

QUALITY of SERVICE in VoIP COMMUNICATION

**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**

151200

by

Utku ERGÜL

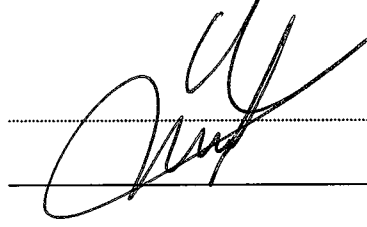
151200

August, 2004

İZMİR


M.Sc THESIS EXAMINATION RESULT FORM

We certify that we have read this thesis and “**QUALITY OF SERVICE IN VOIP COMMUNICATION**” completed by **UTKU ERGUL** under supervision of **ASSOC. PROF. DR. ZAFER DİCLE** and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

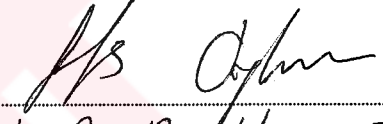


Assoc.Prof Dr. Zafer DİCLE

Supervisor

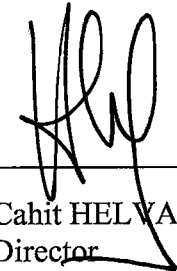

Dr. D. Zafar CEBİ

(Committee Member)


Yrd. Doç. Dr. Hacer Ozturk

(Committee Member)

Approved by the
Graduate School of Natural and Applied Sciences


Prof. Dr. Cahit HELVACI
Director

ACKNOWLEDGEMENTS

I would like to give my sincere thanks to my supervisor, Assoc. Prof. Dr. Zafer Dicle for his guidance, advice and encouragement along the fulfillment of this project. I would like to thank to Tolga Narbay for his discipline, creativity and assistance during this teamwork.

I also say "Thank You" my wife, Hatice, for her support and tolerance during this work and all my life, and to all the friends who have been concerned about us during our study.

ABSTRACT

The absence of solutions ensuring Quality of Service (QoS) has been a deterrent to the widespread adoption of Voice over Internet Protocol (VoIP). Potential users often think that speech quality won't be as good as what they are accustomed to—the familiar public switched telephone network (PSTN). Solutions that improve absolute voice quality and that enable objective quality measurements can easily be incorporated into Service Level Agreements (SLAs). With these solutions in hand, the reluctance of potential users can begin to evaporate, and the adoption of VoIP will stand poised for a period of explosive growth.

Voice quality is subjective because it's a measure of the intelligibility and clarity of speech as perceived by the listener. However, perceptions drive decisions. VoIP service providers must be extremely sensitive to the perceptions of their customers, because a decision to change service can be precipitated from such negative perceptions as;

- When a user perceives unacceptable instantaneous quality, the user is likely to terminate the call prematurely.
- If a user perceives overall poor quality after completing a call, there is likely to be a harboring of residual dissatisfaction.
- If service providers achieve quality by overprovisioning their networks, the resulting high costs undermine the user's perception of value, despite excellent voice quality.

This thesis focuses on methodologies used in IP networks, which aim to ensure a certain level of service and satisfy the customer needs.

Keywords : QoS , VoIP , PSTN , SLA , IP

ÖZET

Servis kalitesinin garanti edilememesinden dolayı,ses taşımasının Internet Protokolü (IP) üzerinden yapılması yeterince benimsenememiştir.Potansiyel kullanıcılar genellikle devre anahtarlama sabit telefon şebekesi (PSTN) üzerinden aldıkları kalitede servis alamayacaklarını düşündükleri için bu servisi kullanmakta tereddüt etmektedirler.

Ses kalitesini artırıcı ve kaliteyi ölçebilmeyi sağlayan çözümler üretilmesi kullanıcı ile servis veren firma arasında Servis Seviyesi Anlaşmaları (SLAs) yapılmasını olanaklı kılmıştır.Bu çözümlerin çerçevesinde kullanıcıların isteksizliği kırılmış ve VoIP kullanımında büyük bir gelişim yaşanmıştır.

Ses kalitesi göreceli bir kavramdır çünkü ölçütü dinleyicinin algıladığı berraklık ve anlaşılabilirlik.Bununla birlikte algı,kararları yönlendirir.Internet protokolü üzerinden ses iletimi (VoIP) servis sağlayıcıları, müşterilerinin algılarına karşı son derece duyarlı olmak zorundadırlar, çünkü verilen servis kalitesindeki bir değişiklik,aşağıdaki negatif sonuçları doğurabilir;

- Eğer kullanıcı kabul edilemez bir servis kalitesiyle karşılaşır,çağrıyı erken sonlandırabilir.
- Kullanıcı çağrıyı sonlandırdıktan sonra,düşük bir servis kalitesi aldığını algılayarsa, servisle ilgili memnuniyetsizliğe varabilir ve bir sonraki görüşmesi için aynı servisi kullanmayabilir.
- Servis sağlayıcı tarafından bakıldığında,eğer servis kalitesinin yükseltilmesi için şebekelerinde iyileşme sağlayacak yatırımlar yaptıklarında,bunu kullanıcıya fiyat olarak yansıtırlarsa,kullanıcı mükemmel kalitede bir servis almasına rağmen yüksek fiyattan dolayı servisi kullanmaktan vazgeçebilir.

Bu tezde odaklanılan nokta,IP şebekelerinde belirli bir servis kalitesinin garanti edilebilmesine olanak veren ve böylece müşteri ihtiyaçlarının tatmin edilmesini sağlayan metodolojilerdir.

Anahtar Kelimeler : Servis Kalitesi , Internet Protokolü Üzerinden Ses İletimi (VoIP) , Devre Anahtarlamaalı Sabit Telefon şebekesi(PSTN) , Servis Seviyesi Anlaşması (SLA) , Internet Protokolü (IP)



CONTENTS

	Page
Contents.....	IV
List of Tables.....	XI
List of Figures.....	XII

Chapter One INTRODUCTION

1	Introduction.....	1
1.1	Levels of QoS	4
1.1.1	Best-effort service.....	4
1.1.2	Differentiated service.....	5
1.1.3	Guaranteed service.....	5
1.2	IP QoS History.....	7
1.3	Performance Measures.....	9
1.3.1	Bandwidth.....	9
1.3.2	Packet Delay and Jitter	9
1.3.3	Packet Loss	11
1.4	QoS Functions	12
1.4.1	Packet Classifier and Marker.....	12
1.4.2	Traffic Rate Management.....	12
1.4.3	Resource Allocation.....	13

1.4.4	Congestion Avoidance and Packet Drop Policy	13
1.4.5	QoS Signaling Protocol	13
1.4.6	Switching	14
1.4.7	Routing	14
1.5	Layer 2 QoS Technologies	14
1.6	Multiprotocol Label Switching.....	15
1.7	End-to-End QoS.....	15
2	Differentiated Services Architecture	17
2.1	Intserv Architecture	17
2.2	Diffserv Architecture.....	18
2.2.1	Network Boundary Traffic Conditioners.....	23
2.2.2	Classifier	23
2.2.3	Marker.....	24
2.2.4	Metering.....	24
2.2.5	Shaper	24
2.2.6	Dropper	24
2.2.7	PHB.....	24
2.2.8	EF PHB	25
2.2.9	AF PHB	27
2.2.10	Resource Allocation Policy	27
2.2.11	IP Precedence Versus DSCP	29
3	Network Boundary Traffic Conditioners: Packet Classifier, Marker, and Traffic Rate Management	31
3.1	Packet Classification.....	32
3.2	Packet Marking.....	33
3.2.1	IP Precedence.....	33
3.2.2	DSCP	34
3.2.3	The QoS Group.....	34
3.3	The Need for Traffic Rate Management.....	35

3.3.1	The Token Bucket Scheme	36
3.4	Traffic Policing	37
3.4.1	The Traffic Matching Specification	38
3.4.2	The Traffic Measurement Instrumentation	39
3.5	Traffic Shaping	44
4	Per-Hop Behavior: Resource Allocation I	48
4.1	Scheduling for Quality of Service (QoS) Support	49
4.1.1	FIFO Queuing	50
4.1.2	The Max-Min Fair-Share Allocation Scheme	50
4.1.3	Generalized Processor Sharing	53
4.2	Sequence Number Computation-Based WFQ	54
4.3	Flow-Based WFQ	59
4.3.1	WFQ Interaction with RSVP	63
4.3.2	WFQ Implementation	63
4.4	Flow-Based Distributed WFQ (DWFQ)	65
4.5	Class-Based WFQ	66
4.6	Priority Queuing	67
4.7	Custom Queuing	68
4.7.1	How Byte Count Is Used in Custom Queuing	68
4.8	Scheduling Mechanisms for Voice Traffic	70
4.8.1	CBWFQ with a Priority Queue	70
5	Per-Hop Behavior: Resource Allocation II	72
5.1	Modified Weighted Round Robin (MWRR)	72
5.1.1	An Illustration of MWRR Operation	73
5.1.2	MWRR Implementation	83
5.2	Modified Deficit Round Robin (MDRR)	84
5.2.1	An MDRR Example	85
5.3	MDRR Implementation	92
5.3.1	MDRR on the RX	92

5.3.2	MDRR on the TX	92
6	Per-Hop Behavior: Congestion Avoidance and Packet Drop Policy.....	94
6.1	TCP Slow Start and Congestion Avoidance.....	95
6.2	TCP Traffic Behavior in a Tail-Drop Scenario	97
6.3	RED—Proactive Queue Management for Congestion Avoidance.....	98
6.3.1	The Average Queue Size Computation	100
6.3.2	Packet Drop Probability.....	101
6.4	WRED.....	102
6.4.1	WRED Implementation	103
6.5	Flow WRED	103
6.6	ECN	106
6.7	SPD.....	107
7	Integrated Services: RSVP	110
7.1	RSVP	111
7.1.1	RSVP Operation	111
7.1.2	RSVP Components	114
7.1.3	RSVP Messages.....	115
7.2	Reservation Styles	117
7.2.1	Distinct Reservations	117
7.2.2	Shared Reservations.....	118
7.3	Service Types.....	120
7.3.1	Guaranteed Bit Rate.....	120
7.4	RSVP Media Support	122
7.5	RSVP Scalability	122
8	Layer 2 QoS: Interworking with IP QoS	124
8.1	ATM	124
8.1.1	ATM Cell Format	125
8.1.2	ATM QoS	128

8.1.3	ATM Service Classes	128
8.1.4	Cell Discard Strategies	130
8.1.5	VP Shaping	131
8.2	ATM Interworking with IP QoS.....	132
8.3	Frame Relay.....	135
8.3.1	Frame Relay Congestion Control	137
8.3.2	Frame Relay Traffic Shaping (FRTS)	138
8.3.3	VC Traffic Shaping.....	139
8.3.4	Adaptive FRTS	140
8.3.5	FECN/BECN Integration.....	141
8.3.6	Frame Relay Fragmentation	141
8.4	Frame Relay Interworking with IP QoS	144
8.5	The IEEE 802.3 Family of LANs	145
8.5.1	Expedited Traffic Capability	146
9	QoS in MPLS-Based Networks	149
9.1	MPLS	149
9.1.1	Forwarding Component.....	150
9.1.2	Control Component	151
9.1.3	Label Binding for Destination-Based Forwarding	153
9.1.4	Downstream Label Allocation.....	153
9.1.5	Downstream Label Allocation on Demand	154
9.1.6	Upstream Label Allocation.....	154
9.1.7	Label Encapsulation.....	155
9.2	MPLS with ATM.....	157
9.3	MPLS QoS	158
9.4	End-to-End IP QoS	161
9.4.1	LER.....	162
9.5	MPLS VPN.....	163
9.6	MPLS VPN QoS.....	166

9.6.1	Differentiated MPLS VPN QoS	166
9.6.2	Guaranteed QoS.....	169
9.6.3	RSVP at VPN Sites Only.....	169
9.6.4	RSVP at VPN Sites and Diff-Serv Across the Service Provider Backbone 170	
9.6.5	End-to-End Guaranteed Bandwidth.....	170
10	MPLS Traffic Engineering	171
10.1	The Layer 2 Overlay Model	172
10.2	RRR	173
10.3	TE Trunk Definition	176
10.4	TE Tunnel Attributes	177
10.4.1	Bandwidth.....	177
10.4.2	Setup and Holding Priorities.....	177
10.4.3	Resource Class Affinity	178
10.4.4	Path Selection Order	179
10.4.5	Adaptability	179
10.4.6	Resilience.....	179
10.5	Link Resource Attributes	180
10.5.1	Available Bandwidth	180
10.5.2	Resource Class.....	180
10.6	Distribution of Link Resource Information	180
10.7	Path Selection Policy	181
10.8	TE Tunnel Setup.....	182
10.9	Link Admission Control	183
10.10	TE Path Maintenance.....	183
10.11	TE-RSVP	184
10.12	IGP Routing Protocol Extensions.....	185
10.12.1	IS-IS Modifications	186
10.12.2	OSPF Modifications	186

10.13	TE Approaches	186
11	Application of VOIP on DEU's Data Network	187
11.1	Integration of Networks	187
11.2	Configuration Issues	188
11.2.1	Routers Compatibility	190
11.2.2	PBX Integration	190
11.3	Proposed Configuration	192
11.4	About Quality of Service	194



LIST OF TABLES

Table 1. 1 Service Levels and Enabling QoS Functions	6
Table 2. 1 Functional Blocks in the diffserv Architecture.....	20
Table 2. 2 Class Selector DSCP	22
Table 2. 3 AF PHB	23
Table 3. 1 IP Precedence Values and Names	33
Table 3. 2 Marking Traffic Using IP Precedence, DSCP, and QoS Groups	35
Table 3. 3 Comparison Between Policing and Shaping Functions.....	36
Table 3. 4 Comparison of TS implementations: GTS and DTS	47
Table 4. 1 Weights Assigned Based on the IP Precedence Value of a Packet Belonging to an Unreserved (Non-RSVP) Flow	59
Table 4. 2 Flow-Based WFQ Example.....	62
Table 5. 1 Weights Associated with Each Queue.....	73
Table 5. 2 MWRR ToS Class Allocation	83
Table 5. 3 Queues 0–2, Along with Their Associated Weights and Quantum Values....	86
Table 7. 1 Different Reservation Filters, Based on Style and Sender Scope.....	119
Table 9. 1 Comparison Between Downstream and Upstream Label Distribution	155
Table 9. 2 MPLS QoS.....	159
Table 9. 3 End-to-End IP QoS Across an MPLS Network.....	162
Table 9. 4 MPLS VPN Terminology	164
Table 9. 5 MPLS VPN QoS Functions	167
Table 10. 1 Implications of the Low and High Values for the TE Tunnel Setup and Holding Priorities.....	178
Table 10. 2 Policy on Including or Excluding a Link in a TE Tunnel Path Selection ..	179
Table 10. 3 New or Modified RSVP Objects for TE and Their Functions	185

LIST OF FIGURES

Figure 1. 1 Delay Components of a 1500-byte Packet on a Transcontinental U.S. Link with Increasing Bandwidths	11
Figure 2. 1 Diffserv Overview	20
Figure 2. 2 General QoS Operational Model	21
Figure 2. 3 ToS Byte as of RFC 1349	22
Figure 2. 4 DS Byte	22
Figure 2. 5 RSVP Signaling Across a Differentiated Services Network	28
Figure 3. 1 The Evaluation Flow of Rate-Limit Statements	38
Figure 3. 2 Standard Token Bucket for CAR	40
Figure 3. 3 Action Based on the Burst Counter Value	42
Figure 3. 4 CAR Packet Drop Probability	43
Figure 3. 5 Traffic Shaping Operation	45
Figure 3. 6 The Token Bucket Scheme for the Traffic Shaping Function	46
Figure 4. 1 FIFO Queue	50
Figure 4. 2 Resource Allocation for Users A and B	52
Figure 4. 3 Resource Allocation for User C	52
Figure 4. 4 Resource Allocation for Users D and E	53
Figure 4. 5 An Example Illustrating the Byte-by-Byte Round-Robin GPS Scheduler Simulation for FQ	57
Figure 4. 6 Illustration of FQ Scheduler Behavior; Packet D1 Arriving After Packet A1 Is Scheduled	58
Figure 4. 7 Illustration of the Flow-Based WFQ Example	62
Figure 4. 8 Illustration of the Flow-Based WFQ Example (continued).	63

Figure 5. 1 WRR Queues with Their Deficit Counters Before Start of Service.....	74
Figure 5. 2 MWRR After Serving Queue 0 in the First Round.....	75
Figure 5. 3 MWRR After Serving Queue 1 in the First Round.....	76
Figure 5. 4 MWRR After Serving Queue 2 in the First Round.....	77
Figure 5. 5 MWRR After Serving Queue 0 in the Second Round	78
Figure 5. 6 MWRR After Serving Queue 1 in the Second Round	79
Figure 5. 7 MWRR After Serving Queue 2 in the Second Round	80
Figure 5. 8 MWRR After Serving Queue 0 in the Third Round	81
Figure 5. 9 MWRR After Serving Queue 1 in the Third Round	82
Figure 5. 10 Queues 0–2, Along with Their Deficit Counters.....	86
Figure 5. 11 MDRR After Serving Queue 2, Its First Pass	87
Figure 5. 12 MDRR After Serving Queue 0, Its First Pass	88
Figure 5. 13 MDRR After Serving Queue 2, Its Second Pass.....	89
Figure 5. 14 MDRR After Serving Queue 1, Its First Pass	90
Figure 5. 15 MDRR After Serving Queue 0, Its Second Pass.....	91
Figure 6. 1 TCP Congestion Window Showing Slow Start and Congestion Avoidance Operations.....	96
Figure 6. 2 Global Synchronization.....	98
Figure 6. 3 RED Packet Drop Probability	102
Figure 6. 4 Packet Drop Probability with Flow WRED	105
Figure 6. 5 SPD Packet Drop Modes.....	109
Figure 7. 1 Data and Control Flow Information of a Client and Router Running RSVP	112
Figure 7. 2 RSVP Reservation Setup Mechanism	114
Figure 7. 3 Examples of the Three Reservation Filter Styles.....	119
Figure 8. 1 ATM Cell UNI and NNI Header Formats.....	125
Figure 8. 2 Connectivity Between Routers R1 and R2 Across an ATM Network.....	127
Figure 8. 3 Bundling of Multiple VCs in a VP.....	131
Figure 8. 4 IP over ATM	132
Figure 8. 5 ATM VC Bundle: IP Precedence to VC Mapping.....	134

Figure 8. 6 IP-ATM QoS Interworking	135
Figure 8. 7 An Example of a Frame Relay Header.....	136
Figure 8. 8 Use of FECN and BECN Bits	138
Figure 8. 9 Relationship Between Traffic Shaping Parameters.....	140
Figure 8. 10 Frame Relay Fragmentation	143
Figure 8. 11 A Conceptual View of FRF.12 Operation with Multiple PVCs	144
Figure 8. 12 Ethernet and IEEE 802.3 Frame Formats.....	146
Figure 8. 13 802.1Q Frame Showing 802.1p Bits	147
Figure 8. 14 An Ethernet Frame to a Tagged 802.1Q Frame	148
Figure 8. 15 Use of 802.1p in the Absence of VLANs.....	148
Figure 9. 1 MPLS Network	151
Figure 9. 2 MPLS Network Operation	152
Figure 9. 3 MPLS Label in Ethernet and PPP Frame	156
Figure 9. 4 MPLS Label Carried in the VPI/VCI Fields in an ATM Header.....	156
Figure 9. 5 MPLS Label Format.....	157
Figure 9. 6 QoS in an MPLS Network	159
Figure 9. 7 MPLS VPN Differentiated Services (Diff-Serv) QoS	168
Figure 9. 8 Guaranteed MPLS VPN QoS.....	169
Figure 10. 1 Layer 2 Overlay Model for TE.....	173
Figure 10. 2 TE Tunnel from the San Francisco Router to the New York Router	174
Figure 10. 3 Block Diagram of TE Operation	176
Figure 11. 1 DEU's Existing Network	189
Figure 11. 2 DEU's Proposed Network.....	192

CHAPTER ONE

INTRODUCTION

1 Introduction

Service providers and enterprises used to build and support separate networks to carry their voice, video, mission-critical, and non-mission-critical traffic. There is a growing trend, however, toward convergence of all these networks into a single, packet-based Internet Protocol (IP) network.

In this thesis, we will first define what QoS is and why do we need to use QoS functions in an IP network, specially in a VoIP network. And then on Chapter Eleven we will focus on the possible VoIP application on existing data network of Dokuz Eylul University (DEU).

The largest IP network is, of course, the global Internet. The Internet has grown exponentially during the past few years, as has its usage and the number of available Internet-based applications. As the Internet and corporate intranets continue to grow, applications other than traditional data, such as Voice over IP (VoIP) and video-conferencing, are envisioned. More and more users and applications are coming on the Internet each day, and the Internet needs the functionality to support both existing and emerging applications and services. Today, however, the Internet offers only best-effort service. A best-effort service makes no service guarantees regarding when or whether a packet is delivered to the receiver, though packets are usually dropped only during network congestion.

In a network, packets are generally differentiated on a flow basis by the five flow fields in the IP packet header—source IP address, destination IP address, IP protocol

field, source port, and destination port. An individual flow is made of packets going from an application on a source machine to an application on a destination machine, and packets belonging to a flow carry the same values for the five IP packet header flow fields.



To support voice, video, and data application traffic with varying service requirements from the network, the systems at the IP network's core need to differentiate and service the different traffic types based on their needs. With best-effort service, however, no differentiation is possible among the thousands of traffic flows existing in the IP network's core. Hence, no priorities or guarantees are provided for any application traffic. This essentially precludes an IP network's capability to carry traffic that has certain minimum network resource and service requirements with service guarantees. IP quality of service (QoS) is aimed at addressing this issue.

IP QoS functions are intended to deliver guaranteed as well as differentiated Internet services by giving network resource and usage control to the network operator. QoS is a set of service requirements to be met by the network in transporting a flow. QoS provides end-to-end service guarantees and policy-based control of an IP network's performance measures, such as resource allocation, switching, routing, packet scheduling, and packet drop mechanisms.

The following are some main IP QoS benefits:

It enables networks to support existing and emerging multimedia service/application requirements. New applications such as Voice over IP (VoIP) have specific QoS requirements from the network.

It gives the network operator control of network resources and their usage.

It provides service guarantees and traffic differentiation across the network. It is required to converge voice, video, and data traffic to be carried on a single IP network.

It enables service providers to offer premium services along with the present best-effort Class of Service (CoS). A provider could rate its premium services to customers as Platinum, Gold, and Silver, for example, and configure the network to differentiate the traffic from the various classes accordingly.

It enables application-aware networking, in which a network services its packets based on their application information within the packet headers.

It plays an essential role in new network service offerings such as Virtual Private Networks (VPNs).

1.1 Levels of QoS

Traffic in a network is made up of flows originated by a variety of applications on end stations. These applications differ in their service and performance requirements. Any flow's requirements depend inherently on the application it belongs to. Hence, understanding the application types is key to understanding the different service needs of flows within a network.

The network's capability to deliver service needed by specific network applications with some level of control over performance measures—that is, bandwidth, delay/jitter, and loss—is categorized into three service levels:

1.1.1 Best-effort service

Basic connectivity with no guarantee as to whether or when a packet is delivered to the destination, although a packet is usually dropped only when the router input or output buffer queues are exhausted.

Best-effort service is not really a part of QoS because no service or delivery guarantees are made in forwarding best-effort traffic. This is the only service the Internet offers today.

Most data applications, such as File Transfer Protocol (FTP), work correctly with best-effort service, albeit with degraded performance. To function well, all applications require certain network resource allocations in terms of bandwidth, delay, and minimal packet loss.

1.1.2 Differentiated service

In differentiated service, traffic is grouped into classes based on their service requirements. Each traffic class is differentiated by the network and serviced according to the configured QoS mechanisms for the class. This scheme for delivering QoS is often referred to as COS.

Note that differentiated service doesn't give service guarantees per se. It only differentiates traffic and allows a preferential treatment of one traffic class over the other. For this reason, this service is also referred as soft QoS.

This QoS scheme works well for bandwidth-intensive data applications. It is important that network control traffic is differentiated from the rest of the data traffic and prioritized so as to ensure basic network connectivity all the time.

1.1.3 Guaranteed service

A service that requires network resource reservation to ensure that the network meets a traffic flow's specific service requirements.

Guaranteed service requires prior network resource reservation over the connection path. Guaranteed service also is referred to as hard QoS because it requires rigid guarantees from the network.

Path reservations with a granularity of a single flow don't scale over the Internet backbone, which services thousands of flows at any given time. Aggregate reservations, however, which call for only a minimum state of information in the Internet core routers, should be a scalable means of offering this service.

Applications requiring such service include multimedia applications such as audio and video. Interactive voice applications over the Internet need to limit latency to 100 ms to meet human ergonomic needs. This latency also is acceptable to a large spectrum

of multimedia applications. Internet telephony needs at a minimum an 8-Kbps bandwidth and a 100-ms round-trip delay. The network needs to reserve resources to be able to meet such guaranteed service requirements.

Layer 2 QoS refers to all the QoS mechanisms that either are targeted for or exist in the various link layer technologies. Chapter 8, "Layer 2 QoS: Interworking with IP QoS," covers Layer 2 QoS. Layer 3 QoS refers to QoS functions at the network layer, which is IP. Table 1.1 outlines the three service levels and their related enabling QoS functions at Layers 2 and 3. These QoS functions are discussed in detail in the rest of this book.(Shenker et al.,1999)

Table 1.1 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

1.2 IP QoS History

IP QoS is not an afterthought. The Internet's founding fathers envisioned this need and provisioned a Type of Service (ToS) byte in the IP header to facilitate QoS as part of the initial IP specification. It described the purpose of the ToS byte as follows:

The Type of Service provides an indication of the abstract parameters of the quality of service desired. These parameters are to be used to guide the selection of the actual service parameters when transmitting a datagram through the particular network.

Until the late 1980s, the Internet was still within its academic roots and had limited applications and traffic running over it. Hence, ToS support wasn't necessarily important, and almost all IP implementations ignored the ToS byte. IP applications didn't specifically mark the ToS byte, nor did routers use it to affect the forwarding treatment given to an IP packet.

The importance of QoS over the Internet has grown with its evolution from its academic roots to its present commercial and popular stage. The Internet is based on a connectionless end-to-end packet service, which traditionally provided best-effort means of data transportation using the Transmission Control Protocol/Internet Protocol (TCP/IP) Suite. Although the connectionless design gives the Internet its flexibility and robustness, its packet dynamics also make it prone to congestion problems, especially at routers that connect networks of widely different bandwidths. The congestion collapse problem was discussed by John Nagle during the Internet's early growth phase in the mid-1980s.

The initial QoS function set was for Internet hosts. One major problem with expensive wide-area network (WAN) links is the excessive overhead due to small Transmission Control Protocol (TCP) packets created by applications such as telnet and rlogin. The Nagle algorithm, which solves this issue, is now supported by all IP host

implementations. The Nagle algorithm heralded the beginning of Internet QoS-based functionality in IP.

In 1986, Van Jacobson developed the next set of Internet QoS tools, the congestion avoidance mechanisms for end systems that are now required in TCP implementations. These mechanisms—slow start and congestion avoidance—have helped greatly in preventing a congestion collapse of the present-day Internet. They primarily make the TCP flows responsive to the congestion signals (dropped packets) within the network. Two additional mechanisms—fast retransmit and fast recovery—were added in 1990 to provide optimal performance during periods of packet loss.

Though QoS mechanisms in end systems are essential, they didn't complete the end-to-end QoS story until adequate mechanisms were provided within routers to transport traffic between end systems. Hence, around 1990 QoS's focus was on routers. Routers, which are limited to only first-in, first-out (FIFO) scheduling, don't offer a mechanism to differentiate or prioritize traffic within the packet-scheduling algorithm. FIFO queuing causes tail drops and doesn't protect well-behaving flows from misbehaving flows. WFQ, a packet scheduling algorithm, and WRED, a queue management algorithm, are widely accepted to fill this gap in the Internet backbone.

Internet QoS development continued with standardization efforts in delivering end-to-end QoS over the Internet. The Integrated Services (intserv) Internet Engineering Task Force (IETF) Working Group aims to provide the means for applications to express end-to-end resource requirements with support mechanisms in routers and subnet technologies. RSVP is the signaling protocol for this purpose. The Intserv model requires per-flow states along the path of the connection, which doesn't scale in the Internet backbones, where thousands of flows exist at any time. Chapter 7, "Integrated Services: RSVP," provides a discussion on RSVP and the intserv service types.

The IP ToS byte hasn't been used much in the past, but it is increasingly used lately as a way to signal QoS. The ToS byte is emerging as the primary mechanism for delivering

diffserv over the Internet, and for this purpose, the IETF differentiated services (diffserv) Working Group is working on standardizing its use as a diffserv byte. Chapter 2, "Differentiated Services Architecture," discusses the diffserv architecture in detail.

1.3 Performance Measures

QoS deployment intends to provide a connection with certain performance bounds from the network. Bandwidth, packet delay and jitter, and packet loss are the common measures used to characterize a connection's performance within a network. They are described in the following sections.

1.3.1 Bandwidth

The term bandwidth is used to describe the rated throughput capacity of a given medium, protocol, or connection. It effectively describes the "size of the pipe" required for the application to communicate over the network.

Generally, a connection requiring guaranteed service has certain bandwidth requirements and wants the network to allocate a minimum bandwidth specifically for it. A digitized voice application produces voice as a 64-kbps stream. Such an application becomes nearly unusable if it gets less than 64 kbps from the network along the connection's path.

1.3.2 Packet Delay and Jitter

Packet delay, or latency, at each hop consists of serialization delay, propagation delay, and switching delay. The following definitions describe each delay type:

Serialization delay— The time it takes for a device to clock a packet at the given output rate. Serialization delay depends on the link's bandwidth as well as the size of the packet being clocked. A 64-byte packet clocked at 3 Mbps, for example, takes about 0,171 ms to transmit. Notice that serialization delay

depends on bandwidth: The same 64-byte packet at 19.2 kbps takes 26 ms. Serialization delay also is referred to as transmission delay.

Propagation delay— The time it takes for a transmitted bit to get from the transmitter to a link's receiver. This is significant because it is, at best, a fraction of the speed of light. Note that this delay is a function of the distance and the media but not of the bandwidth. For WAN links, propagation delays of milliseconds are normal. Transcontinental U.S. propagation delay is in the order of 30 ms.

Switching delay— The time it takes for a device to start transmitting a packet after the device receives the packet. This is typically less than 10 ms.

