different packet-loss guarantees. The AF PHB groups are named as AF1, AF2, AF3, and AF4. Three drop precedence levels are supported by each of these groups . If the group exhausts its allocated resources such as bandwidth and buffers, DiffServ node will drop the packet when it has the higher the drop precedence.

3.8 MPLS Support for DiffServ

MPLS supports DiffServ with least possible adjustments to the MPLS and DiffServ architectures. The traffic conditioning and PHB concepts, which were described in DiffServ, are not introduced any modification. The same traffic management mechanisms are used such as metering, marking, shaping, policing, queuing to condition and implement the different PHBs for MPLS traffic by a Label Switching Router (LSR). Traffic engineering can be used to complement its DiffServ implementation. RFC 3270 describes MPLS support for the DiffServ architecture. DiffServ may be implemented to support a different range of QoS needs and services in a scalable manner. MPLS DiffServ is not specific to the transmitting of IP traffic over an MPLS network. An MPLS DiffServ implementation is interested in with supporting the PHBs that can satisfy the QoS needs of all types of traffic, which is carried. These characteristics are very important for the implementation of large MPLS networks, which can transport a wide range of traffic.

3.9 End-to-End QoS Levels

Service levels refer to the end-to-end QoS capabilities, which mean the capability of a network to deliver service demanded by specific network traffic from end to end. The services differ in their level of QoS that explains how tightly the service can be bound by delay, specific bandwidth, loss characteristics, and jitter.

As shown in Figure 3.6, three fundamental levels of end-to-end QoS can be provided across a heterogeneous network.

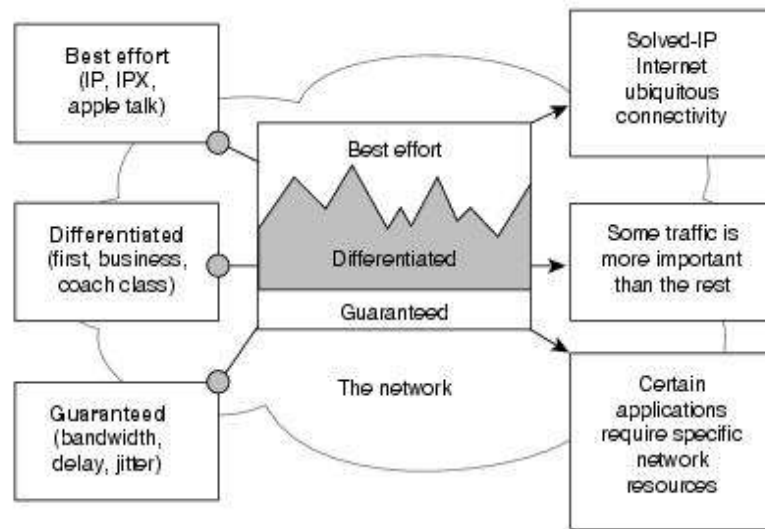


Figure 3.6 The three levels of end-to-end QoS

3.9.1 Best-Effort Service

It is a basic connectivity without delivery guarantee. When the router input or output buffer queues are exhausted, the packet is commonly dropped. Best-Effort service has no service or delivery guarantees while forwarding best-effort traffic; it is not really a part of QoS. It is just a service that Internet offers today. Most data applications, such as File Transfer Protocol (FTP), are forwarded with Best-Effort service with degraded performance. All applications need definite network resource allocations in terms of bandwidth, delay, and minimal packet loss to function well.

3.9.2 Differentiated Service

In Differentiated Service, based on service demands, traffic is grouped into classes. The network differentiates each traffic class and services according to the configured QoS mechanisms for the class. This design for delivering QoS is often referred to as CoS (Class of Service). Differentiated Service does not give service guarantees. It only differentiates traffic and provides a preferential treatment of one traffic class over the other one. Therefore, this service is also referred as soft QoS. For the bandwidth-intensive data applications, this QoS scheme is very suitable.

Network control traffic is prioritized and differentiated from the rest of the data traffic to guarantee fundamental network connectivity all the time.

3.9.3 *Guaranteed Service*

Guaranteed Service needs network resource reservation to guarantee that traffic flow's specific service requirements are met by the network. Prior network resource reservation is required over the connection path by the Guaranteed Service. Rigid guarantees are required from the network, so Guaranteed Service is referred to as hard QoS. With a granularity of a single flow, path reservations do not scale over the Internet backbone. Aggregate reservations should be a scalable means of offering this service. Like audio and video multimedia applications are included by applications requiring such service. For the interactive voice applications, which are transmitted over the Internet, it is needed to limit latency to 100 ms to satisfy human ergonomic needs. This latency is suitable for large spectrum of multimedia applications. For example, internet telephones need at a minimum an 8-Kbps bandwidth and a 100- ms round-trip delay. Resources are needed to reserve to meet such guaranteed service requirements by the network.

Service levels and enabling QoS functions are shown in Table 3.1.

Table 3.1 Service levels and enabling QoS functions

Service Levels	Enabling Layer 3 QoS	Enabling Layer 2 QoS
Best-effort	Basic connectivity	Asynchronous Transfer Mode (ATM), Unspecified Bit Rate (UBR), Frame Relay Committed Information Rate (CIR)=0
Differentiated	CoS Committed Access Rate (CAR), Weighted Fair Queuing (WFQ), Weighted Random Early Detection (WRED)	IEEE 802.1p
Guaranteed	Resource Reservation Protocol (RSVP)	Subnet Bandwidth Manager (SBM), ATM Constant Bit Rate (CBR), Frame Relay CIR

3.10 QoS Functions

There are some basic functions for QoS implementation and these functions are mentioned in the following sections. (Figure 3.7)

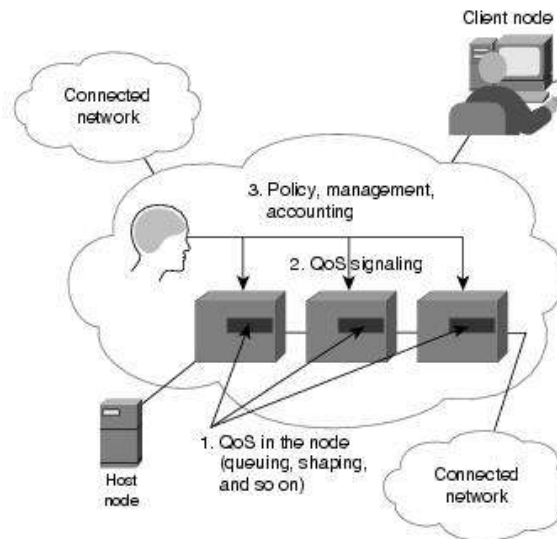


Figure 3.7 Components of a basic QoS implementation

3.10.1 Classification

Firstly, the traffic must be identified, for providing preferential service to it. Second, the packet may be marked or may not be. These two procedures compose classification. The identification process can range from simple to complex. The different classification can be made by identification based on IP protocol field, Source IP Address, Destination IP Address, Source Port Number, and Destination Port number, IP Precedence or DSCP field and source and destination Media Access Control (MAC) addresses.

Classification has effect on policing, marking, queuing and scheduling process (Figure 3.8). The suitable handling for the packets can be chosen by the router, with the classification as a basis.

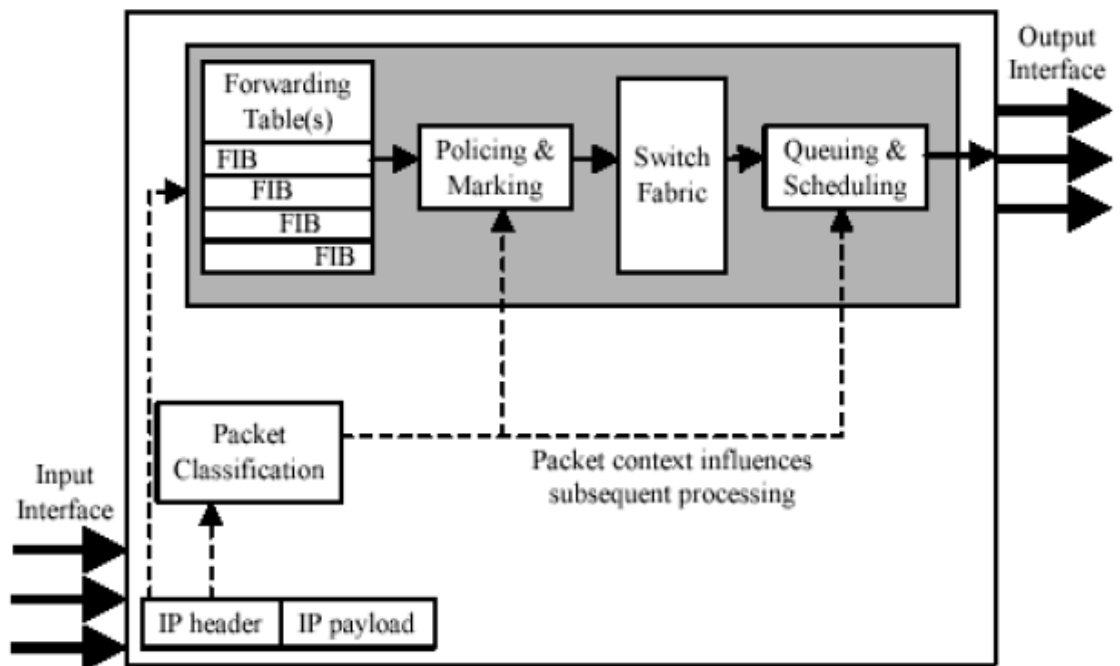


Figure 3.8 An overview to how classification influences forwarding (Bingöl, 2005)

3.10.2 Marking

Packet marking includes assigning a new value to the QoS field in the header of the packet. The packet is associated with a class or a drop precedence by marking. To indicate the PHB for each packet, the DiffServ architecture depends on packet marking. Various Layer-2 technologies utilize the packet marking for QoS purposes, for example Ethernet uses a 3-bit priority field in the VLAN header, ATM uses a 1-bit field to mark the drop precedence of a cell, Frame Relay uses an equivalent 1-bit field to manage the drop precedence of a frame.

3.10.3 Policing

Traffic policing is used usually for rate control. The amount of a particular traffic stream might be needed to control by a network node in different cases. The traffic is measured and then the measurement is compared with a predefined traffic profile by a policer. According to the comparison result, the action that the policer takes on the packet is determined. Transmitting, marking, or dropping the packet are the main

actions. The marking action shows that the packet will be transmitted after the node marks it. Policing is necessary for the traffic conditioning function in DiffServ but it is not exclusive for this architecture. Traffic policing is used by many technologies such as ATM and Frame Relay. Generally, traffic policing is a common mechanism at boundaries between administrative domains.

Policing has common qualities with shaping, but it is different from shaping in one important way that traffic, which exceeds the configured rate, is discarded. It is not buffered.

3.10.4 Shaping

For rate control, shaping is commonly used like policing. Traffic is measured and then the measurement with a profile is compared by a shaper, similar to a policer. In this situation, according to the comparison result, the shaper action is determined. The shaper can delay the packet or permit further processing. Hence, shaping needs the buffering or queuing of packets, which exceed the profile. If the traffic stream largely exceeds the profile, shaping might result in packet loss. If the stream never exceeds the profile, shaping might not smooth the traffic. Shaping is also necessary to traffic conditioning in DiffServ. Traffic shaping operation is shown in Figure 3.9.

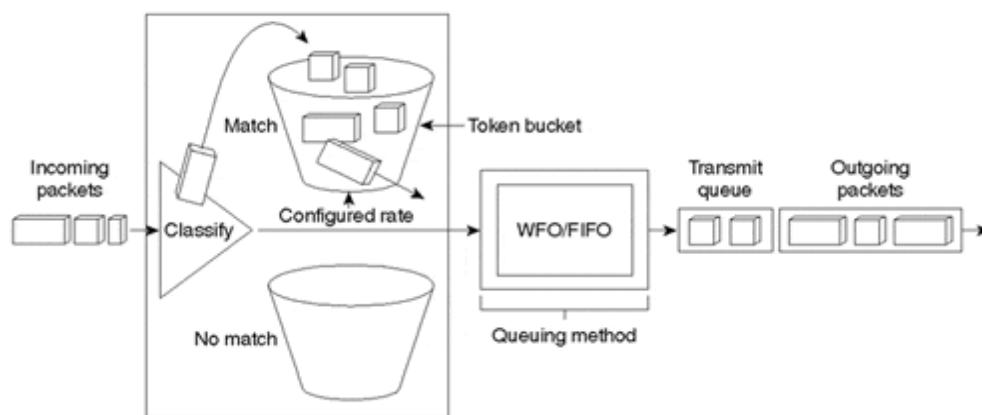


Figure 3.9 Traffic shaping operation

A token bucket is used to measure traffic to classify a packet that it is conforming or nonconforming by traffic shaping. The sum of conformed burst size, BC and the extended burst size, BE is equal to the maximum size of the token bucket. Tokens, which is equal to BC , are added to the bucket every measuring interval T , where T is equal to the division of BC to CIR ($T = BC / CIR$). CIR (Committed Information Rate) is the allowed average rate of traffic flow. Any added tokens overflow, if the bucket becomes full. The procedure is like that, when a packet reaches, the token bucket is controlled to see if enough tokens are available to send the packet. The packet is marked compliant, if enough tokens are available and then number of the tokens that is equal to the packet size are removed from the bucket. The packet is marked non-compliant, and is queued for later transmission, if enough tokens are not available. Traffic shaping token bucket is described in Figure 3.10.

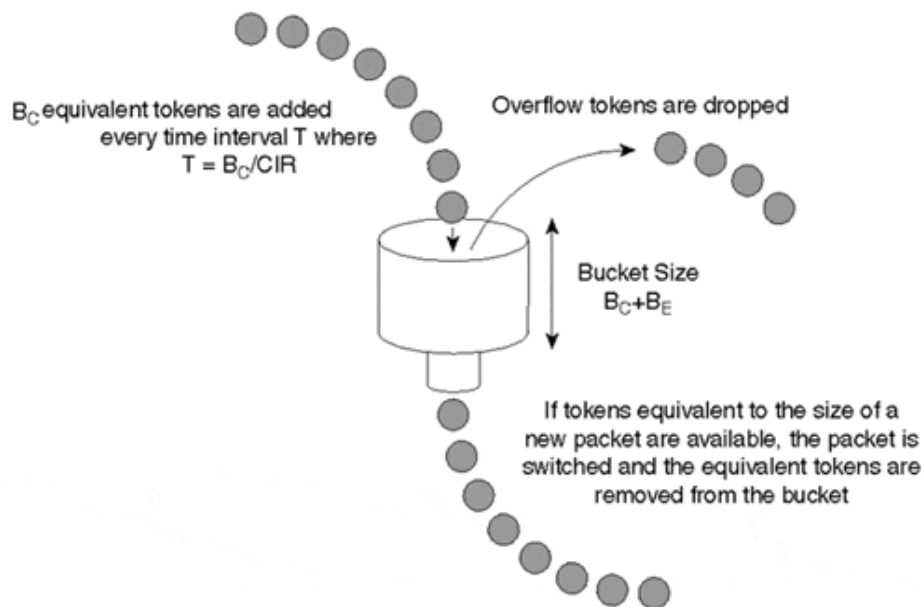


Figure 3.10 The token bucket scheme for the traffic shaping function

3.10.5 Queuing and Scheduling

The manner of the packets while passing on to the output queue is controlled by queuing function. Closely related scheduling mechanisms are needed by nodes to guarantee that different connections obtain their promised share of the resources and it guarantees that any spare capacity is delivered in a fair manner.