All packets in a flow don't experience the same delay in the network. The delay seen by each packet can vary based on transient network conditions.

If the network is not congested, queues will not build at routers, and serialization delay at each hop as well as propagation delay account for the total packet delay. This constitutes the minimum delay the network can offer. Note that serialization delays become insignificant compared to the propagation delays on fast link speeds.

If the network is congested, queuing delays will start to influence end-to-end delays and will contribute to the delay variation among the different packets in the same connection. The variation in packet delay is referred to as packet jitter.

Packet jitter is important because it estimates the maximum delays between packet reception at the receiver against individual packet delay. A receiver, depending on the application, can offset the jitter by adding a receive buffer that could store packets up to the jitter bound. Playback applications that send a continuous information stream—including applications such as interactive voice calls, videoconferencing, and distribution—fall into this category.

Figure 1.1 illustrates the impact of the three delay types on the total delay with increasing link speeds. Note that the serialization delay becomes minimal compared to propagation delay as the link's bandwidth increases. The switching delay is negligible if the queues are empty, but it can increase drastically as the number of packets waiting in the queue increases.

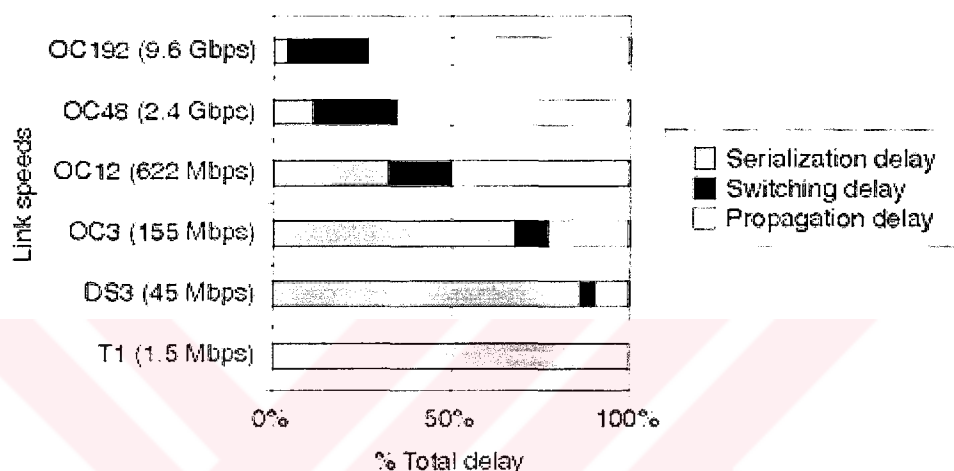


Figure 1. 1 Delay Components of a 1500-byte Packet on a Transcontinental U.S. Link with Increasing Bandwidths

1.3.3 Packet Loss

Packet loss specifies the number of packets being lost by the network during transmission. Packet drops at network congestion points and corrupted packets on the transmission wire cause packet loss. Packet drops generally occur at congestion points when incoming packets far exceed the queue size limit at the output queue. They also occur due to insufficient input buffers on packet arrival. Packet loss is generally specified as a fraction of packets lost while transmitting a certain number of packets over some time interval.

Certain applications don't function well or are highly inefficient when packets are lost. Such loss-intolerant applications call for packet loss guarantees from the network.

Packet loss should be rare for a well-designed, correctly subscribed or under-subscribed network. It is also rare for guaranteed service applications for which the network has already reserved the required resources. Packet loss is mainly due to packet drops at network congestion points with fiber transmission lines, with a Bit Error Rate (BER) of $10E-9$ being relatively loss-free. Packet drops, however, are a fact of life when transmitting best-effort traffic, although such drops are done only when necessary. Keep in mind that dropped packets waste network resources, as they already consumed certain network resources on their way to the loss point.

1.4 QoS Functions

This section briefly discusses the various QoS functions, their related features, and their benefits. The functions are discussed in further detail in the rest of the book.

1.4.1 Packet Classifier and Marker

Routers at the network's edge use a classifier function to identify packets belonging to a certain traffic class based on one or more TCP/IP header fields. A marker function is then used to color the classified traffic by setting either the IP precedence or the Differentiated Services Code Point (DSCP) field.

1.4.2 Traffic Rate Management

Service providers use a policing function to meter the customer's traffic entering the network against the customer's traffic profile. At the same time, an enterprise accessing its service provider might need to use a traffic shaping function to meter all its traffic and send it out at a constant rate such that all its traffic passes through the service provider's policing functions. Token bucket is the common traffic-metering scheme used to measure traffic.

1.4.3 Resource Allocation

FIFO scheduling is the widely deployed, traditional queuing mechanism within routers and switches on the Internet today. Though it is simple to implement, FIFO queuing has some fundamental problems in providing QoS. It provides no way to enable delay-sensitive traffic to be prioritized and moved to the head of the queue. All traffic is treated exactly the same, with no scope for traffic differentiation or service differentiation among traffic.

For the scheduling algorithm to deliver QoS, at a minimum it needs to be able to differentiate among the different packets in the queue and know the service level of each packet. A scheduling algorithm determines which packet goes next from a queue. How often the flow packets are served determines the bandwidth or resource allocation for the flow.

1.4.4 Congestion Avoidance and Packet Drop Policy

In traditional FIFO queuing, queue management is done by dropping all incoming packets after the packets in the queue reach the maximum queue length. This queue management technique is called tail drop, which signals congestion only when the queue is completely full. In this case, no active queue management is done to avoid congestion, or to reduce the queue sizes to minimize queuing delays. An active queue management algorithm enables routers to detect congestion before the queue overflows.

1.4.5 QoS Signaling Protocol

RSVP is part of the IETF intserv architecture for providing end-to-end QoS over the Internet. It enables applications to signal per-flow QoS requirements to the network. Service parameters are used to specifically quantify these requirements for admission control.

1.4.6 Switching

A router's primary function is to quickly and efficiently switch all incoming traffic to the correct output interface and next-hop address based on the information in the forwarding table. The traditional cache-based forwarding mechanism, although efficient, has scaling and performance problems because it is traffic-driven and can lead to increased cache maintenance and poor switching performance during network instability.

The topology-based forwarding method solves the problems involved with cache-based forwarding mechanisms by building a forwarding table that exactly matches the router's routing table. (Shenker et al., 1999)

1.4.7 Routing

Traditional routing is destination-based only and routes packets on the shortest path derived in the routing table. This is not flexible enough for certain network scenarios. Policy routing is a QoS function that enables the user to change destination-based routing to routing based on various user-configurable packet parameters.

Current routing protocols provide shortest-path routing, which selects routes based on a metric value such as administrative cost, weight, or hop count. Packets are routed based on the routing table, without any knowledge of the flow requirements or the resource availability along the route. QoS routing is a routing mechanism that takes into account a flow's QoS requirements and has some knowledge of the resource availability in the network in its route selection criteria.

1.5 Layer 2 QoS Technologies

Support for QoS is available in some Layer 2 technologies, including ATM, Frame Relay, Token Ring, and recently in the Ethernet family of switched LANs. As a connection-oriented technology, ATM offers the strongest support for QoS and could

provide a specific QoS guarantee per connection. Hence, a node requesting a connection can request a certain QoS from the network and can be assured that the network delivers that QoS for the life of the connection. Frame Relay networks provide connections with a minimum CIR, which is enforced during congestion periods. Token Ring and a more recent Institute of Electrical and Electronic Engineers (IEEE) standard, 802.1p, have mechanisms enabling service differentiation.

If the QoS need is just within a subnetwork or a WAN cloud, these Layer 2 technologies, especially ATM, can provide the answer. But ATM or any other Layer 2 technology will never be pervasive enough to be the solution on a much wider scale, such as on the Internet.

1.6 Multiprotocol Label Switching

The Multiprotocol Label Switching (MPLS) Working Group[9] at the IETF is standardizing a base technology for using a label-swapping forwarding paradigm (label switching) in conjunction with network-layer routing. The group aims to implement that technology over various link-level technologies, including Packet-over-Sonet, Frame Relay, ATM, and 10 Mbps/100 Mbps/1 Gbps Ethernet. The MPLS standard is based mostly on Cisco's tag switching ¹¹.

MPLS also offers greater flexibility in delivering QoS and traffic engineering. It uses labels to identify particular traffic that needs to receive specific QoS and to provide forwarding along an explicit path different from the one constructed by destination-based forwarding. MPLS, MPLS-based VPNs, and MPLS traffic engineering are aimed primarily at service provider networks.

1.7 End-to-End QoS

Layer 2 QoS technologies offer solutions on a smaller scope only and can't provide end-to-end QoS simply because the Internet or any large scale IP network is made up of a large group of diverse Layer 2 technologies. In a network, end-to-end connectivity

starts at Layer 3 and, hence, only a network layer protocol, which is IP in the TCP/IP-based Internet, can deliver end-to-end QoS.

The Internet is made up of diverse link technologies and physical media. IP, being the layer providing end-to-end connectivity, needs to map its QoS functions to the link QoS mechanisms, especially of switched networks, to facilitate end-to-end QoS.

Some service provider backbones are based on switched networks such as ATM or Frame Relay. In this case, you need to have ATM and Frame Relay QoS-to-IP interworking to provide end-to-end QoS. This enables the IP QoS request to be honored within the ATM or the frame cloud.

Switched LANs are an integral part of Internet service providers (ISPs) that provide Web-hosting services and corporate intranets. IEEE 802.1p and IEEE 802.1Q offer priority-based traffic differentiation in switched LANs. Interworking these protocols with IP is essential to making QoS end to end.

MPLS facilitates IP QoS delivery and provides extensive traffic engineering capabilities that help provide MPLS-based VPNs. For end-to-end QoS, IP QoS needs to interwork with the QoS mechanisms in MPLS and MPLS-based VPNs.

CHAPTER TWO

DIFFERENTIATED SERVICES ARCHITECTURE

2 Differentiated Services Architecture

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

This chapter primarily focuses on Differentiated Services (diffserv) architecture for delivering QoS in the Internet. The other architectural model, Integrated Services (intserv) is introduced.

2.1 Intserv Architecture

The Internet Engineering Task Force (IETF) set up the intserv Working Group (WG) in 1994 to expand the Internet's service model to better meet the needs of emerging, diverse voice/video applications. It aims to clearly define the new enhanced Internet service model as well as to provide the means for applications to express end-to-end resource requirements with support mechanisms in routers and subnet technologies. It follows the goal of managing those flows separately that requested specific QoS.

Two services—guaranteed and controlled load—are defined for this purpose. Guaranteed service provides deterministic delay guarantees, whereas controlled load service provides a network service close to that provided by a best-effort network under lightly loaded conditions. (Postel, 1981)

Resource Reservation Protocol (RSVP) is suggested as the signaling protocol that delivers end-to-end service requirements.

The intserv model requires per-flow guaranteed QoS on the Internet. With the thousands of flows existing on the Internet today, the amount of state information required in the routers can be enormous. This can create scaling problems, as the state information increases as the number of flows increases. This makes intserv hard to deploy on the Internet.

In 1998, the diffserv Working Group was formed under IETF. Diffserv is a bridge between intserv's guaranteed QoS requirements and the best-effort service offered by the Internet today. Diffserv provides traffic differentiation by classifying traffic into a few classes, with relative service priority among the traffic classes.

2.2 Diffserv Architecture

The diffserv approach to providing QoS in networks employs a small, well-defined set of building blocks from which you can build a variety of services. Its aim is to define the differentiated services (DS) byte, the Type of Service (ToS) byte from the Internet Protocol (IP) Version 4 header and the Traffic Class byte from IP Version 6, and mark the standardized DS byte of the packet such that it receives a particular forwarding treatment, or per-hop behavior (PHB), at each network node.

The diffserv architecture provides a framework within which service providers can offer customers a range of network services, each differentiated based on performance. A customer can choose the performance level needed on a packet-by-packet basis by simply marking the packet's Differentiated Services Code Point (DSCP) field to a specific value. This value specifies the PHB given to the packet within the service provider network. Typically, the service provider and customer negotiate a profile describing the rate at which traffic can be submitted at each service level. Packets

submitted in excess of the agreed profile might not be allotted the requested service level.

The diffserv architecture only specifies the basic mechanisms on ways you can treat packets. You can build a variety of services by using these mechanisms as building blocks. A service defines some significant characteristic of packet transmission, such as throughput, delay, jitter, and packet loss in one direction along a path in a network. In addition, you can characterize a service in terms of the relative priority of access to resources in a network. After a service is defined, a PHB is specified on all the network nodes of the network offering this service, and a DSCP is assigned to the PHB. A PHB is an externally observable forwarding behavior given by a network node to all packets carrying a specific DSCP value. The traffic requiring a specific service level carries the associated DSCP field in its packets.

All nodes in the diffserv domain observe the PHB based on the DSCP field in the packet. In addition, the network nodes on the diffserv domain's boundary carry the important function of conditioning the traffic entering the domain. Traffic conditioning involves functions such as packet classification and traffic policing and is typically carried out on the input interface of the traffic arriving into the domain. Traffic conditioning plays a crucial role 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 2.1. The two major functional blocks in this architecture are shown in Table 2.1.

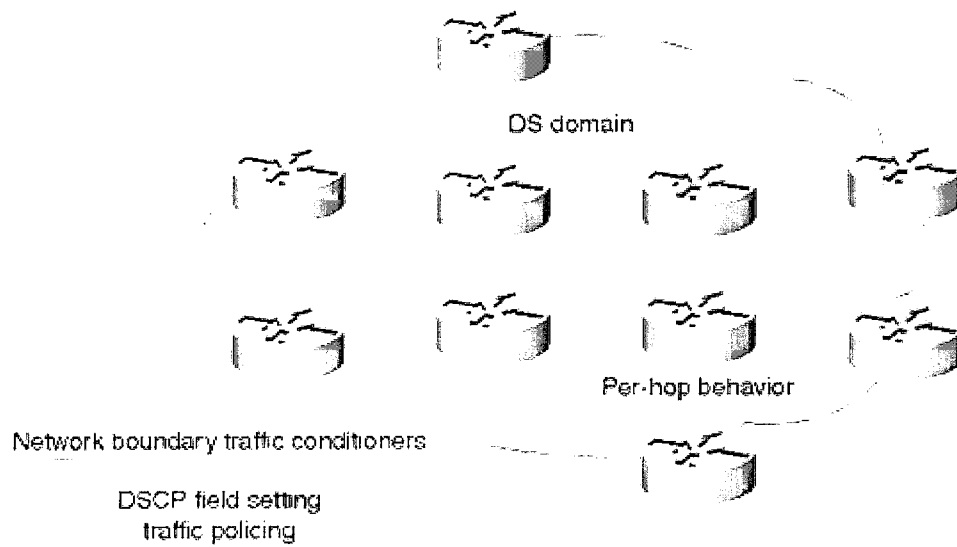


Figure 2. 1 Diffserv Overview

Table 2. 1 Functional Blocks in the diffserv Architecture

Functional Blocks	Location	Enabling Functions	Action
Traffic Conditioners	Typically, on the input interface on the diffserv domain boundary router	Packet Classification, Traffic Shaping, and Policing (Chapter 3)	Polices incoming traffic and sets the DSCP field based on the traffic profile
PHB	All routers in the entire diffserv domain	Resource Allocation (Chapters 4 and 5) Packet Drop Policy (Chapter 6)	PHB applied to packets based on service characteristic defined by DSCP

Apart from these two functional blocks, resource allocation policy plays an important role in defining the policy for admission control, ratio of resource overbooking, and so on.

A general QoS operational model is shown in Figure 2.2.

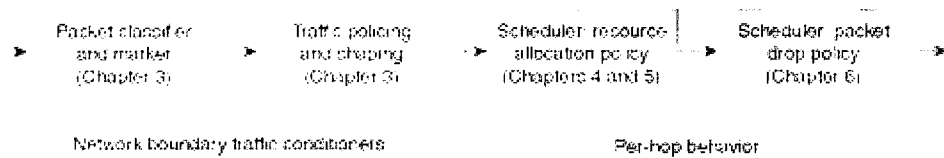


Figure 2. 2 General QoS Operational Model

DSCP

The IETF diffserv group is in the process of standardizing an effort that enables users to mark 6 bits of the ToS byte in the IP header with a DSCP. The lowest-order 2 bits are currently unused (CU). DSCP is an extension to 3 bits used by IP precedence. Like IP precedence, you can use DSCP to provide differential treatment to packets marked appropriately. Figure 2.3 shows the ToS byte. The ToS byte is renamed the DS byte with the standardization of the DSCP field. Figure 2.4 shows the DS byte.

P2	P1	P0	T3	T2	T1	T0	CU
----	----	----	----	----	----	----	----

IP precedence: 3 bits (P2-P0)

Type of service (ToS): 4 bits (T3-T0)

Currently unused (CU): 1 bit

Figure 2. 3 ToS Byte as of RFC 1349

DS5	DS4	DS3	DS2	DS1	DS0	CU	CU
-----	-----	-----	-----	-----	-----	----	----

Differentiated services code point (DSCP): 6 bits (DS5-DS0)

Currently unused (CU): 2 bits

Figure 2. 4 DS Byte

The DSCPs defined thus far by the IETF Working Group are as follows:

Default DSCP— It is defined to be 000 000.

Class Selector DSCPs— They are defined to be backward-compatible with IP precedence and are tabulated in Table 2.2.

Table 2. 2 Class Selector DSCP

Class Selectors	DSCP
Precedence 1	001 000
Precedence 2	010 000
Precedence 3	011 000
Precedence 4	100 000

Precedence 6	110 000
Precedence 7	111 000

- **Expedited Forwarding (EF) PHB**— It defines premium service. Recommended DSCP is 101110.
- **Assured Forwarding (AF) PHB**— It defines four service levels, with each service level having three drop precedence levels. As a result, AF PHB recommends 12 code points, as shown in Table 2.3.

Table 2.3 AF PHB

Drop Precedence	Class 1	Class 2	Class 3	Class 4
Low	001010	010010	011010	100010
Medium	001100	010100	011100	100100
High	001110	010110	011110	100110

2.2.1 Network Boundary Traffic Conditioners

Traffic conditioners are various QoS functions needed on a network boundary. The edge functions classify or mark traffic by setting the DSCP field and monitor incoming traffic into the network for profile compliance.

DSCP is the field indicating what treatment the packet should receive in a diffserv domain. The function can be that of packet classifier, DSCP marker, or traffic metering function, with either the shaper or dropper action.

2.2.2 Classifier

The classifier selects a packet in a traffic stream based on the content of some portion of the packet header. The most common way to classify traffic is based on the DSCP

field, but you can classify traffic based on the other fields in the packet headers. This function identifies a packet's traffic class.

2.2.3 Marker

This function helps write/rewrite the DSCP field of a packet based on its traffic class.

2.2.4 Metering

The metering function checks compliance to traffic profile, based on a traffic descriptor such as a token bucket, and passes the result to the marker function and either a shaper or dropper function to trigger particular action for in-profile and out-of-profile packets.

2.2.5 Shaper

The shaper function delays traffic by buffering some packets so that they comply with the profile. This action is also referred to as traffic shaping.

2.2.6 Dropper

The dropper function drops all traffic that doesn't comply with the traffic profile. This action is also referred to as traffic policing.

2.2.7 PHB

Network nodes with diffserv support use the DSCP field in the IP header to select a specific PHB for a packet. A PHB is a description of the externally observable forwarding behavior of a diffserv node applied to a set of packets with the same DSCP.

You can define a PHB in terms of its resource priority relative to other PHBs, or to some observable traffic service characteristics, such as packet delay, loss, or jitter. You

can view a PHB as a black box, as it defines some externally observable forwarding behavior without mandating a particular implementation.

In a diffserv network, best-effort behavior can be viewed as the default PHB. Diffserv recommends specific DSCP values for each PHB, but a network provider can choose to use a different DSCP than the recommended values in his or her network. The recommended DSCP for best-effort behavior is 000000.

The PHB of a specific traffic class depends on a number of factors:

Arrival rate or load for the traffic class— This is controlled by the traffic conditioning at the network boundary.

Resource allocation for the traffic class— This is controlled by the resource allocation on the nodes in the diffserv domain.

Traffic loss— This depends on the packet discard policy on the nodes in the diffserv domain.

Two PHBs, EF and AF, are standardized. They are discussed in the following sections.

2.2.8 EF PHB

You can use the EF PHB to build a low-loss, low-latency, low-jitter, assured-bandwidth, end-to-end service through diffserv domains. EF PHB targets applications such as Voice over IP (VoIP) and video conferencing, and services such as virtual leased line, as the service looks like a point-to-point connection for a diffserv network's end nodes. Such service is also often termed as premium service.

The main contributors to high packet delays and packet jitter are queuing delays caused by large, accumulated queues. Such queues are typical at network congestion points. Network congestion occurs when the arrival rate of packets exceeds their

departure rate. You can essentially eliminate queuing delays if the maximum arrival rate is less than the minimal departure rate. The EF service sets the departure rate, whereas you can control the traffic arrival rate at the node by using appropriate traffic conditioners at the network boundary.

An EF PHB needs to assure that the traffic sees no or minimal queues and, hence, needs to configure a departure rate for traffic that is equal to or less than the packet arrival rate. The departure rate or bandwidth should be independent of the other traffic at any time. The packet arrival and departure rates are typically measured at intervals equal to the time it takes for a link's maximum transmission unit (MTU)-sized packet to be transmitted.

A router can allocate resources for a certain departure rate on an interface by using different EF functionality implementations. Packet scheduling techniques—such as Class-Based Weighted Fair Queuing (CBWFQ), Weighted Round Robin (WRR), and Deficit Round Robin (DRR)—provide this functionality when the EF traffic can be carried over a highly weighted queue; that is, a weight that allocates a much higher rate to EF traffic than the actual EF traffic arrival rate. Further, you can modify these scheduling techniques to include a priority queue to carry EF traffic. The scheduling techniques are discussed in detail in Chapter 5.

When EF traffic is implemented using a priority queue, it is important to ensure that a busy EF priority queue does not potentially starve the remaining traffic queues beyond a certain configurable limit. To alleviate this problem, a user can set up a maximum rate against which the traffic serviced by the priority queue is policed. If the traffic exceeds the configured rate limit, all excess EF traffic is dropped. The network boundary traffic conditioners should be configured such that the EF traffic never exceeds its maximum configured rate at any hop in the network.

The recommended DSCP to be used for EF traffic in the network is 101110.

2.2.9 AF PHB

AF PHB is a means for a service provider to offer different levels of forwarding assurances for IP packets received from a customer diffserv domain. It is suitable for most Transmission Control Protocol (TCP)-based applications.

An AF PHB provides differentiated service levels among the four AF traffic classes. Each AF traffic class is serviced in its own queue, enabling independent capacity management for the four traffic classes. Within each AF class are three drop precedence levels (Low, Medium, and High) for Random Early Detection (RED)-like queue management.

2.2.10 Resource Allocation Policy

The last section discussed the defined diffserv PHBs in a network. How is the traffic conditioned at the edge and the resources allocated in the network to achieve the desired PHB? Three solutions are suggested: network provisioning, signaled QoS, and policy manager.

Network Provisioning

One resource allocation solution is to provision resources within the network using heuristic methods or systematic modeling techniques. This method can work only in a small network environment where the QoS policies and network traffic profile don't change often.

Signaled QoS

In this method, applications signal QoS requests using a signaling protocol such as RSVP. For RSVP, the diffserv domain is treated as another link in the network for admission control. QoSes are mapped between RSVP and diffserv classes. You can map RSVP guaranteed service to diffserv EF service, for example.

Signaled QoS can be a scalable solution in a large-scale network environment because RSVP is run only at the network edges with diffserv used in the network core, as shown in Figure 2.5. Mapping between RSVP reservation and a diffserv class happens at the edge of the diffserv network. With widespread RSVP support (for instance, the Microsoft Windows 2000 operating system supports RSVP), policy-aware applications at the network edges can use RSVP to signal QoS across the network without any scalability concerns. The solution suits well in some large-scale enterprise networks.

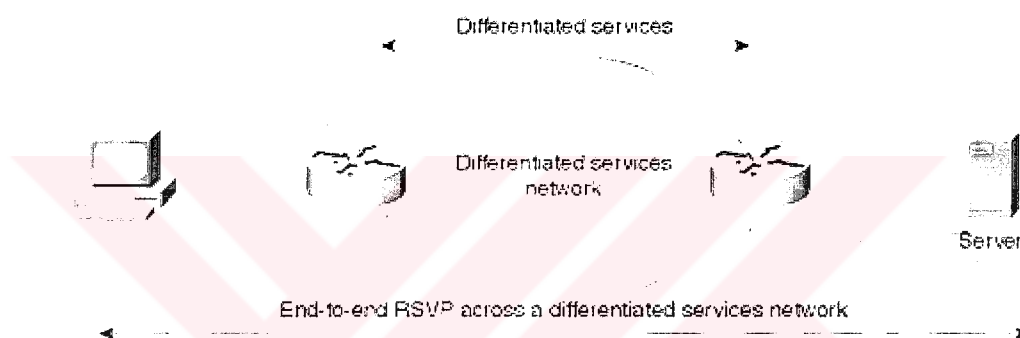


Figure 2. 5 RSVP Signaling Across a Differentiated Services Network

RSVP can scale well to support a few thousand per-flow sessions running in parallel. In addition, work is ongoing to provide aggregated RSVP. Multiple RSVP reservations are aggregated into a single aggregate reservation for large-scale RSVP deployment across a core network backbone that requires topology-aware admission control. Aggregated RSVP reservation is a fat, slowly adjusting reservation state that results in a reduced state signaling information in the network core. As a normal RSVP reservation, you can map the aggregate reservation to a diffserv class.

QoS Policy Manager

The policy definition determines the QoS applied on a traffic flow. The policy identifies the critical application traffic in the network and specifies its QoS level. Policy

is simply the configuration needed in all the individual network nodes to enable QoS. How does a QoS node get its policy?

In simple terms, a network engineer can configure the policies by making changes to a router's configuration. On a large-scale network, however, the process becomes tedious and unmanageable. To deliver end-to-end QoS on a large-scale network, the policies applied across all the individual nodes in the network should be consistent. As such, a centralized policy manager to define policies makes the task less daunting. This policy manager can distribute the policy to all the network devices.

Common Open Policy Service (COPS) is an IETF protocol for distributing policy. In COPS terminology, the centralized policy server is called the Policy Decision Point (PDP). The network node implementing or enforcing the policy is called the Policy Enforcement Point (PEP). The PDP uses the COPS protocol for downloading the policies into the PEPs in the network. A PEP device can generate a message informing the PDP if it cannot implement a policy that it was given by PDP.

2.2.11 IP Precedence Versus DSCP

As discussed in this chapter, diffserv architecture needs traffic conditioners at the network boundary and resource allocation and packet discard functions in the network core to provide EF and AF services. Because DSCP field definitions were not fully clear until recently, the diffserv architecture was initially supported using the 3-bit IP precedence because historically, the IP precedence field is used to mark QoS or precedence in IP packets. Cisco IOS is fully aligned with the diffserv architecture and provides all required network edge and core QoS functions based on the 3-bit IP precedence field.

Both 3-bit IP precedence and 6-bit DSCP fields are used in exactly the same purpose in a diffserv network: for marking packets at the network edge and triggering specific packet queuing/discard behavior in the network. Further, the

DSCP field definition is backward-compatible with the IP precedence values. Hence, DSCP field support doesn't require any change in the existing basic functionality and architecture. Soon, all IP QoS functions will support the DSCP field along with IP precedence.

Cisco introduced modular QoS CLI to provide a clean separation and modular configuration of the different enabling QoS functions. DSCP support for the various QoS functions is part of the modular QoS architecture.



CHAPTER THREE

NETWORK BOUNDARY TRAFFIC CONDITIONERS

3 Network Boundary Traffic Conditioners: Packet Classifier, Marker, and Traffic Rate Management

Traffic conditioning functions at the network boundary are vital to delivering differentiated services within a network domain. These functions provide packet classifier, marker, and traffic rate management.

In a network, packets are generally differentiated on a flow basis by the five flow fields in the Internet Protocol (IP) packet header: source IP address; destination IP address; IP protocol field; and source and destination ports. An individual flow is made of packets going from an application on a source machine to an application on a destination machine, and packets belonging to a flow carry the same values for the five packet header flow fields. Quality of service (QoS) applied on an individual flow basis, however, is not scalable because the number of flows can be large. So, routers at the network boundary perform classifier functions to identify packets belonging to a certain traffic class based on one or more Transmission Control Protocol/Internet Protocol (TCP/IP) header fields. A marker function is used to color the classified traffic by setting either the IP Precedence or the Differentiated Services Code Point (DSCP) field. Within the network core, you can apply a per-hop behavior (PHB) to the packets based on either the IP Precedence or the DSCP field marked in the packet header.

Another important traffic conditioner at the network boundary is traffic rate management. It enables a service provider to meter the customer's traffic entering the network against the customer's traffic profile using a policing function. Conversely, an enterprise accessing its service provider can meter all its traffic to shape the traffic and

send out at a constant rate such that all its traffic passes through the service provider's policing functions.

Network boundary traffic conditioners are essential to delivering differentiated services in a network. (Nagle,1984)

3.1 Packet Classification

Packet classification is a means of identifying packets to be of a certain class based on one or more fields in a packet. The identification function can range from straightforward to complicated. The different classification support types include:

- IP flow identification based on the five flow parameters: Source IP Address, Destination IP Address, IP protocol field, Source Port Number, and Destination Port number.
- Identification based on IP Precedence or DSCP field.
- Packet identification based on other TCP/IP header parameters, such as packet length.
- Identification based on source and destination Media Access Control (MAC) addresses.
- Application identification based on port numbers, Web Universal Resource Locator (URL) addresses, and so on. This functionality is available in Cisco products as Network Based Application Recognition (NBAR).

You can use access lists to match packets based on the various flow parameters. Access lists can also identify packets based on the IP Precedence or DSCP field. NBAR enables a router to recognize traffic flows as belonging to a specific application enabling packet classification based on application.

Packet classification also can be done based on information internal to the router. Examples of such classification are identification based on the arrived input interface and the QoS group field in the internal packet data structure. All the preceding

classification mechanisms are supported across all QoS functions as part of Modular QoS command-line interface (CLI).

The classification action is referred to as packet marking or packet coloring. Packets identified to belong to a class are colored accordingly.

3.2 Packet Marking

You can mark classified packets to indicate their traffic class. You can color packets by marking the IP Precedence or the DSCP field in the packet's IP header, or the QoS group field in the packet's internal data structure within a router.

3.2.1 IP Precedence

The IP Precedence field in the packet's IP header is used to indicate the relative priority with which a particular packet should be handled. It is made up of three bits in the IP header's Type of Service (ToS) byte. Apart from IP Precedence, the ToS byte contains ToS bits. ToS bits were designed to contain values indicating how each packet should be handled in a network, but this particular field is never used much in the real world.

Table 3. 1 IP Precedence Values and Names

IP Precedence Value	IP Precedence Bits	IP Precedence Names
0	000	Routine
1	001	Priority
2	010	Immediate
3	011	Flash
4	100	Flash Override
5	101	Critical
6	110	Internetwork Control

7	111	Network Control
---	-----	-----------------

3.2.2 DSCP

DSCP field is used to indicate a certain PHB in a network. It is made up of 6 bits in the IP header and is being standardized by the Internet Engineering Task Force (IETF) Differentiated Services Working Group. The original ToS byte containing the DSCP bits has been renamed the DSCP byte.

The DSCP field is part of the IP header, similar to IP Precedence. In fact, the DSCP field is a superset of the IP Precedence field. Hence, the DSCP field is used and is set in ways similar to what was described with respect to IP Precedence.

Note that the DSCP field definition is backward-compatible with the IP Precedence values.

3.2.3 The QoS Group

The QoS group is a field in the packet data structure internal to the router. The QoS group is used to mark packets matching certain user-specified classification criteria. It is important to note that a QoS group is an internal label to the router and is not part of the IP packet header.

Modular QoS CLI enables packet marking by any of the three mechanisms discussed in this section. Table 3.2 compares packet coloring using IP Precedence, DSCP, and QoS groups. (Postel,1981)

Table 3. 2 Marking Traffic Using IP Precedence, DSCP, and QoS Groups

Attributes	IP Precedence	DSCP	QoS Groups
Scope of the Classification	Entire network. Carried within the packet's IP header.	Entire network. Carried within the packet's IP header.	Internal to the router only. Not carried within the IP packet.
Number of Classes	8 classes (0–7)	64 classes (0–63)	100 classes (0–99)

Often, packets arrive at a network boundary carrying a set IP Precedence or DSCP field. Even in such situations when the packet arriving into the network is already marked, a network operator wants to enforce the right marking at the network edge based on the packet's class and its offered service level before the traffic enters the network.

3.3 The Need for Traffic Rate Management

To offer QoS in a network, traffic entering the service provider network needs to be policed on the network boundary routers to make sure the traffic rate stays within the service limit. Even if a few routers at the network boundary start sending more traffic than what the network core is provisioned to handle, the increased traffic load leads to network congestion. The degraded performance in the network makes it impossible to deliver QoS for all the network traffic.

Traffic policing functions using the CAR feature and shaping functions using the traffic shaping (TS) feature manage traffic rate but differ in how they treat traffic when tokens are exhausted. The concept of tokens comes from the token bucket scheme, a traffic metering function discussed in the next section. Table 3.3 compares the policing and shaping functions.

Table 3.3 Comparison Between Policing and Shaping Functions

Policing Function (CAR)	Shaping Function (TS)
Sends conforming traffic up to the line rate and allows bursts.	Smooths traffic and sends it out at a constant rate.
When tokens are exhausted, it can drop packets.	When tokens are exhausted, it buffers packets and sends them out later, when tokens are available.
Works for both input and output traffic.	Implemented for output traffic only.
Transmission Control Protocol (TCP) detects the line at line speed but adapts to the configured rate when a packet drop occurs by lowering its window size.	TCP can detect that it has a lower speed line and adapt its retransmission timer accordingly. This results in less scope of retransmissions and is TCP-friendly.

3.3.1 The Token Bucket Scheme

Traffic rate management requires a traffic metering function to measure the traffic. Token bucket is a common scheme used to measure traffic. It is used in both the policing and shaping algorithms as a means to report whether a packet is compliant or noncompliant with the rate parameters configured for it.

Depending on whether a packet is conforming, you can perform an appropriate action (transmit, drop, and so on).

Token bucket has three key parameters:

Mean rate or Committed Information Rate (CIR), in bits per second—
On average, traffic does not exceed CIR.

Conformed burst size (B_C), in number of bytes— This is the amount of traffic allowed to exceed the token bucket on an instantaneous basis. It is also occasionally referred to as normal burst size.

Extended burst size (B_E), in number of bytes— This is the bonus round. It allows a reduced percentage of traffic to conform between the conform burst and extended burst.

A fourth relevant parameter, time interval (TI), depends on the mean rate and the B_C , where $TI = B_C \div CIR$.

Token bucket implementations for policing and shaping functions are discussed in detail in the rest of this chapter.

3.4 Traffic Policing

The traffic policing or rate-limiting function is provided by CAR. CAR offers two primary functions: packet coloring by setting IP Precedence, and rate-limiting.

As a traffic policing function, CAR doesn't buffer or smooth traffic and might drop packets when the allowed bursting capability is exceeded. CAR is implemented as a list of rate-limit statements. You can apply it to both the output and input traffic on an interface. A rate-limit statement is implemented as follows:

```
rate-limit <input/output> access-group rate-limit # "CIR"
"conformed burst" "extended burst" conform-action "action desired"
exceed-action "action desired"
```

Each rate-limit statement is made up of three elements. The rate-limit statements are processed, as shown in Figure 3.1.

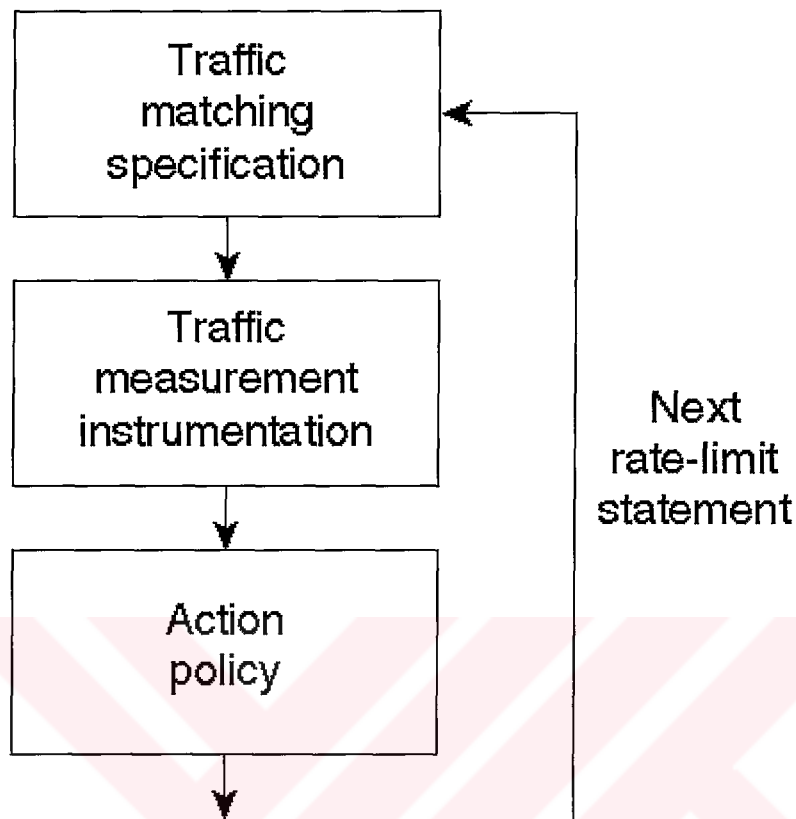


Figure 3. 1 The Evaluation Flow of Rate-Limit Statements

3.4.1 The Traffic Matching Specification

The traffic matching specification defines what packets are of interest for CAR. Each rate-limit statement is checked sequentially for a match. When a match occurs, the token bucket is evaluated. If the action is a continue action, it resumes looking for matches in subsequent rate limits. Note that you can define a match specification to match every packet.

Any packet that reaches the end of the list of rate-limit statements is transmitted. You can configure a "catch all" rate-limit statement at the end that drops everything, if needed. (Nagle,1984)

You can define a matching specification in four ways:

Match all traffic

Match on an IP Precedence value using a rate-limit access list

Match on a MAC address using a rate-limit access list

Match by using an IP standard or extended access list

Two special rate-limit access lists are provided to match IP Precedence and MAC addresses. Examples of the rate-limit access lists and other traffic match specification usage are given in the case studies in this chapter.

3.4.2 The Traffic Measurement Instrumentation

A token bucket is used as a means of measuring traffic. Figure 3-2 depicts the function of token bucket in CAR.

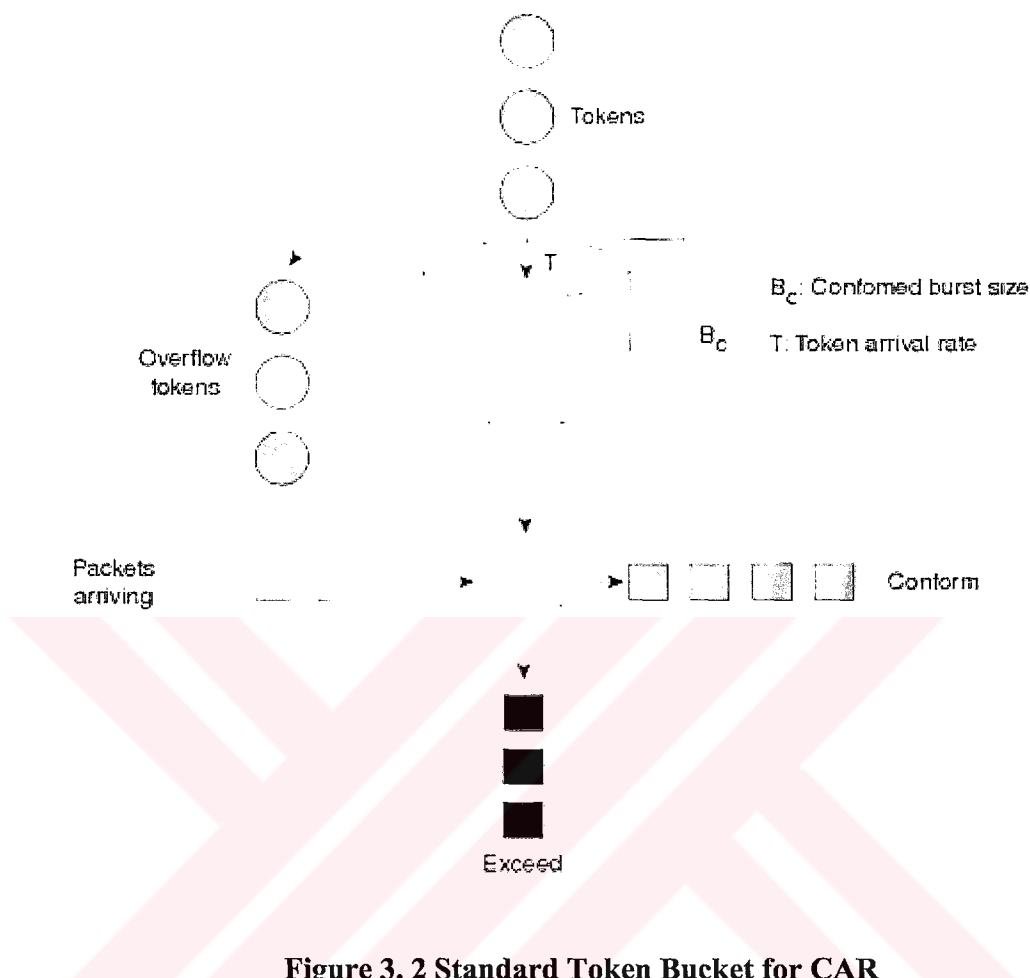


Figure 3. 2 Standard Token Bucket for CAR

The size of the token bucket (the maximum number of tokens it can hold) is equal to the conformed burst (B_C). For each packet for which the CAR limit is applied, tokens are removed from the bucket in accordance to the packet's byte size. When enough tokens are available to service a packet, the packet is transmitted. If a packet arrives and the number of available tokens in the bucket is less than the packet's byte size, however, the extended burst (B_E) comes into play. Consider the following two cases:

Standard token bucket, where $B_E = B_C$

A standard token bucket has no extended burst capability, and its extended burst is equal to its B_C . In this case, you drop the packet when tokens are unavailable.

Token bucket with extended burst capability, where $B_E > B_C$

A token bucket with an extended burst capability allows a stream to borrow more tokens, unlike the standard token bucket scheme. Because this discussion concerns borrowing, this section introduces two terms related to debt—actual debt (D_A) and compounded debt (D_C)—that are used to explain the behavior of an extended burst-capable token bucket.

D_A is the number of tokens the stream currently borrowed. This is reduced at regular intervals, determined by the configured committed rate by the accumulation of tokens. Say you borrow 100 tokens for each of the three packets you send after the last packet drop. The D_A is 100, 200, and 300 after the first, second, and third packets are sent, respectively.

D_C is the sum of the D_A of all packets sent since the last time a packet was dropped. Unlike D_A , which is an actual count of the borrowed tokens since the last packet drop, D_C is the sum of the actual debts for all the packets that borrowed tokens since the last CAR packet drop. Say, as in the previous example, you borrow 100 tokens for each of the three packets you send after the last packet drop. D_C equals 100, 300 ($= 100 + 200$), and 600 ($= 100 + 200 + 300$) after the first, second, and third packets are sent, respectively. Note that for the first packet that needs to borrow tokens after a packet drop, D_C is equal to D_A .

The D_C value is set to zero after a packet is dropped, and the next packet that needs to borrow has a new value computed, which is equal to the D_A . In the example, if the fourth packet gets dropped, the next packet that needs to borrow tokens (for example, 100) has its $D_C = D_A = 100$. Note that unlike D_C , D_A is not forgiven after a packet drop. If D_A is greater than the extended limit, all packets are dropped until D_A is reduced through accumulation of tokens.

The need for a token bucket with extended burst capability is not to immediately enter into a tail-drop scenario such as a standard token bucket, but rather, to gradually drop packets in a more Random Early Detection (RED)-like fashion. If a packet arrives and needs to borrow some tokens, a comparison is made between B_E and D_C . If D_C is greater than B_E , the packet is dropped and D_C is set to zero. Otherwise, the packet is sent, and D_A is incremented by the number of tokens borrowed and D_C with the newly computed D_A value.

Note that if a packet is dropped because the number of available tokens exceeds the packet size (in bytes), tokens will not be removed from the bucket (for example, dropped packets do not count against any rate or burst limits).

It is important to note that CIR is a rate expressed in bytes per second. The bursts are expressed in bytes. A burst counter counts the current burst size. The burst counter can either be less than or greater than the B_C . When the burst counter exceeds B_C , the burst counter equals $B_C + D_A$. When a packet arrives, the burst counter is evaluated, as shown in Figure 3-3.

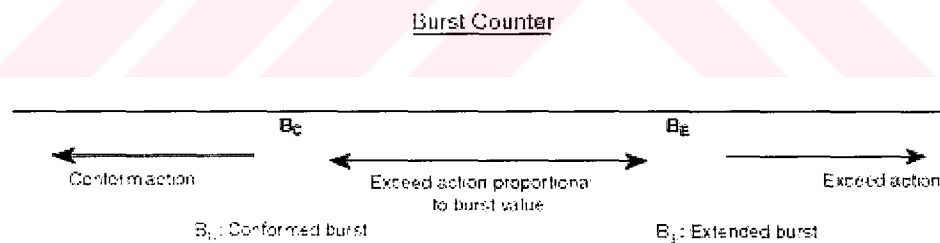


Figure 3. 3 Action Based on the Burst Counter Value

For cases when the burst counter value is between B_C and B_E , you can approximately represent the exceed action probability as:

$$(Burst\ counter - B_C) \div (B_E - B_C)$$

Based on this approximation, the CAR packet drop probability is shown in Figure 3-4. The concept of exceed action packet drop probability between the conformed and extended burst is similar to the packet drop probability of RED between the minimum and maximum thresholds.

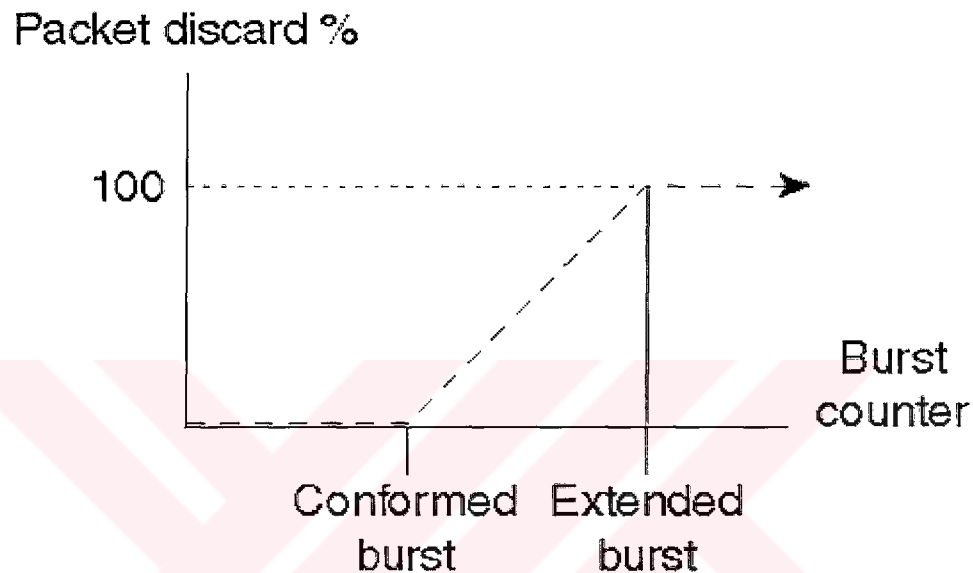


Figure 3. 4 CAR Packet Drop Probability

In a simple rate-limit statement where the B_C and B_E are the same value, no variable drop probability exceed region exists.

CAR implementation puts the following constraints on the token bucket parameters:

Rate (bps) should be in increments of 8 Kbps, and the lowest value allowed for conformed and extended burst size is 8000 bytes.

The minimum value of B_C size is Rate (bps) divided by 2000. It should be at least 8000 bytes.

The B_E is always equal to or greater than the B_C value.

3.5 The Action Policy

You can define separate action policies for conforming and exceeding traffic for each rate-limit statement. A **conform-action** or **exceed-action** could be one of the following:

Transmit

Drop

Continue (go to next rate limit statement in the list)

Set precedence and transmit

Set precedence and continue

Set **qos-group** and transmit

Set **qos-group** and continue

3.6 Traffic Shaping

TS is a mechanism to smooth the traffic flow on an interface to avoid congestion on the link and to meet a service provider's requirements. TS smoothes bursty traffic to meet the configured CIR by queuing or buffering packets exceeding the mean rate. The queued packets are transmitted as tokens become available. The queued packets' transmission is scheduled in either the first-in, first-out (FIFO) or Weighted Fair Queuing (WFQ) order. TS operation is illustrated in Figure 3.5.

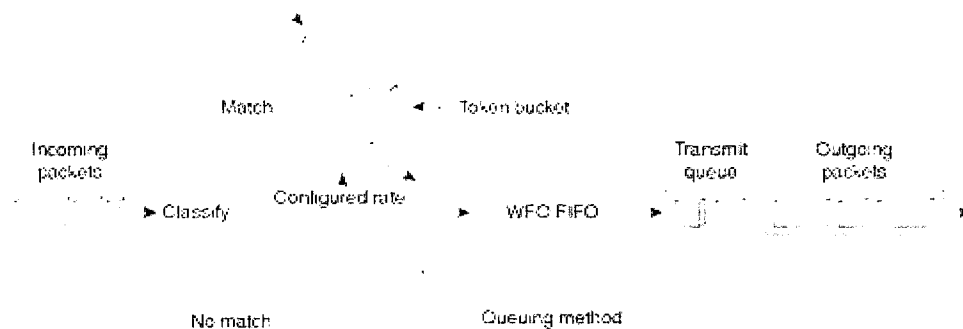


Figure 3. 5 Traffic Shaping Operation

TS also can be configured in an adaptive mode on a frame-relay interface. In this mode, TS estimates the available bandwidth by Backward Explicit Congestion Notification (BECN)/Forward Explicit Congestion Notification (FECN) field and Discard Eligible(DE) bit integration (discussed in Chapter 8, "Layer 2 QoS: Interworking with IP QoS").

This section covers traffic shaping on all interfaces/subinterfaces, regardless of the interface encapsulation. Traffic-shaping on an individual Frame Relay permanent virtual circuit (PVC)/switched virtual circuit (SVC) is covered in Chapter 8.

Traffic Measuring Instrumentation

TS uses a token bucket to measure traffic to classify a packet to be either conforming or nonconforming.

The maximum size of the token bucket is set to be the sum of conformed burst size, B_C and the extended burst size, B_E . Tokens equivalent to B_C are added to the bucket every measuring interval T , where $T = B_C \div CIR$. CIR is the allowed mean rate of traffic flow. If the bucket becomes full, any added tokens overflow. When a packet arrives, the

token bucket is checked to see if enough tokens are available to send the packet. If enough tokens are available, the packet is marked compliant, and the tokens equivalent to the packet size are removed from the bucket. If enough tokens are not available, the packet is marked non-compliant and is queued for later transmission. The TS token bucket is depicted in Figure 3.6.

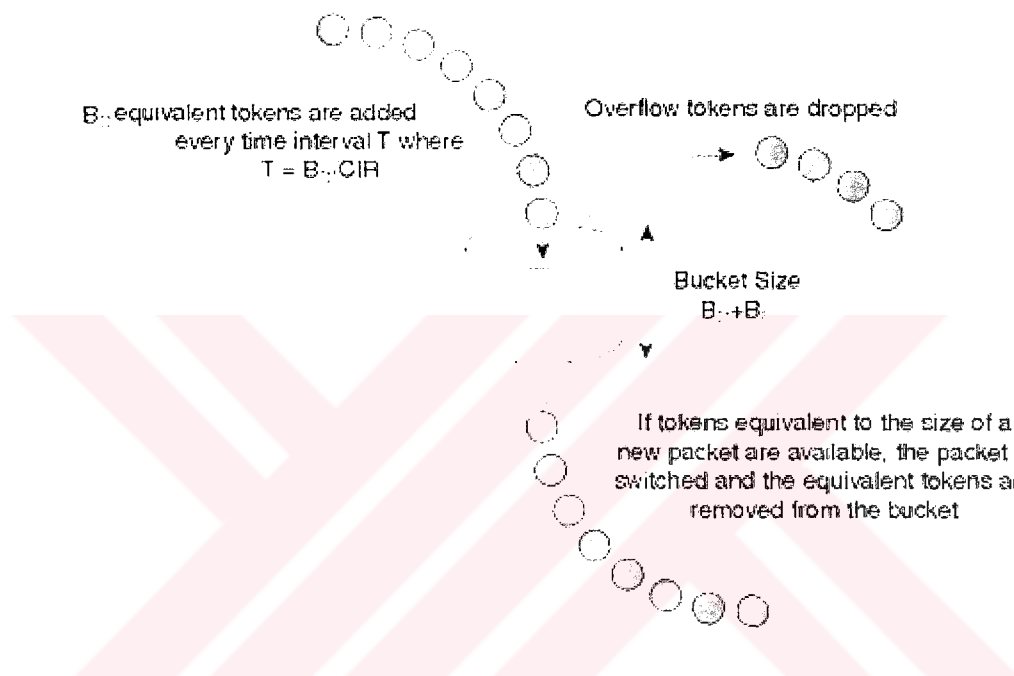


Figure 3. 6 The Token Bucket Scheme for the Traffic Shaping Function

You can accomplish traffic shaping on any generic interface using one of two implementations—Generic Traffic Shaping (GTS) and Distributed Traffic Shaping (DTS). Table 3.4 compares the two TS implementations.

Table 3. 4 Comparison of TS implementations: GTS and DTS

Feature Attributes	GTS	DTS
Order of Transmission of Buffered Packets	Uses WFQ as a scheduling algorithm.	Can use either FIFO or Distributed WFQ (DWFQ) as a scheduling algorithm.
Traffic Matching Specification	Has two modes: either all traffic, or traffic matched by a simple or extended IP access list.	Traffic classes as defined by a user by means of one of the classification features (CAR or QPPB).
Per Frame Relay Data-Link Identifier Support	Doesn't support per-PVC/SVC traffic-shaping on a Frame Relay interface.	Supports per-PVC/SVC traffic-shaping on a Frame Relay interface.
Availability	All single-processor (non-distributed) router platforms.	VIP-based 7500 series routers.
Protocol Support	All protocols.	IP only.

CHAPTER FOUR

PER HOP BEHAVIOR : RESOURCE ALLOCATION I

4 Per-Hop Behavior: Resource Allocation I

At times of network congestion, resource allocation for a flow on a router is determined by the router's scheduling discipline for the packets queued in the queuing system. The scheduling behavior determines which packet goes next from a queue. How often a flow's packets are served determines its bandwidth, or resource allocation.

The traditional packet scheduling mechanism on the Internet has been first-in, first-out (FIFO) scheduling, by which packets are transmitted in the same order in which they arrive in the output queue. FIFO is simple and easy to implement but cannot differentiate among the many flows; hence, FIFO cannot allocate specific performance bounds for a flow, or prioritize one flow over the others.

Weighted Fair Queuing (WFQ) is a scheduling discipline in which flow differentiation occurs in scheduling. In WFQ, each flow or traffic class is assigned a weight, and the rate at which a flow or a traffic class is serviced is proportional to its assigned weight. WFQ provides prioritization among unequally weighted traffic flows and fairness and protection among equally weighted traffic flows as per the max-min fair-share allocation scheme. This chapter discusses max-min fair-share allocation and how Fair Queuing (FQ) simulates this allocation scheme. It also covers WFQ in detail.

Later in this chapter, priority queuing and custom queuing schemes are discussed. Priority queuing and custom queuing also help define flows into different packet flows and schedule them based on an absolute priority and round-robin basis, respectively.

The chapter ends with a section on scheduling disciplines for voice. Note that this chapter covers only packet scheduling issues that decide which packet is served next.

4.1 Scheduling for Quality of Service (QoS) Support

Packet dynamics in a network can make the network prone to occasional or constant congestion, especially at routers connecting networks of widely different bandwidths. At times when a network doesn't see traffic congestion, any scheduling scheme works, because no queues build at the routers. When some network congestion exists, however, queues build upon the routers, and the scheduling mechanism on a router determines the order in which the packets in the queue are serviced.

For the scheduling algorithm to deliver QoS, at a minimum it needs to be able to differentiate among the different packets in the queue and know each packet's service level. Such a scheduling algorithm should allow guarantees on performance bounds by allocating resources on a flow basis and/or by prioritizing one flow over the other. You can do this at a granularity of a single flow or a traffic class that might be made up of packets from different traffic flows.

In addition, a scheduling algorithm is needed that provides fairness and protection among the flows with the same priority, such as all best-effort traffic flows.

Other requirements for such a scheduling algorithm include ease of implementation and admission control for flows requiring resource guarantees.

Although WFQ is more difficult to implement than FIFO queuing, it supports all the other requirements for QoS support that FIFO cannot deliver. WFQ also can work in conjunction with Resource Reservation Protocol (RSVP) to provide admission control for flows signaling resource requirements using RSVP. (Mankin,1997)

4.1.1 FIFO Queuing

FIFO queuing is a queuing mechanism in which the order of the packets coming into a queue is the same as the order in which they are serviced or transmitted out of the queue. Figure 4.1 illustrates a FIFO queue.

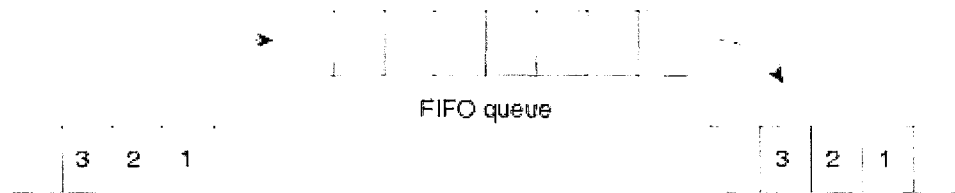


Figure 4. 1 FIFO Queue

In FIFO queuing, the order in which packets arrive in a queue is the same as the order in which they are transmitted from the queue. FIFO, the most common scheduling mechanism in routers today, is easy to implement. It has no mechanism to differentiate between flows, however, and hence cannot prioritize among them. Not only can FIFO queuing not prioritize one flow over the other, but it also offers neither protection nor fairness to equal-priority traffic flows because a large, badly behaving flow can take the share of resources of well-behaving flows with end-to-end, adaptive flow-control schemes, such as Transmission Control Protocol (TCP) dynamic window control. With FIFO, flows receive service approximately in proportion to the rate at which they send data into the network. Such a scheme is obviously not fair, because it rewards greedy flows over well-behaved ones. Any fairness algorithm by its nature offers protection against greedy flows.

4.1.2 The Max-Min Fair-Share Allocation Scheme

If FIFO doesn't do fair-share allocation among flows, how do you define a fair allocation scheme in which each flow gets its fair share of resources? A widely accepted fair-share allocation scheme is called the max-min fair-share scheme.

Various users' demands for a resource usually differ. So it is possible to classify users in the order of their increasing demand for a resource. The max-min fair-share allocation is defined as follows:

Resources are allocated in order of increasing demand.

No user gets a resource share larger than its demand.

Users with unsatisfied demands get an equal share of the resource.

Consider an example in which a resource has a capacity of 14, servicing five users, A, B, C, D, and E, with demands 2, 2, 3, 5, and 6, respectively. Initially, the source with the smallest demand is given a resource equal to the resource capacity divided by the total number of users. In this case, User A and User B are given a resource of $14 \div 5 = 2.8$. But Users A and B actually need only 2. So the unused excess, 1.6 (0.8 each from Users A and B), is distributed evenly among the other three users. So Users C, D, and E each get a resource of $2.2 + (1.6 \div 3) = 3.33$. Now, the user with the next-smaller demand is serviced. In this case, it is User C. The resource allocated to User C is 0.33 units in excess of its demand for 3. This unused excess is distributed evenly between Users D and E so that each now has a resource of $3.33 + (0.33 \div 2) = 3.5$.

We can calculate fair allocation as follows:

Fair allocation = (resource capacity – resource capacity already allocated to users) ÷ number of users who still need resource allocation

In this step, the demands of C, D, and E exceed fair allocation of 2.8 and, hence, cannot be allocated. In the next step, the unused excess bandwidth of A and B's fair allocation is equally distributed among the three remaining users, C, D, and E.