The order in which the packets in the queue are serviced is determined by the scheduling mechanism on a router, when some network congestion exists. The packet, which goes next from a queue, is decided by the scheduling. FIFO (first-in, first-out) scheduling is the traditional packet scheduling mechanism on the Internet.

The scheduling algorithm needs to be able to differentiate among the different packets in the queue and know each packet's service level to deliver QoS. By allocating resources on flow basis and by prioritizing one flow according to the others, scheduling algorithm enables guarantees on performance bounds.

Moreover, to provide fairness and protection among the flows with the same priority, such as all best-effort traffic flows, the scheduling algorithm is needed.

3.10.5.1 FIFO (first-in, first-out)

FIFO (first-in, first-out) queuing provides storing packets when the network is congested and when the network is no longer congested, forwarding them in order of arrival. (Figure 3.11)



Figure 3.11 FIFO queue

FIFO is used for default queuing algorithm in some services. Although it does not require configuration, it has several shortcomings. Most importantly, FIFO queuing does not decide about packet priority; parameters such as bandwidth, promptness, and buffer allocation are determined by the order of packets arrival.

In FIFO queuing, all the bandwidth of a link can be consumed when a host starts a file transfer. This causes disadvantage of real-time traffic and it is an important problem with FIFO queuing. One source sends a “train” of packets to its destination and packets that send from other hosts are got behind the train, so the phenomenon is referred to as packet trains. For large links, which have little delay and minimal congestion, FIFO queuing can be efficient.

3.10.5.2 WFQ (Weighted Fair Queuing)

WFQ (Weighted Fair Queuing) is the best known and the most studied queuing method. In WFQ, a queue is assigned for each flow and to decide how much bandwidth for each flow is allowed relative to other, priority is applied to identified traffic. WFQ assigns a weight to each flow or traffic class, and the rate, which a flow or a traffic class is serviced, is proportional to its assigned weight. Prioritization among unequally weighted traffic flows, fairness and protection among equally weighted traffic flows are provided by WFQ. Due to that, WFQ can prevent other flows to have direct effect on one certain flow.

Packets are reordered and this provides that low-volume flows are moved forward and high-volume flows are moved towards the tail of the queue, in WFQ (Figure 3.12). This reordering also results to break up packet trains and low-volume flows receive preferential service.

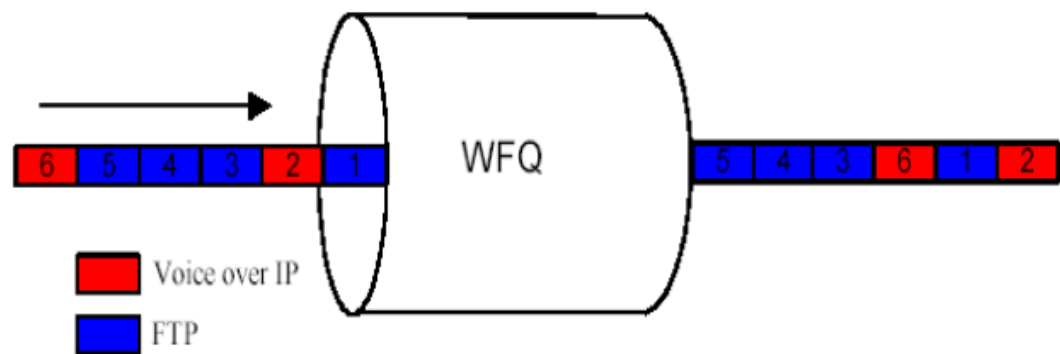


Figure 3.12 An example on how WFQ reorders packets

The problem of round-trip delay variability also is addressed by the WFQ algorithm. It makes multiple high volume transfer rates of conversation and interarrival periods much more predictable, if they are active. Delay variation or jitter stabilizes, if conversations are serviced in a consistent manner with every round-robin approach. (Figure 3.13)

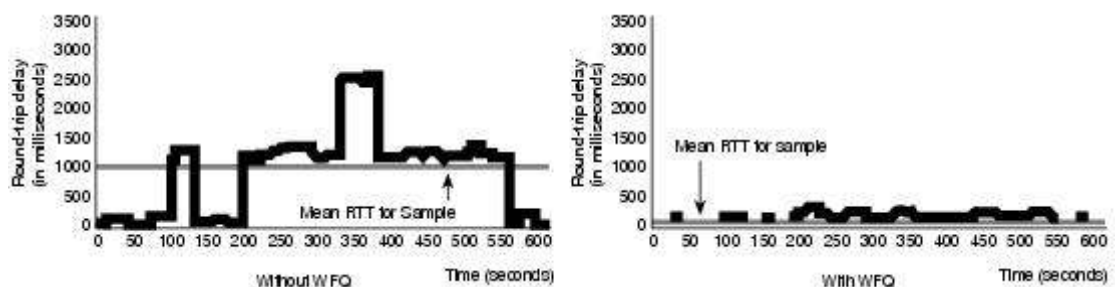


Figure 3.13 An example of interactive traffic delay

3.10.5.3 PQ (Priority Queuing)

PQ guarantees that important traffic is handled fastestly at each point where it is used. Based on input interface, IP access lists, packet size, and application, packets are classified into priority queues by PQ. It was designed to give strict priority to important traffic. In order of decreasing priority, output subqueues named high, medium, normal, and low are maintained by PQ (Figure 3.14). In PQ, packets, which are on the highest priority queue are forwarded firstly. Packets, which are on the next

highest priority queue, are forwarded when the highest priority queue empties and this continues. While packets in the high priority queue are waiting for service, packets in the medium-priority queue are not serviced. PQ algorithm gives to higher-priority queues a definite preferential treatment over low-priority queues, during sending packets.

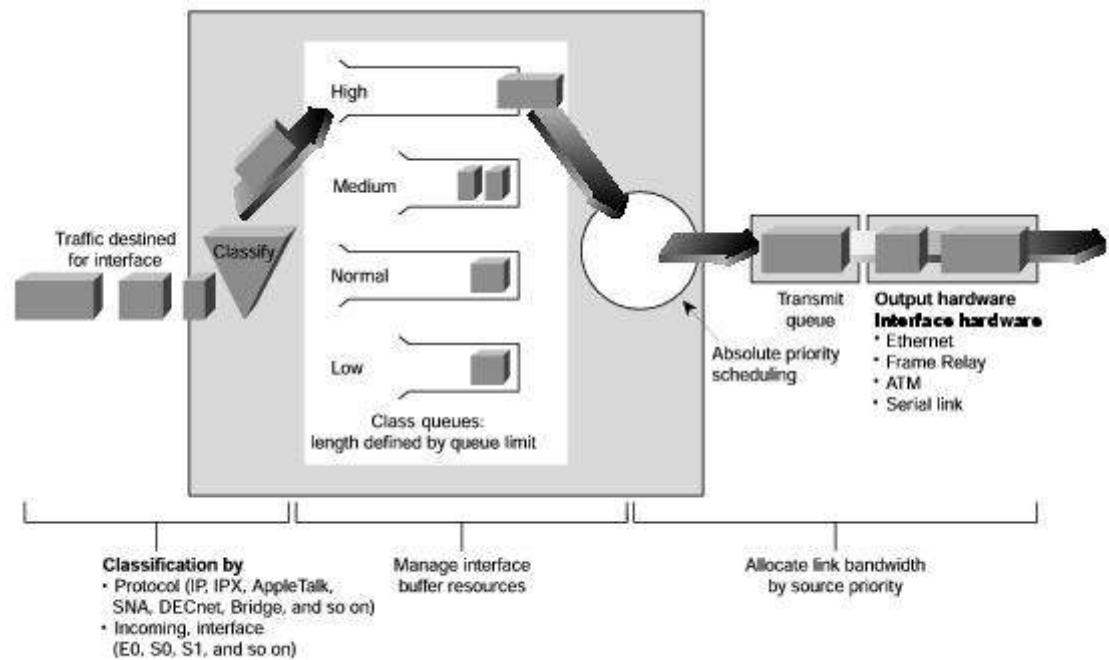


Figure 3.14 Priority queuing places data into four levels of queues

3.10.5.4 CQ (Custom Queuing)

CQ was designed to enable different applications to share the network among applications with latency needs or specific minimum bandwidth. Between users and applications, bandwidth must be allocated proportionally in these conditions. Traffic is handled by assigning a definite quantity of queue space to each class of packets and then servicing the queues in a round-robin fashion by CQ. (Figure 3.15)

Each nonempty queue is serviced sequentially in a round-robin fashion; a configurable percentage of traffic is transmitted on each queue by this bandwidth reservation method. While assuring predictable throughput for other traffic, CQ

ensures that mission-critical data is always assigned a definite percentage of the bandwidth.

Ensuring a certain bandwidth to a set of places selected by an access list is one of the popular use of CQ. The byte count must be specified for each queue to allocate bandwidth to different queues.

CQ is configured statically and cannot adapt to changing network conditions automatically, like PQ.

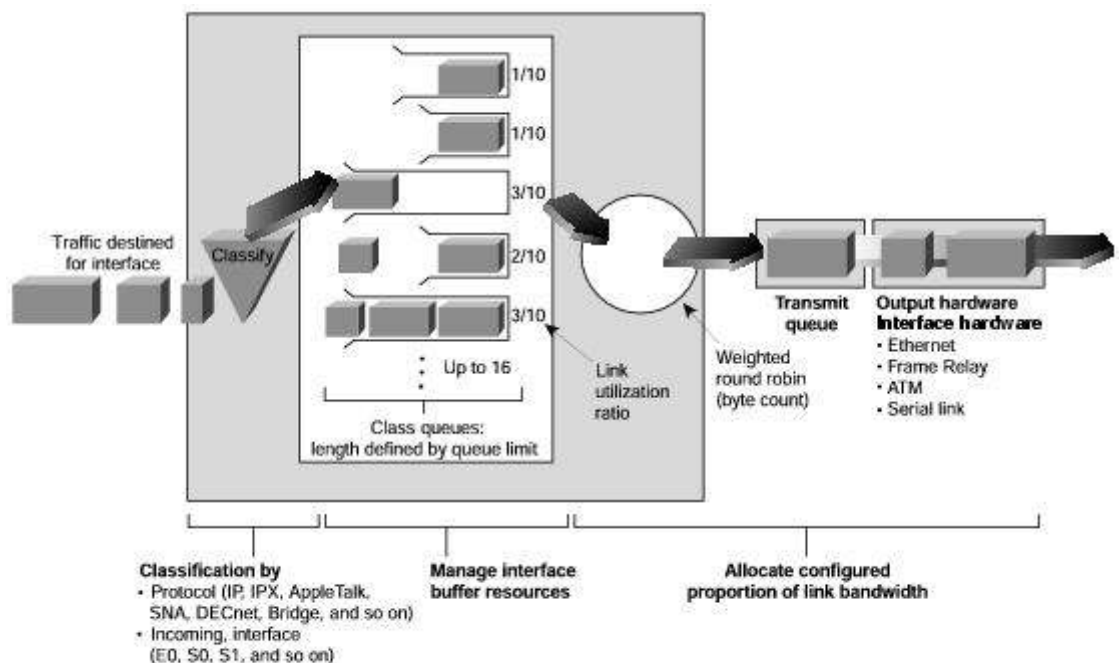


Figure 3.15 Custom queuing

3.11 Queue Management

Congestion avoidance is a form of queue management. Network traffic loads is monitored to predict and avoid congestion at common network bottlenecks by congestion avoidance techniques. It is different form the congestion management techniques, which are used to control congestion after it occurs.

3.11.1 Random Early Detection (RED)

Random Early Detection (RED) is a congestion avoidance method proposed by Van Jacobson and Sally Floyd. RED is an active queue management technique and it ensures performance advantages over a traditional tail drop approach. RED behaves proactively to congestion state. After the average queue size exceeds a definite minimum threshold, packets are started to be dropped, instead of waiting until the queue is filled to capacity. For avoiding global synchronization, RED drops packets randomly from only a few flows and this is guaranteed by a drop probability.

3.11.2 Weighted Random Early Detection (WRED)

The capabilities of the RED algorithm with IP precedence are combined by Weighted Random Early Detection (WRED). For higher-priority packets, preferential traffic handling is provided through this combination. When congestion begins to start, lower-priority traffic can be discarded selectively and differentiated performance characteristics can be provided for different classes of service by WRED. As a result, specific precedence level packets are dropped more aggressively and other precedence level packets are dropped less aggressively by WRED.

CHAPTER FOUR

ANALYSIS OF DATA TRANSMITTING IN MPLS NETWORK AND QoS EFFECTS

4.1 Topology of the Thesis Work

The topology of the thesis work is shown in Figure 4.1. Two videophones, two Layer-2 switches and two MPLS router were used to create data transmitting; these videophones communicate each other over the MPLS network.