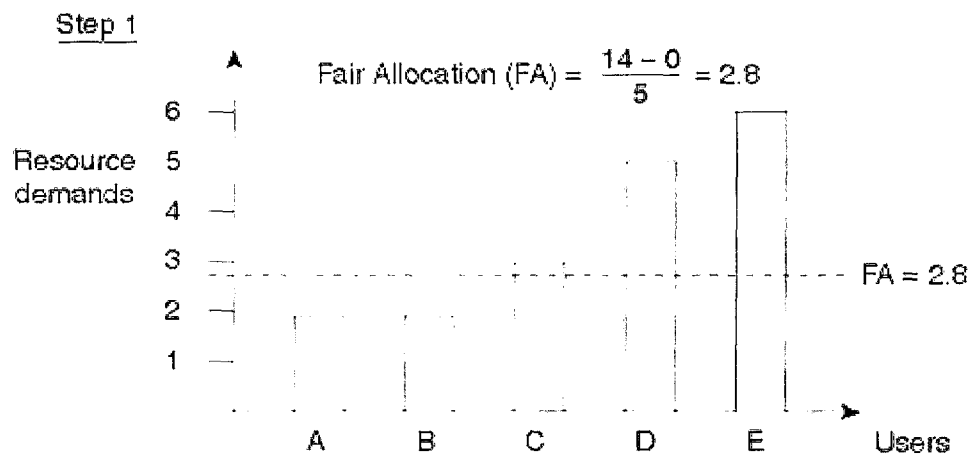


Figure 4. 2 Resource Allocation for Users A and B

In Step 2, shown in Figure 4.3, the demand of User C is fully allocated because its resource request falls within the fair allocation. In this step, the demands of D and E exceed fair allocation of 3.33 and, hence, cannot be allocated. In the next step, the unused excess bandwidth of C's fair allocation is equally distributed between the two remaining users, D and E.

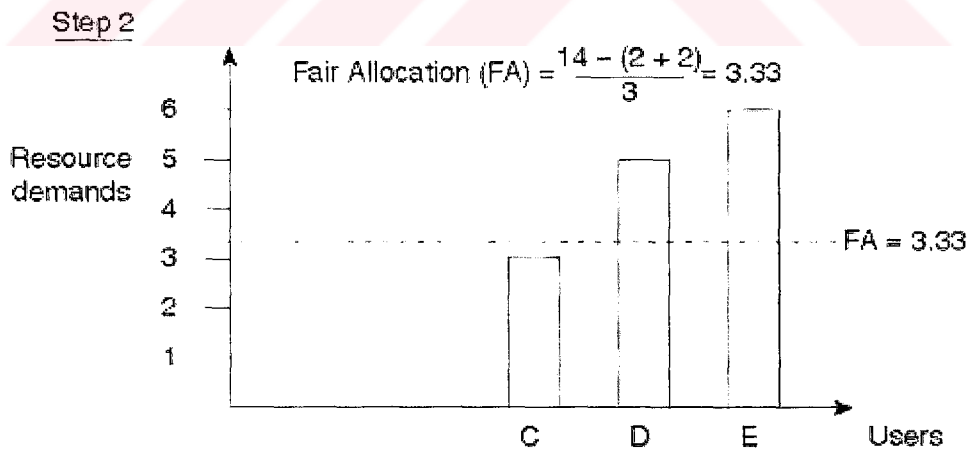


Figure 4. 3 Resource Allocation for User C

In Step 3, shown in Figure 4.4, the fair allocation of 3.5 falls below the requests of both Users D and E, which are each allocated 3.5 and have unsatisfied demands of 1.5 and 2.5, respectively.

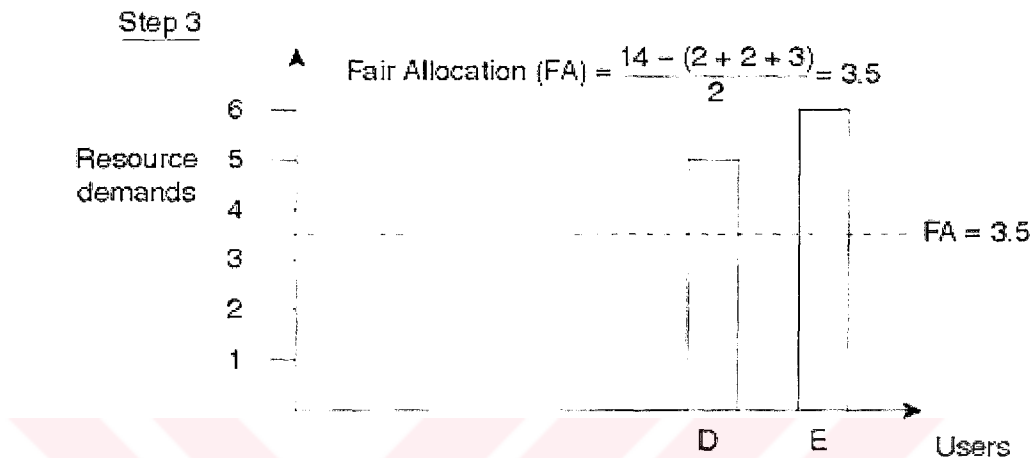


Figure 4. 4 Resource Allocation for Users D and E

This scheme allocates resources according to the max-min fair-share scheme. Note that all users with unsatisfied demands (beyond what is their max-min fair share) get equal allocation. So, you can see that this scheme is referred to as max-min fair-share allocation because it maximizes the minimum share of a user whose demand is not fully satisfied.

Consider an extension to the max-min fair-share allocation scheme in which each user is assigned a weight. Such a scheme is referred to as weighted max-min fair-share allocation, in which a user's fair share is proportional to its assigned weight.

4.1.3 Generalized Processor Sharing

For best-effort traffic flows and other equally weighted flow classes, the right scheduling discipline is one that provides fairness as described by the max-min fair-share allocation. Generalized processor sharing (GPS) is an ideal scheduling mechanism that achieves this objective.

GPS puts each flow in its own logical queue and services an infinitesimally small amount of data from each nonempty queue in a round-robin fashion. It services only an infinitesimally small amount of data at each turn so that it visits all the nonempty queues at any finite time interval, thus being fair at any moment in time.

If you assign a weight per flow, in each round of service, GPS services an amount of data from a flow in proportion to the assigned weight. This GPS extension provides weighted max-min fair share.

GPS, though an ideal model for max-min fair share, is not possible to implement. The right scheduling algorithm for practical purposes is one that approximates GPS and can be implemented.

4.2 Sequence Number Computation-Based WFQ

WFQ is an approximation of the GPS scheme, because it attempts to simulate a GPS scheduler behavior without making its impractical infinitesimal packet size assumption. Sequence number computation-based WFQ simulates a GPS server that services 1 byte at a time. WFQ works well with variable-size packets, because it doesn't need to know a flow's mean packet size in advance. FQ is a WFQ technique that considers all flows to be the same—that is, to be of equal weight.

FQ simulates GPS by computing a sequence number for each arriving packet. The assigned sequence numbers are essentially service tags, which define the relative order in which the packets are to be serviced. The service order of packets using sequence number computation emulates the service order of a GPS scheduler.

To intuitively understand how GPS simulation is done, consider a variable called round number, which denotes the number of rounds of service a byte-by-byte round-robin scheduler has completed at a given time. The computation of a sequence number depends on the round number.

To illustrate how GPS is simulated by FQ, consider three flows, A, B, and C, with packet sizes 128, 64, and 32 bytes, respectively. Packets arrive back-to-back on a busy FQ server in the order A1, A2, A3, B1, C1, with A1 arriving first, followed by A2, and so on.

A flow is said to be active if any outstanding packets of that flow are awaiting service, and inactive otherwise.

For this example, assume Packet A1 arrived on an inactive flow in the FQ system. Assuming service by a byte-by-byte round-robin scheduler, an entire 128-byte packet is sent when the scheduler completes 128 rounds of service since the packet arrived. If the round number at the time Packet A1 arrived is 100, the entire packet is transmitted when the round number becomes $100 + 128 = 228$. Hence, the sequence number of a packet for an inactive flow is calculated by adding the round number and the packet size in bytes. Essentially, it is the round in which the last byte of the packet is transmitted. Because, in reality, a scheduler transmits a packet and not 1 byte at a time, it services the entire packet whenever the round number becomes equal to the sequence number.

When Packet A2 arrives, the flow is already active with A1 in the queue, waiting for service, with a sequence number of 228. The sequence number of Packet A2 is $228 + 128 = 356$, because it needs to be transmitted after A1. Hence, the sequence number of a packet arriving on an active flow is the highest sequence number of the packet in the flow queue, plus its packet size in bytes.

Similarly, Packet A3 gets a sequence number of $356 + 128 = 484$. Because Packets B1 and C1 arrive on an inactive flow, their sequence numbers are 164 (that is, $100 + 64$) and 132 (that is, $100 + 32$), respectively.

Sequence Number (SN) assignment for a packet is summarized based on whether it arrives on an active or an inactive flow as follows:

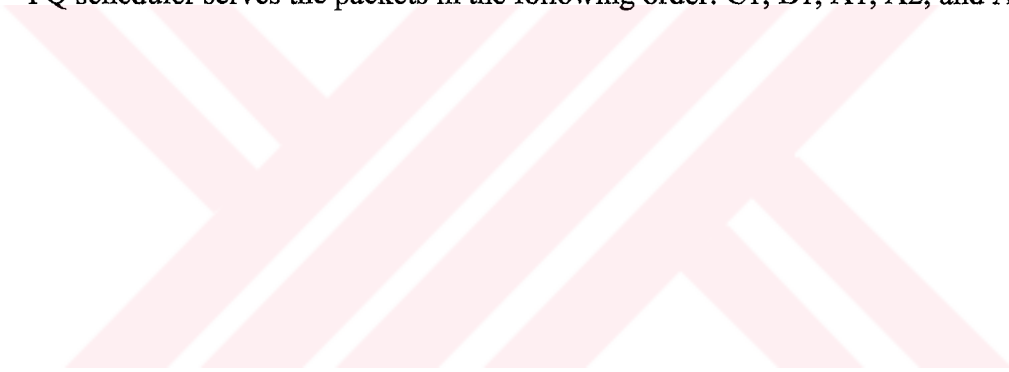
Packet arrives on an inactive flow:

$SN = \text{size of the packet in bytes} + \text{the round number at the time the packet arrived}$
(The round number is the sequence number of the last packet serviced.)

Packet arrives on an active flow:

$SN = \text{size of the packet in bytes} + \text{the highest sequence number of the packet already in the flow queue}$

Packets in their flow queues, along with their computed sequence numbers, are shown in Figure 4.5 to illustrate how the FQ scheduler emulated GPS. A GPS scheduler will have completed scheduling the entire Packet A1 in the 228th round. The sequence number denotes the relative order in which the packets are served by the scheduler. The FQ scheduler serves the packets in the following order: C1, B1, A1, A2, and A3.



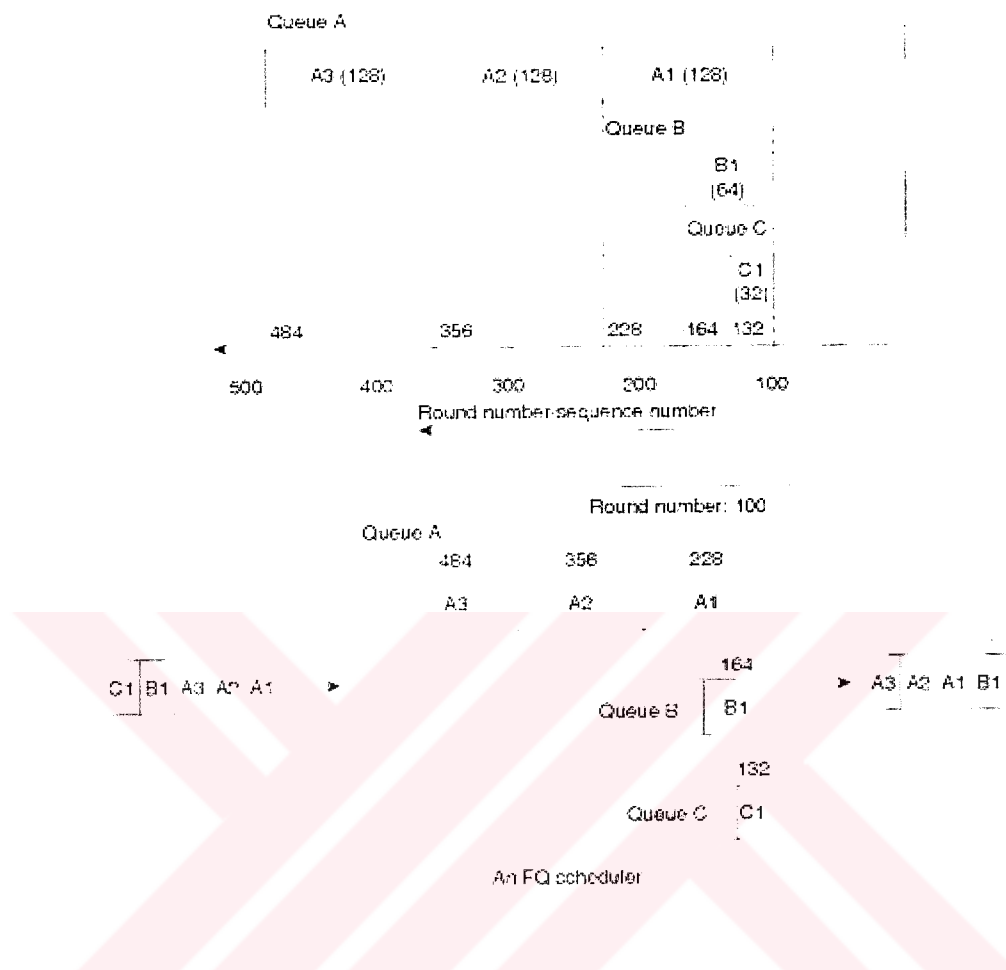


Figure 4. 5 An Example Illustrating the Byte-by-Byte Round-Robin GPS Scheduler Simulation for FQ

Round numbers are used only for calculating sequence numbers if the arriving packet belongs to a new flow. Otherwise, the sequence number is based on the highest sequence number of a packet in that flow awaiting service. If Packet A4 arrives at any time before A3 is serviced, it has a sequence number of $484 + 128 = 612$.

Note that the round number is updated every time a packet is scheduled for transmission to equal the sequence number of the packet being transmitted. So if Packet D1 of size 32 bytes, belonging to a new flow, arrives when A1 is being transmitted, the round number is 228 and the sequence number of D1 is 260 ($228 + 32$). Because D1 has

a lower sequence number than A2 and A3, it is scheduled for transmission before A2 and A3. Figure 4.6 depicts this change in scheduling order.

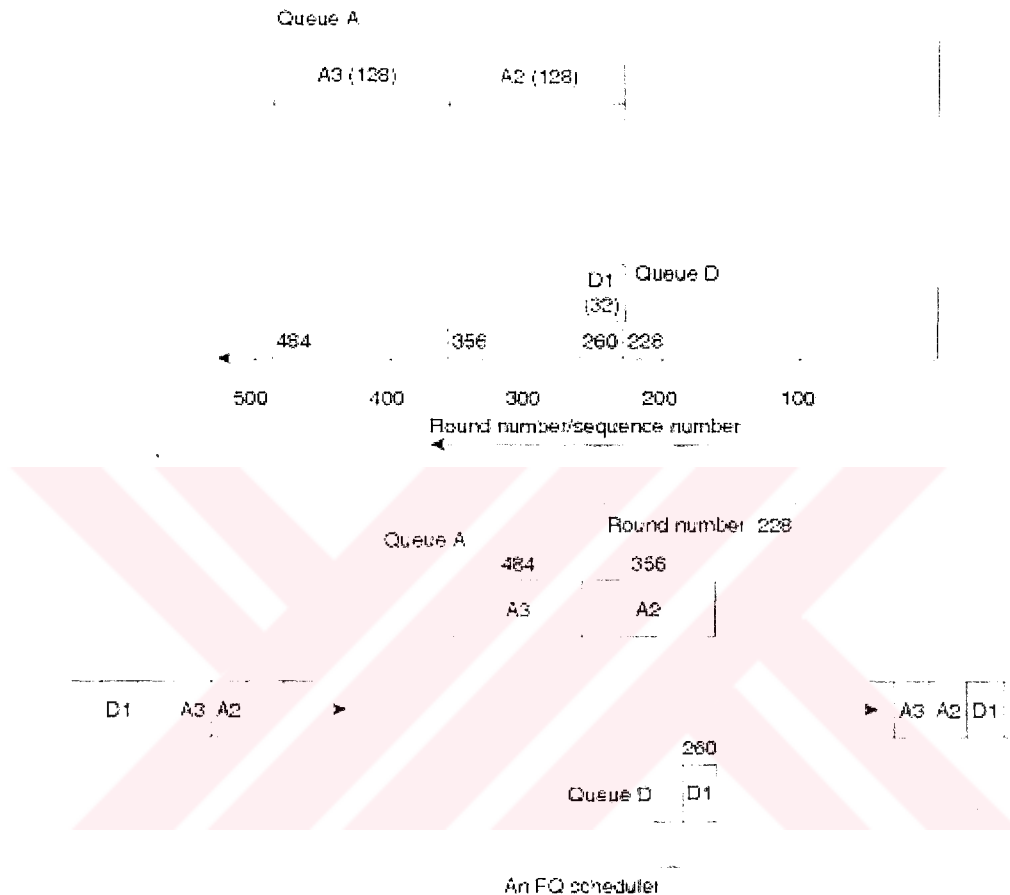


Figure 4. 6 Illustration of FQ Scheduler Behavior; Packet D1 Arriving After Packet A1 Is Scheduled

Most often, some flows are considered more important or mission-critical than others. Such flows need to be preferred over the others by the scheduler. You can expand the FQ concept to assign weights per flow so that each flow is serviced in proportion to its weight. Such a fair queuing system is called flow-based WFQ and is discussed in the next section.

4.3 Flow-Based WFQ

In WFQ, weights are assigned based on their precedence value in the Internet Protocol (IP) header. They are calculated as follows:

$$\text{Weight} = 4096 \div (\text{IP precedence} + 1)$$

Table 4.1 tabulates a packet's weight based on its IP precedence and Type of Service (ToS) byte value.

Table 4.1 Weights Assigned Based on the IP Precedence Value of a Packet Belonging to an Unreserved (Non-RSVP) Flow

IP Precedence	ToS Byte Value	Weight	
		IOS Versions Prior to 12.0(5)T	IOS Versions 12.0(5)T and Higher
0	0 (0x00)	4096	32768
1	32	2048	16384

	(0x20)		
2	64	1365	10920
	(0x40)		
3	96	1024	8192
	(0x60)		
4	128	819	6552
	(0x80)		
5	160 (0xA0)	682	5456
6	192 (0xC0)	585	4680
7	224 (0xE0)	512	4096

The weight of an RSVP flow with the largest bandwidth reservation is 4 until IOS Version 12.0(5)T and is 6 for 12.0(5)T and higher. The weight of all other RSVP flow reservations is derived based on the largest bandwidth reservation, as shown here:

Weight for RSVP flow or conversation = highest bandwidth reservation on the link \times (greatest bandwidth reservation on the link \div conservation bandwidth)

For the purpose of the discussion in the remainder of this section, weights prior to 12.0(5)T are used. It is important to note that the exact weight calculation scheme used doesn't matter in illustrating the working of WFQ.

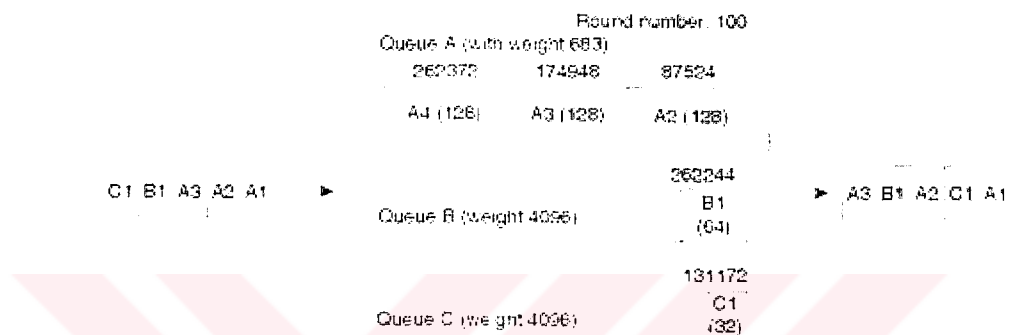
WFQ uses two packet parameters to determine its sequence number. Like FQ, WFQ uses the packet's byte size. In addition, however, WFQ uses the weight assigned to the packet. The packet's weight is multiplied by its byte size for calculating the sequence number. This is the only difference between WFQ and FQ.

Note that the direct correlation between a byte-by-byte round-robin scheduler and FQ is lost with WFQ, because the packet's byte count is multiplied by its weight before its sequence number is calculated. Consider a sequence number in WFQ as a number calculated to determine the relative order of a packet in a WFQ scheduler, and consider the round number as the sequence number of the last packet served in the WFQ scheduler.

Using the same example discussed in the FQ section, assume that packets of Flow A have precedence 5, whereas Flows B and C have precedence 0. This results in a weight of 683 for packets in Flow A and 4096 for packets in Flows B and C. [Table 4-2](#) shows all the flow parameters in this example. The sequence number of Packet A1 is calculated as $100 + (683 \times 128) = 87524$. Similarly, you can calculate sequence numbers for packets A2, A3, B1, and C1 as 174948, 262372, 262244, and 131172, respectively. So the order in which the scheduler services them is A1, C1, A2, B1, and A3, as illustrated in Figure 4.7

Table 4.2 Flow-Based WFQ Example

Queue	Packet Size	Precedence	Weight = $4096 \div (\text{Precedence} + 1)$
Queue A	128	5	683
Queue B	64	0	4096
Queue C	32	0	4096

**Figure 4.7 Illustration of the Flow-Based WFQ Example**

Note that with WFQ, you can prioritize Flow A, but you can't accommodate Flows B and C fairly. A WFQ scheduler simulates a max-min weighted GPS.

If Packets A4 and D1 (a new flow with a precedence 0 and size 32 bytes) arrive after A1 is scheduled, A4 and D1 get a sequence number of 349,796 ($((683 \times 128) + 262,372)$) and 218,596 ($((4096 \times 32) + 87,524)$), respectively. The discussion of calculating the sequence numbers for A4 and D1 with FQ still applies here. Now, the scheduling order of the remaining packets is changed to C1, A2, D1, B1, A3, and A4. This is shown in Figure 4.8.

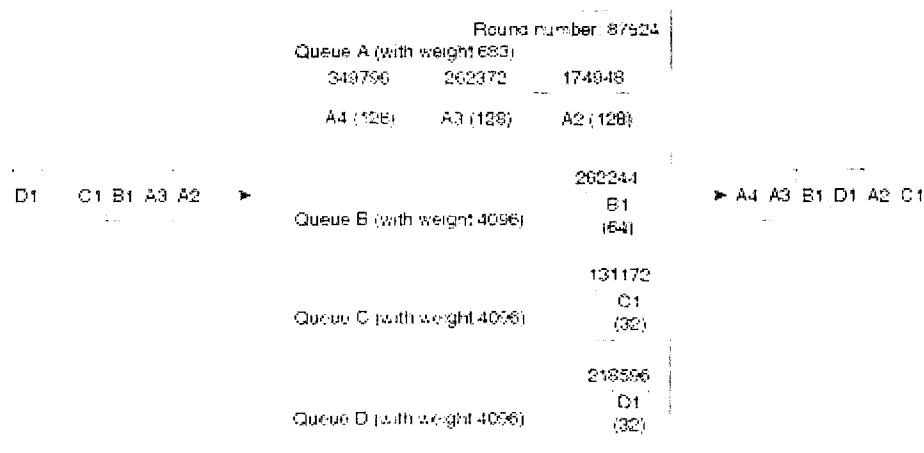


Figure 4. 8 Illustration of the Flow-Based WFQ Example (continued).

In Figure 4.8, packet D1 comes closely after packet A1 has been scheduled for transmission. Packet D1 is transmitted before packets A3 and A4, which arrived in the queue early.

4.3.1 WFQ Interaction with RSVP

RSVP requires scheduler support to guarantee bandwidth reservations. WFQ interacts with RSVP reservation request (RESV) messages requesting resource allocations. WFQ maintains reserved conversation queues with weights assigned to match the bandwidth allocation reserved based on RSVP requests. RSVP is discussed in detail in Chapter 7. The number of reserved conversations allowed is a configurable parameter.

4.3.2 WFQ Implementation

In flow-based WFQ implementation, weights are based strictly on precedence and cannot be changed. Though FQ in itself is not available, WFQ becomes FQ for all practical purposes when all traffic arriving at the scheduler carries the same precedence value.

With flow-based WFQ, packets with different IP precedence in a single flow are not scheduled out of order. In this regard, a flow is implemented as a hash defined by source and destination IP addresses, IP protocol field, Transmission Control Protocol/User Datagram Protocol (TCP/UDP) port numbers and the 5 bits (excluding the 3 IP precedence bits) in the ToS byte. Due to this flow description, packets of the same flow, but with different precedence values, fall into the same queue. Packets within a flow queue are serviced in FIFO order.

In general, WFQ limits its drops to the most active flows, whereas FIFO might drop from any flow. Therefore, WFQ should encourage the most active flows to scale back without affecting the smaller flows. Because the median flow duration in the Internet is 10–20 packets in length, a fairly small percentage of the flows should be taking the lion's share of the drops with WFQ, while FIFO drops should be distributed across all flows. Hence, the effects of global synchronization with FIFO are less pronounced with WFQ for traffic with adaptive flow control such as TCP traffic.

In general, a flow-based WFQ uses a subqueue for each flow. As such, flow-based WFQ queues are referred to as conversation queues. Because memory is a finite resource, the default number of conversation queues allocated is restricted to 256. This parameter is configurable when enabling the **fair-queue** interface command, however. Note that increasing the number of queues increases the memory taken by the queue data structures and the amount of state information maintained by the router. If the number of flows is greater than the number of queues, multiple flows can share a single queue. Configuring a large number of queues increases the chances of having only one flow per queue.

Flow-based WFQ can also work in conjunction with Weighted Random Early Detection (WRED), a proactive packet drop policy to avoid congestion. Flow-based WFQ implementation is done using list sorting. The complexity is of the order of $O(n)$, where n is the number of packets waiting for service from a WFQ scheduler. List sorting

can become prohibitively expensive on high-bandwidth links where the number of flows and the number of packets to be serviced per second is high.

4.4 Flow-Based Distributed WFQ (DWFQ)

Flow-based DWFQ operates in a distributed mode in the 7500 series routers supporting Versatile Interface Processor (VIP) line cards with built-in processors. When flow-based DWFQ is enabled on an interface, the feature runs on the interface's individual VIP line card, unlike flow-based WFQ, which runs on the router's central processor. Distributed Cisco Express Forwarding (DCEF) switching is required to run DWFQ.

Flow-based WFQ uses a sorted linked list to maintain packets, and newly arriving packets are inserted into the sorted list based on the sequence number assigned to the packet. In contrast, flow-based DWFQ uses calendar queues to perform the sorting function required by WFQ. Flow-based DWFQ implements calendar queues that approximate GPS with a less-complex algorithm than simple list sorting. Calendar queues do an $O(1)$ insertion, which is important for higher-speed interfaces, as opposed to $O(n)$ algorithms. Calendar queues are more efficient in terms of CPU utilization, but calendar queues have larger memory requirements. It is a trade-off between the larger memory costs versus limitations imposed by a smaller set of calendar queues.

Timestamps are computed for each arriving packet and are sorted using a calendar queue. Any calendar queue-based implementation has its timestamp granularity constrained by the number of calendar queues in the system. For the calendar queue system to behave the same as the flow-based WFQ implementation, you need $4096 \times \text{maximum transmission unit (MTU)}$ -size calendar queues!

Because the number of calendar queues prevents the timestamp granularity from allowing unique timestamp values for packets ranging from 1 byte to MTU bytes, packets with different sizes from flows/classes with the same weight cannot have the

same timestamp. To serve these variable-size packets with the same timestamp, it is necessary to run a deficit-like algorithm on the calendar queues to ensure proper bandwidth allocation. Thus, part of the DWFQ implementation has Deficit Round Robin (DRR)-like characteristics. The DRR algorithm is described in Chapter 5.

DWFQ implementation has an individual queue limit and an aggregate queue limit across all the individual queues. The individual queue limits are enforced only when the aggregate queue limit is reached.

Flow-based DWFQ is actually FQ. Under FQ, all flows are the same, and flows are not weighted. In the case of flow-based DWFQ, a flow is described by a hash function consisting of the source and destination IP addresses, IP field, and TCP/UDP port numbers. All non-IP traffic is treated as a single flow and, therefore, placed in the same queue.

Because of the reasons discussed in the section on flow-based WFQ, the number of subqueues allowed in flow-based DWFQ is restricted. The total number of subqueues for a flow-based DWFQ is 512. If more than 512 flows exist on an interface, some flows will share the same subqueue.

4.5 Class-Based WFQ

The last two sections discussed flow-based WFQ mechanisms running on the IOS router platforms' central processor, and flow-based DWFQ mechanisms running on the 7500 platform's VIP line cards. This section studies a CBWFQ mechanism that is supported in both nondistributed and distributed operation modes.

CBWFQ allocates a different subqueue for each traffic class compared with a subqueue per each flow in the flow-based versions of WFQ. So, you can use the existing flow-based implementations of WFQ to deliver CBWFQ in both nondistributed and distributed modes of operation by adding a traffic classification module in which each WFQ subqueue carries a traffic class rather than a traffic flow. Hence, CBWFQ is still

based on sequence number computation when run on the router's central processor and on calendar queue implementation when run on the 7500 platform's VIP line cards.

CBWFQ enables a user to directly specify the required minimum bandwidth per traffic class. This functionality is different from flow-based WFQ, where a flow's minimum bandwidth is derived indirectly based on the assigned weights to all active flows in the WFQ system.

DWFQ and CBWFQ differ in that you can run FQ within any DWFQ class, but in the case of CBWFQ, only default classes can run WFQ.

4.6 Priority Queuing

Priority queuing maintains four output subqueues—high, medium, normal, and low—in decreasing order of priority. A network administrator can classify flows to fall into any of these four queues. Packets on the highest-priority queue are transmitted first. When that queue empties, traffic on the next-highest-priority queue is transmitted, and so on. No packets in the medium-priority queue are serviced if packets in the high-priority queue are waiting for service.

Priority queuing is intended for environments where mission-critical data needs to be categorized as the highest priority, even if it means starving the lower-priority traffic at times of congestion. During congestion, mission-critical data can potentially take 100 percent of the bandwidth. If the high-priority traffic equals or exceeds the line rate for a period of time, priority queuing always lets the highest-priority traffic go before the next-highest-priority traffic and, in the worst case, drops important control traffic.

Priority queuing is implemented to classify packets into any of the priority queues based on input interface, simple and extended IP access lists, packet size, and application.

Note that unclassified traffic, which isn't classified to fall into any of the four priority queues, goes to the normal queue. The packets within a priority queue follow FIFO order of service.

4.7 Custom Queuing

Whereas priority queuing potentially guarantees the entire bandwidth for mission-critical data at the expense of low-priority data, custom queuing guarantees a minimum bandwidth for each traffic classification.

This bandwidth reservation discipline services each nonempty queue sequentially in a round-robin fashion, transmitting a configurable percentage of traffic on each queue. Custom queuing guarantees that mission-critical data is always assigned a certain percentage of the bandwidth, while assuring predictable throughput for other traffic. You can think of custom queuing as CBWFQ with lots of configuration details.

You can classify traffic into 16 queues. Apart from the 16 queues is a special 0 queue, called the system queue. The system queue handles high-priority packets, such as keepalive packets and control packets. User traffic cannot be classified into this queue. Custom queuing is implemented to classify IP packets into any of the 16 queues based on input interface, simple and extended IP access lists, packet size, and application type.

A popular use of custom queuing is to guarantee a certain bandwidth to a set of places selected by an access list. To allocate bandwidth to different queues, you must specify the byte count for each queue.

4.7.1 How Byte Count Is Used in Custom Queuing

In custom queuing, the router sends packets from a particular queue until the byte count is exceeded. Even after the byte count value is exceeded, the packet currently being transmitted is completely sent. Therefore, if you set the byte count to 100 bytes

and your protocol's packet size is 1024 bytes, every time this queue is serviced, 1024 bytes are sent, not 100 bytes.

Assume that one protocol has 500-byte packets, another has 300-byte packets, and a third has 100-byte packets. If you want to split the bandwidth evenly across all three protocols, you might choose to specify byte counts of 200, 200, and 200 for each queue. This configuration does not result in a 33/33/33 ratio, however. When the router services the first queue, it sends a single 500-byte packet; when it services the second queue, it sends a 300-byte packet; and when it services the third queue, it sends two 100-byte packets. The effective ratio is 50/30/20. Thus, setting the byte count too low can result in an unintended bandwidth allocation.

Large byte counts produce a "jerky" distribution, however. That is, if you assign 10 KB, 10 KB, and 10 KB to three queues in the example given, each protocol is serviced promptly when its queue is the one being serviced, but it might be a long time before the queue is serviced again. A better solution is to specify 500-byte, 600-byte, and 500-byte counts for the queue. This configuration results in a ratio of 31:38:31, which might be acceptable.

To service queues in a timely manner, and to ensure that the configured bandwidth allocation is as close as possible to the required bandwidth allocation, you must determine the byte count based on each protocol's packet size. Otherwise, your percentages might not match what you configure.

4.8 Scheduling Mechanisms for Voice Traffic

Voice traffic requires minimum delay and jitter (delay variation) to be intelligible to the listener. Although CBWFQ and custom queuing scheduling mechanisms can give bandwidth guarantees for voice, they cannot provide the jitter bounds acceptable for voice traffic. Voice has relatively low bandwidth demands (typically 64 kbps) but more stringent delay and jitter needs. Hence, CBWFQ and custom queuing are modified to implement a strict priority queue(s) to carry voice and thereby minimize drastically delay and jitter for voice traffic. A strict priority queue is also called a low latency queue.

Apart from voice, you can use the strict priority queues to carry other real-time, delay-sensitive application traffic.

4.8.1 CBWFQ with a Priority Queue

CBWFQ with a priority queue affords a scheduling mechanism providing a strict priority queuing scheme for delay-sensitive traffic, such as voice, and CBWFQ scheduling for differentiation and bandwidth guarantees among the other traffic classes. CBWFQ with a priority queue is also called low latency queuing (LLQ).

A voice priority queue in CBWFQ provides a single priority queue behaving similar to the high-priority queue discussed in the priority queuing section of this chapter. The remaining queues are CBWFQ queues (one queue per traffic class) delivering differentiation and bandwidth guarantees among the queues based on the configured weight or the bandwidth allocated for each queue.

You can identify voice traffic by its Real-time Transport Protocol (RTP) port numbers and classify it into a priority queue by using the **ip rtp priority** command on an output interface.

A busy priority queue can potentially starve the remaining queues on an interface, making the performance seen by the CBWFQ queues less than desirable. To alleviate this problem, a user can set up a maximum bandwidth against which the traffic serviced by the priority queue is policed. If the traffic exceeds the configured bandwidth, all excess traffic is dropped. Note that this might not always be the right way of policing traffic, because it is based on bandwidth rather than voice calls.

The sum of the bandwidths for the priority queue and the CBWFQ queues is not allowed to exceed 75 percent of the interface bandwidth. This is done to provide room for other unclassified traffic and Layer 2 encapsulations. However, using the **max-reserved-bandwidth** command, you can modify the default maximum reservable bandwidth.

Even with the priority queue in CBWFQ, the queuing system can't service and transmit a voice packet arriving on an empty priority queue immediately, because it needs to finish scheduling the packet it is already servicing and transmit it on the wire. Voice packets are small in size, but the data packet sizes carried in the CBWFQ queues can potentially be large. The larger the packet sizes, the greater the possible delay seen by the packets in the voice traffic.

This delay can be more perceivable on low-speed interfaces. To reduce voice traffic delay, Multilink Point-to-Point (MLPP) fragmentation needs to be configured for low-speed interfaces to fragment the large data packets so that you can interleave the small voice packets between the data fragments that make up a large data packet.

CHAPTER FIVE

PER HOP BEHAVIOR : RESOURCE ALLOCATION II

5 Per-Hop Behavior: Resource Allocation II

5.1 Modified Weighted Round Robin (MWRR)

Round-robin scheduling that serves a packet rather than an infinitesimal amount from each nonempty queue is the simplest way to simulate GPS. It works well in representing a GPS scheduler if all packets are the same size. Weighted Round Robin (WRR) is an extension of round-robin scheduling in which each flow is assigned a weight. WRR serves a flow in proportion to its weight.

WRR scheduling is well suited when an Asynchronous Transfer Mode (ATM) switch fabric is used for switching. Internally, the switch fabric treats packets as cells, and WRR is used to schedule the cells in the queues. WRR is essentially a cell-based round robin, whereby the weight determines how many cells are scheduled in each round robin. Hence, each queue shares the interface bandwidth of the ratio of the weights independent of packet sizes.

You can schedule only packets, not cells. Therefore, all cells of a packet are served in the same pass, even when you need to borrow some weight from the future. To support variable-size packets, MWRR uses a deficit counter associated with each WRR queue. This gives MWRR some characteristics of the Deficit Round Robin (DRR) algorithm described in the next section.

Before a queue is serviced, its deficit counter is initialized to the queue's weight. A packet from a queue is scheduled only if the deficit counter is greater than zero. After serving the n -cell packet, the resulting counter is decremented by n . Packets are scheduled as long as the counter is greater than 0. Otherwise, you skip to the next queue. In each coming round, the queue's deficit counter is incremented by the queue's weight. No packet is scheduled, however, if the deficit counter is still not greater than 0. If the counter becomes greater than 0, a packet is scheduled. After serving the packet, the deficit counter is decremented by the number of cells in the packet. By using a deficit counter, MWRR works independent of the variable-length packet sizes in the long run.

The effective bandwidth for each queue is proportional to its weight:

Effective queue bandwidth = (Queue weight x Interface bandwidth) ÷ Sum of all active queue weights)

5.1.1 An Illustration of MWRR Operation

In this example, consider three queues with the assigned weights shown in Table 5.1. Figure 5.1 depicts the queues along with their deficit counters. Deficit counters are used to make WRR support variable packet sizes.

Table 5. 1 Weights Associated with Each Queue

Queue	Weight
2	4
1	3
0	2

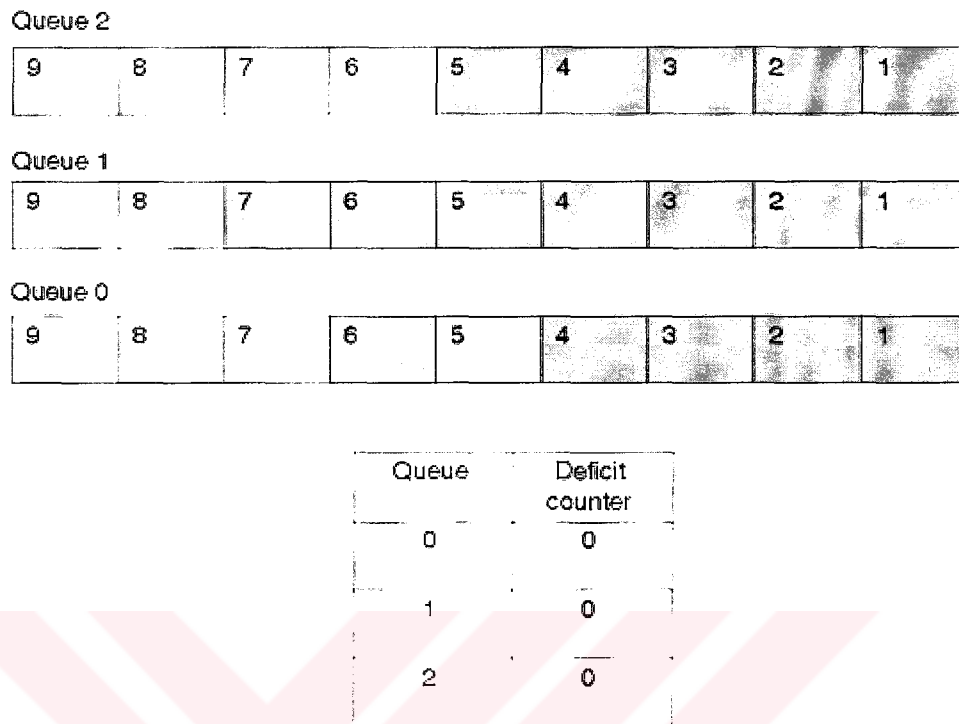


Figure 5.1 WRR Queues with Their Deficit Counters Before Start of Service

The queues show the cells queued, and the cells making up a packet are marked in the same shade of black. Queue 2, for example, has a 2-cell, 3-cell, and 4-cell packet in its queue.

Queue 0 is the first queue being served. The deficit counter is initialized to 2, the queue's weight. At the head of the queue is a 4-cell packet. Therefore, the deficit counter becomes $2 - 4 = -2$ after serving the packet. Because the deficit counter is negative, the queue cannot be served until it accumulates to a value greater than zero, as in Figure 5.2.

Queue 2

9	8	7	6	5	4	3	2	1
---	---	---	---	---	---	---	---	---

Queue 1

9	8	7	6	5	4	3	2	1
---	---	---	---	---	---	---	---	---

Queue 0

9	8	7	6	5
---	---	---	---	---

Queue	Deficit counter
2	0
1	0
0	-2

Figure 5. 2 MWRR After Serving Queue 0 in the First Round

Queue 1 is the next queue to be served. Its deficit counter is initialized to 3. The 3-cell packet at the head of the queue is served, which makes the deficit counter become $3 - 3 = 0$. Because the counter is not greater than zero, you skip to the next queue, as in Figure 5.3.

Queue 2

9	8	7	6	5	4	3	2	1
---	---	---	---	---	---	---	---	---

Queue 1

9	8	7	6	5	4
---	---	---	---	---	---

Queue 0

9	8	7	6	5
---	---	---	---	---

Queue	Deficit counter
2	0
1	0
0	-2

Figure 5.3 MWRR After Serving Queue 1 in the First Round

Now it is Queue 2's turn to be serviced. Its deficit counter is initialized to 4. The 2-cell packet at the head of the queue is served, which makes the deficit counter $4 - 2 = 2$. The next 3-cell packet is also served, as the deficit counter is greater than zero. After the 3-cell packet is served, the deficit counter is $2 - 3 = -1$, as in Figure 5.4.

Queue 2

9	8	7	6
---	---	---	---

Queue 1

9	8	7	6	5	4
---	---	---	---	---	---

Queue 0

9	8	7	6	5
---	---	---	---	---

Queue	Deficit counter
2	-1
1	0
0	-2

Figure 5. 4 MWRR After Serving Queue 2 in the First Round

Queue 0 is now served in the second round. The deficit counter from the last round was -2. Incrementing the deficit counter by the queue's weight makes the counter $-2 + 2 = 0$. No packet can be served because the deficit counter is still not greater than zero, so you skip to the next queue, as in Figure 5.5.

Queue 2

9	8	7	6
---	---	---	---

Queue 1

9	8	7	6	5	4
---	---	---	---	---	---

Queue 0

9	8	7	6	5
---	---	---	---	---

Queue	Deficit counter
2	-1
1	0
0	0

Figure 5. 5MWRR After Serving Queue 0 in the Second Round

Queue 1 has a deficit counter of zero in the first round. For the second round, the deficit counter is $0 + 3 = 3$. The 4-cell packet at the head of the queue is served, making the deficit counter $3 - 4 = -1$, as in Figure 5.5.

Queue 2

9	8	7	6
---	---	---	---

Queue 1

9	8
---	---

Queue 0

9	8	7	6	5
---	---	---	---	---

Queue	Deficit counter
2	-1
1	-1
0	0

Figure 5. 6 MWRR After Serving Queue 1 in the Second Round

In the second round, Queue 2's deficit counter from the first round is incremented by the queue's weight, making it $-1 + 4 = 3$. The 4-cell packet at the head of Queue 2 is served, making the deficit counter $3 - 4 = -1$. Because Queue 2 is now empty, the deficit counter is initialized to zero, as in Figure 5.7.

Queue 2

Queue 1

9	8
---	---

Queue 0

9	8	7	6	5
---	---	---	---	---

Queue	Deficit counter
2	0
1	-1
0	0

Figure 5. 7 MWRR After Serving Queue 2 in the Second Round

Now, it is again Queue 0's turn to be served. Its deficit counter becomes $0 + 2 = 2$. The 2-cell packet at the head of the queue is served, which results in a deficit counter of $2 - 2 = 0$. Now skip to Queue 1, as in Figure 5.8.

Queue 2

Queue 1

9	8
---	---

Queue 0

9	8	7
---	---	---

Queue	Deficit counter
2	0
1	-1
0	0

Figure 5. 8 MWRR After Serving Queue 0 in the Third Round

Queue 1's new deficit counter is $-1 + 3 = 2$. The 2-cell packet at the head of Queue 1 is served, resulting in a deficit counter of $2 - 2 = 0$. The resulting Queue 1 is now empty. Because Queue 2 is already empty, skip to Queue 0, as in Figure 5.9.

Queue 2

Queue 1

Queue 0

9	8	7
---	---	---

Queue	Deficit counter
2	0
1	0
0	0

Figure 5. 9 MWRR After Serving Queue 1 in the Third Round

Queue 0's deficit counter in the fourth round becomes 2. The 3-cell packet is served, which makes the deficit counter equal to -1. Because Queue 0 is now empty, reset the deficit counter to zero.

5.1.2 MWRR Implementation

MWRR is implemented in the Cisco Catalyst family of switches and the Cisco 8540 routers. These switches and routers differ in terms of the number of available MWRR queues and in the ways you can classify traffic into the queues.

MWRR in 8540 routers offers four queues between any interface pair based on Type of Service (ToS) group bits. Table 5.2 shows the ToS class allocation based on the IP precedence bits

Table 5.2 MWRR ToS Class Allocation

IP Precedence Bits	ToS Class Bits	ToS Class Assigned
000	00	0
001	00	1
010	01	2
011	01	3
100	10	0
101	10	1
110	11	2
111	11	3

Catalyst 6000 and 6500 series switches use MWRR with two queues, Queue 1 and Queue 2, based on the Layer 2 Institute of Electrical and Electronic Engineers (IEEE) 802.1p Class of Service (CoS) field. Frames with CoS values of 0–3 go to Queue 1, and frames with CoS values of 4–7 go to Queue 2. 6500 series switches also implement strict priority queues as part of MWRR to support the low-latency requirements of voice and other real-time traffic.

5.2 Modified Deficit Round Robin (MDRR)

This section discusses the MDRR algorithm for resource allocation available in the Cisco 12000 series routers. Within a DRR scheduler, each service queue has an associated quantum value—an average number of bytes served in each round—and a deficit counter initialized to the quantum value. Each nonempty flow queue is served in a round-robin fashion, scheduling on average packets of quantum bytes in each round. Packets in a service queue are served as long as the deficit counter is greater than zero. Each packet served decreases the deficit counter by a value equal to its length in bytes. A queue can no longer be served after the deficit counter becomes zero or negative. In each new round, each nonempty queue's deficit counter is incremented by its quantum value.

After a queue is served, the queue's deficit counter represents the amount of debit it incurred during the past round, depending on whether it was served equal to or more than its allocated quantum bytes. The amount the queue is entitled to be served in a subsequent round is reduced from the quantum bytes by a value equal to the deficit counter.

For efficiency, you should make the quantum size equal to the maximum packet size in the network. This ensures that the DRR scheduler always serves at least one packet from each nonempty flow queue.

The general DRR algorithm described in this section is modified to allow a low-latency queue. In MDRR, all queues are serviced in a round-robin fashion with the exception of the low-latency queue. You can define this queue to run in either one of two ways: in strict priority or alternate priority mode.

In strict priority mode, the low-latency queue is serviced whenever the queue is nonempty. This allows the lowest possible delay for this traffic. It is conceivable,

however, for the other queues to starve if the high-priority, low-latency queue is full for long periods of time because it can potentially take 100 percent of the interface bandwidth.

In alternate priority mode, the low-latency queue is serviced alternating between the low-latency queue and the remaining CoS queues. In addition to a low-latency queue, MDRR supports up to seven other queues, making the total number of queues to eight. Assuming that 0 is the low-latency queue, the queues are served in the following order: 0, 1, 0, 2, 0, 3, 0, 4, 0, 5, 0, 6, 0, 7.

In alternate priority mode the largest delay for Queue 0 is equal to the largest single quantum for the other queues rather than the sum of all the quanta for the queues if Queue 0 were served in traditional round-robin fashion.

In addition to being DRR-draining, MDRR is not conventional round-robin scheduling. Instead, DRR is modified in such a way that it limits the latency on one user-configurable queue, thus providing better jitter characteristics.

5.2.1 An MDRR Example

This example, which illustrates an alternate-priority low-latency queue, defines three queues—Queue 2, Queue 1, and Queue 0, with weights of 1, 2, and 1, respectively. Queue 2 is the low-latency queue running in alternate-priority mode. All the queues, along with their deficit counters, are shown in Figure 5.10.

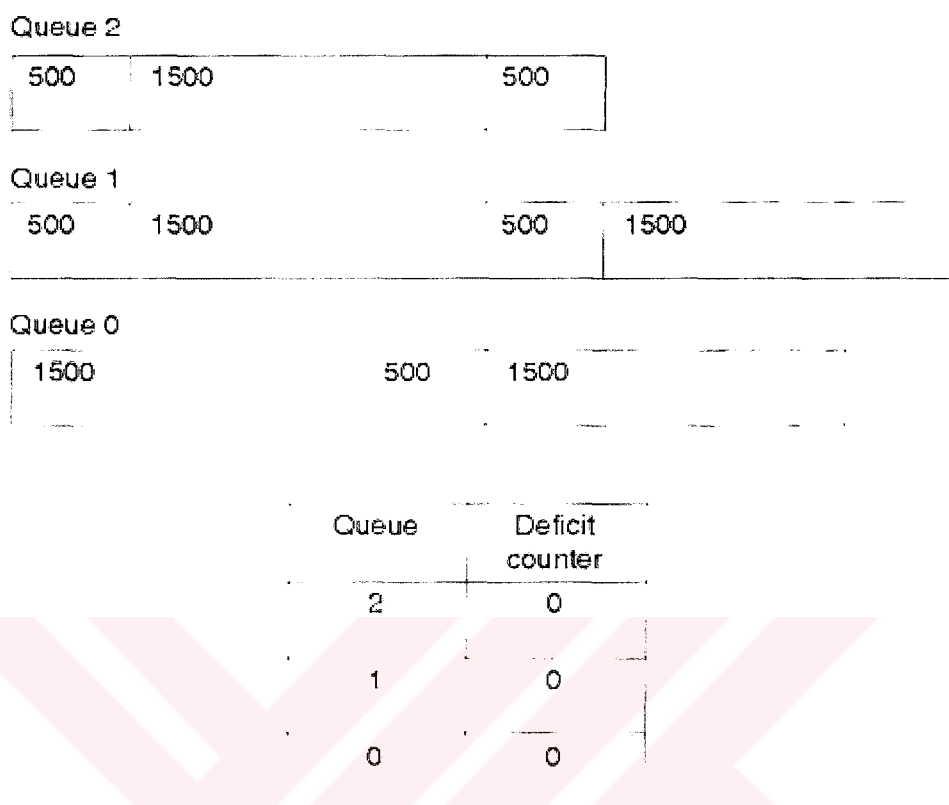


Figure 5. 10 Queues 0–2, Along with Their Deficit Counters

Table 5.4 provides the weight and quantum associated with each queue. When MDRR is run on the output interface queue, the interface maximum transmission unit (MTU) is used. When MDRR is run, the fabric queues.

Table 5. 3 Queues 0–2, Along with Their Associated Weights and Quantum Values

Queue Number	Weight	Quantum = Weight x MTU (MTU = 1500 Bytes)
Queue 2	1	1500
Queue 1	2	3000
Queue 0	1	1500

On the first pass, Queue 2 is served. Queue 2's deficit counter is initialized to equal its quantum value, 1500. Queue 2 is served as long as the deficit counter is greater than 0.

After serving a packet, Queue 2's size is subtracted from the deficit counter. The first 500-byte packet from the queue gets served because the deficit counter is 1500. Now, the deficit counter is updated as $1500 - 500 = 1000$. Therefore, the next packet is served. After the 1500-byte packet is served, the deficit counter becomes -500 and Queue 2 can no longer be served. Figure 5.11 shows the three queues and the deficit counters after Queue 2 is served.

Queue 2

500

Queue 1

500

1500

500

1500

Queue 0

1500

1000

1500

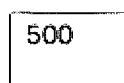
Queue	Deficit counter
2	-500
1	0
0	0

Figure 5. 11 MDRR After Serving Queue 2, Its First Pass

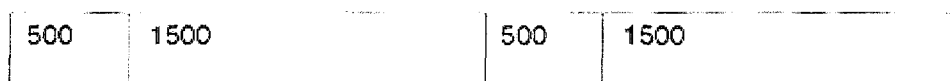
Because you are in alternate-priority mode, you alternate between serving Queue 2 and another queue. This other queue is selected in a round-robin fashion. Consider that in the round robin, it is now Queue 0's turn. The deficit counter is initialized to 1500, the quantum value for the queue. The first 1500-byte packet is served. After serving the first packet, its deficit counter is updated as $1500 - 1500 = 0$. Hence, no other packet can be

served in this pass. Figure 5.12 shows the three queues and their deficit counters after Queue 0 is served.

Queue 2



Queue 1



Queue 0



Queue	Deficit counter
2	-500
1	0
0	0

Figure 5. 12 MDRR After Serving Queue 0, Its First Pass

Because you alternate between the low-latency queue and the other queues served in the round robin, Queue 2 is served next. Queue 2's deficit counter is updated to $-500 + 1500 = 1000$. This allows the next packet in Queue 2 to be served. After sending the 500-byte packet, the deficit counter becomes 500. It could have served another packet, but Queue 2 is empty. Therefore, its deficit counter is reset to 0. An empty queue is not attended, and the deficit counter remains 0 until a packet arrives on the queue. Figure 5.13 shows the queues and the counters at this point.

Queue 2

Queue 1

500	1500	500	1500
-----	------	-----	------

Queue 0

1500	1000
------	------

Queue	Deficit counter
2	0
1	0
0	0

Figure 5. 13 MDRR After Serving Queue 2, Its Second Pass

Queue 1 is served next. Its deficit counter is initialized to 3000. This allows three packets to be sent, leaving the deficit counter to be $3000 - 1500 - 500 - 1500 = -500$. Figure 5.14 shows the queues and the deficit counters at this stage.

Queue 2

Queue 1

500

Queue 0

1500 1000

Queue	Deficit counter
2	0
1	-500
0	0

Figure 5. 14 MDRR After Serving Queue 1, Its First Pass

Queue 0 is the next queue serviced and sends two packets, making the deficit counter $1500 - 1000 - 1500 = -500$. Because the queue is now empty, the deficit counter is reset to 0. Figure 5.15 depicts the queues and counters at this stage.

Queue 2

Queue 1

500

Queue 0

Queue	Deficit counter
2	0
1	-500
0	0

Figure 5. 15 MDRR After Serving Queue 0, Its Second Pass

Queue 1 serves the remaining packet in a similar fashion in its next pass. Because the queue becomes empty, its deficit counter is reset to 0.

5.3 MDRR Implementation

Cisco 12000 series routers support MDRR. MDRR can run on the output interface queue (transmit [TX] side) or on the input interface queue (receive [RX] side) when feeding the fabric queues to the output interface.

Different hardware revisions of line cards termed as engine 0, 1, 2, 3, and so on, exist for Cisco 12000 series routers. The nature of MDRR support on a line card depends on the line card's hardware revision. Engine 0 supports MDRR software implementation. Line card hardware revisions, Engine 2 and above, support MDRR hardware implementation. (Coltun,1998)

5.3.1 MDRR on the RX

MDRR is implemented in either software or hardware on a line card. In a software implementation, each line card can send traffic to 16 destination slots because the 12000 series routers use a 16x16 switching fabric. For each destination slot, the switching fabric has eight CoS queues, making the total number of CoS queues 128 (16 x 8). You can configure each CoS queue independently.

In the hardware implementation, each line card has eight CoS queues per destination interface. With 16 destination slots and 16 interfaces per slot, the maximum number of CoS queues is $16 \times 16 \times 8 = 2048$. All the interfaces on a destination slot have the same CoS parameters.

5.3.2 MDRR on the TX

Each interface has eight CoS queues, which you can configure independently in both hardware- and software-based MDRR implementations.

Flexible mapping between IP precedence and the eight possible queues is offered in the MDRR implementation. MDRR allows a maximum of eight queues so that each IP

precedence value can be made its own queue. The mapping is flexible, however. The number of queues needed and the precedence values mapped to those queues are user-configurable. You can map one or more precedence values into a queue.

MDRR also offers individualized drop policy and bandwidth allocation. Each queue has its own associated Random Early Detection (RED) parameters that determine its drop thresholds and DRR quantum, the latter which determines how much bandwidth it gets. The quantum (in other words, the average number of bytes taken from the queue for each service) is user-configurable.



CHAPTER SIX

PER HOP BEHAVIOR : CONGESTION AVOIDANCE AND PACKET DROP POLICY

6 Per Hop Behavior: Congestion Avoidance and Packet Drop Policy

A packet drop policy is a queue management algorithm that manages the packets and queue length in a queuing system. Traditional first-in, first-out (FIFO) queue management uses a simple tail-drop policy, which drops any packet arriving on a full queue.

Transmission Control Protocol (TCP) is currently the dominant transport protocol used on the Internet. This chapter discusses TCP congestion control mechanisms and how TCP traffic reacts in a tail-drop scenario. It calls for an active queue management algorithm, Random Early Detection (RED), that avoids network congestion by dropping packets proactively to signal congestion to the TCP sources with end-to-end adaptive feedback control. (Stevens,1997)

Weighted Random Early Detection (WRED), also discussed in this chapter, allows different RED parameters based on the packet's Internet Protocol (IP) precedence value or traffic class. Flow WRED, a WRED extension that applies an increased non-zero drop probability to penalize flows taking more than their fair share of queue resources, and Explicit Congestion Notification (ECN), which enables congestion notification for incipient congestion by marking packets rather than dropping them, are also discussed. The chapter ends with a section on Selective Packet Discard (SPD), a selective packet drop policy on the IP input queue that queues packets for the IP process in a router.

6.1 TCP Slow Start and Congestion Avoidance

A TCP source uses a congestion window (cwnd) to perform congestion avoidance. When a new TCP session is established, the congestion window is initialized, based on the slow start mechanism, to one segment (the maximum segment size [MSS] is announced by the other end, or set to the default, and is typically 536 bytes or 512 bytes). The congestion window indicates the maximum amount of data the sender can send on a TCP session without receiving an acknowledgment.

When the first packet is acknowledged, the TCP source increases the window size to 2, at which point two packets can be sent. When the two packets are acknowledged, the window size increases to 4. In this manner, the congestion window size grows exponentially. Note that the increase in window size might not be exactly exponential, however, because a TCP receiver need not acknowledge each packet, as it typically uses delayed acknowledgments and sends an acknowledgment for every two packets it receives. The TCP source's behavior follows the slow start algorithm, which operates by sending new packets into the network at a rate of acknowledgments received from the other end. This makes TCP self-clocking.

In TCP, packet loss is an indicator of congestion. A TCP source detects congestion when it fails to receive an acknowledgment for a packet within the estimated retransmit timer timeout (RTT) period. In such a situation, it resets the congestion window to one segment and restarts the slow start algorithm. It also decreases the slow start threshold (sssthresh) to half the congestion window size at the time the retransmit was required. Note that when the TCP session is established, sssthresh is set to the size of the receiver window announced by the other end, or the default 65,535 bytes.

After an RTT timeout, a sender follows the slow start algorithm until the window size reaches sssthresh. Afterward, the window size increases linearly (by $1/\text{cwnd}$) per acknowledgment received. The window size is increased slowly after it reaches sssthresh

because ssthresh estimates the bandwidth-delay product for the TCP connection. TCP's slow start algorithm and congestion avoidance[1] behavior are depicted in [Figure 6-1](#).