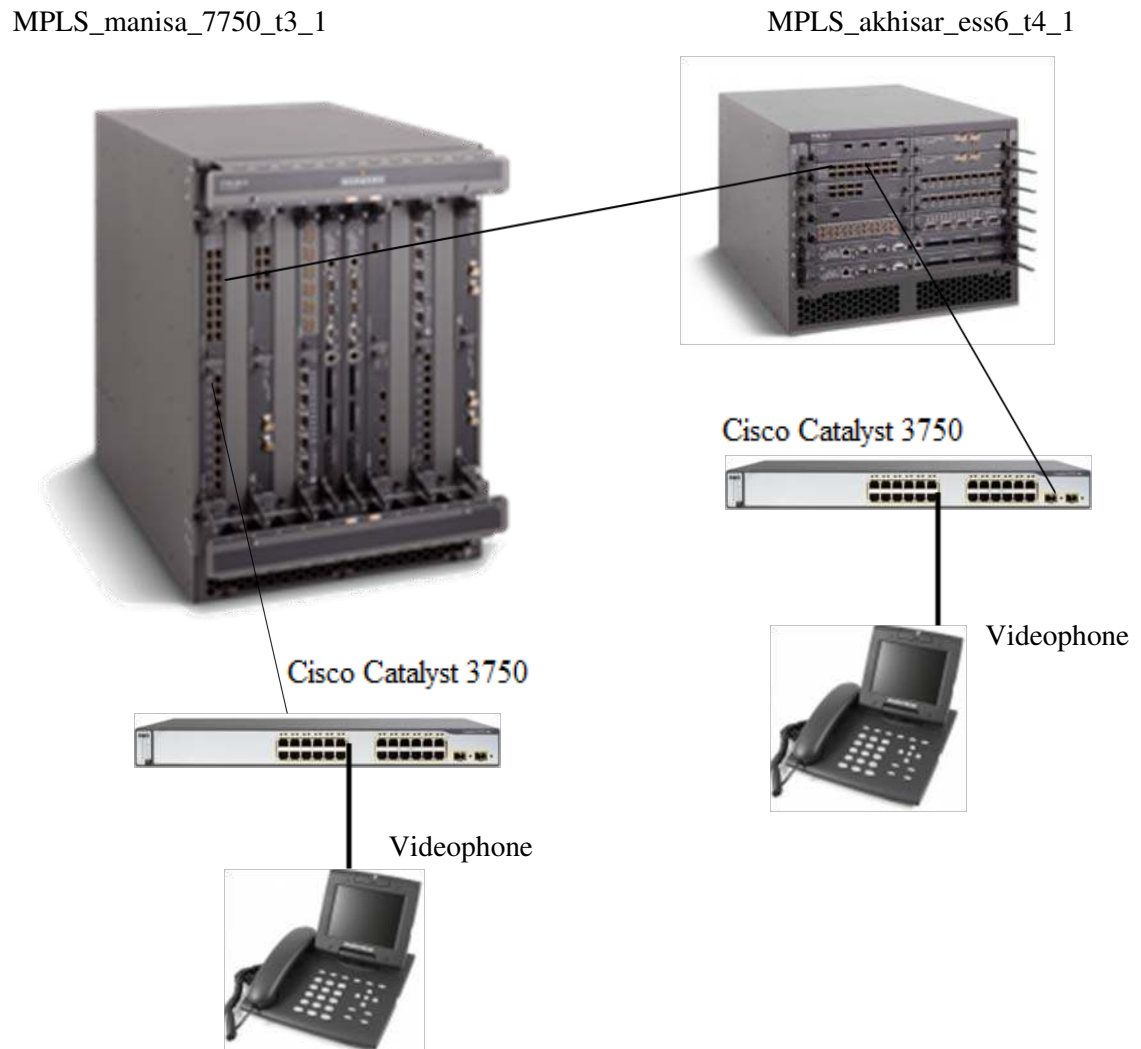


Figure 4.1 Topology of the thesis work

Firstly, one port of MPLS device was configured. It was configured as Internet-Enhanced Service (IES) circuit, assigned a Virtual Local Area Network (VLAN) and an IP address. This part of port configuration is shown below.

```
A:45_manisa_t3_1>show> port 6/2/20
ies 1 customer 1 create
    interface "ies-1||tezdeneme" create
        shutdown
    exit
    interface "tezdeneme" create
        address 195.175.73.81/30
        sap 6/2/20:10 create
            ingress
                qos 20001
            exit
            egress
                qos 20001
            exit
        exit
    exit
    no shutdown
exit
```

sap 6/2/20 shows the port of MPLS device and sap 6/2/20:10 mean this port has vlan 10.

According to the port of MPLS device's configuration, port of Cisco 3750 switch was configured. The Ethernet port of switch, which was used to connect the videophone was assigned vlan 10 and configured as access mode. The Gigabit Ethernet ports were configured as trunk mode. This part of switch configuration is shown below.

```

vlan 10
  name DATA
vlan 60
  name MANAGEMENT
!
interface range GigabitEthernet1/0/1-2
  switchport trunk encapsulation dot1q
  switchport mode trunk
!
!
interface range FastEthernet1/0/1-24
  switchport access vlan 10
  switchport mode access
exit
!
interface Vlan99
  ip addr 10.45.34.4 255.255.255.0
  no sh
!
ip default-gateway 10.45.34.1
ip classless

```

Videophone device was connected to switch's ethernet port, which is configured as access mode. According to the ip address that was assigned to the MPLS device port; videophone's IP address, gateway, subnet mask and DNS information were manually configured.

The same configurations were applied for the other videophone side. Thus, two videophones can communicate each other and data transmitting can be provided through the MPLS network.

In order to analyze the QoS effects, different QoS configurations were applied and the results of data packets were monitored. Firstly to understand the QoS effects fully; the port bandwidth was limited 1 Mbps to create congestion state. Then different QoS level configurations were applied.

4.1.1 General explanations of the commands

The general meanings of the port, QoS configurations and the monitoring results are explained in this title (Figure 4.2, Figure 4.3, and Figure 4.4). Therefore, meanings of configurations and results will not be mentioned in every different level QoS configuration.

The explanations of the port configuration commands are given below.

A:45_manisa_t3_1>config>service>ies# info

```
-----
interface "tezdeneme" create
  address 195.175.73.81/30 → tells the IP and subnet mask info of circuit
  sap 6/2/20:10 create → tells the port and vlan info of circuit
  ingress → tells the ingress QoS of the circuit
    qos 10001 → the first 1 means port has fc (forwarding class) of
best effort and last 1 means port has 1 Mbps bandwidth
  exit
  egress → tells the egress QoS of the circuit
    qos 10001 → the first 1 means port has fc best effort and last 1
means port has 1 Mbps bandwidth
  exit
exit
exit
no shutdown
```

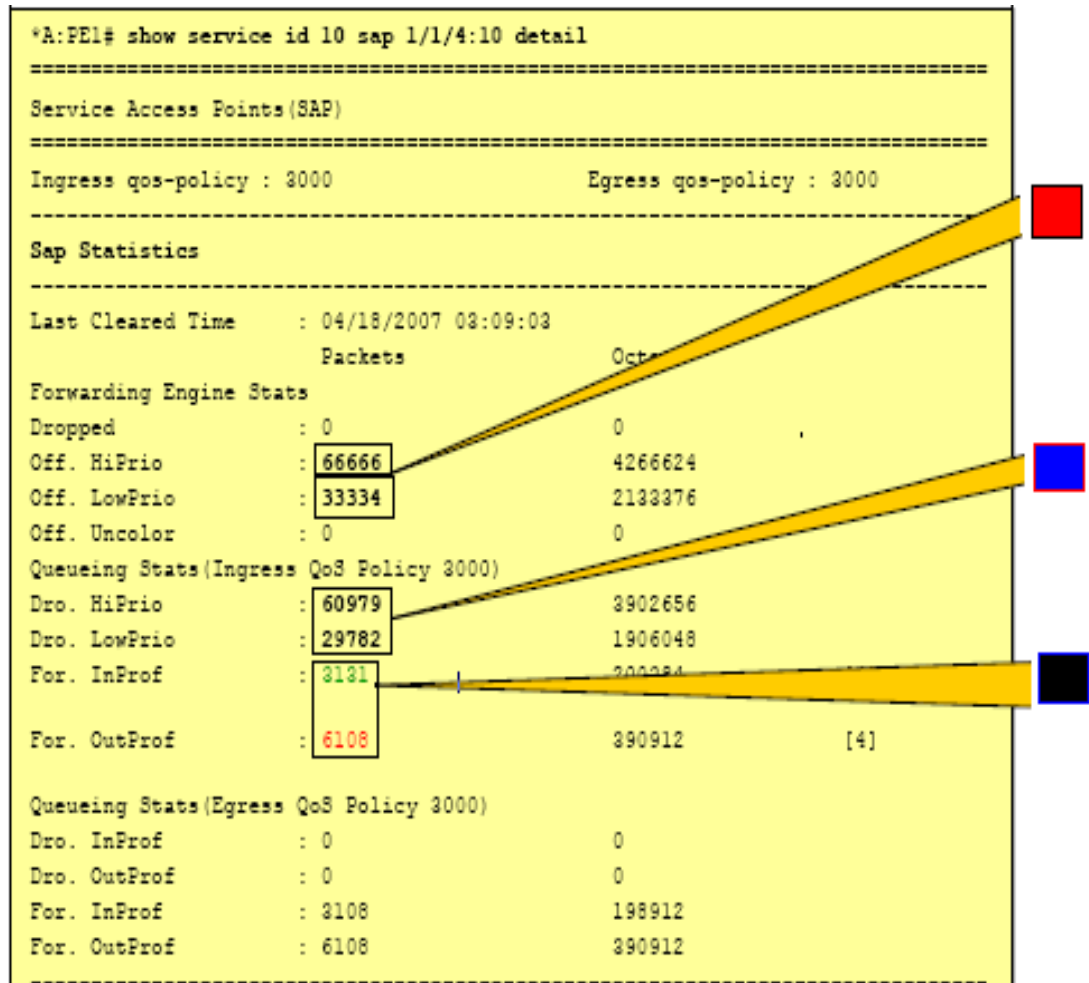


Figure 4.2 Explanations of service router monitoring result

■ Number of the total high and low enqueueing priority packets offered.

■ Number of the total high and low enqueueing priority packet dropped.

■ Number of the total in-profile and out-profile packets numbers.

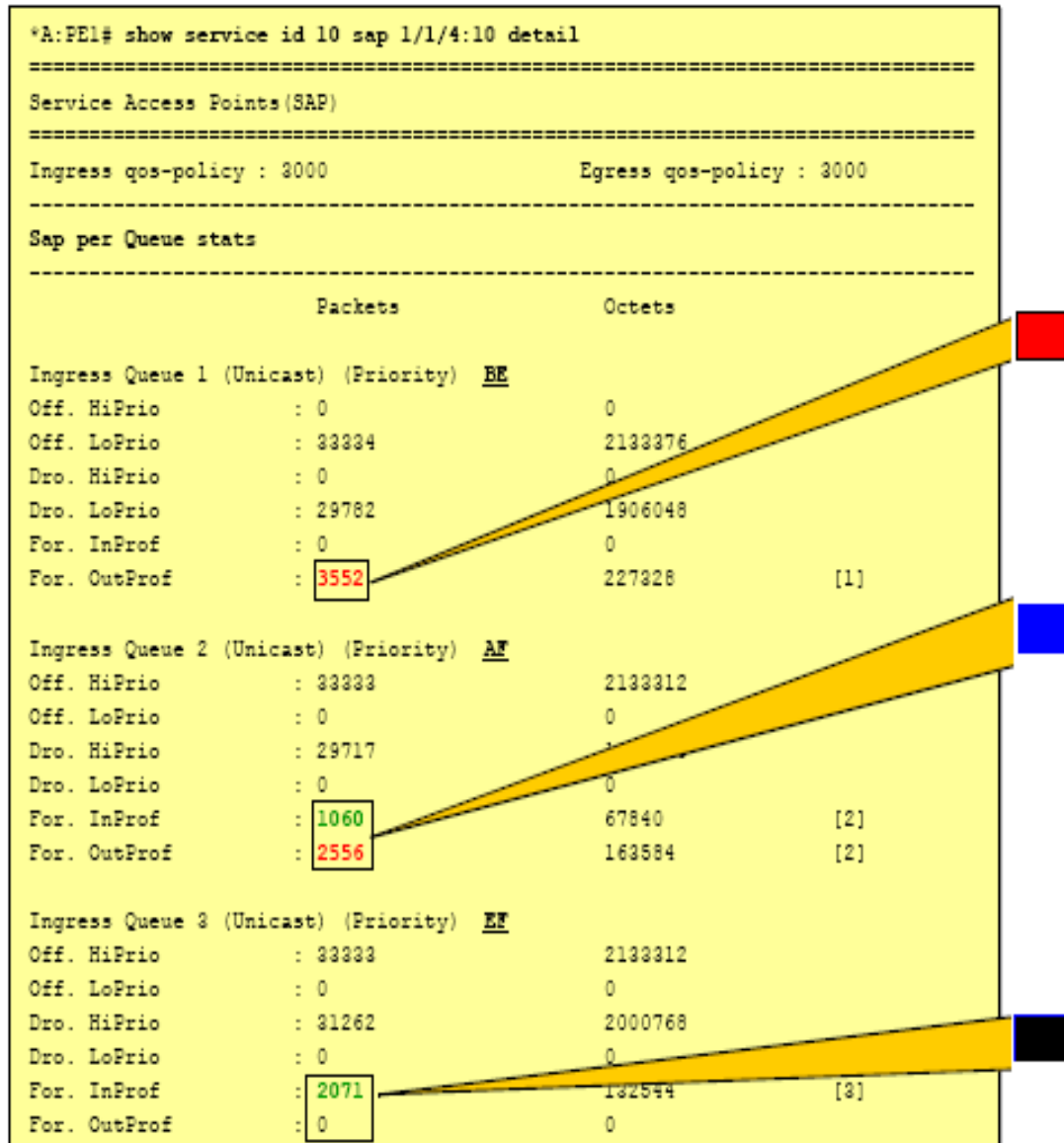


Figure 4.3 Explanations of service router monitoring result

■ The traffic ingressing queue 1 is all out-profile. This is the BE traffic that has no CIR defined, hence it is all out-profile.

■ The traffic ingressing queue 2 has some in-profile and some out-profile. This is the AF traffic that has a CIR and PIR defined, hence the <CIR is in-profile and the >CIR is out-of profile.

■ The traffic ingressing queue 3 is all in-profile. This is the EF traffic, which has CIR=PIR, hence all traffic is considered in-profile.

```

A:PE2# show service id 10 sap 1/1/4:10 detail
=====
Service Access Points(SAP)
=====
Ingress qos-policy : 3000                      Egress qos-policy : 3000
=====
Sap per Queue stats
=====

```

	Packets	Octets	
Ingress Queue 1 (Unicast) (Priority)			
...			
Ingress Queue 2 (Unicast) (Priority)			
...			
Ingress Queue 3 (Unicast) (Priority)			
...			
Egress Queue 1 <u>BE</u>			
For. InProf	: 0	0	
For. OutProf	: 3552	227328	[11]
Dro. InProf	: 0	0	
Dro. OutProf	: 0	0	
Egress Queue 2 <u>AF</u>			
For. InProf	: 1060	67840	[11]
For. OutProf	: 2556	163584	[11]
Dro. InProf	: 0	0	
Dro. OutProf	: 0	0	
Egress Queue 3 <u>EF</u>			
For. InProf	: 2071	132544	[11]
For. OutProf	: 0	0	
Dro. InProf	: 0	0	
Dro. OutProf	: 0	0	

```

=====

```

Figure 4.4 Explanations of service router monitoring result

■ The BE/AF traffic exits the SAP on the same queues, and with the same number of packets in each, as was accepted on the ingress SAP.