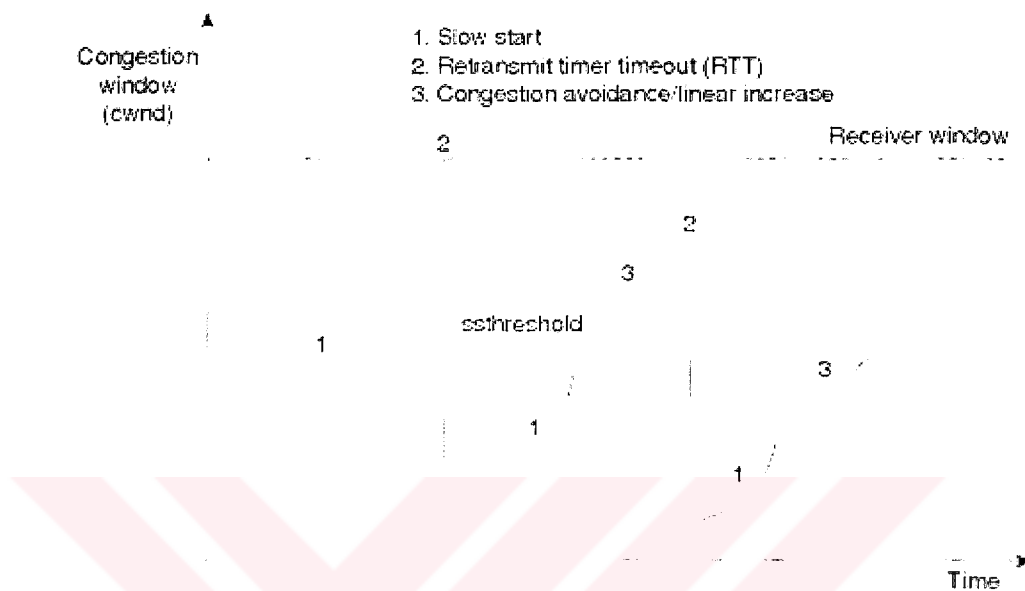


Figure 6. 1 TCP Congestion Window Showing Slow Start and Congestion Avoidance Operations

When packet loss occurs for reasons other than network congestion, waiting for the RTT times to expire can have an adverse performance impact, especially in high-speed networks. To avoid this scenario, TCP fast retransmit and recovery algorithms are used.

6.2 TCP Traffic Behavior in a Tail-Drop Scenario

Traditionally, packets arriving at the queue when the queue reaches its maximum queue length are dropped. This behavior continues until the queue decreases because of a packet transmission. This queue management technique is called tail-drop.

Because packet drop signals congestion to a TCP source, the tail-drop mechanism signals congestion only when the queue is completely full. A packet drop causes a TCP source to drop its window size to one segment and enter the slow start mode, which drastically slows down the traffic from the source.

Because many thousands of TCP flows transit a typical core router on the Internet or on a large-scale IP network, a tail-drop scenario will lead to packet loss for a large number of TCP sessions. The TCP sources of all these TCP sessions now slow down simultaneously, resulting in a significant drop in the traffic seen by the queue, and thereby reducing the queue size drastically.

All the TCP sources that went into slow start with a window size set to 1 segment now start to increase their window sizes exponentially, increasing the traffic at the queue. The increased traffic steadily causes the queue to build up again, leading to packet drops. The tail-drops again cause a large number of TCP sources to slow down, provoking an immediate drop in traffic. As these TCP sources increase their window sizes steadily, they can again lead to congestion and packet drops. (Stevens, 1997)

This cyclic behavior of significant traffic slowdown and congestion leads to a wave-type effect in the queue size often termed as global synchronization. Global synchronization behavior is depicted in Figure 6.2. It is called so because it synchronizes the behavior of a large number of TCP sources leading to undesirable queue fluctuations. It also leads to high delay jitter for the traffic and lowers the network's overall throughput.

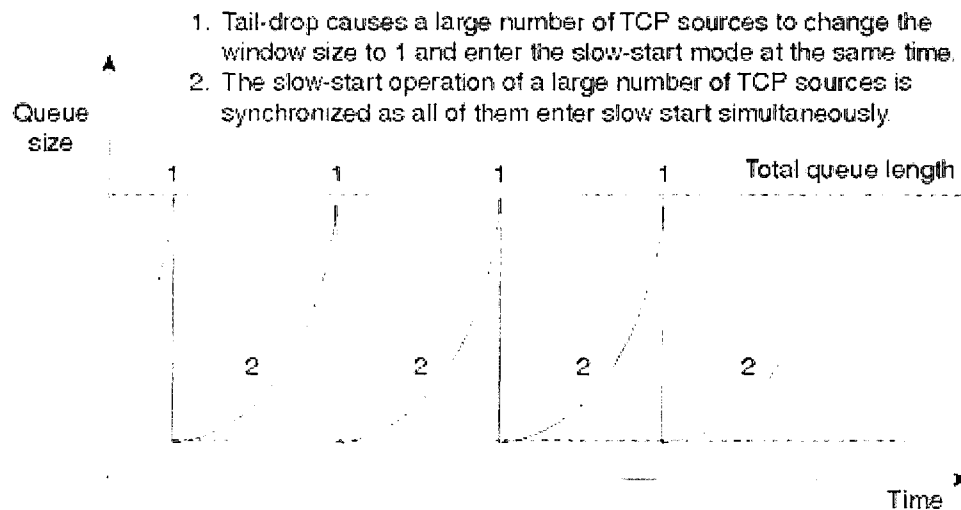


Figure 6. 2 Global Synchronization

6.3 RED—Proactive Queue Management for Congestion Avoidance

The behavior of TCP sources due to tail-drop underscores the need for proactive queue management to signal congestion before the queue is full, and to control queue sizes to minimize queuing delays. RED is a congestion avoidance mechanism proposed by Sally Floyd and Van Jacobson. It is an active queue management technique intended to provide considerable performance advantages over a traditional tail-drop approach.

RED takes a proactive approach to congestion. Instead of waiting until the queue is completely filled up, RED starts dropping packets with a non-zero drop probability after the average queue size exceeds a certain minimum threshold. A drop probability ensures that RED randomly drops packets from only a few flows, avoiding global synchronization. A packet drop is meant to signal the TCP source to slow down. Responsive TCP flows slow down after packet loss by going into a slow-start mode.

If the average queue continues to rise in spite of the random drops, the packet drop probability increases linearly to control the average queue size. As such, the packet drop

rate increases linearly as the average queue size increases from the minimum to the maximum threshold. The average queue size is strictly enforced to be the maximum threshold value because you drop all newly arriving packets (with a 100 percent probability, similar to tail-drop) when the average queue size exceeds maximum threshold. Thus, RED aims to reduce the average queue size, hence reducing queuing delay.

If the average queue length is already short, or below the minimum threshold, RED provides no actual benefit. On the other hand, if congestion is sustained for long periods of time, RED—with a deep queue and a high maximum threshold value—still exhibits tail-drop behavior. RED's main purpose is to accommodate temporary bursts and to detect and prevent sustained congestion by signaling sources to slow down. This results in congestion avoidance if the sources cooperate and reduce the traffic. If sources don't cooperate, any packet coming over the queue of maximum threshold length is dropped.

RED's main goals include:

- Minimizing the packets' packet delay jitter by controlling the average queue size

- Avoiding global synchronization for TCP traffic

- Supporting bursty traffic without bias

- Strictly enforcing the upper limit on the average queue limit.

RED is implemented by means of two different algorithms:

- Average queue size computation**— This determines the degree of burstiness allowed in the queue.

- Packet drop probability**— For a given average queue size, the probability that a packet is dropped determines how frequently the router drops packets.

These algorithms are discussed in the following sections.

6.3.1 The Average Queue Size Computation

RED calculates an exponentially weighted average queue size, rather than the current queue size, when deciding the packet drop probability. The current average queue length depends on the previous average and on the queue's current actual size. In using an average queue size, RED achieves its goal to not react to momentary burstiness in the network and react only to persistent congestion. The formula is

$$\text{average} = (\text{old_average} \times (1 - 1/2^n)) + (\text{current_queue_size} \times 1/2^n)$$

where n is the exponential weight factor, a user-configurable variable.

The exponential weight factor is the key parameter determining the significance of the old average and the current queue size values in the average queue size computation. A default value of 9 for the exponential weight factor n is seen to show best results. In calculating the average queue size, high values of n increase the significance of the old average queue size over the current queue size in computing the average queue size; low values of n increase the significance of the current queue size over the old average.

With high values of n , the average queue size closely tracks the old average queue size and more freely accommodates changes in the current queue size, resulting in the following RED behavior:

The average queue size moves slowly and is unlikely to change quickly, avoiding drastic swings in size.

RED accommodates temporary bursts in traffic, smoothing out the peaks and lows in the current queue size.

RED is slow to start dropping packets, but it can continue dropping packets for a time after the actual queue size falls below the minimum threshold.

If n is too high, RED does not react to congestion, as the current queue size becomes insignificant in calculating the average queue size. Packets are transmitted or dropped as if RED were not in effect.

With low values of n , the average queue size closely tracks the current queue size, resulting in the following RED behavior:

The average queue size moves rapidly and fluctuates with changes in the traffic levels.

The RED process responds quickly to long queues. When the queue falls below the minimum threshold, the process stops dropping packets.

If n is too low, RED overreacts to temporary traffic bursts and drops traffic unnecessarily.

6.3.2 Packet Drop Probability

Packet drop probability is a linear function of the average queue size. It also is based on the minimum threshold, maximum threshold, and mark probability denominator, which is the fraction of packets dropped when the average queue depth is at the maximum threshold. If the mark probability denominator is 10, for example, 1 out of every 10 packets is dropped when the average queue is at the maximum threshold. The packet drop probability formula is as follows:

$$\text{packet drop probability} = \left(\frac{(\text{average queue length} - \text{minimum threshold})}{\text{threshold}} \right) \times \frac{\text{mark probability}}{\text{denominator}}$$

When the average queue depth is above the minimum threshold, RED starts dropping packets. The packet drop rate increases linearly as the average queue size increases, until the average queue size reaches the maximum threshold.

When the average queue size is above the maximum threshold, all packets are dropped. Packet drop probability is illustrated in Figure 6.3.

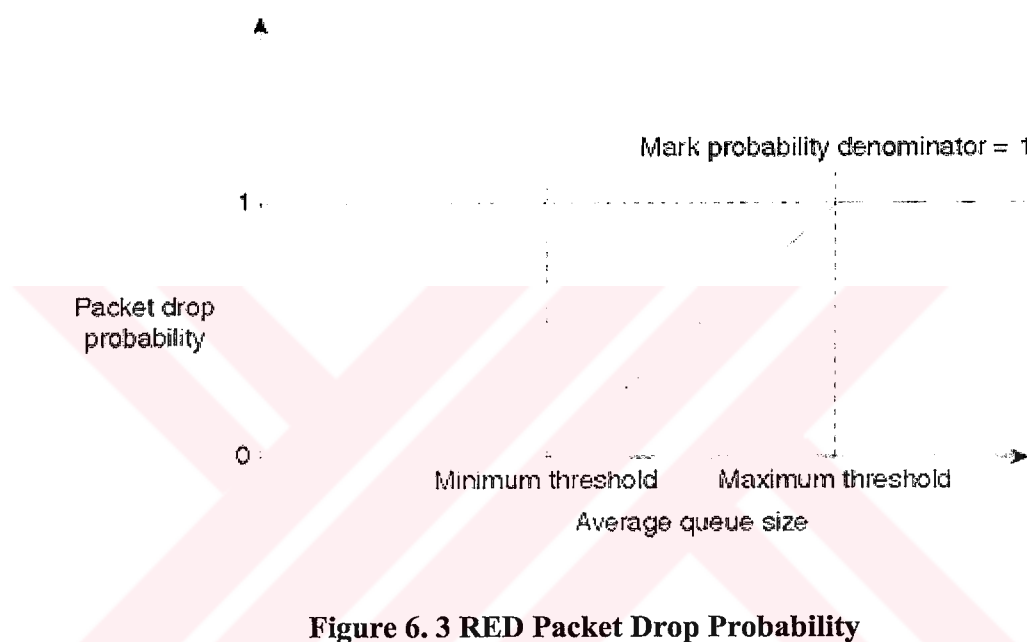


Figure 6.3 RED Packet Drop Probability

6.4 WRED

WRED introduces grades of service among packets based on a packet's drop probability and allows selective RED parameters based on IP precedence. As a result, WRED drops more aggressively for certain-precedence-level packets and less aggressively for other-precedence-level packets.

6.4.1 WRED Implementation

WRED can be run on the central processor of the router or on a distributed mode on Cisco's 7500 series routers with Versatile Interface Processors (VIPs). By default, the maximum threshold is the same for all precedence levels, but the minimum threshold varies with packet precedence. Hence, you drop the lower precedence packets more aggressively than the higher precedence packets. The default minimum threshold value for precedence 0 traffic is half the value of the maximum threshold.

You can enable WRED based on traffic classes by using modular QoS CLI.

In Cisco 12000 series routers, WRED is available in either hardware- or software-based implementations, depending on the hardware revision of the line card. Cisco 12000 series routers allow eight class of service (CoS) queues. You can map a CoS queue to carry packets of one or more precedence value(s). After the CoS queues are defined, RED parameters can be applied independently to the different CoS queues. Because this router platform uses a switch-based architecture, you can enable WRED on both the fabric queues on the receive side and the interface queues on the transmit side.

6.5 Flow WRED

Only adaptive TCP flows respond to a congestion signal and slow down, while nonadaptive UDP flows, which do not respond to congestion signals, don't slow down. For this reason, nonadaptive flows can send packets at a much higher rate than adaptive flows at times of congestion. Hence, greedy, nonadaptive flows tend to use a higher queue resource than the adaptive flows that slow down in response to congestion signals. Flow WRED modifies WRED such that it penalizes flows taking up more than their fair share of queue resources.

To provide fairness among the active traffic flows in the queue, WRED classifies all arriving packets into the queue based on their flow and precedence. It also maintains state for all active flows, or flows that have packets in the queue. This state information

is used to determine the fair amount of queue resources for each flow (queue size/number of active flows), and flows taking more than their fair share are penalized more than the others are.

To accommodate for a flow's traffic burstiness, you can increase each flow's fair share by a scaling factor before it gets penalized using the following formulas:

Fair share of queue resources per active flow = queue size/number of active flows

Scaled fair share of queue resources per flow = (queue size/number of active flows) x scaling factor

A flow exceeding the scaled fair share of queue resources per flow in the queue is penalized by an increase in the non-zero drop probability for all the newly arriving packets in the queue.

As an example, consider a packet arriving on a queue running Flow WRED. Flow WRED considers both the IP precedence value in the packet and the flow state information to determine the packet's drop probability. The packet's IP precedence determines the configured (or the default) minimum and maximum WRED thresholds for the packet. If the average queue size is below the minimum threshold, the packet gets zero drop probability (in other words, it is not dropped). If the average queue size is in between the packet's minimum and maximum threshold (as determined by the packet's IP precedence), the flow state information is taken into consideration. If the packet belongs to a flow that exceeded the scaled fair share of queue resources per flow in the queue, you increase the packet's drop probability by decreasing the WRED maximum threshold as follows:

New Maximum threshold = Minimum threshold + ((Maximum threshold – Minimum threshold)/2)

The non-zero probability is then derived based on the minimum threshold and the new maximum threshold values. Because the drop probability curve is much steeper now, as shown in Figure 6.4, you apply a higher drop probability on the packet. If the flow is within its fair allocation of queue resources, the packet gets a non-zero drop probability determined by the normal WRED calculation.

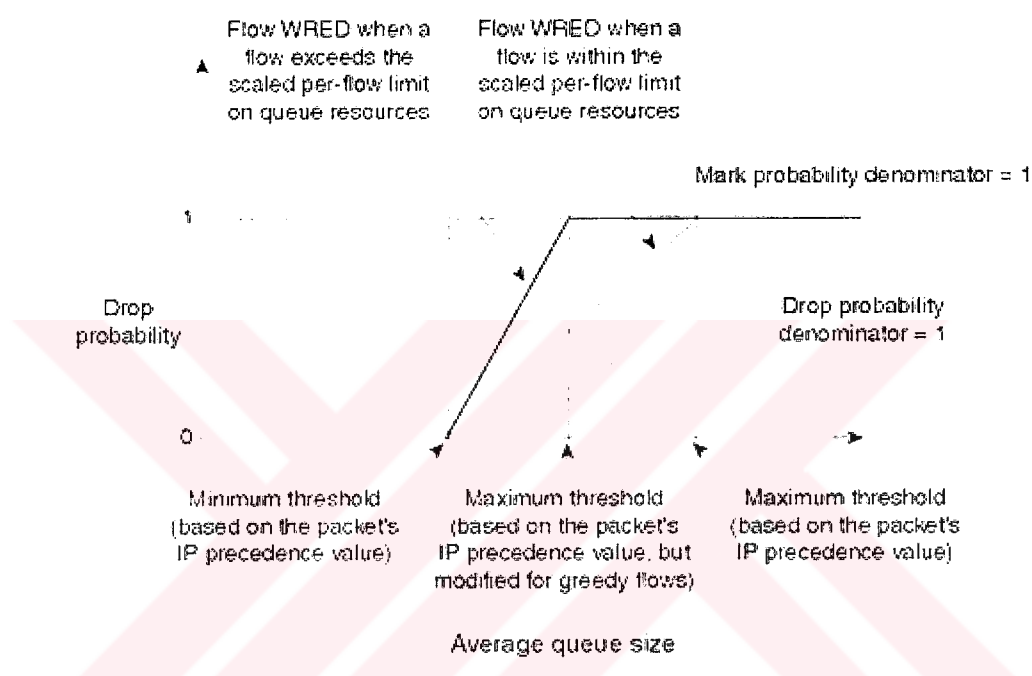


Figure 6. 4 Packet Drop Probability with Flow WRED

If the average queue size exceeds the maximum threshold, continue to drop packets using a process similar to that used in the WRED operation.

Flow WRED increases the probability of a packet getting dropped only if the packet belongs to a flow whose packets in the queue exceed the scaled per-flow limit. Otherwise, Flow WRED operates similar to WRED.

6.6 ECN

Thus far, in active queue management using WRED, packets have been dropped to signal congestion to a TCP source. ECN provides routers the functionality to signal congestion to a TCP source by marking a packet header rather than dropping the packet. In the scenarios where a WRED-enabled router dropped a packet to signal congestion, it can now set the ECN bit in the packet, avoiding potential delays due to packet retransmissions caused by packet loss.

ECN functionality requires support for a Congestion Experienced (CE) bit in the IP header and a transport protocol that understood the CE bit. An ECN field of 2 bits in the IP header is provided for this purpose. The ECN-Capable Transport (ECT) bit is set by the TCP source to indicate that the transport protocol's end-points are ECN-capable. The CE bit is set by the router to indicate congestion to the end nodes. (Stevens, 1997)

Bits 6 and 7 in the IPv4 Type of Service (ToS) byte form the ECN field and are designated as the ECT bit and the CE bit, respectively. Bits 6 and 7 are listed in differentiated services architecture as currently unused.

For TCP, ECN requires three new mechanisms:

ECN Capability Negotiation— The TCP endpoints negotiate during setup to determine if they are both ECN-capable.

ECN-Echo flag in the TCP header— The TCP receiver uses the ECN-Echo flag to inform the TCP source that a CE packet has been received.

Congestion Window Reduced (CWR) flag in the TCP header— The TCP source uses the CWR flag to inform the TCP receiver that the congestion window has been reduced.

ECN functionality is still under discussion in the standard bodies.

6.7 SPD

SPD helps differentiate important control traffic (such as routing protocol packets) over normal data traffic to the router. This enables the router to keep its Interior Gateway Protocol (IGP) and Border Gateway Protocol (BGP) routing information during congestion by enqueueing routing protocol packets over the normal data traffic.

SPD implements a selective packet drop policy on the router's IP process queue. Therefore, it applies to only process switched traffic. Even when the router is using route-cache forwarding (also called fast switching), some of the transit data traffic still needs to be process switched in order to create a route-cache entry. When a router is using CEF, though, all transit data traffic is usually CEF switched and the only packets that reach the IP process input queue are the important control packets such as routing and keepalives, normal data traffic destined to the router, and transit traffic that is not CEF supported. (Coltun,1998)

Traffic arriving at the IP process input queue is classified in three ways:

Important IP control traffic (routing protocol packets), often called priority traffic.

Normal IP traffic, such as telnet/ping packets to a router interface, IP packets with options, and any IP feature or encapsulation not supported by CEF.

Aggressive dropable packets. These are IP packets that fail the IP sanity check; that is, they might have incorrect checksums, invalid versions, an expired Time-to-Live (TTL) value, an invalid UDP/TCP port number, an invalid IP protocol field, and so on. Most of these packets trigger an Internet Control Message Protocol (ICMP) packet to notify the sender of the bad packet. A small number of these packets are generated due to normal utilities such as a trace route. Such packets in large numbers, however, can be part of a malicious smurf attack intended to cripple the router by filling up the IP process queue. It is

essential to selectively drop these packets without losing the important control information.

SPD operates in the following modes:

Disabled— The SPD feature is disabled on the router.

Normal— The IP input queue is less than the queue minimum threshold. No packets are dropped.

Random drop— The IP input queue is more than the minimum threshold but less than the maximum threshold. Normal IP packets are dropped in this mode, with a drop probability shown in the following formula:

$$\text{drop probability} = (\text{queue length} - \text{min. threshold}) / (\text{max. threshold} - \text{min. threshold})$$

Random drops are called SPD flushes. Important IP control traffic is still enqueued.

Full drop— The IP input queue is above the maximum threshold. All normal IP traffic is dropped. Important IP control traffic is still received to a special process level queue, termed the priority queue, that is drained before the normal one.

Aggressive drop— This is a special aggressive drop mode for IP packets failing the sanity check. All bad IP packets are dropped when the input queue is above the minimum threshold. You can enable this special drop mechanism using the **ip spd mode aggressive** command.

Figure 6.5 illustrates SPD operation and its modes.

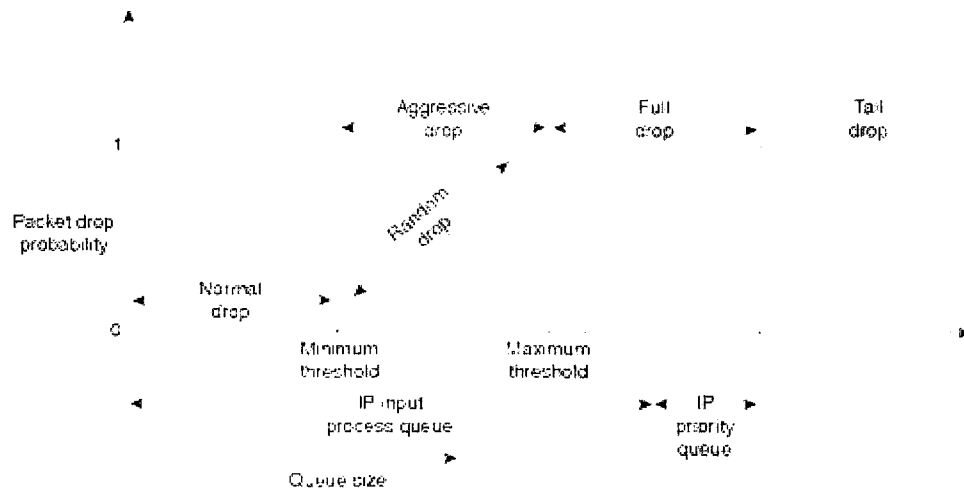


Figure 6. 5 SPD Packet Drop Modes

CHAPTER SEVEN

INTEGRATED SERVICES : RSVP

7 Integrated Services: RSVP

The previous chapters discussed the Differentiated Services (diffserv) architecture and its enabling functions. They discussed how the Differentiated Services Code Point (DSCP) and Internet Protocol (IP) precedence in a packet's IP header are used to classify traffic based on the traffic's service level to indicate the required per-hop behavior within a network. Now, in the Integrated Services (intserv) architecture, we will discuss how the network is informed about the various traffic flows' divergent needs?

In intserv, a quality of service (QoS) signaling protocol, Resource Reservation Protocol (RSVP) is used for this purpose. RSVP is a QoS signaling protocol that enables end applications requiring certain guaranteed services to signal their end-to-end QoS requirements to obtain service guarantees from the network.

This chapter discusses the RSVP protocol, its control messages, its operation, and other details. The two integrated service types—controlled load and guaranteed service—also are covered.

7.1 RSVP

The Internet Engineering Task Force (IETF) specified RSVP as a signaling protocol for the intserv architecture. RSVP enables applications to signal per-flow QoS requirements to the network. Service parameters are used to specifically quantify these requirements for admission control.

RSVP is used in multicast applications such as audio/video conferencing and broadcasting. Although the initial target for RSVP is multimedia traffic, there is a clear interest in reserving bandwidth for unicast traffic such as Network File System (NFS), and for Virtual Private Network (VPN) management.

RSVP signals resource reservation requests along the routed path available within the network. It does not perform its own routing; instead, it is designed to use the Internet's current robust routing protocols. Like other IP traffic, it depends on the underlying routing protocol to determine the path for both its data and its control traffic. As the routing protocol information adapts to network topology changes, RSVP reservations are carried over to the new path. This modularity helps RSVP to function effectively with any underlying routing service. RSVP provides opaque transport of traffic control and policy control messages, and provides transparent operation through nonsupporting regions. (Mankin,1997)

7.1.1 RSVP Operation

End systems use RSVP to request a specific QoS from the network on behalf of an application data stream. RSVP requests are carried through the network, visiting each node the network uses to carry the stream. At each node, RSVP attempts to make a resource reservation for the stream.

RSVP-enabled routers help deliver the right flows to the right locations. Figure 7.1 gives an overview of the important modules and the data and control flow information of a client and router running RSVP.

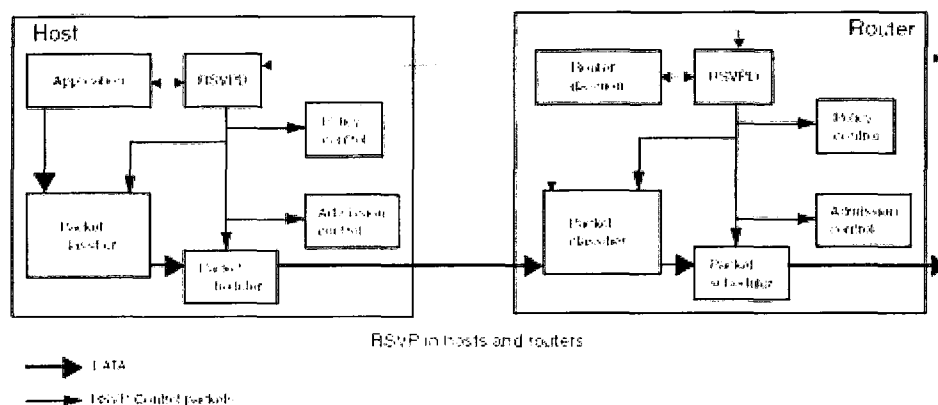


Figure 7.1 Data and Control Flow Information of a Client and Router Running RSVP

The RSVP daemon in a router communicates with two local decision modules—admission control and policy control—before making a resource reservation. Admission control determines whether the node has sufficient available resources to supply the requested QoS. Policy control determines whether the user has administrative permission to make the reservation. If either check fails, the RSVP daemon sends an error notification to the application process that originated the request. If both checks succeed, the RSVP daemon sets parameters in a packet classifier and a packet scheduler to obtain the desired QoS. The packet classifier determines the QoS class for each packet, and the packet scheduler orders packet transmission based on its QoS class. The Weighted Fair Queuing (WFQ) and Weighted Random Early Detection (WRED) disciplines provide scheduler support for QoS.

During the admission control decision process, a reservation for the requested capacity is put in place if sufficient capacity remains in the requested traffic class. Otherwise, the admission request is refused, but the traffic is still forwarded with the default service for that traffic's traffic class. In many cases, even an admission request that failed at one or more routers can still supply acceptable quality, as it might have succeeded in installing a reservation in all the routers suffering congestion. This is because other reservations might not be fully utilizing their reserved capacity.

Reservations must follow on the same unicast path or on the multicast tree at all times. In case of link failures, the router should inform the RSVP daemon so that RSVP messages are generated on a new route.

You can break down the process of installing a reservation into five distinct steps:

Data senders send RSVP PATH control messages the same way they send regular data traffic. These messages describe the data they are sending or intend to send.

Each RSVP router intercepts the PATH messages, saves the previous hop IP address, writes its own address as the previous hop, and sends the updated message along the same route the application data is using.

Receiver stations select a subset of the sessions for which they are receiving PATH information and request RSVP resource reservations from the previous hop router using an RSVP RESV message. The RSVP RESV messages going from a receiver to a sender take an exact reverse path when compared to the path taken by the RSVP PATH messages.

The RSVP routers determine whether they can honor those RESV requests. If they can't, they refuse the reservations. If they can, they merge reservation requests being received and request a reservation from the previous hop router.

The senders receive reservation requests from the next hop routers indicating that reservations are in place. Note that the actual reservation allocation is made by the RESV messages.

Figure 7.2 shows the RSVP reservation setup mechanism.

QoS, including

Guaranteed service— PATH messages also describe the worst-case delays in the network.

Controlled load service— The routers guarantee only that network delays will be maximized.

7.1.3 RSVP Messages

RSVP uses seven message types for its operation: two required message types—PATH and RESV—and five optional message types—PATH ERROR, PATH TEARDOWN, RESV ERROR, RESV CONFIRM, and RSV TEARDOWN. The RSVP routers and clients use them to create and maintain reservation states.

RSVP usually runs directly over the IP. As such, RSVP messages are unreliable datagrams. They help create soft states within the routers, and a periodic refresh is needed.

The following are the sender message types:

PATH messages are sent periodically by senders. The senders describe the flows in terms of the source and destination IP addresses, the IP protocol, and the User Datagram Protocol (UDP) or Transmission Control Protocol (TCP) ports, if applicable. They quantify the expected resource requirements for this data by specifying its mean rate and burst size. (Mankin,1997)

They are sent to the multicast group or unicast destination of the flow for which the reservation is being made; RSVP routers detect them because they are sent in UDP messages to a particular UDP port, or because they have the IP Router Alert option in their IP header. A router creates a Path State Block (PSB) when the PATH messages are received.

PATH messages contain a periodic hello interval indicating how frequently the sender sends them. The default hello interval is 30 seconds. It is important to keep the hello interval small, or to have a fast retransmit scheme, because lost PATH messages can result in poor performance for VoIP, as that would delay the establishment of an RSVP reservation along the path of the VoIP call. The PSB is discarded upon a PATH TEARDOWN or ingress link failure, or when the PSB has not been refreshed by a new PATH message after four hello intervals.

When error(s) in a PATH message are found, the optional PATH ERROR message is sent by the receiver or router, notifying the sender of the problem. Typically, this is a fundamental format or integrity check fault.

PATH TEARDOWN messages are sent to the multicast group with the sender's source address when the PATH must be flushed from the database, either due to a link failure or because the sender is exiting the multicast group.