4.1.2 Queue 1 Configuration and Monitoring Results

Queue 1 means Best Effort (BE) Forwarding Class in our topology. The configurations are shown below. Also with these commands, traffic is limited to 1 Mbps, Queue 1 is created at the ingress and the egress and QoS profile is assigned to the circuit. Traffic, which has 1 Mbps bandwidth, is transmitted as low priority since it is assigned to Queue 1.

Port config	Sap-ingress config	Sap-egress config
<pre> :45_manisa_t3_1>config>service>ies# info ----- interface "tezdeneme" create address 195.175.73.81/30 sap 6/2/20:10 create ingress qos 10001 exit egress qos 10001 exit exit exit no shutdown </pre>	<pre> A:45_manisa_t3_1>config>qos# sap-ingress 10001 A:45_manisa_t3_1>config>qos>sap-ingress# info ----- description "tez-deneme" queue 1 create exit queue 11 multipoint create exit fc "be" create queue 1 exit exit </pre>	<pre> A:45_manisa_t3_1>config>qos# sap-egress 10001 A:45_manisa_t3_1>config>qos>sap-egress# info ----- description "tez-deneme" queue 1 create exit fc be create queue 1 exit exit </pre>

The result of monitoring port traffic is shown below.

```
*A:45_manisa_t3_1# monitor service id 111517 sap 6/2/20:10
```

```
=====
```

```
Monitor statistics for Service 111517 SAP 6/2/20:10
```

```
=====
```

```
At time t = 0 sec (Base Statistics)
```

```
-----
```

```
Sap Statistics
```

```
-----
```

	Packets	Octets
Forwarding Engine Stats		
Dropped	: 0	0
Off. HiPrio	: 0	0
Off. LowPrio	: 591921	183253342
Off. Uncolor	: 0	0

Queueing Stats(Ingress QoS Policy 10001)

Dro. HiPrio	: 0	0
Dro. LowPrio	: 147940	44677880
For. InProf	: 0	0
For. OutProf	: 443981	135414205

Queueing Stats(Egress QoS Policy 10001)

Dro. InProf	: 0	0
Dro. OutProf	: 120305	46918950
For. InProf	: 0	0
For. OutProf	: 361271	152588971

Sap per Queue Stats

	Packets	Octets
Ingress Queue 1 (Unicast) (Priority)		
Off. HiPrio	: 0	0
Off. LoPrio	: 591921	183253342
Dro. HiPrio	: 0	0
Dro. LoPrio	: 147940	44677880
For. InProf	: 0	0

For. OutProf : 443981 135414205

Egress Queue 1

For. InProf : 0 0

For. OutProf : 361271 152588971

Dro. InProf : 0 0

Dro. OutProf : 120305 46918950

At time t = 11 sec (Mode: Delta)

Sap Statistics

	Packets	Octets
Forwarding Engine Stats		
Dropped	: 0	0
Off. HiPrio	: 0	0
Off. LowPrio	: 683	203661
Off. Uncolor	: 0	0

Queueing Stats(Ingress QoS Policy 10001)

Dro. HiPrio	: 0	0
Dro. LowPrio	: 165	49830
For. InProf	: 0	0
For. OutProf	: 518	160580

Queueing Stats(Egress QoS Policy 10001)

Dro. InProf	: 0	0
Dro. OutProf	: 153	46665
For. InProf	: 0	0
For. OutProf	: 513	158004

Sap per Queue Stats

	Packets	Octets
Ingress Queue 1 (Unicast) (Priority)		
Off. HiPrio	: 0	0
Off. LoPrio	: 683	203661
Dro. HiPrio	: 0	0
Dro. LoPrio	: 165	49830
For. InProf	: 0	0
For. OutProf	: 518	160580

Egress Queue 1

For. InProf	: 0	0
For. OutProf	: 513	158004
Dro. InProf	: 0	0
Dro. OutProf	: 153	46665

At time t = 22 sec (Mode: Delta)

Sap Statistics

	Packets	Octets
Forwarding Engine Stats		
Dropped	: 0	0
Off. HiPrio	: 0	0
Off. LowPrio	: 693	211536
Off. Uncolor	: 0	0

Queueing Stats(Ingress QoS Policy 10001)

Dro. HiPrio	: 0	0
Dro. LowPrio	: 170	51510
For. InProf	: 0	0
For. OutProf	: 523	158992

Queueing Stats(Egress QoS Policy 10001)

Dro. InProf	: 0	0
Dro. OutProf	: 161	48461
For. InProf	: 0	0
For. OutProf	: 507	155142

Sap per Queue Stats

	Packets	Octets
--	---------	--------

Ingress Queue 1 (Unicast) (Priority)

Off. HiPrio	: 0	0
Off. LoPrio	: 693	211536
Dro. HiPrio	: 0	0
Dro. LoPrio	: 170	51510
For. InProf	: 0	0
For. OutProf	: 523	158992

Egress Queue 1

For. InProf	: 0	0
For. OutProf	: 507	155142
Dro. InProf	: 0	0
Dro. OutProf	: 161	48461

At time t = 33 sec (Mode: Delta)

Sap Statistics

	Packets	Octets
Forwarding Engine Stats		
Dropped	: 0	0
Off. HiPrio	: 0	0
Off. LowPrio	: 1386	422451
Off. Uncolor	: 0	0

Queueing Stats(Ingress QoS Policy 10001)

Dro. HiPrio	: 0	0
Dro. LowPrio	: 339	102039
For. InProf	: 0	0
For. OutProf	: 1047	319335

Queueing Stats(Egress QoS Policy 10001)

Dro. InProf	: 0	0
Dro. OutProf	: 151	45753
For. InProf	: 0	0
For. OutProf	: 512	157184

Sap per Queue Stats

	Packets	Octets
Ingress Queue 1 (Unicast) (Priority)		
Off. HiPrio	: 0	0
Off. LoPrio	: 1386	422451
Dro. HiPrio	: 0	0
Dro. LoPrio	: 339	102039
For. InProf	: 0	0
For. OutProf	: 1047	319335

Egress Queue 1

For. InProf	: 0	0
For. OutProf	: 512	157184
Dro. InProf	: 0	0
Dro. OutProf	: 151	45753

At time t = 44 sec (Mode: Delta)

Sap Statistics

	Packets	Octets
Forwarding Engine Stats		
Dropped	: 0	0
Off. HiPrio	: 0	0
Off. LowPrio	: 703	215714
Off. Uncolor	: 0	0

Queueing Stats(Ingress QoS Policy 10001)

Dro. HiPrio	: 0	0
Dro. LowPrio	: 170	51680
For. InProf	: 0	0
For. OutProf	: 533	163631

Queueing Stats(Egress QoS Policy 10001)

Dro. InProf	: 0	0
Dro. OutProf	: 143	43186
For. InProf	: 0	0
For. OutProf	: 515	156560

Sap per Queue Stats

	Packets	Octets
Ingress Queue 1 (Unicast) (Priority)		
Off. HiPrio	: 0	0
Off. LoPrio	: 703	215714
Dro. HiPrio	: 0	0
Dro. LoPrio	: 170	51680
For. InProf	: 0	0
For. OutProf	: 533	163631

Egress Queue 1

For. InProf	: 0	0
For. OutProf	: 515	156560
Dro. InProf	: 0	0
Dro. OutProf	: 143	43186

At time t = 55 sec (Mode: Delta)

Sap Statistics

	Packets	Octets
Forwarding Engine Stats		
Dropped	: 0	0
Off. HiPrio	: 0	0
Off. LowPrio	: 686	201111
Off. Uncolor	: 0	0

Queueing Stats(Ingress QoS Policy 10001)

Dro. HiPrio	: 0	0
Dro. LowPrio	: 170	51170
For. InProf	: 0	0
For. OutProf	: 516	156348

Queueing Stats(Egress QoS Policy 10001)

Dro. InProf	: 0	0
Dro. OutProf	: 331	98969
For. InProf	: 0	0
For. OutProf	: 1003	304912

Sap per Queue Stats

	Packets	Octets
Ingress Queue 1 (Unicast) (Priority)		
Off. HiPrio	: 0	0

Off. LoPrio	: 686	201111
Dro. HiPrio	: 0	0
Dro. LoPrio	: 170	51170
For. InProf	: 0	0
For. OutProf	: 516	156348

Egress Queue 1

For. InProf	: 0	0
For. OutProf	: 1003	304912
Dro. InProf	: 0	0
Dro. OutProf	: 331	98969

4.1.3 Queue 2 Configuration and Monitoring Results

Queue 2 means Low-2 (L2) Forwarding Class in our topology. With the port and ingress and egress configurations, traffic is limited to 1 Mbps, Queue 2 is created at the ingress and the egress and Queue 2 is assigned Low-2 Forwarding Class. Traffic is also transmitted as low priority in this QoS level since it is assigned Queue 2.

Port config

```
interface "tezdeneme" create
address 195.175.73.81/30
sap 6/2/20:10 create
  ingress
    qos 20001
  exit
  egress
    qos 20001
  exit
exit
```

Sap-ingress config

```
*A:45_manisa_t3_1>config>qos# sap-ingress 20001
*A:45_manisa_t3_1>config>qos>sap-ingress# info
-----
description "p2p 1 Mbps-out"
queue 1 create
exit
queue 2 create
  rate 1000 cir 1000
exit
queue 11 multipoint create
exit
fc "l2" create
  queue 2
exit
default-fc "l2"
```

Sap-egress config

```
A:45_manisa_t3_1>config>qos# sap-egress 20001
*A:45_manisa_t3_1>config>qos>sap-egress# info
-----
description "p2p 1 Mbps-out"
queue 1 create
exit
queue 2 create
  rate 1000 cir 1000
exit
fc l2 create
  queue 2
exit
```

The result of monitoring port traffic is shown below.

:45_manisa_t3_1# monitor service id 1 sap 6/2/20:10

```
=====
=====
```

Monitor statistics for Service 1 SAP 6/2/20:10

```
=====
=====
```

At time t = 0 sec (Base Statistics)

```
-----
-----
```

Sap Statistics

```
-----
```

	Packets	Octets
Forwarding Engine Stats		
Dropped	: 0	0
Off. HiPrio	: 0	0
Off. LowPrio	: 520533	179915184
Off. Uncolor	: 0	0

Queueing Stats(Ingress QoS Policy 20001)

Dro. HiPrio	: 0	0
Dro. LowPrio	: 109297	33882070
For. InProf	: 411236	143932600
For. OutProf	: 0	0

Queueing Stats(Egress QoS Policy 20001)

Dro. InProf	: 105654	36978900
-------------	----------	----------

Dro. OutProf	: 0	0
For. InProf	: 398028	166619586
For. OutProf	: 0	0

Sap per Queue Stats

	Packets	Octets
--	---------	--------

Ingress Queue 1 (Unicast) (Priority)

Off. HiPrio	: 0	0
Off. LoPrio	: 0	0
Dro. HiPrio	: 0	0
Dro. LoPrio	: 0	0
For. InProf	: 0	0
For. OutProf	: 0	0

Ingress Queue 2 (Unicast) (Priority)

Off. HiPrio	: 0	0
Off. LoPrio	: 520533	179915184
Dro. HiPrio	: 0	0
Dro. LowPrio	: 109297	33882070
For. InProf	: 411236	143932600
For. OutProf	: 0	0

Egress Queue 1

For. InProf	: 0	0
For. OutProf	: 0	0
Dro. InProf	: 0	0

Dro. OutProf : 0 0

Egress Queue 2

For. InProf : 398028 166619586

For. OutProf : 0 0

Dro. InProf : 105654 36978900

Dro. OutProf : 0 0

At time t = 11 sec (Mode: Delta)

Sap Statistics

	Packets	Octets
Forwarding Engine Stats		
Dropped	: 0	0
Off. HiPrio	: 0	0
Off. LowPrio	: 698	241704
Off. Uncolor	: 0	0

Queueing Stats(Ingress QoS Policy 20001)

Dro. HiPrio	: 0	0
Dro. LowPrio	: 143	47905
For. InProf	: 555	181485
For. OutProf	: 0	0

Queueing Stats(Egress QoS Policy 20001)

Dro. InProf	: 138	42642
-------------	-------	-------

Dro. OutProf	: 0	0
For. InProf	: 531	181602
For. OutProf	: 0	0

Sap per Queue Stats

	Packets	Octets
--	---------	--------

Ingress Queue 1 (Unicast) (Priority)

Off. HiPrio	: 0	0
Off. LoPrio	: 0	0
Dro. HiPrio	: 0	0
Dro. LoPrio	: 0	0
For. InProf	: 0	0
For. OutProf	: 0	0

Ingress Queue 2 (Unicast) (Priority)

Off. HiPrio	: 0	0
Off. LoPrio	: 698	241704
Dro. HiPrio	: 0	0
Dro. LowPrio	: 143	47905
For. InProf	: 555	181485
For. OutProf	: 0	0

Egress Queue 1

For. InProf	: 0	0
For. OutProf	: 0	0
Dro. InProf	: 0	0

Dro. OutProf	: 0	0
--------------	-----	---

Egress Queue 2

For. InProf	: 531	181602
For. OutProf	: 0	0
Dro. InProf	: 138	42642
Dro. OutProf	: 0	0

At time t = 22 sec (Mode: Delta)

Sap Statistics

	Packets	Octets
Forwarding Engine Stats		
Dropped	: 0	0
Off. HiPrio	: 0	0
Off. LowPrio	: 696	240458
Off. Uncolor	: 0	0

Queueing Stats(Ingress QoS Policy 20001)

Dro. HiPrio	: 0	0
Dro. LowPrio	: 144	48960
For. InProf	: 552	186576
For. OutProf	: 0	0

Queueing Stats(Egress QoS Policy 20001)

Dro. InProf	: 139	47121
-------------	-------	-------

Dro. OutProf	: 0	0
For. InProf	: 540	177660
For. OutProf	: 0	0

Sap per Queue Stats

	Packets	Octets
--	---------	--------

Ingress Queue 1 (Unicast) (Priority)

Off. HiPrio	: 0	0
Off. LoPrio	: 0	0
Dro. HiPrio	: 0	0
Dro. LoPrio	: 0	0
For. InProf	: 0	0
For. OutProf	: 0	0

Ingress Queue 2 (Unicast) (Priority)

Off. HiPrio	: 0	0
Off. LoPrio	: 696	240458
Dro. HiPrio	: 0	0
Dro. LowPrio	: 144	48960
For. InProf	: 552	186576
For. OutProf	: 0	0

Egress Queue 1

For. InProf	: 0	0
For. OutProf	: 0	0
Dro. InProf	: 0	0

Dro. OutProf : 0 0

Egress Queue 2

For. InProf : 540 177660

For. OutProf : 0 0

Dro. InProf : 139 47121

Dro. OutProf : 0 0

At time t = 33 sec (Mode: Delta)

Sap Statistics

	Packets	Octets
Forwarding Engine Stats		
Dropped	: 0	0
Off. HiPrio	: 0	0
Off. LowPrio	: 1400	489466
Off. Uncolor	: 0	0

Queueing Stats(Ingress QoS Policy 20001)

Dro. HiPrio : 0 0

Dro. LowPrio : 294 99960

For. InProf : 1106 363874

For. OutProf : 0 0

Queueing Stats(Egress QoS Policy 20001)

Dro. InProf : 139 47677

Dro. OutProf	: 0	0
For. InProf	: 534	179424
For. OutProf	: 0	0

Sap per Queue Stats

	Packets	Octets
Ingress Queue 1 (Unicast) (Priority)		
Off. HiPrio	: 0	0
Off. LoPrio	: 0	0
Dro. HiPrio	: 0	0
Dro. LoPrio	: 0	0
For. InProf	: 0	0
For. OutProf	: 0	0

Ingress Queue 2 (Unicast) (Priority)		
Off. HiPrio	: 0	0
Off. LoPrio	: 1400	489466
Dro. HiPrio	: 0	0
Dro. LowPrio	: 294	99960
For. InProf	: 1106	363874
For. OutProf	: 0	0

Egress Queue 1

For. InProf	: 0	0
For. OutProf	: 0	0
Dro. InProf	: 0	0

Dro. OutProf	: 0	0
--------------	-----	---

Egress Queue 2

For. InProf	: 534	179424
For. OutProf	: 0	0
Dro. InProf	: 139	47677
Dro. OutProf	: 0	0

At time t = 44 sec (Mode: Delta)

Sap Statistics

	Packets	Octets
Forwarding Engine Stats		
Dropped	: 0	0
Off. HiPrio	: 0	0
Off. LowPrio	: 697	240728
Off. Uncolor	: 0	0

Queueing Stats(Ingress QoS Policy 20001)

Dro. HiPrio	: 0	0
Dro. LowPrio	: 145	49590
For. InProf	: 552	188232
For. OutProf	: 0	0

Queueing Stats(Egress QoS Policy 20001)

Dro. InProf	: 142	47854
Dro. OutProf	: 0	0
For. InProf	: 538	178078
For. OutProf	: 0	0

Sap per Queue Stats

	Packets	Octets
--	---------	--------

Ingress Queue 1 (Unicast) (Priority)

Off. HiPrio	: 0	0
Off. LoPrio	: 0	0
Dro. HiPrio	: 0	0
Dro. LoPrio	: 0	0
For. InProf	: 0	0
For. OutProf	: 0	0

Ingress Queue 2 (Unicast) (Priority)

Off. HiPrio	: 0	0
Off. LoPrio	: 697	240728
Dro. HiPrio	: 0	0
Dro. LowPrio	: 145	49590
For. InProf	: 552	188232
For. OutProf	: 0	0

Egress Queue 1

For. InProf	: 0	0
-------------	-----	---

For. OutProf	: 0	0
Dro. InProf	: 0	0
Dro. OutProf	: 0	0

Egress Queue 2

For. InProf	: 538	1780780
For. OutProf	: 0	0
Dro. InProf	: 142	47854
Dro. OutProf	: 0	0

At time t = 55 sec (Mode: Delta)

Sap Statistics

	Packets	Octets
Forwarding Engine Stats		
Dropped	: 0	0
Off. HiPrio	: 0	0
Off. LowPrio	: 712	253762
Off. Uncolor	: 0	0

Queueing Stats(Ingress QoS Policy 20001)

Dro. HiPrio	: 0	0
Dro. LowPrio	: 147	51597
For. InProf	: 565	194925
For. OutProf	: 0	0

Queueing Stats(Egress QoS Policy 20001)

Dro. InProf	: 281	95821
Dro. OutProf	: 0	0
For. InProf	: 1065	361035
For. OutProf	: 0	0

Sap per Queue Stats

	Packets	Octets
--	---------	--------

Ingress Queue 1 (Unicast) (Priority)

Off. HiPrio	: 0	0
Off. LoPrio	: 0	0
Dro. HiPrio	: 0	0
Dro. LoPrio	: 0	0
For. InProf	: 0	0
For. OutProf	: 0	0

Ingress Queue 2 (Unicast) (Priority)

Off. HiPrio	: 0	0
Off. LoPrio	: 712	253762
Dro. HiPrio	: 0	0
Dro. LowPrio	: 147	51597
For. InProf	: 565	194925
For. OutProf	: 0	0

Egress Queue 1

For. InProf	: 0	0
-------------	-----	---

```

For. OutProf      : 0      0
Dro. InProf       : 0      0
Dro. OutProf      : 0      0

```

Egress Queue 2

```

For. InProf       : 1065    361035
For. OutProf      : 0      0
Dro. InProf       : 281    95821
Dro. OutProf      : 0      0

```

4.1.4 Queue 3 Configuration and Monitoring Results

Queue 3 means Assured (AF) Forwarding Class in our topology. It is intended for assured traffic but not high priority traffic normally. With the Service Access Point (SAP) ingress configuration, AF Forwarding Class is assigned to high priority. Thus, traffic is transmitted as high priority. Also in this configuration, traffic is limited to 1Mbps bandwidth. The configurations are shown below.

Port config

```

interface "tezdeneme" create
  address 195.175.73.81/30
  sap 6/2/20:10 create
    ingress
      qos 30001
    exit
    egress
      qos 30001
    exit
  exit
  exit
  no shutdown

```

Sap-ingress config

```

A:45_manisa_t3_1# configure qos sap-ingress 30001
*A:45_manisa_t3_1>config>qos>sap-ingress# info
-----
queue 1 create
exit
queue 3 create
  rate 1000 cir 1000
exit
queue 11 multipoint create
exit
fc "af" create
  queue 3
exit
default-fc "af"
default-priority high

```

Sap-egress config

```

*A:45_manisa_t3_1>config>qos# sap-egress
30001
*A:45_manisa_t3_1>config>qos>sap-egress# info
-----
queue 1 create
exit
queue 3 create
  rate 1000 cir 1000
exit
fc af create
  queue 3
exit

```


The result of monitoring port traffic is shown below.

*A:45_manisa_t3_1# monitor service id 111517 sap 6/2/20:10

=====
Monitor statistics for Service 111517 SAP 6/2/20:10
=====

At time t = 0 sec (Base Statistics)

Sap Statistics

	Packets	Octets
Forwarding Engine Stats		
Dropped	: 0	0
Off. HiPrio	: 1119803	337114307
Off. LowPrio	: 0	0
Off. Uncolor	: 0	0

Queueing Stats(Ingress QoS Policy 30001)

Dro. HiPrio	: 134376	40581552
Dro. LowPrio	: 0	0
For. InProf	: 985427	293657246
For. OutProf	: 0	0

Queueing Stats(Egress QoS Policy 30001)

Dro. InProf	: 180545	55066225
Dro. OutProf	: 0	0
For. InProf	: 1323998	395875402
For. OutProf	: 0	0

Sap per Queue Stats

	Packets	Octets
Ingress Queue 1 (Unicast) (Priority)		
Off. HiPrio	: 0	0
Off. LoPrio	: 0	0
Dro. HiPrio	: 0	0
Dro. LoPrio	: 0	0
For. InProf	: 0	0
For. OutProf	: 0	0

Ingress Queue 3 (Unicast) (Priority)

Off. HiPrio	: 1119803	337114307
Off. LoPrio	: 0	0
Dro. HiPrio	: 134376	40581552
Dro. LowPrio	: 0	0
For. InProf	: 985427	293657246
For. OutProf	: 0	0

Egress Queue 1

For. InProf	: 0	0
For. OutProf	: 0	0
Dro. InProf	: 0	0
Dro. OutProf	: 0	0

Egress Queue 3

For. InProf	: 1323998	395875402
For. OutProf	: 0	0
Dro. InProf	: 180545	55066225
Dro. OutProf	: 0	0

At time t = 11 sec (Mode: Delta)

Sap Statistics

	Packets	Octets
Forwarding Engine Stats		
Dropped	: 0	0
Off. HiPrio	: 628	164098
Off. LowPrio	: 0	0
Off. Uncolor	: 0	0

Queueing Stats(Ingress QoS Policy 30001)

Dro. HiPrio	: 75	21750
Dro. LowPrio	: 0	0
For. InProf	: 553	158158
For. OutProf	: 0	0

Queueing Stats(Egress QoS Policy 30001)

Dro. InProf	: 160	48160
Dro. OutProf	: 0	0
For. InProf	: 1203	357291
For. OutProf	: 0	0

Sap per Queue Stats

	Packets	Octets
Ingress Queue 1 (Unicast) (Priority)		
Off. HiPrio	: 0	0
Off. LoPrio	: 0	0
Dro. HiPrio	: 0	0
Dro. LoPrio	: 0	0
For. InProf	: 0	0
For. OutProf	: 0	0

Ingress Queue 3 (Unicast) (Priority)		
Off. HiPrio	: 628	164098
Off. LoPrio	: 0	0
Dro. HiPrio	: 75	21750
Dro. LowPrio	: 0	0
For. InProf	: 553	158158
For. OutProf	: 0	0

Egress Queue 1		
For. InProf	: 0	0
For. OutProf	: 0	0
Dro. InProf	: 0	0
Dro. OutProf	: 0	0

Egress Queue 3

For. InProf	: 1203	357291
For. OutProf	: 0	0
Dro. InProf	: 160	48160
Dro. OutProf	: 0	0

At time t = 22 sec (Mode: Delta)

Sap Statistics

	Packets	Octets
Forwarding Engine Stats		
Dropped	: 0	0
Off. HiPrio	: 605	133784
Off. LowPrio	: 0	0
Off. Uncolor	: 0	0

Queueing Stats(Ingress QoS Policy 30001)

Dro. HiPrio	: 74	17834
Dro. LowPrio	: 0	0
For. InProf	: 531	134343
For. OutProf	: 0	0

Queueing Stats(Egress QoS Policy 30001)

Dro. InProf	: 79	18802
Dro. OutProf	: 0	0
For. InProf	: 588	155232

For. OutProf	: 0	0
--------------	-----	---

Sap per Queue Stats

	Packets	Octets
--	---------	--------

Ingress Queue 1 (Unicast) (Priority)

Off. HiPrio	: 0	0
-------------	-----	---

Off. LoPrio	: 0	0
-------------	-----	---

Dro. HiPrio	: 0	0
-------------	-----	---

Dro. LoPrio	: 0	0
-------------	-----	---

For. InProf	: 0	0
-------------	-----	---

For. OutProf	: 0	0
--------------	-----	---

Ingress Queue 3 (Unicast) (Priority)

Off. HiPrio	: 605	133784
-------------	-------	--------

Off. LoPrio	: 0	0
-------------	-----	---

Dro. HiPrio	: 74	17834
-------------	------	-------

Dro. LowPrio	: 0	0
--------------	-----	---

For. InProf	: 531	134343
-------------	-------	--------

For. OutProf	: 0	0
--------------	-----	---

Egress Queue 1

For. InProf	: 0	0
-------------	-----	---

For. OutProf	: 0	0
--------------	-----	---

Dro. InProf	: 0	0
-------------	-----	---

Dro. OutProf	: 0	0
--------------	-----	---

Egress Queue 3

For. InProf	: 588	155232
For. OutProf	: 0	0
Dro. InProf	: 79	18802
Dro. OutProf	: 0	0

At time t = 33 sec (Mode: Delta)

Sap Statistics

	Packets	Octets
Forwarding Engine Stats		
Dropped	: 0	0
Off. HiPrio	: 622	153283
Off. LowPrio	: 0	0
Off. Uncolor	: 0	0

Queueing Stats(Ingress QoS Policy 30001)

Dro. HiPrio	: 75	18825
Dro. LowPrio	: 0	0
For. InProf	: 547	141673
For. OutProf	: 0	0

Queueing Stats(Egress QoS Policy 30001)

Dro. InProf	: 78	20670
Dro. OutProf	: 0	0
For. InProf	: 591	152478
For. OutProf	: 0	0

Sap per Queue Stats		

	Packets	Octets
Ingress Queue 1 (Unicast) (Priority)		
Off. HiPrio	: 0	0
Off. LoPrio	: 0	0
Dro. HiPrio	: 0	0
Dro. LoPrio	: 0	0
For. InProf	: 0	0
For. OutProf	: 0	0
Ingress Queue 3 (Unicast) (Priority)		
Off. HiPrio	: 622	153283
Off. LoPrio	: 0	0
Dro. HiPrio	: 75	18825
Dro. LowPrio	: 0	0
For. InProf	: 547	141673
For. OutProf	: 0	0
Egress Queue 1		
For. InProf	: 0	0
For. OutProf	: 0	0
Dro. InProf	: 0	0
Dro. OutProf	: 0	0

Egress Queue 3

For. InProf	: 591	152478
For. OutProf	: 0	0
Dro. InProf	: 78	20670
Dro. OutProf	: 0	0

At time t = 44 sec (Mode: Delta)

Sap Statistics

	Packets	Octets
Forwarding Engine Stats		
Dropped	: 0	0
Off. HiPrio	: 630	165680
Off. LowPrio	: 0	0
Off. Uncolor	: 0	0

Queueing Stats(Ingress QoS Policy 30001)

Dro. HiPrio	: 77	21098
Dro. LowPrio	: 0	0
For. InProf	: 553	146545
For. OutProf	: 0	0

Queueing Stats(Egress QoS Policy 30001)

Dro. InProf	: 81	22680
Dro. OutProf	: 0	0
For. InProf	: 593	169598
For. OutProf	: 0	0

Sap per Queue Stats

	Packets	Octets
Ingress Queue 1 (Unicast) (Priority)		
Off. HiPrio	: 0	0
Off. LoPrio	: 0	0
Dro. HiPrio	: 0	0
Dro. LoPrio	: 0	0
For. InProf	: 0	0
For. OutProf	: 0	0

Ingress Queue 3 (Unicast) (Priority)		
Off. HiPrio	: 630	165680
Off. LoPrio	: 0	0
Dro. HiPrio	: 77	21098
Dro. LowPrio	: 0	0
For. InProf	: 553	146545
For. OutProf	: 0	0

Egress Queue 1

For. InProf	: 0	0
For. OutProf	: 0	0
Dro. InProf	: 0	0
Dro. OutProf	: 0	0

Egress Queue 3

For. InProf	: 593	169598
For. OutProf	: 0	0
Dro. InProf	: 81	22680
Dro. OutProf	: 0	0

At time t = 55 sec (Mode: Delta)

Sap Statistics

	Packets	Octets
Forwarding Engine Stats		
Dropped	: 0	0
Off. HiPrio	: 605	133690
Off. LowPrio	: 0	0
Off. Uncolor	: 0	0

Queueing Stats(Ingress QoS Policy 30001)

Dro. HiPrio	: 71	16188
Dro. LowPrio	: 0	0
For. InProf	: 534	123354
For. OutProf	: 0	0

Queueing Stats(Egress QoS Policy 30001)

Dro. InProf	: 76	17100
Dro. OutProf	: 0	0
For. InProf	: 594	136026
For. OutProf	: 0	0

Sap per Queue Stats

	Packets	Octets
Ingress Queue 1 (Unicast) (Priority)		
Off. HiPrio	: 0	0
Off. LoPrio	: 0	0
Dro. HiPrio	: 0	0
Dro. LoPrio	: 0	0
For. InProf	: 0	0
For. OutProf	: 0	0

Ingress Queue 3 (Unicast) (Priority)		
Off. HiPrio	: 605	133690
Off. LoPrio	: 0	0
Dro. HiPrio	: 71	16188
Dro. LowPrio	: 0	0
For. InProf	: 534	123354
For. OutProf	: 0	0

Egress Queue 1

For. InProf	: 0	0
For. OutProf	: 0	0
Dro. InProf	: 0	0
Dro. OutProf	: 0	0

Egress Queue 3

For. InProf	: 594	136026
For. OutProf	: 0	0
Dro. InProf	: 76	17100
Dro. OutProf	: 0	0

4.1.5 Queue 5 Configuration and Monitoring Results

Queue 5 means High-2 (H2) Forwarding Class in our topology. It is intended for delay/jitter sensitive traffic. With the configurations, Queue 5 is created, traffic is limited to 1 Mbps like all the other configurations and High-2 Forwarding Class is assigned as high priority. The configurations are shown below.