The following are the receiver message types:

RESV messages are sent periodically by receivers. The receivers describe the flows and resource guarantees they need using information derived from the PATH messages, in terms of the source and destination IP addresses, the IP protocol, and the UDP or TCP ports, if applicable. They also describe the bit rate and delay characteristics they need, using flow specifications. They traverse through all RSVP routers along the routed path to the sender for which the reservation is being made. Routers create Reservation State Blocks (RSBs) when RESV messages (FlowSpec, FilterSpec) are granted.

RESV messages contain a periodic hello interval indicating how frequently the receiver sends them. The RSB is discarded upon a RESV TEARDOWN or ingress link failure, or when they have not been refreshed by a new RESV message after four hello intervals.

When error(s) in an RESV message are found, an RESV ERROR message is sent by a sender or router informing the receiver of a problem. Typically, it is due to a fundamental format or integrity check fault, or because insufficient resources were available to make the requested guarantees.

When the effect of an RESV message applies end to end and a receiver requests notification of the fact, RESV CONFIRM messages are sent to the receivers or merge point routers.

RESV TEARDOWN messages are sent when an RSB must be flushed from the database, either due to a link failure or because the sender is exiting the multicast group.

7.2 Reservation Styles

You can categorize RSVP flow reservations into two major types—distinct and shared—which are discussed in the following sections.

7.2.1 Distinct Reservations

Distinct reservations are appropriate for those applications in which multiple data sources are likely to transmit simultaneously. In a video application, each sender emits a distinct data stream requiring separate admission control and queue management on the routers along its path to the receiver. Such a flow, therefore, requires a separate reservation per sender on each link along the path.

Distinct reservations are explicit about the sender and are installed using a Fixed Filter (FF) reservation style. Symbolically, you can represent an FF-style reservation request by FF (S,Q), where the S represents the sender selection and the Q represents the FlowSpec.

Unicast applications form the simplest case of a distinct reservation, in which there is one sender and one receiver.

7.2.2 Shared Reservations

Shared reservations are appropriate for those applications in which multiple data sources are unlikely to transmit simultaneously. Digitized audio applications, such as VoIP, are suitable for shared reservations. In this case, as a small number of people talk at any time, a limited number of senders send at any given time. Such a flow, therefore, does not require a separate reservation per sender; it requires a single reservation that you can apply to any sender within a set, as needed.

RSVP refers to such a flow as a shared flow and installs it using a shared explicit or wildcard reservation scope. These two reservation styles are discussed below.

The Shared Explicit (SE) reservation style specifically identifies the flows that reserve network resources.

Symbolically, you can represent an SE-style reservation request by $SE((S1,S2)\{Q\})$, where the $S1, S2, \dots$ represent specific senders for the reservation and the Q represents the FlowSpec.

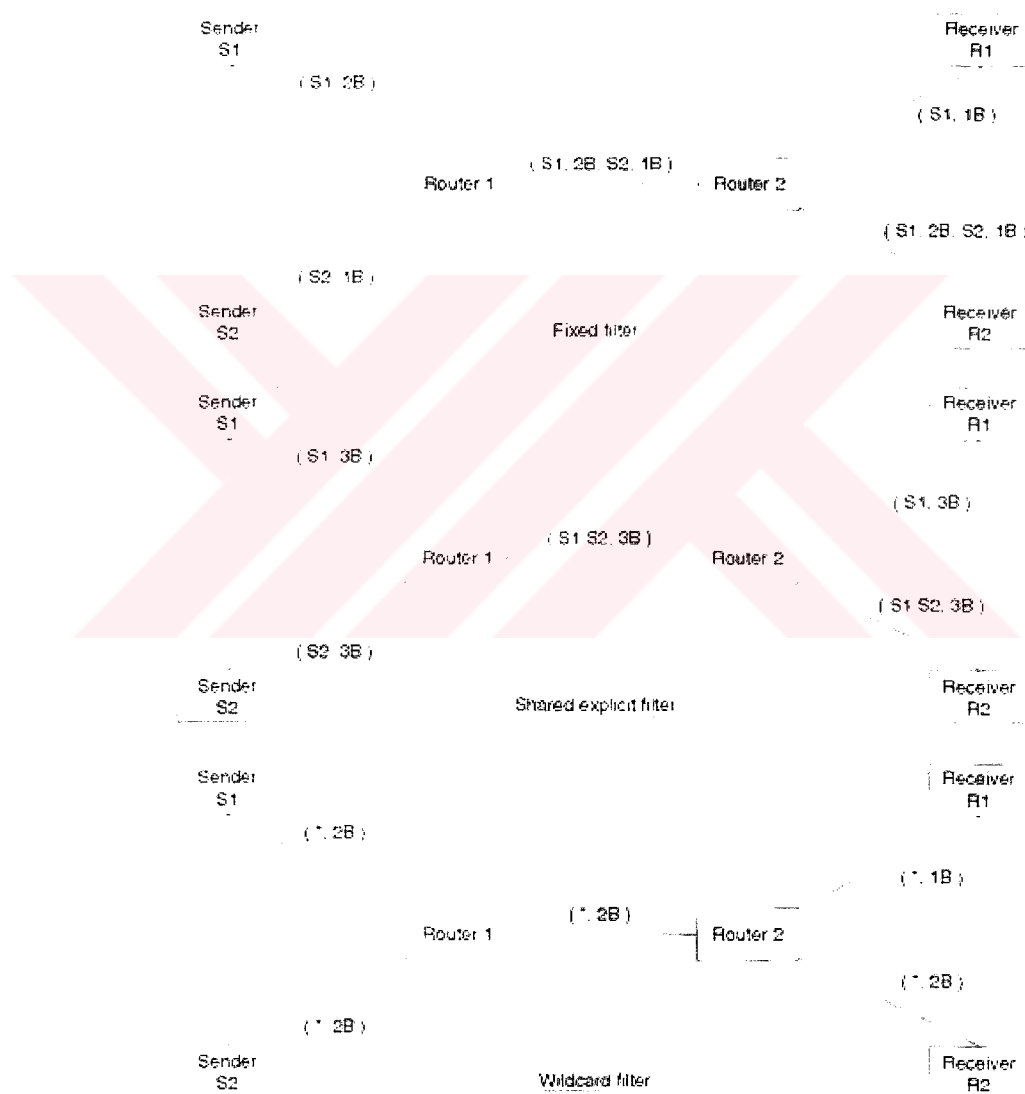
The Wildcard Filter (WF) reserves bandwidth and delay characteristics for any sender. It does not admit the sender's specification; it accepts all senders, which is denoted by setting the source address and port to zero.

Symbolically, you can represent a WF-style reservation request by $WF(*\{Q\})$, where the asterisk represents the wildcard sender selection and the Q represents the FlowSpec.

Table 7.1 shows the different reservation filters based on the reservation styles and sender's scope, and Figure 7.3 illustrates the three reservation filter styles described previously.

Table 7. 1 Different Reservation Filters, Based on Style and Sender Scope

Sender Selection Scope	Reservation Styles	
	Distinct	Shared
Explicit	FF	SE
Wildcard	None defined	WF

**Figure 7. 3 Examples of the Three Reservation Filter Styles**

7.3 Service Types

RSVP provides two types of integrated services that the receivers can request through their RSVP RESV messages: controlled load service and guaranteed bit rate.

7.3.1 Controlled Load

Under controlled load service, the network guarantees that the reserved flow will reach its destination with a minimum of interference from the best-effort traffic. In addition, Cisco's implementation of the service offers isolation between the reserved flows. Flow isolation allows a flow reservation to operate unaffected by the presence of any other flow reservations that might exist in the network.

The controlled load service is primarily intended for a broad class of applications running on the Internet today that are sensitive to overloaded conditions. These applications work well on unloaded nets, but they degrade quickly under overloaded conditions. An example of such an application is File Transfer Protocol (FTP).

7.3.2 Guaranteed Bit Rate

Guaranteed bit rate service provides a delay-bounded service with no queuing loss for all conforming datagrams, assuming no failure of network components or changes in routing during the life of the flow. Under this service, the network guarantees a minimum of interference from best-effort traffic, isolation between reserved flows, and a quantified worst-case delay. (Mankin, 1997)

Guaranteed service only guarantees the worst-case queuing delay, not the datagrams' minimal or average delay. Furthermore, to compute the maximum delay a datagram will experience, the path's fixed latency (propagation delay and transmission delay) must be determined and added to the guaranteed worst-case queuing delay. It is important to note that the guaranteed service guarantees maximum queuing delay, but not the maximum

overall end-to-end delay, because the total delay's remaining components, such as propagation and transmission delay, depend entirely on the traffic path.

The worst-case queuing delay that the guaranteed service promises is the cumulative delay as seen by the PATH message before it reaches the receiver. PATH messages carry the delay information along the path from the source to the receiver and provide the receiver an accurate estimation of the exact delay conditions along the path at any time. A receiver uses this delay information while making a request for a guaranteed service. (Coltun, 1998)

Guaranteed service suits well for playback and real-time applications. Playback applications use a jitter buffer to offset the delay variations of the packet arrivals to function well. Guaranteed service, by guaranteeing the worst-case queuing delay, aids in the estimation of required jitter buffer size. Real-time applications get guaranteed bandwidth and delay service.

Both controlled load and guaranteed bit rate use a token bucket to describe a data flow's traffic parameters. The token bucket is a rate control mechanism that identifies a mean rate (how much can be sent or forwarded per unit time on average), a burst size (how much can be sent within a given unit of time without scheduling concerns), and a measurement interval (the time quantum). Details regarding the token bucket rate control mechanism are discussed in Chapter 3.

Under both services, a receiver requests for a certain bit rate and burst size in an RESV message. The WFQ scheduler and WRED queue management techniques with preferential weights assure that traffic to the receiver has a bounded latency. The latency bound is not specified, however. The controlled load service only promises "good service," and the guaranteed service provides information (in the PATH messages) from which the delay bounds can be calculated.

7.4 RSVP Media Support

Point-to-point media, such as serial lines, are well modeled for RSVP, but shared media, such as Ethernet, Asynchronous Transfer Mode (ATM), Frame Relay, and X.25, are not.

In shared media, there is no way to ensure that others on the same segment do not send traffic that might fill up the segment. It can work fine, however, in a segment that is lightly loaded or has only a few sources. To address this problem of resource reservation on shared media, SBM, a protocol for RSVP-based admission control over Institute of Electrical and Electronic Engineers (IEEE) 802-style networks, was developed. SBM defines some RSVP extensions that provide a method of mapping RSVP onto Layer 2 devices and networks. It is discussed further in Chapter 8.

In the Cisco-specific implementation, to support RSVP over ATM, RSVP creates a Variable Bit Rate (VBR) ATM switched virtual circuit (SVC) for every reservation across an ATM network. It then redirects all reserved traffic down the corresponding SVCs and relies on the ATM interface to police the traffic.

7.5 RSVP Scalability

One drawback of RSVP is that the amount of state information required increases with the number of per-flow reservations. As many hundreds of thousands of real-time unicast and multicast flows can exist in the Internet backbone at any time, state information on a per-flow granularity is considered a nonscalable solution for Internet backbones.

RSVP with per-flow reservations scales well for medium-size corporate intranets with link speeds of DS3 or less. For large intranets and for Internet service provider (ISP) backbones, you can make RSVP scale well when you use it with large multicast groups, large static classes, or an aggregation of flows at the edges rather than per-flow reservations. RSVP reservation aggregation proposes to aggregate several end-to-end

reservations sharing common ingress and egress routers into one large, end-to-end reservation. Another approach is to use RSVP at the edges and diffserv in the network backbone to address RSVP scalability issues in the core of a large network. RSVP-to-diffserv mapping is discussed in Chapter 2, "Differentiated Services Architecture."

The service provider networks and the Internet of the future are assumed to have, for the most part, sufficient capacity to carry normal telephony traffic. If a network is engineered with sufficient capacity, you can provision all telephony traffic as a single class. Depending on available network capacity, telephony traffic can require relatively modest capacity, which is given some fraction of the capacity overall, without the need for resource allocation per individual call. (Mankin, 1997)



CHAPTER EIGHT

LAYER 2 QoS : INTERWORKING WITH IP QoS

8 Layer 2 QoS: Interworking with IP QoS

Most networks use diverse network technologies. Popular link-layer technologies, such as Asynchronous Transfer Mode (ATM), Frame Relay, and Ethernet, offer quality of service (QoS) functionality at Layer 2. This chapter discusses these multi-access Layer 2 technologies and their QoS offerings. Because QoS is only as good as its weakest link, it is necessary to ensure that Internet Protocol (IP) QoS is seamless across diverse link technologies. To provide end-to-end IP QoS, it is necessary that QoS at Layer 2 map to IP QoS, and vice versa.

8.1 ATM

ATM is a fixed-size cell-switching and multiplexing technology. It is connection-oriented, and a virtual circuit (VC) must be set up across the ATM network before any user data can be transferred between two or more ATM attached devices. Primarily, ATM has two types of connections, or VCs: permanent virtual circuits (PVCs) and switched virtual circuits (SVCs). PVCs are generally static and need a manual or external configuration to set them up. SVCs are dynamic and are created based on demand. Their setup requires a signaling protocol between the ATM endpoints and ATM switches. (Romanow et al.,1995)

An ATM network is composed of ATM switches and ATM end nodes, or hosts. The cell header contains the information the ATM switches use to switch ATM cells. The

data link layer is broken down into two sublayers: the ATM Adaptation Layer (AAL) and the ATM layer. You map the different services to the common ATM layer through the AAL. Higher layers pass down the user information in the form of bits to the AAL. User information gets encapsulated into an AAL frame, and then the ATM layer breaks the information down into ATM cells. The reverse is done at the receiver end. This process is known as segmentation and reassembly (SAR).

8.1.1 ATM Cell Format

Each ATM cell contains 53 bytes—5 bytes of cell header information and 48 bytes of user information, or payload. Two ATM cell formats exist: User-to-Network Interface (UNI), which defines the format for cells between a user and an ATM switch; and Network-to-Network Interface (NNI), which defines the format for cells between the switching nodes. The cell formats are shown in Figure 8.1.

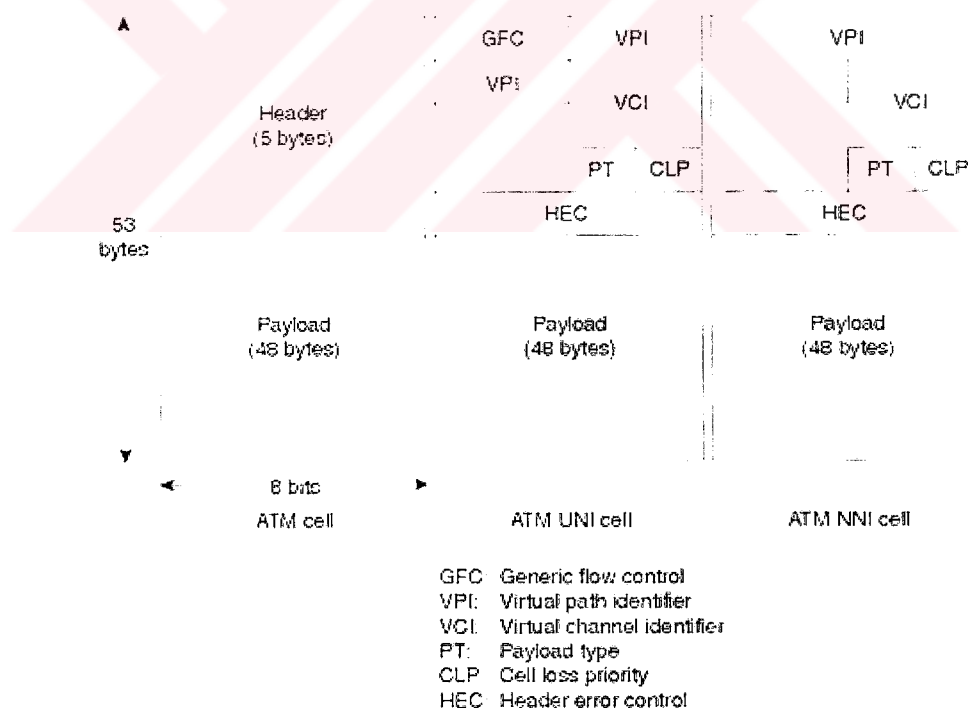


Figure 8.1 ATM Cell UNI and NNI Header Formats

Generic flow control (GFC) has local flow-control significance to the user for flow control. The GFC mechanism is used to control traffic flow from end stations to the network. The GFC field is not present in the NNI cell format. Two modes of operation are defined: uncontrolled access and controlled access. Traffic enters the network without GFC-based flow control in uncontrolled access mode. In controlled access mode, ATM-attached end nodes shape their transmission in accordance with the value present in GFC. Most UNI implementations don't use this field.

A virtual path (VP) consists of a bundle of VCs and is assigned to a virtual path identifier (VPI). ATM switches can switch VPIs, along with all the VCs within them. VCs are the paths over which user data is sent. Each VC within a VP is assigned a virtual channel identifier (VCI). VPI and VCI fields are used in ATM switches to make switching decisions.

Figure 8.2 shows a VC being set up between routers R1 and R2 through a network of ATM switches. All the cells leaving R1 for R2 are tagged with a VPI of 0 and a VCI of 64. The ATM switch S1 looks at the VPI and VCI pair on Port 0 and looks them up in its translation table. Based on the lookup, the ATM switch switches the cells out of Port 1 with a VPI of 1 and a VCI of 100. Similarly, Switch S4 switches cells on Port 4 with a VPI of 1 and a VCI of 100 onto Port 3 with a VPI of 2 and a VCI of 100 based on its translation table.

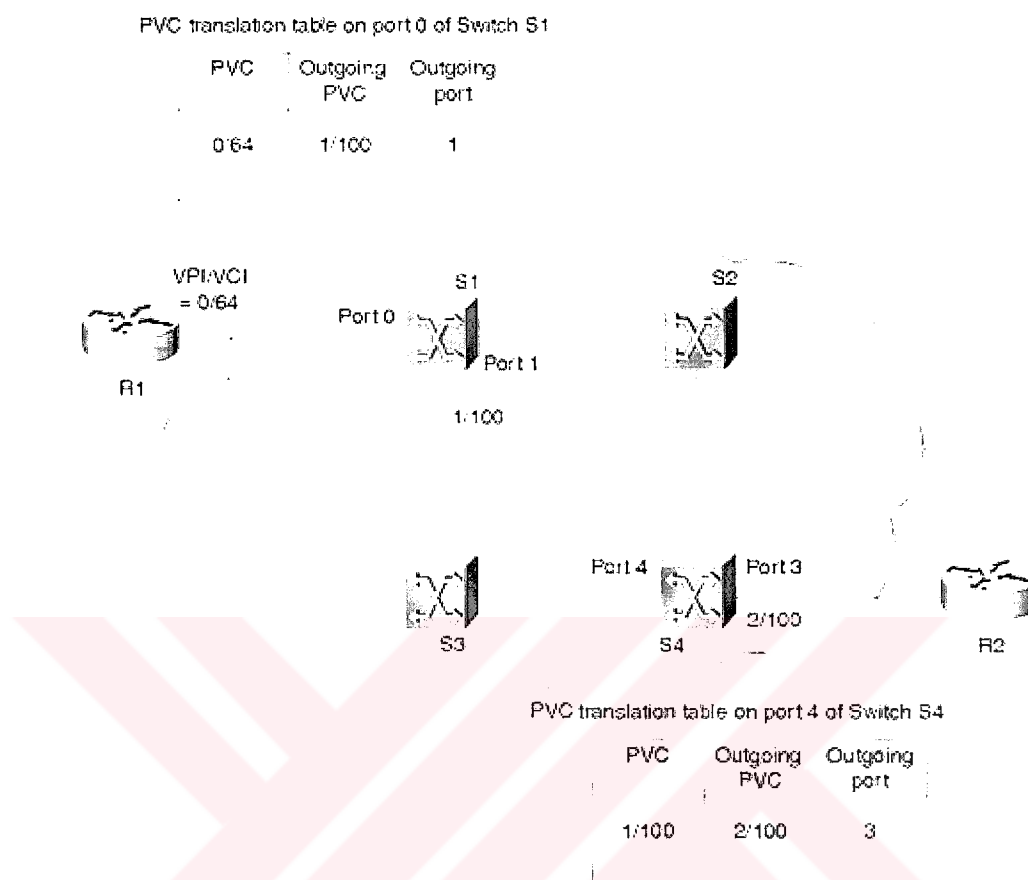


Figure 8. 2 Connectivity Between Routers R1 and R2 Across an ATM Network

The remaining fields in the ATM header are as follows:

Payload type identifier (PTI)— This 3-bit field is used to identify the kind of payload carried in the cell. It is used to differentiate between operation, administration, and maintenance (OAM) information and user data.

Cell loss priority (CLP)— CLP defines a cell's priority. If CLP is not set (CLP = 0), it is considered at a higher priority than cells with CLP set (CLP = 1). With CLP set, the cell has a higher chance of being discarded at times of network congestion.

Header error control (HEC)— HEC is used for detecting and correcting the errors in the ATM header.

8.1.2 ATM QoS

ATM offers QoS guarantees by making the ATM end system explicitly specify a traffic contract describing its intended traffic flow characteristics. The flow descriptor carries QoS parameters, such as Peak Cell Rate (PCR), Sustained Cell Rate (SCR), and burst size.

ATM end systems are responsible for making sure the transmitted traffic meets the QoS contract. The ATM end system shapes traffic by buffering data and transmitting it within the contracted QoS parameters. The ATM switches police each user's traffic characteristics and compare them to their QoS contract. If certain traffic is found to exceed the QoS contract, a switch can set the CLP bit on the nonconforming traffic. A cell with the CLP bit set has a higher drop probability at times of congestion.

8.1.3 ATM Service Classes

The AAL layer is what gives ATM the flexibility to carry entirely different traffic services within the same format. ATM defines five different AAL types based on the supported service type. AAL1 and AAL2 are designed for Constant Bit Rate (CBR) and Variable Bit Rate (VBR) services, respectively. AAL3 and AAL4, meant for connection-oriented and connectionless data traffic, respectively, have been absorbed by a more generalized AAL5 service for packetized data. (Romanow et al., 1995)

ATM services exist based on timing between source and destination, type of bit rate, and connection mode. The ATM Forum defines five service categories:

- **CBR**— CBR is meant for real-time data with tight constraints on delay and jitter and consistent bandwidth availability. CBR is synonymous with Circuit Emulation. This type of traffic requires reserved bandwidth that is available on

demand. Examples include standard digitized voice at 64 Kbps and video (H.320 standard).

- **Real-Time Variable Bit Rate (RT-VBR)**— As the name implies, this service is meant for real-time data similar to CBR, but under this service, the sources are expected to be bursty. Packetized video and data traffic with real-time requirements, such as interactive multimedia traffic, fall into this category. RT-VBR is specified by PCR, SCR, and maximum burst size (MBS).

- **Non-Real Time Variable Bit Rate (nRT-VBR)**— nRT-VBR is intended for applications that have burst traffic characteristics and do not have tight constraints on delay and jitter. Interactive data transactions fall into this category. nRT-VBR is specified by PCR, SCR, and MBS.

- **Available Bit Rate (ABR)**— ABR sources regulate their sending rate according to feedback provided by the network. An ABR service doesn't make bandwidth reservations as such, but it adapts the traffic allowed into the network based on the feedback mechanism implemented by using the Resource Management (RM) cells. It uses a rate-based approach where the sending rate on each VC is adaptive to the explicit rate indications in the network as conveyed by the RM cells. Data cell transmission is preceded by sending ABR RM cells. The source rate is controlled by the return of these RM cells, which are looped back by the destination or by a virtual destination. RM cells introduce a new source of overhead. This service works well with Transmission Control Protocol (TCP) sources, which implement adaptive flow control.

ABR service specifies a PCR and a guaranteed Minimum Cell Rate (MCR) per VC.

- **Unspecified Bit Rate (UBR)**— UBR is the ATM equivalent of best-effort service in IP. There is no bandwidth reservation, nor are there delay and jitter bounds. No congestion control is performed at the ATM layer. An example of such traffic is massive file transfers, such as system backups. TCP over UBR is not expected to perform well, because cell loss can occur on a UBR circuit, but

you can improve the performance by using the cell discard strategies described in the next section.

8.1.4 Cell Discard Strategies

As discussed earlier in this chapter, the ATM header has a CLP bit to indicate CLP. Cells with $CLP = 1$ are dropped before a cell with $CLP = 0$. The ATM traffic policing mechanism can tag the CLP bit as 1 in a cell when a cell exceeds the VC's traffic contract specifications. When congestion occurs in some part of the ATM network, cells with a CLP bit of 0 or 1 might be dropped, though cells whose CLP bits are tagged will be dropped first. Partial packet discard (PPD) and early packet discard (EPD) are ATM cell discard strategies that increase the effective throughput in an ATM network.

The ATM SAR function takes care of segmenting larger packets into cells. When a dropped ATM cell is part of a larger packet, it is not necessary to send the remaining cells belonging to the segmented packet, because this would cause unnecessary traffic on an already congested link. In this case, reassembling the cells into the original packet at the destination is not possible, and the entire packet needs to be retransmitted anyway. Therefore, when a cell of a larger packet is discarded, PPD starts discarding the packet's remaining cells. The improvement is limited, however, because the switch begins to drop cells only when the buffer overflows.

Implementing PPD is straightforward with AAL5. PPD can be signaled on a per-VC basis. With PPD, after the switch drops cells from the VC, the switch continues dropping cells from the same VC until the switch sees the parameter set in the ATM cell header indicating the end of the AAL packet. The end-of-packet cell itself is not dropped. Because AAL5 does not support the simultaneous multiplexing of packets on a single VC, you can use this cell to delimit packet boundaries.

Because a cell discard of a larger packet (and, hence, PPD) can occur after some of the earlier cells of the larger packet find space in the output queue, PPD does only a

partial packet discard. Thus, the congested link can still transmit a significant fraction of cells belonging to corrupted packets. Note, however, that if the discarded cell happens to be the packet's first cell, PPD is effectively doing a full packet discard. On the other hand, EPD occurs before any cell is admitted into the output queue. When a new packet arrives, EPD checks the output buffer usage. If the buffer usage is below the configured threshold value, the ATM switch knows the buffer space is not about to be exhausted, and all the packet's cells can be queued. Otherwise, the switch assumes the buffer is close to exhaustion and the entire packet might not be queued; hence, the switch throws the entire packet away. Therefore, EPD either queues all the cells of a packet or drops the entire packet altogether. Because EPD does a full packet discard before any cell belonging to the packet is queued, this process is called early packet discard.

8.1.5 VP Shaping

Similar to the ATM services for an ATM VC, an ATM service—or shaping—can be applied on a VP for traffic contract restrictions on an entire VP. Within a shaped VP, all VCs still can be UBR without strict traffic constrictions carrying best-effort traffic. Figure 8.3 shows a logical diagram of how VCs are bundled within a VP.

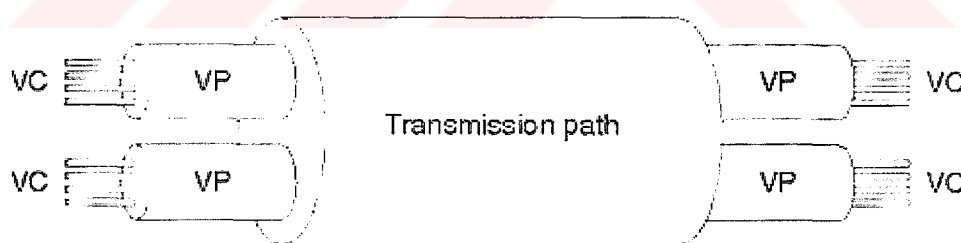


Figure 8.3 Bundling of Multiple VCs in a VP

8.2 ATM Interworking with IP QoS

In an IP network where a certain part of the network uses ATM as the underlying Layer 2 technology (as in Figure 8.4), the IP traffic is carried across the ATM backbone using a VC provisioned with certain ATM QoS, which suits the IP traffic it carries. Due to congestion at the ingress to the ATM network, however, queues can build up and certain packets might be tail-dropped. Without IP QoS, such packet drops are random and happen without any knowledge of the packet's IP precedence value, which fails to deliver end-to-end IP QoS across an ATM network. Hence, a need exists for IP QoS in an ATM network. You can apply the Weighted Random Early Detection (WRED) and Weighted Fair Queuing (WFQ) IP QoS techniques on the queues at the ingress to an ATM network, so the packet drops and scheduling on the output queue are based on the packets' IP QoS. Chapter 4, "Per-Hop Behavior: Resource Allocation I," and Chapter 6, "Per-Hop Behavior: Congestion Avoidance and Packet Drop Policy," discuss WFQ and WRED functions, respectively.

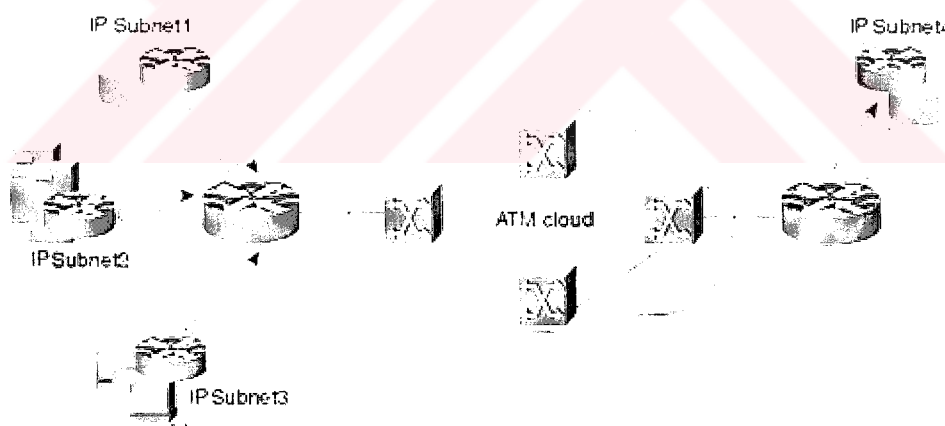


Figure 8. 4 IP over ATM

Note that from the IP QoS perspective, the IP traffic is transported across the ATM network without any loss (as any drops in the ATM network are IP QoS-unaware) by using an ATM service that suits the traffic's IP service needs. For IP QoS, each VC in an

ATM network maintains a separate queue. You can apply the WRED and WFQ IP QoS functions on each VC queue.

Two scenarios are discussed in this section as a way to preserve IP QoS over an ATM network:

A single PVC carrying all the IP traffic to its destination— IP traffic exceeding the ATM PVC parameters and service at the ingress to the ATM network gets queued, and IP QoS techniques such as WRED and WFQ are applied on the queue as it builds up due to congestion conditions.