Port config	Sap-ingress config	Sap-egress config
<pre>interface "tezdeneme" create address 195.175.73.81/30 sap 6/2/20:10 create ingress qps 45001 exit egress qps 45001 exit exit exit exit no shutdown</pre>	<pre>*A:45_manisa_t3_1# configure qps sap-ingress 45001 *A:45_manisa_t3_1>config>qps>sap-ingress# info ----- queue 1 create exit queue 5 create rate 1000 cir 1000 exit queue 11 multipoint create exit fc "h2" create queue 5 exit default-fc "h2" default-priority high</pre>	<pre>*A:45_manisa_t3_1>config>qps# sap-egress 45001 *A:45_manisa_t3_1>config>qps>sap-egress# info ----- queue 1 create exit queue 5 create rate 1000 cir 1000 exit fc h2 create</pre>

The result of monitoring port traffic is shown below.

=====

Monitor statistics for Service 111517 SAP 6/2/20:10

=====

At time t = 0 sec (Base Statistics)

Sap Statistics

	Packets	Octets
Forwarding Engine Stats		
Dropped	: 0	0
Off. HiPrio	: 181023	62925320
Off. LowPrio	: 0	0
Off. Uncolor	: 0	0
Queueing Stats(Ingress QoS Policy 45001)		
Dro. HiPrio	: 9056	3124320
Dro. LowPrio	: 0	0
For. InProf	: 171967	58984681
For. OutProf	: 0	0

Queueing Stats(Egress QoS Policy 45001)

Dro. InProf	: 53289	18011682
Dro. OutProf	: 0	0
For. InProf	: 1012500	355387500
For. OutProf	: 0	0

Sap per Queue Stats

	Packets	Octets
Ingress Queue 1 (Unicast) (Priority)		
Off. HiPrio	: 0	0
Off. LoPrio	: 0	0

Dro. HiPrio	: 0	0
Dro. LoPrio	: 0	0
For. InProf	: 0	0
For. OutProf	: 0	0

Ingress Queue 5 (Unicast) (Priority)

Off. HiPrio	: 181023	62925320
Off. LoPrio	: 0	0
Dro. HiPrio	: 9056	3124320
Dro. LowPrio	: 0	0
For. InProf	: 171967	58984681
For. OutProf	: 0	0

Egress Queue 1

For. InProf	: 0	0
For. OutProf	: 0	0
Dro. InProf	: 0	0
Dro. OutProf	: 0	0

Egress Queue 5

For. InProf	: 1012500	355387500
For. OutProf	: 0	0
Dro. InProf	: 53289	18011682
Dro. OutProf	: 0	0

At time t = 11 sec (Mode: Delta)

Sap Statistics

	Packets	Octets
Forwarding Engine Stats		
Dropped	: 0	0
Off. HiPrio	: 701	235398
Off. LowPrio	: 0	0
Off. Uncolor	: 0	0

Queueing Stats(Ingress QoS Policy 45001)

Dro. HiPrio	: 33	11121
Dro. LowPrio	: 0	0
For. InProf	: 668	221108
For. OutProf	: 0	0

Queueing Stats(Egress QoS Policy 45001)

Dro. InProf	: 31	10199
Dro. OutProf	: 0	0
For. InProf	: 635	213995
For. OutProf	: 0	0

Sap per Queue Stats

	Packets	Octets
Ingress Queue 1 (Unicast) (Priority)		
Off. HiPrio	: 0	0
Off. LoPrio	: 0	0

Dro. HiPrio	: 0	0
Dro. LoPrio	: 0	0
For. InProf	: 0	0
For. OutProf	: 0	0

Ingress Queue 5 (Unicast) (Priority)

Off. HiPrio	: 701	235398
Off. LoPrio	: 0	0
Dro. HiPrio	: 33	11121
Dro. LowPrio	: 0	0
For. InProf	: 668	221108
For. OutProf	: 0	0

Egress Queue 1

For. InProf	: 0	0
For. OutProf	: 0	0
Dro. InProf	: 0	0
Dro. OutProf	: 0	0

Egress Queue 5

For. InProf	: 635	213995
For. OutProf	: 0	0
Dro. InProf	: 31	10199
Dro. OutProf	: 0	0

At time t = 22 sec (Mode: Delta)

Sap Statistics

	Packets	Octets
Forwarding Engine Stats		
Dropped	: 0	0
Off. HiPrio	: 690	235819
Off. LowPrio	: 0	0
Off. Uncolor	: 0	0

Queueing Stats(Ingress QoS Policy 45001)

Dro. HiPrio	: 33	11319
Dro. LowPrio	: 0	0
For. InProf	: 657	226008
For. OutProf	: 0	0

Queueing Stats(Egress QoS Policy 45001)

Dro. InProf	: 32	10944
Dro. OutProf	: 0	0
For. InProf	: 631	217064
For. OutProf	: 0	0

Sap per Queue Stats

	Packets	Octets
Ingress Queue 1 (Unicast) (Priority)		
Off. HiPrio	: 0	0
Off. LoPrio	: 0	0

Dro. HiPrio	: 0	0
Dro. LoPrio	: 0	0
For. InProf	: 0	0
For. OutProf	: 0	0

Ingress Queue 5 (Unicast) (Priority)

Off. HiPrio	: 690	235819
Off. LoPrio	: 0	0
Dro. HiPrio	: 33	11319
Dro. LowPrio	: 0	0
For. InProf	: 657	226008
For. OutProf	: 0	0

Egress Queue 1

For. InProf	: 0	0
For. OutProf	: 0	0
Dro. InProf	: 0	0
Dro. OutProf	: 0	0

Egress Queue 5

For. InProf	: 631	217064
For. OutProf	: 0	0
Dro. InProf	: 32	10944
Dro. OutProf	: 0	0

At time t = 33 sec (Mode: Delta)

Sap Statistics

	Packets	Octets
Forwarding Engine Stats		
Dropped	: 0	0
Off. HiPrio	: 1407	484111
Off. LowPrio	: 0	0
Off. Uncolor	: 0	0

Queueing Stats(Ingress QoS Policy 45001)

Dro. HiPrio	: 72	24840
Dro. LowPrio	: 0	0
For. InProf	: 1335	457905
For. OutProf	: 0	0

Queueing Stats(Egress QoS Policy 45001)

Dro. InProf	: 66	22506
Dro. OutProf	: 0	0
For. InProf	: 1274	439530
For. OutProf	: 0	0

Sap per Queue Stats

	Packets	Octets
Ingress Queue 1 (Unicast) (Priority)		
Off. HiPrio	: 0	0
Off. LoPrio	: 0	0

Dro. HiPrio	: 0	0
Dro. LoPrio	: 0	0
For. InProf	: 0	0
For. OutProf	: 0	0

Ingress Queue 5 (Unicast) (Priority)

Off. HiPrio	: 1407	484111
Off. LoPrio	: 0	0
Dro. HiPrio	: 72	24840
Dro. LowPrio	: 0	0
For. InProf	: 1335	457905
For. OutProf	: 0	0

Egress Queue 1

For. InProf	: 0	0
For. OutProf	: 0	0
Dro. InProf	: 0	0
Dro. OutProf	: 0	0

Egress Queue 5

For. InProf	: 1274	439530
For. OutProf	: 0	0
Dro. InProf	: 66	22506
Dro. OutProf	: 0	0

At time t = 44 sec (Mode: Delta)

Sap Statistics

	Packets	Octets
Forwarding Engine Stats		
Dropped	: 0	0
Off. HiPrio	: 696	237966
Off. LowPrio	: 0	0
Off. Uncolor	: 0	0

Queueing Stats(Ingress QoS Policy 45001)

Dro. HiPrio	: 34	11662
Dro. LowPrio	: 0	0
For. InProf	: 662	227728
For. OutProf	: 0	0

Queueing Stats(Egress QoS Policy 45001)

Dro. InProf	: 33	11319
Dro. OutProf	: 0	0
For. InProf	: 635	215265
For. OutProf	: 0	0

Sap per Queue Stats

	Packets	Octets
Ingress Queue 1 (Unicast) (Priority)		
Off. HiPrio	: 0	0
Off. LoPrio	: 0	0

Dro. HiPrio	: 0	0
Dro. LoPrio	: 0	0
For. InProf	: 0	0
For. OutProf	: 0	0

Ingress Queue 5 (Unicast) (Priority)

Off. HiPrio	: 696	237966
Off. LoPrio	: 0	0
Dro. HiPrio	: 34	11662
Dro. LowPrio	: 0	0
For. InProf	: 662	215256
For. OutProf	: 0	0

Egress Queue 1

For. InProf	: 0	0
For. OutProf	: 0	0
Dro. InProf	: 0	0
Dro. OutProf	: 0	0

Egress Queue 5

For. InProf	: 635	215265
For. OutProf	: 0	0
Dro. InProf	: 33	11319
Dro. OutProf	: 0	0

At time t = 55 sec (Mode: Delta)

Sap Statistics

	Packets	Octets
Forwarding Engine Stats		
Dropped	: 0	0
Off. HiPrio	: 706	246007
Off. LowPrio	: 0	0
Off. Uncolor	: 0	0

Queueing Stats(Ingress QoS Policy 45001)

Dro. HiPrio	: 32	11072
Dro. LowPrio	: 0	0
For. InProf	: 674	233878
For. OutProf	: 0	0

Queueing Stats(Egress QoS Policy 45001)

Dro. InProf	: 34	11696
Dro. OutProf	: 0	0
For. InProf	: 632	220568
For. OutProf	: 0	0

Sap per Queue Stats

	Packets	Octets
Ingress Queue 1 (Unicast) (Priority)		
Off. HiPrio	: 0	0
Off. LoPrio	: 0	0

Dro. HiPrio	: 0	0
Dro. LoPrio	: 0	0
For. InProf	: 0	0
For. OutProf	: 0	0

Ingress Queue 5 (Unicast) (Priority)

Off. HiPrio	: 706	246007
Off. LoPrio	: 0	0
Dro. HiPrio	: 32	11072
Dro. LowPrio	: 0	0
For. InProf	: 674	233878
For. OutProf	: 0	0

Egress Queue 1

For. InProf	: 0	0
For. OutProf	: 0	0
Dro. InProf	: 0	0
Dro. OutProf	: 0	0

Egress Queue 5

For. InProf	: 632	220568
For. OutProf	: 0	0
Dro. InProf	: 34	11696
Dro. OutProf	: 0	0

4.2 Analysis of the Results

According to the monitoring results, it can easily be seen that packets are transmitted as low or high priority through the MPLS network at different QoS levels. The percentage of dropped packets differs in accordance with the QoS level. Traffic bandwidth is limited to 1 Mbps at all QoS level configurations, so percentage of the dropped packets is not affected by bandwidth. In best effort traffic, packets transmitted as low priority. In queue 1; the BE traffic that has no CIR defined, hence it is all out-profile. It has approximately 25% dropped packets at the ingress queue. In queue 2, CIR=PIR configuration is chosen, so the traffic is considered in-profile. Percentage of dropped packets is less than queue 1 configuration but more than other forwarding classes. In assured traffic; packets are transmitted as default priority but it can be transmitted as high priority with the configuration. In our configuration queue 3 (af forwarding class) traffic transmitted as high priority traffic. In queue 3, dropped packet numbers are less than best effort traffic but more than “h2” forwarding class. In high priority traffic; packets transmitted as high priority, traffic is considered in-profile and dropped packets number is much less than other forwarding class traffic.

For the 1Mbps limited port bandwidth, the graph of the percentage of dropped packets to total packet in ingress queues at can be seen in Figure 4.5.

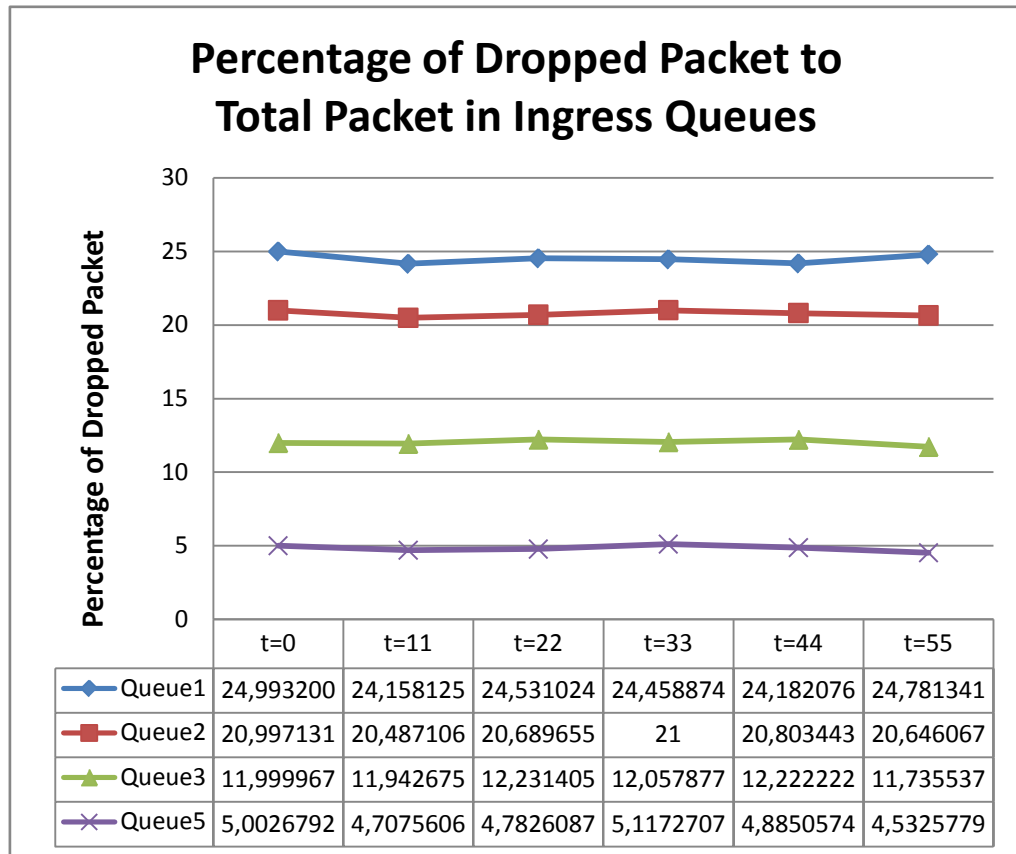


Figure 4.5 Percentage of dropped packets

Difference between percentages of dropped packets at different forwarding classes, beginning at the $t=0$ until $t=55$ sec can be seen in Figure 4.5. For the queue 1 (best effort forwarding class); it has approximately 25% dropped packets in the ingress queue. For the queue 2 (12 forwarding class); it has approximately 21% dropped packets in the ingress queue. Packets are transmitted as low priority in these two forwarding classes. For the queue 3 (af forwarding class); it has approximately 12% dropped packets in the ingress queue. For the queue 5 (h2 forwarding class); it has approximately 5% dropped packets in the ingress queue. Packets are transmitted as high priority in these two forwarding classes, so percentage of dropped packets is much less than the other two forwarding classes. The most important thing in this comparison is the differences between the percentages at the different queues and forwarding classes. The percentage values are not much important because percentages can change according to the port bandwidth, traffic congestion state etc.

On these grounds with QoS configurations, the important traffics such as network control traffic or delay/jitter sensitive traffic can be transmitted without drop, delay or jitter through the MPLS network.

CHAPTER FIVE

CONCLUSION

In this thesis, data transmitting in Multi Protocol Label Switching (MPLS) and Quality of Service (QoS) effects are discussed. The aim of this thesis was to analyze data transmitting through the MPLS network and effects of QoS to the packets with the different QoS configurations.

In the post-bubble economy, the most important objectives for service providers moreover, carriers are to:

- reduce operating costs
- preserve existing services
- introduce new services efficiently

MPLS technology has proven its value for delivering new services while at the same time allowing migration from old to new networks. By converging new and legacy network services over an MPLS network, providers and carriers can introduce efficiencies that promise great savings in operating costs. As a result, MPLS is well into mainstream deployment in networks around the world, as a standard backbone technology for converged networks.

However, MPLS has proven to be an extremely complex technology, encompassing a wide range of functionality and applications. Vendors, who develop MPLS technology, as well as organizations looking to deploy MPLS, must consider the complexity of the technology, its continually evolving state, and its impact on network performance and scalability.

To manage the complexity of MPLS, a wide range of protocols, services, applications, and hardware must be tested and validated. For network managers and vendors of MPLS-related products and services, a comprehensive and well-designed conformance and performance testing solution is crucial to the success of MPLS technology.

Quality of Service refers to the ability of a network to recognize the differing service requirements of different application traffic flowing through it and to comply with service level agreements negotiated for each of the application services. In today's service delivery climate, all service providers are expected to offer personalized media rich application services. In order to reduce operational costs and to enrich user experience, providers are migrating toward offering all killer applications over a single IP/MPLS core infrastructure. Quality of Service features enable networks to handle traffic for efficient multi-service delivery.

MPLS is a standards-based technology that can improve network performance and QoS for selected traffic. Service providers enhance the variety of the services, the class of the services and customer portfolio through MPLS. Service providers can market various services such as Metro Ethernet, IP/TV, Layer-3 and Layer-2 VPNs, VOIP. MPLS offers multiple classes of service, each associated with different types of traffic. For instance, an enterprise's mission-critical applications (such as VOIP applications) might be in a gold class of service, less-important applications might be in a silver service, recreational applications (such as games, instant messaging, and P2P) might be in a best effort service.

QoS mechanisms present the industry with a true end-to-end QoS solution, allowing providers to guarantee SLA compliance. MPLS makes QoS services more scalable and extends their reach end-to-end across multiple technologies.

As deeply focused in this thesis, most important point about MPLS is the Quality of Service. If a service provider could not offer differentiated services to the

customers and provide desired QoS, the satisfaction goes down and the subscription will not continue.

To observe the importance of QoS for MPLS network, different QoS configurations are performed in the thesis. For all QoS levels, 1 Mbps port bandwidth is used, in order to investigate only the effects of QoS levels. Because with a unlimited bandwidth condition, there is no congestion state and all packets (high or low priority) can be transmitted without drops. To analyze QoS effects, firstly Queue 1 configuration was applied and the results were monitored. In these circumstances, it was seen that the low priority packets are transmitted so it caused packets being dropped. Secondly, Queue 2 configuration was applied to software, and then result is monitored. Also in this configuration, packets are dropped since they are transmitted with low priority. However, the percentage of dropped packets is less than Queue 1 configuration. After that; Queue 3 and Queue 5 configurations were applied and the results were monitored. In these configurations, packets are transmitted as high priority so the percentage of dropped packets is less than the other QoS configurations; especially in Queue 5 configuration. Least percentage of dropped packets is obtained in Queue 5 configuration.

The analyzing results are provided from live network, a simulation program or a lab is not used in the thesis. Therefore, results are more correct than the results of simulation programs and labs, since the live network conditions are much more realistic.

For further study, other parameters, which change QoS effects, such as Committed Information Rate (CIR) and Peak Information Rate (PIR) can be investigated. At different CIR and PIR values configurations, QoS effects can be analyzed.

REFERENCES

- Alvarez, S. (2006). *QoS for IP/MPLS networks*. USA: Cisco Press
- Alwayn, V. (2002). *Advanced MPLS design and implementation*. USA: Cisco Press
- Barreiros, M., & Lundqvist, P. (2011). *QoS-Enabled networks: Tools and foundations*. UK: John Wiley & Sons, Ltd
- Bingöl, E. (2005). *QoS for real-time IP traffic*. Master Thesis, Department of Electrical and Electronics Engineering, Dokuz Eylül University
- Blake, S., et al. (1998). *An architecture for differentiated services*. RFC2475, Retrieved March 3, 2011, from <http://www.ietf.org/rfc/rfc2475>
- Duysburgh, B., Lambrecht, T., Turck F, Dhoedt, B., & Demeester, P. (January 30, 2004). *An active networking based service for media transcoding in multicast sessions*. Retrieved April 11, 2011, from <http://ieeexplore.ieee.org/>
- Ergül, U. (2004). *Quality of service in VoIP communication*. Master Thesis, Department of Electrical and Electronics Engineering, Dokuz Eylül University
- Fineberg, V., Chen, C., Xiao, X. (December 10, 2002). *An end-to-end QoS architecture with the MPLS-based core*. Retrieved April 9, 2011, from <http://ieeexplore.ieee.org/>
- Flannagan, M., et al. (2003). *Cisco catalyst QoS: quality of service in campus networks*. USA: Cisco Press
- Grossman, D. (2002). *New terminology and clarifications for diffServ*. RFC3260, Retrieved March 9, 2011, from <http://www.ietf.org/rfc/rfc3260>
- Kaarthick, B., Nagarajan, N., Rajeev, S., Angeline, R.J. (February 2, 2009). *Improving QoS of audio and video packets in MPLS using network processors*. Retrieved April 9, 2011, from <http://ieeexplore.ieee.org/>

- Nichols, K., et al. (1998). *Definition of the differentiated services field (DS Field) in the IPv4 and IPv6 headers*. RFC 2474, Retrieved February 16, 2011, from <http://www.ietf.org/rfc/rfc2474>
- Osborne, E., & Simha, A. (2003). *Traffic engineering with MPLS*. USA: Cisco Press
- Pepelnjak, I., et al. (2003). *MPLS and VPN architecture, Volume II*. USA: Cisco Press
- Rouhana, N., Horlait, E. (August 6, 2002). *Differentiated services and integrated services use of MPLS*. Retrieved April 10, 2011, from <http://ieeexplore.ieee.org/>
- Saad, T., et al. (August 6, 2002). *DiffServ-enabled adaptive traffic engineering over MPLS*. Retrieved April 10, 2011, from <http://ieeexplore.ieee.org/>
- Szigeti, T. (2005). *End-to-End QoS network design*. USA: Cisco Press
- Vegesna, S. (2001). *IP quality of service*. USA: Cisco Press
- Wang, Z. (2001). *Internet QoS: Architectures and mechanisms for quality of service*. USA: Morgan Kaufmann
- Welzl, M. (2005). *Network congestion control*. England: John Wiley & Sons, Ltd
- Xipeng, X., & Lionel M. Ni. (1999). *Internet Qos: The big picture*. IEEE Network 1999, Retrieved January 21, 2011, from <http://www.cs.columbia.edu/~zwb/my/oral/qos/netmag/qos.pdf>
- Rosen, E., et al. (2001). *Multiprotocol label switching architecture*. RFC 3031, Retrieved January 10, 2011, from <http://www.ietf.org/rfc/rfc3031>
- Zhang, D., & Ionescu, D. (July 30, 2007). *QoS performance analysis in deployment of diffServ aware MPLS traffic engineering*. Retrieved April 9, 2011, from <http://ieeexplore.ieee.org/>

APPENDIX A- Properties of Service Router

MPLS Label Stack

A stack can carry several labels, organized in a last in/first out order. The top of the label stack appears first in the packet and the bottom of the stack appears last



Figure A.1 Label organization

Service Router OS uses labels for MPLS, RSVP-TE, and LDP, as well as packet-based services such as VLL and VPLS. Label values 16 through 1,048,575 are defined as follows:

- Label values 16 through 31 are reserved for future use.
- Label values 32 through 1,023 are available for static assignment.
- Label values 1,024 through 2,047 are reserved for future use.
- Label values 2,048 through 18,431 are statically assigned for services.
- Label values 28,672 through 131,071 are dynamically assigned for both MPLS and services.
- Label values 131,072 through 1,048,575 are reserved for future use.

Forwarding Classes

Service Router routers support multiple forwarding classes and class-based queuing, so the concept of forwarding classes is common to all of the Service Router QoS policies. Each forwarding class (also called Class of Service (CoS)) is important only in relation to the other forwarding classes. A forwarding class provides network elements a method to weigh the relative importance of one packet over another in a different forwarding class. Queues are created for a specific forwarding class to determine the manner in which the queue output is scheduled into the switch fabric and the type of parameters the queue accepts. The forwarding class of the packet,

along with the in-profile or out-of-profile state, determines how the packet is queued and handled (the per hop behavior (PHB)) at each hop along its path to a destination egress point. Service Router supports eight (8) forwarding classes (Table A.1).

Table A.1 Forwarding classes

FC-ID	FC Name	FC Designation	DiffServ Name	Class Type	Notes
7	Network Control	NC	NC2	High-Priority	Intended for network control traffic.
6	High-1	H1	NC1		Intended for a second network control class or delay/jitter sensitive traffic.
5	Expedited	EF	EF		Intended for delay/jitter sensitive traffic.
4	High-2	H2	AF4		Intended for delay/jitter sensitive traffic.
3	Low-1	L1	AF2	Assured	Intended for assured traffic. Also is the default priority for network management traffic.
2	Assured	AF	AF1		Intended for assured traffic.
1	Low-2	L2	CS1	Best Effort	Intended for BE traffic.
0	Best Effort	BE	BE		

Note that Table A.1 presents the default definitions for the forwarding classes. The forwarding class behavior, in terms of ingress marking interpretation and egress marking, can be changed by a Network QoS Policies. All forwarding class queues support the concept of in-profile and out-of-profile although some forwarding classes by default do not differentiate for egress marking.

The forwarding classes can be classified into three class types (Figure A.2):

- High-priority/Premium
- Assured
- Best effort

High-Priority Classes

The high-priority forwarding classes are Network Control (nc), Expedited (ef), High 1 (h1), and High 2 (h2). High-priority forwarding classes are always serviced at congestion points over other forwarding classes. With a strict PHB at each network

hop, service latency is mainly affected by the amount of high-priority traffic at each hop. These classes are intended to be used for network control traffic or for delay or jitter-sensitive services.

If the service core network is over-subscribed, a mechanism to traffic engineer a path through the core network and reserve bandwidth must be used to apply strict control over the delay and bandwidth requirements of high-priority traffic.

If the core network has sufficient bandwidth, it is possible to effectively support the delay and the jitter characteristics of high-priority traffic without utilizing traffic engineered paths, as long as the core treats high-priority traffic with the proper PHB.

Assured Classes

The assured forwarding classes are Assured (af) and Low 1 (ll). Assured forwarding classes provide services with a committed rate and a peak rate much like Frame Relay. Packets transmitted through the queue at or below the committed transmission rate are marked in-profile.

If the core service network has sufficient bandwidth along the path for the assured traffic, all aggregate in profile service packets will reach the service destination. Packets transmitted out the service queue that are above the committed rate will be marked out-of-profile. When an assured out-of-profile service packet is received at a congestion point in the network, it will be discarded before in profile assured service packets.

Multiple assured classes are supported with relative weighting between them. In DiffServ, the code points for the various Assured classes are AF4, AF3, AF2 and AF1. Typically, AF4 has the highest weight of the four and AF1 the lowest. The Assured and Low 1 classes are differentiated based on the default DSCP mappings. Note that all DSCP and EXP mappings can be modified by the user.

Best-Effort Classes

The best-effort classes are Low 2 (l2) and Best-Effort (be). The best-effort forwarding classes have no delivery guarantees. All packets within this class are treated, at best, like out-of-profile assured service packets.

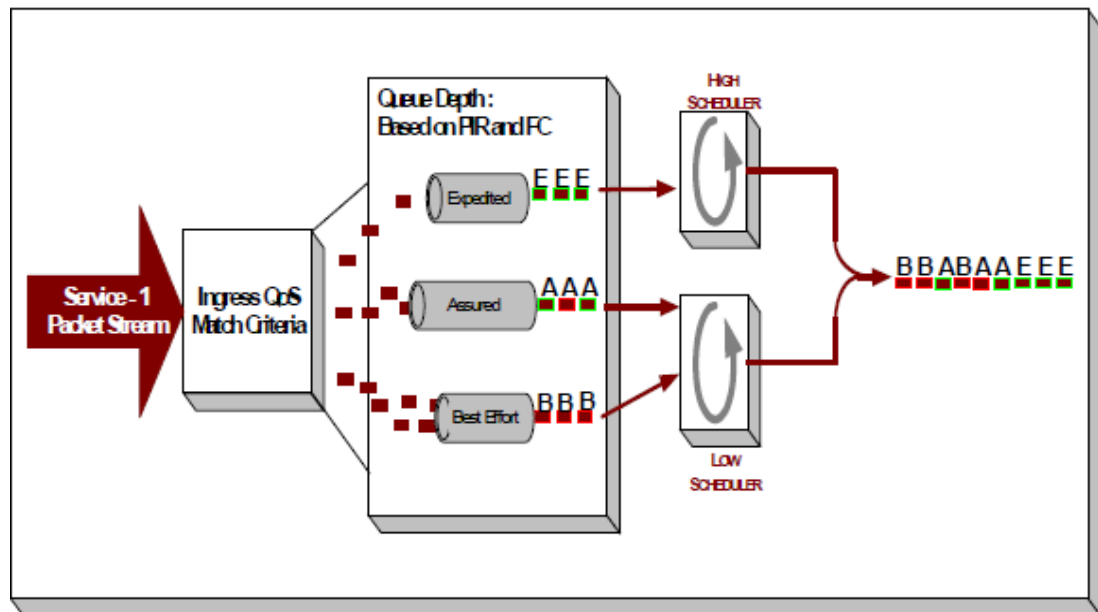


Figure A.2 Traffic queuing model for 3 queues and 3 classes

QoS Policies

There are several types of QoS policies:

- Service ingress
- Service egress
- Network (for ingress and egress)
- Network queue (for ingress and egress)
- ATM traffic descriptor profile
- Scheduler
- Shared queue
- Slope

Service ingress QoS policies are applied to the customer-facing Service Access Points (SAPs) and map traffic to forwarding class queues on ingress. The mapping of traffic to queues can be based on combinations of customer QoS marking (IEEE 802.1p bits, DSCP, and TOS precedence), IP and MAC criteria. The characteristics of the forwarding class queues are defined within the policy as to the number of forwarding class queues for unicast traffic and the queue characteristics. There can be up to eight (8) unicast forwarding class queues in the policy; one for each forwarding class.

Service egress QoS policies are applied to SAPs and map forwarding classes to service egress queues for a service. Up to 8 queues per service can be defined for the 8 forwarding classes. A service egress QoS policy also defines how to remark the forwarding class to IEEE 802.1p bits in the customer traffic.

Network QoS policies are applied to IP interfaces. On ingress, the policy maps incoming DSCP and EXP values to forwarding class and profile state for traffic received from the core network. On egress, the policy maps forwarding class and profile state to DSCP and EXP values for traffic to be transmitted into the core network.

Network Queue policies are applied on egress to network ports and channels and on ingress to MDAs. The policies define the forwarding class queue characteristics for these entities.

For network ingress, Table A.2 lists the default mapping of DSCP name and LSP EXP values to forwarding class and profile state for the default network QoS policy.

Table A.2 Default network QoS policy DSCP to forwarding class mappings

Ingress DSCP		Forwarding Class			
dscp-name	dscp-value (binary - decimal)	FC ID	Name	Label	Profile State
Default ^a		0	Best-Effort	be	Out
ef	101110 - 46	5	Expedited	ef	In
nc1	110000 - 48	6	High-1	h1	In
nc2	111000 - 56	7	Network Con- trol	nc	In
af11	001010 - 10	2	Assured	af	In
af12	001100 - 12	2	Assured	af	Out
af13	001110 - 14	2	Assured	af	Out
af21	010010 - 18	3	Low-1	l1	In
af22	010100 - 20	3	Low-1	l1	Out
af23	010110 - 22	3	Low-1	l1	Out
af31	011010 - 26	3	Low-1	l1	In
af32	011100 - 28	3	Low-1	l1	Out
af33	011110 - 30	3	Low-1	l1	Out
af41	100010 - 34	4	High-2	h2	In
af42	100100 - 36	4	High-2	h2	Out
af43	100110 - 38	4	High-2	h2	Out

For network egress, traffic remarking in the default network QoS policy is disabled. Table A.3 lists the default mapping of forwarding class to DSCP name and LSP EXP values.

Table A.3 Default network QoS policy egress marking

FC-ID	FC Name	FC Label	DiffServ Name	Egress DSCP Marking		Egress LSP EXP Marking	
				In-Profile Name	Out-of-Profile Name	In-Profile	Out-of-Profile
7	Network Control	nc	NC2	nc2 111000 - 56	nc2 111000 - 56	111 - 7	111 - 7
6	High-1	h1	NC1	nc1 110000 - 48	nc1 110000 - 48	110 - 6	110 - 6
5	Expedited	ef	EF	ef 101110 - 46	ef 101110 - 46	101 - 5	101 - 5
4	High-2	h2	AF4	af41 100010 - 34	af42 100100 - 36	100 - 4	100 - 4
3	Low-1	l1	AF2	af21 010010 - 18	af22 010100 - 20	011 - 3	010 - 2
2	Assured	af	AF1	af11 001010 - 10	af12 001100 - 12	011 - 3	010 - 2
1	Low-2	l2	CS1	cs1 001000 - 8	cs1 001000 - 8	001 - 1	001 - 1
0	Best Effort	be	BE	be 000000 - 0	be 000000 - 0	000 - 0	000 - 0

APPENDIX B- Basic Configurations in the SR/ESS Services for QoS

To create a network QoS policy:

CLI Syntax: config>qos#

network policy-id

description description-string

scope {exclusive|template}

egress

remarking

fc {be|l2|af|l1|h2|ef|h1|nc}

dot1p-in-profile dot1p-priority

dot1p-out-profile dot1p-priority

dscp-in-profile dscp-name

dscp-out-profile dscp-name

lsp-exp-in-profile mpls-exp-value

lsp-exp-out-profile mpls-exp-value

ingress

default-action fc {be|l2|af|l1|h2|ef|h1|nc}

profile{in|out}

dot1p dot1p-priority fc {fc-name} profile {in|out}

dscp dscp-name fc {be|l2|af|l1|h2|ef|h1|nc}
profile{in|out}

ler-use-dscp

lsp-exp lsp-exp-value fc fc-name profile {in|out}

The following displays the network policy configuration:

```
>config>qos# info

network 600 create

description "Network Egress Policy"

ingress

default-action fc ef profile in

exit

egress

remarking

exit

exit
```

To apply network policies to router interfaces:

CLI Syntax: config>router

```
interface interface-name

qos network-policy-id
```

The following output displays the configuration for router interface with network policy 600 applied to the interface.

```
XX>config>router# info

interface "name of the router"

address X.X.X.X/X

qos 600

exit
```

To create a network queue QoS policy:

CLI Syntax: config>qos

network-queue policy-name

description description-string

fc fc-name

multicast-queue queue-id

queue queue-id

queue queue-id [multipoint] [queue-type]

cbs percent

high-prio-only percent

mbs percent

port-parent [weight weight] [level level] [cir-weight cir-weight] [cir-level cir-level]

rate percent[cir percent]

The following displays the network queue policy configuration:

```
>config>qos# network-queue default
```

```
>config>qos>network-queue# info detail
```

```
description "Default network queue QoS policy."
```

```
queue 1 create
```

```
mbs 50
```

```
cbs 1
```

high-prio-only 10

exit

queue 2 create

rate 100 cir 25

mbs 50

cbs 3

high-prio-only 10

exit

queue 3 create

rate 100 cir 25

mbs 50

cbs 1

high-prio-only 10

exit

queue 4 create

rate 100 cir 25

mbs 25

cbs 3

high-prio-only 10

exit

queue 5 create

rate 100 cir 100

mbs 50

cbs 1

high-prio-only 10

exit

queue 6 create

rate 100 cir 100

mbs 50

cbs 1

high-prio-only 10

exit

queue 7 create

rate 100 cir 10

mbs 25

cbs 3

high-prio-only 10

exit

queue 8 create

rate 100 cir 10

mbs 25

cbs 3

high-prio-only 10

exit

7.6.3.1 To apply a network queue policy to an MDA network ingress port:

CLI Syntax: config>card

mda mda-slot

network

ingress

queue-policy name

To apply a network queue policy to an Ethernet port:

CLI Syntax: config>port#

ethernet

network

queue-policy name

To create Service Egress and Ingress QoS Policies

Configuring and applying QoS policies is optional. If no QoS policy is explicitly applied to a SAP or IP interface, a default QoS policy is applied.

The following displays an egress QoS policy configuration:

XX>config>qos# info

#-----

echo "QoS Policy Configuration"

#-----

...

sap-egress **105** create

description "**SAP egress policy**"

queue **1** create

exit

queue **2** create

exit

```

queue 3 expedite create
    parent test1
exit
fc af create
    queue 1
    exit
    fc ef create
        queue 2
    exit
exit

```

The following displays an service ingress policy configuration:

```

XX>config>qos>sap-ingress# info
sap-ingress 100 create
    description "Used on VPN sap"
        queue 1 create
        exit
        queue 2 multipoint create
        exit
        queue 10 create
            parent VPN_be
            rate 11000
        exit
        queue 12 create
            parent VPN_priority

```

```

                                rate 11000

                                exit

                                queue 13 create

                                parent VPN_reserved

                                rate 1

                                exit
                                queue 15 create

                                parent VPN_video

                                rate 1500 cir 1500

                                exit
                                ...
                                #-----
                                XX>config>qos#

```

SAP Ingress Forwarding Class (FC)

The following displays a forwarding class and precedence configurations:

```

xx>config>qos# info

#-----

fc af create

    queue 12

    broadcast-queue 22

    multicast-queue 22

    unknown-queue 22

exit

fc be create

    queue 10

```


broadcast-queue 20

multicast-queue 20

unknown-queue 20

exit

fc ef create

queue 13

broadcast-queue 23

multicast-queue 23

unknown-queue 23

exit

fc h1 create

queue 15

broadcast-queue 25

multicast-queue 25

unknown-queue 25

exit

fc h2 create

queue 16

broadcast-queue 26

multicast-queue 26

unknown-queue 26

exit

```

fc nc create

queue 17

broadcast-queue 27

multicast-queue 27

unknown-queue 27

exit

prec 0 fc be

prec 2 fc af

prec 3 fc ef

prec 5 fc h1

prec 6 fc h2

prec 7 fc nc

#-----

```

Service Ingress IP Match Criteria

When specifying SAP ingress match criteria, only one match criteria type can be configured in the SAP ingress QoS policy.

The following displays an ingress IP criteria configuration:

```

XX>config>qos# info

#-----

echo "QoS Policy Configuration"

#-----

```

```
sap-ingress 100 create

    ip-criteria

        entry 10 create

            description "Entry 10-FC-AF"

            match protocol 6

                src-ip 10.10.10.103/24

            exit

            action fc af priority high

        exit

    entry 20 create

        description "Entry 20-FC-BE"

        match protocol 17

            dst-port eq 255

        exit

        no action

    exit

exit

#-----

XX>config>qos#
```

Service Ingress MAC Match Criteria

Both IP criteria and MAC criteria cannot be configured in the same SAP ingress QoS policy.

The following displays an ingress MAC criteria configuration:

```
XX>config>qos# info
#-----
echo "QoS Policy Configuration"
#-----

    sap-ingress 101 create
        mac-criteria
            entry 10 create
                description "Entry10-low prio"
                match
                    dst-mac 04-67-ff-00-00-01 ff-ff-ff-ff-ff-ff
                    dot1p 7 7
                exit
            action fc be priority low
        exit
    exit
exit
XX>config>qos#
```

The following sample output displays an IES service configuration with SAP ingress policy 100 and SAP egress 105 applied to the SAP.

```
XX>config>service# info
-----

    ies 88 customer 8 vpn 88 create
```

```
interface "Sector A" create
  sap 1/1/1.2.2 create
    ingress
      qos 100
    exit
    egress
      qos 105
    exit
  exit
exit
no shutdown
exit
```

APPENDIX C- GLOSSARY

Term	Definition
QoS	Quality of Service
ATM	Asynchronous Transfer Mode
AF	Assured Forwarding
BA	Behavior Aggregate
BE	Best Effort
BGP	Border Gateway Protocol
CE	Customer Edge
CIR	Committed Information Rate
CoS	Class of Service
CQ	Custom Queuing
CSPF	Constrained Shortest Path First
DiffServ	Differentiated Services
DSCP	Differentiated Services Code Point
EF	Expedited Forwarding
ELER	Egress Label Edge Router
FEC	Forwarding Equivalence Class
FIFO	First-in First-out
IETF	Internet Engineering Task Force
IGP	Interior Gateway Protocols
IntServ	Integrated Services
IS-IS	Intermediate System to Intermediate System Protocol.
ISP	Internet Service Provider
LDP	Label Distribution Protocol
LER	Label Edge Router
LIFO	Last-In-First-Out
LSP	Label Switched Path
LSR	Label Switch Router
L2TP	Layer 2 Tunneling Protocol
MPLS	Multiprotocol Label Switching

OSPF	Open Shortest Path First
PE	Provider Edge
PHB	Per-Hop Behaviors
PIR	Peak Information Rate
PQ	Priority Queuing
RED	Random Early Detection
RSVP	Resource Reservation Protocol
RSVP-TE	Resource Reservation Protocol-Traffic Engineering
SAP	Service Access Point
SLA	Service Level Agreements
SONET	Synchronous Optical Network
SP	Service Provider
TDM	Time Division Multiplexing
TE	Traffic Engineering
ToS	Type of Service
TTL	Time-to-Live
WFQ	Weighted Fair Queuing
WRED	Weighted Random Early Detection
VLAN	Virtual Local Area Network
VLL	Virtual Leased Lines
VoIP	Voice over IP
VPDN	Virtual Private Dial Networks
VPLS	Virtual Private LAN Segments
VPN	Virtual Private Network
VPRN	Virtual Private Routed Networks