WRED ensures that high-precedence traffic has low loss relative to lower-precedence traffic. WFQ ensures that high-precedence traffic gets a higher bandwidth relative to the lower-precedence traffic, because it schedules high-precedence traffic more often. Note that when Class-Based Weighted Fair Queuing (CBWFQ) is run on a PVC, you can make bandwidth allocations based on traffic class. CBWFQ is discussed in Chapter 4.

A VC bundle (made of multiple PVCs) carrying IP traffic to its destination— When carrying traffic with different QoS (real-time, non-real-time, best-effort) to the same destination, it is a good idea to provision multiple PVCs across the ATM network to the destination so that each IP traffic class is carried by a separate PVC. Each PVC is provisioned to an ATM service class based on the IP traffic it is mapped to carry. Figure 8.5 depicts IP-ATM QoS using a VC bundle. Some VC bundle characteristics are as follows:

- Each VC in the bundle is mapped to carry traffic with certain IP precedence value(s). You can map a VC to one or more IP precedence values. Note, however, that only one routing peering or adjacency is made per PVC bundle.

- You can monitor VC integrity by using ATM OAM or Interim Local Management Interface (ILMI). If a bundle's high-precedence VC fails, you can either "bump" its traffic to a lower-precedence VC in the bundle, or the entire bundle can be declared down.
- A separate queue exists for each VC in the bundle. You can apply the WRED and WFQ IP QoS techniques on each VC queue.

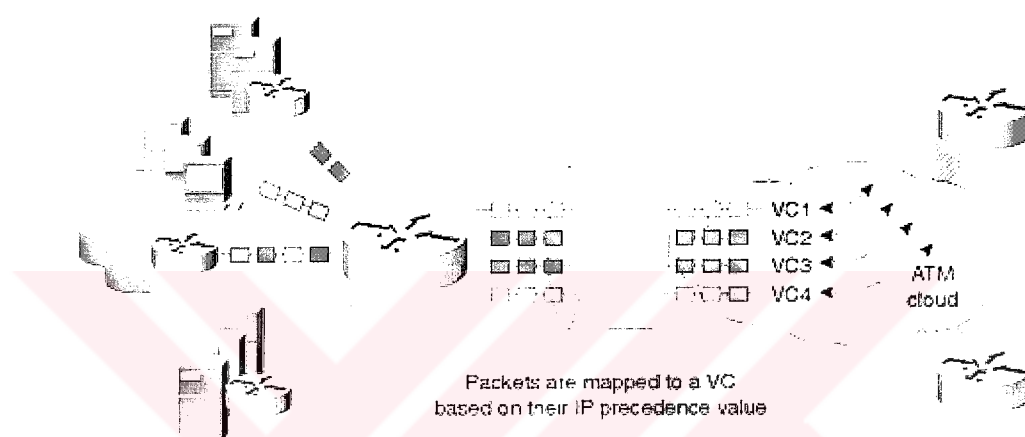


Figure 8.5 ATM VC Bundle: IP Precedence to VC Mapping

ATM service for the IP traffic is expected to be above UBR (best-effort) class so that no packets (cells) are dropped as part of ATM QoS within the ATM network. Figure 8.6 illustrates the two IP-ATM QoS scenarios.

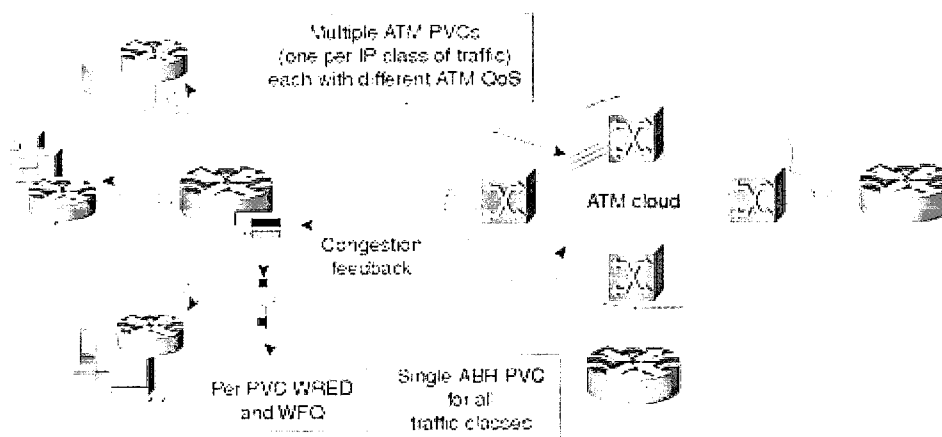


Figure 8. 6 IP-ATM QoS Interworking

The difference between using a single PVC or a PVC bundle for IP QoS interworking depends on the cost and traffic needs in the network. Although a PVC bundle can be more expensive than a single PVC, a PVC bundle provides traffic isolation for critical traffic classes such as voice. On the other hand, a PVC bundle requires prior traffic engineering so that all the PVCs in the bundle are utilized optimally. Otherwise, you can run into conditions in which the PVC in the bundle carrying the high-precedence traffic gets congested while the PVC carrying the lower-precedence traffic is running relatively uncongested. Note that you cannot automatically bump high-precedence traffic to a different member PVC when the PVC carrying high precedence gets congested. When using a single PVC, you can enable CBWFQ with a priority queue on it so that voice traffic is prioritized over the rest of the traffic carried by the PVC. CBWFQ with a priority queue is discussed in Chapter 4.

8.3 Frame Relay

Frame Relay is a popular wide-area network (WAN) packet technology well suited for data traffic. It is a simple protocol that avoids link-layer flow control and error correction functions within the Frame Relay network. These functions are left for the

applications in the end stations. The protocol is best suited for data traffic, because it can carry occasional bursts.

Frame Relay operates using VCs. A VC offers a logical connection between two end points in a Frame Relay network. A network can use a Frame Relay VC as a replacement for a private leased line. You can use PVCs and SVCs in Frame Relay networks. A PVC is set up by a network operator through a network management station, whereas an SVC is set up dynamically on a call-by-call basis. PVCs are most commonly used, and SVC support is relatively new.

A user frame is placed in a Frame Relay header to be sent on a Frame Relay network. The Frame Relay header is shown in Figure 8.7.

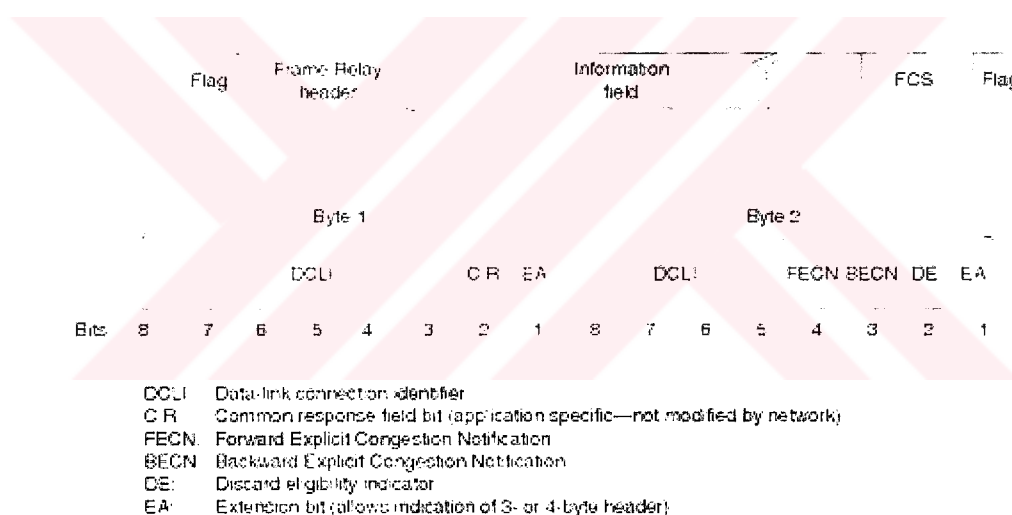


Figure 8. 7 An Example of a Frame Relay Header

The 10-bit data-link connection identifier (DLCI) is the Frame Relay VC number corresponding to a logical connection to the Frame Relay network. A DLCI has only local significance. A Frame Relay switch maintains the VC mapping tables to switch a frame to its destination DLCI.

The Address Extension (AE) bit indicates a 3- or 4-byte header. It is not supported under the present Cisco Frame Relay implementation. The Command and Response (C/R) bit is not used by the Frame Relay protocol and is transmitted unchanged end to end.

The Frame Check Sequence (FCS) is used to verify the frame's integrity by the switches in the Frame Relay network and the destination station. A frame that fails the FCS test is dropped. A Frame Relay network doesn't attempt to perform any error correction. It is up to the higher-layer protocol at the end stations to retransmit the frame after discovering the frame might have been lost.

8.3.1 Frame Relay Congestion Control

Three bits in the Frame Relay header provide congestion control mechanisms in a Frame Relay network. These 3 bits are referred to as Forward Explicit Congestion Notification (FECN), Backward Explicit Congestion Notification (BECN), and Discard Eligible (DE) bits.

You can set the FECN bit to a value of 1 by a switch to indicate to a destination data terminal equipment (DTE) device, such as a router, that congestion was experienced in the direction of the frame transmission from source to destination.

The BECN bit is set to a value of 1 by a switch to indicate to the destination router that congestion was experienced in the network in the direction opposite the frame transmission from source to destination.

The primary benefit of the FECN and BECN congestion notification bits is the capability of higher-layer protocols to react intelligently to these congestion indicators. The use of FECN and BECN is shown in Figure 8.8.

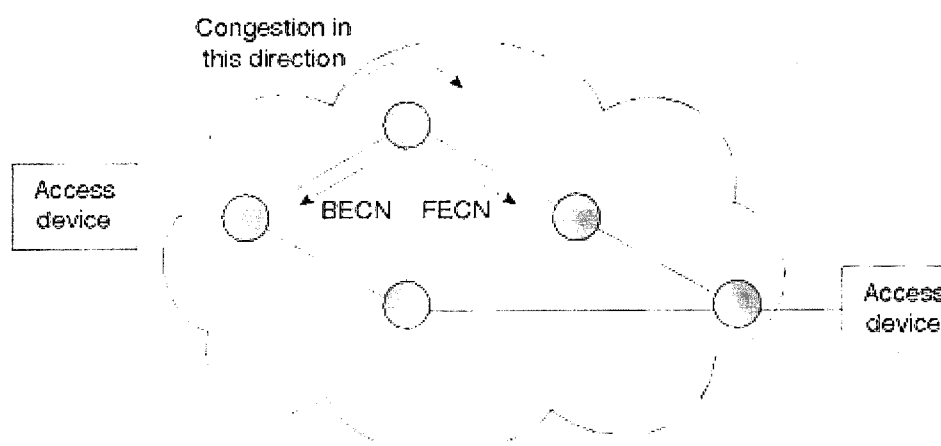


Figure 8. 8 Use of FECN and BECN Bits

The DE bit is set by the router or other DTE device to indicate that the marked frame is of lesser importance relative to other frames being transmitted. It provides a basic prioritization mechanism in Frame Relay networks. Frames with the DE bit set are discarded first, before the frames without the DE bit flagged.

8.3.2 Frame Relay Traffic Shaping (FRTS)

FRTS shapes traffic going out on a Frame Relay VC in accordance with the rate configured. It tries to smooth the bursty traffic by buffering packets exceeding the average rate. The buffered packets are de-queued for transmission when enough resources are available according to the queuing mechanism configured. A queuing algorithm is configurable on a per-VC basis. It can be configured only for outbound traffic on an interface.

FRTS can do traffic shaping to a peak rate configured to be either the Committed Information Rate (CIR) or some other defined value, such as the Excess Information Rate (EIR), on a per-VC basis.

FRTS in adaptive mode also permits output on Frame Relay VCs to be throttled based on received network BECN congestion indicators. It shapes traffic going out on a

PVC in accordance with the bandwidth available in a Frame Relay network. It is able to run at rate X, and when the network sees BECNs, it drops down to rate Y.

8.3.3 VC Traffic Shaping

A token bucket, similar to the one used for traffic shaping in Chapter 3, "Network Boundary Traffic Conditioners: Packet Classifier, Marker, and Traffic Rate Management," is used as a traffic descriptor to measure conforming traffic. The contracted average rate is called the Committed Information Rate (CIR). Burst size (B_C) is the amount of data added to the token bucket of size ($B_C + B_E$) at each measuring time interval, T_C . T_C is defined as $B_C \div \text{CIR}$. Excess burst size (B_E) is the amount of excess burst of data allowed to be sent during the first interval when the token bucket is full. When a new packet arrives, it is queued into the output queue and scheduled for transmission based on the queue scheduler used, such as WFQ or first-in, first-out (FIFO). A packet scheduled for transmission by the scheduler is transmitted only if enough tokens are available in the bucket equivalent to the scheduled packet. After a conforming packet is transmitted, an equivalent amount of tokens are removed from the bucket.

If not all B_C bytes are sent in a T_C interval, you can transmit the unused bytes in the subsequent interval along with the new credit for B_C bytes. Hence, in a T_C interval during which there is less than B_C traffic, the credit can increase to an upper bound of $B_C + B_E$ for the next subsequent interval.

If serious load sets in and the token bucket is full, you can send $B_E + B_C$ bytes in the first interval and B_C bytes in each subsequent interval until congestion conditions ease. As you can see, you can throttle to the CIR equation at times of congestion.

This relationship between the traffic shaping parameters is shown in Figure 8.9.

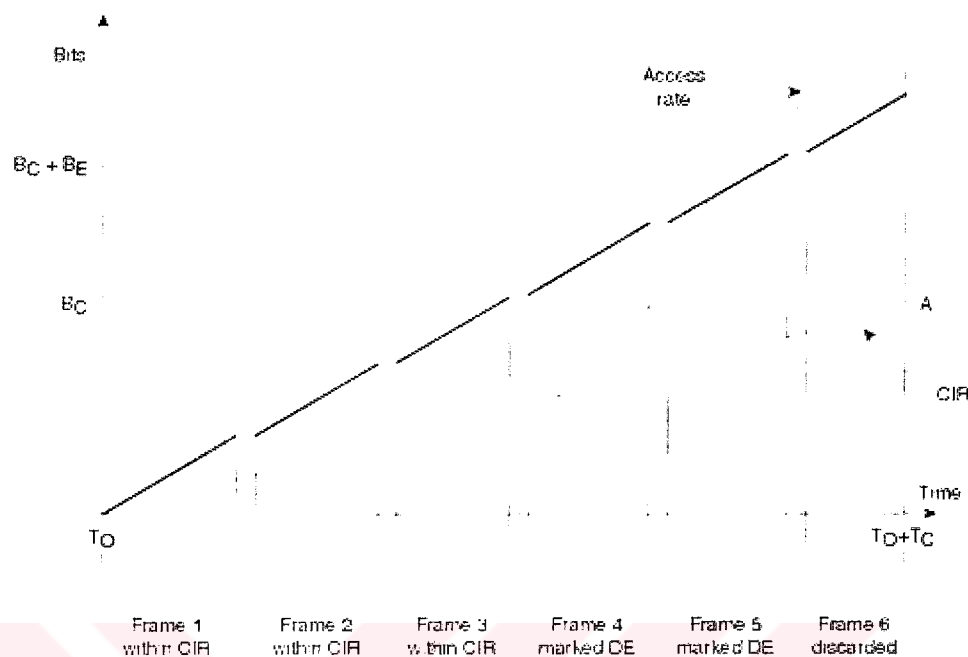


Figure 8.9 Relationship Between Traffic Shaping Parameters

Frame Relay traffic shaping allows occasional bursts over the CIR on a PVC, although the rate throttles to the CIR at times of congestion. A PVC can also be configured for a fixed data rate equal to the CIR ($B_E = 0$).

8.3.4 Adaptive FRTS

At every time interval T_C , a process checks if any BECN was received from the Frame Relay network. If BECN was received in the last T_C interval, the transmission rate is dropped by 25 percent of the current rate. It might continue to drop until the lower limit is reached, which is the minimum CIR (MINCIR). No matter how congested the Frame Relay network is, the router does not drop its transmission rate below MINCIR. The default value for MINCIR is half of the CIR.

After the traffic rate has adapted to the congestion in the Frame Relay network, it takes $16 T_C$ intervals without BECN to start increasing the traffic rate back to the configured CIR.

8.3.5 FECN/BECN Integration

A UDP- or IP-based application can result in a unidirectional traffic flow, but the application might not necessarily have data flowing in the opposite direction because both IP and UDP do not use any acknowledging scheme. If you enable traffic shaping on UDP data, the only congestion notification set by the network is the FECN bit received in the frames arriving at the destination router. The router sourcing all this traffic does not see any BECNs because there's no return traffic. FECN/BECN integration implemented a command to send a Q.922 test frame with the BECN bit set in response to a frame that has the FECN set. Note that you need to clear the DE bit in a BECN frame you send in response to a received FECN. The source router gets a test frame, which is discarded, but the BECN bit is used to throttle the data flow. To have this interaction, traffic shaping commands need to be present on the ingress and egress ports.

8.3.6 Frame Relay Fragmentation

At the output queue of a PVC on a router, large packets that were queued ahead of small, delay-sensitive packets contribute to increased delay and jitter for the small packets. This behavior is due to the larger transmission delay for a large packet compared to a small packet. Transmission delay is discussed in Chapter 1, "Introducing IP Quality of Service." Fragmenting the large data frames into smaller frames, interleaving small, delay-sensitive packets between the fragments of large packets before putting them on the queue for transmission, and reassembling the frame at the destination eases the problem for the small packets. The Frame Relay Forum (FRF) has ratified a new standard for Frame Relay fragmentation: FRF.12.

FRF.12 ensures that voice and similar small packets are not unacceptably delayed behind large data packets. This standard also attempts to ensure that the small packets

are sent in a more regular fashion, thereby reducing jitter. This capability is important in enabling a user to combine voice and other delay-sensitive applications, such as Enterprise Resource Planning (ERP)-based, mission-critical applications with non-time-sensitive applications or other data on a single PVC.

The same standard is applicable at the UNI interface (between the router or a Frame Relay Access Device [FRAD] and the Frame Relay cloud), at the NNI interface (between the switches within the Frame Relay cloud), and between the routers (or FRADs) for end-to-end transmissions, where fragmentation is transparent to the Frame Relay cloud.

Because fragmentation is most useful on slow lines, and the slower links tend to be access links to an FRAD or a router, UNI fragmentation becomes the widely used application. UNI fragmentation is local to the router, and you can optimize its network interface and fragment size based on the DTE device's speed.

NNI and end-to-end fragmentation are not commonly used applications. Fragmentation might not be a good idea across high-speed trunk lines in the Frame Relay cloud because it can carry relatively large frames when compared to the slow speed access links without delaying the voice packets drastically. A speed mismatch between the two end FRAD systems can also preclude running end-to-end fragmentation between the DTE devices. UNI fragmentation can also be used to run fragmentation end to end. For an end DTE that doesn't implement end-to-end fragmentation, UNI fragmentation enables the network to proxy the DTE.

An important driving application for FRF.12 is to enable voice over Frame Relay technology for service approaching toll voice quality. Fragmentation, especially when used with QoS functionality such as WFQ and WRED, is used to ensure a consistent flow of voice information.

Prior to FRF.12, in an IP environment the only way to fragment frames was to set the IP maximum transmission unit (MTU) to a small value. This action introduces undesirable inefficiencies in the system in terms of increased processor overhead and packet overhead, however. In addition, FRF.12 operates below Layer 3 of the Open System Interconnection (OSI) model, and hence it fragments not only IP, but other frames as well. With Voice over IP (VoIP), fragmentation becomes an absolute necessity in low-bandwidth environments.

In addition to FRF.12 fragmentation, Cisco supports FRF.11 Annex C fragmentation and a Cisco proprietary fragmentation. Frame Relay fragmentation is part of Cisco's traffic-shaping functionality. Figure 8.10 illustrates Frame Relay fragmentation when carrying voice and data traffic on a PVC.

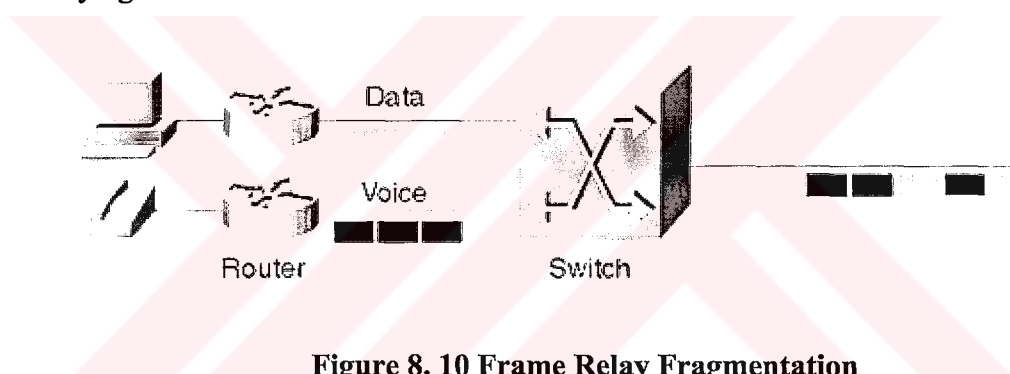


Figure 8.10 Frame Relay Fragmentation

Enabling FRTS is a prerequisite to turning on FRF.12 fragmentation. For voice, FRTS uses a flow-based Weighted Fair Queuing (WFQ) with a priority queue (PQ) on the shaping queue. Each Frame Relay PVC has its own PQ-WFQ structure that is used to schedule packets based on their IP precedence values as per the flow-based WFQ algorithm discussed in Chapter 4. FRTS uses dual FIFO queues at the interface level—the first queue for priority packets such as voice, Local Management Interface (LMI), and related high-priority packets and the second queue for the rest of the traffic. Note that FRTS disallows any queuing other than dual FIFO at the interface level.

In Figure 8.11, three different flows—two data flows and one voice flow—arrive for transmission on an interface. The flows are routed to their respective PVC structures

based on the flow header information. In this case, flows 1 and 2 belong to PVC 1, and flow 3 belongs to PVC 2. Each PVC has its own shaping queue. In this case, PQ-WFQ is enabled on PVC 1 such that all voice flow packets are scheduled first, and all voice packets go to the priority interface FIFO queue. Data packets scheduled by the shaping queue are fragmented based on the fragment threshold value and are put in the normal FIFO interface queue. It is assumed that the fragment threshold is set such that none of the voice packets need to be fragmented. Though a separate shaping queue exists for each PVC, all the PVCs on the interface share the dual FIFO interface queues. Packets are transmitted from the priority FIFO queue with a strict priority over the packets in the normal FIFO interface queue.

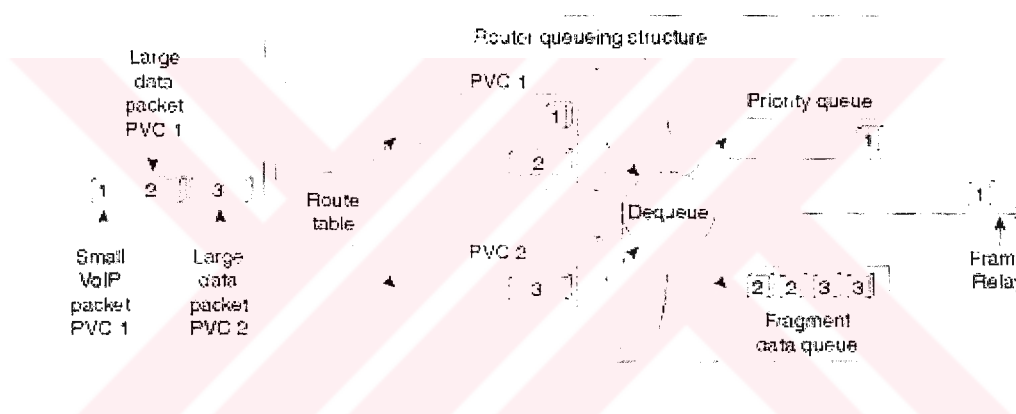


Figure 8.11 A Conceptual View of FRF.12 Operation with Multiple PVCs

8.4 Frame Relay Interworking with IP QoS

Similar to IP-ATM QoS, IP Frame Relay QoS is necessary to provide end-to-end IP QoS. At the ingress to a Frame Relay network, IP traffic queues are maintained for each Frame Relay VC. On each VC queue, you can apply the WFQ and WRED IP QoS techniques. In addition to WFQ, PQ-WFQ, priority, and custom queuing can also be configured as a scheduling mechanism for traffic queued on a VC.

As discussed previously, the Frame Relay uses the DE bit to provide a basic prioritization scheme for its traffic. A frame with a DE bit flagged has a higher drop probability than a frame with the DE bit at 0.

Because IP uses 3 bits to indicate its precedence, there can't be a one-to-one map between the Frame Relay DE bit setting and IP precedence levels. A fairly basic mapping can be worthwhile, however. For example, you can map IP best-effort traffic indicated by an IP precedence of 0 over Frame Relay with a DE bit set, whereas you can send other, better-than-best-effort traffic without flagging the DE bit. This ensures that the best-effort traffic is dropped before the other, more important traffic.

8.5 The IEEE 802.3 Family of LANs

Ethernet is the most common LAN technology used today. The original Ethernet operates at 10 Mbps using carrier sense media access, collision detect (CSMA/CD) for media access. Today, the term "Ethernet" is used to refer to all extensions of the original Ethernet specification that continue to use CSMA/CD.

Ethernet operating at 10 Mbps is standardized as part of the IEEE 802.3 specification. Ethernet Version 2.0, which is compatible with IEEE 802.3, is also commonly used. The higher-speed version of the Ethernet family LANs—100 Mbps fast Ethernet (100BaseT) and Gigabit Ethernet—continue to use the existing IEEE 802.3 CSMA/CD specification or an extension of it. 100BaseT and Gigabit Ethernet are standardized as the IEEE 802.3u and 802.3x specifications, respectively. As a result, all the 802.3 family of Ethernets retain the IEEE 802.3 frame format, size, and error-detection mechanism. The IEEE 802.3 and Ethernet frame formats are shown in Figure 8.12.

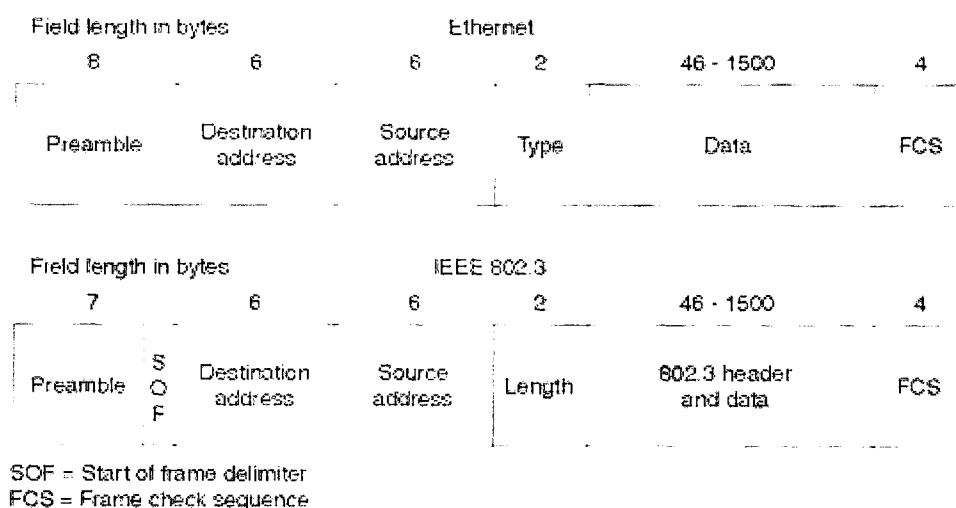


Figure 8. 12 Ethernet and IEEE 802.3 Frame Formats

8.5.1 Expedited Traffic Capability

Expedited traffic capability provides the ability for network prioritization on an Ethernet, virtual LAN (VLAN)-based or otherwise.

The expedited traffic capability for Ethernet is defined as part of the 802.1p standard. 802.1p uses 3 bits within the 4-byte tag defined by 802.1Q to support VLANs. The 4-byte 802.1Q and the 802.1p bits are shown in Figure 8.13.

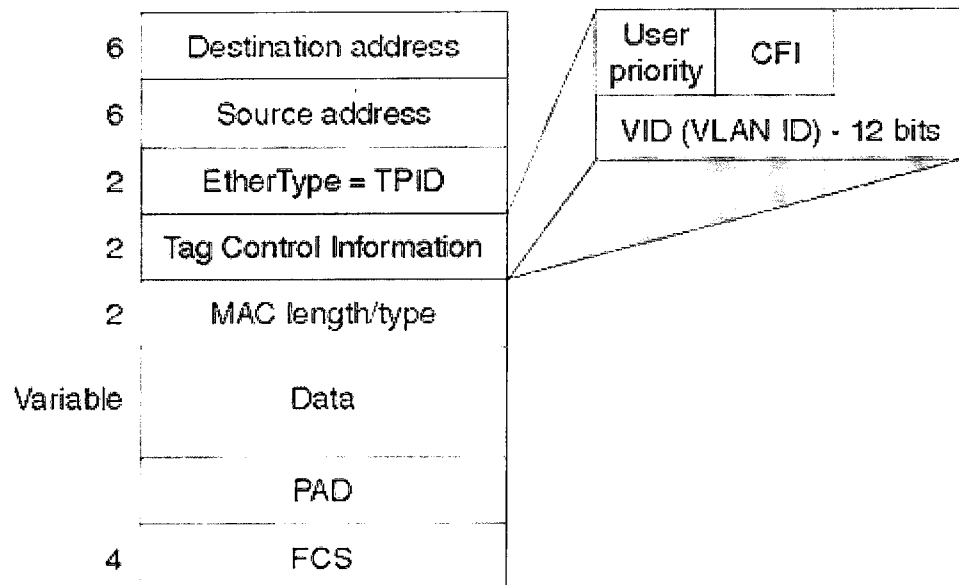


Figure 8. 13 802.1Q Frame Showing 802.1p Bits

802.1Q defines a new tagged frame type by adding a 4-byte tag, which is made up of the following:

- 2 bytes of Tagged Protocol Identifier (TPID)
 - 0x8100 is used to indicate an 802.1Q packet
- 2 bytes of Tagged Control Information (TCI)
 - 3-bit 802.1p bits
 - 1-bit canonical format identifier (CFI)
 - 12-bit VLAN Identifier (ID)

Figure 8.14 shows how the original Ethernet/802.3 frame is changed into a tagged 802.1Q frame. The FCS needs to be recalculated after introducing the 4-byte tag.

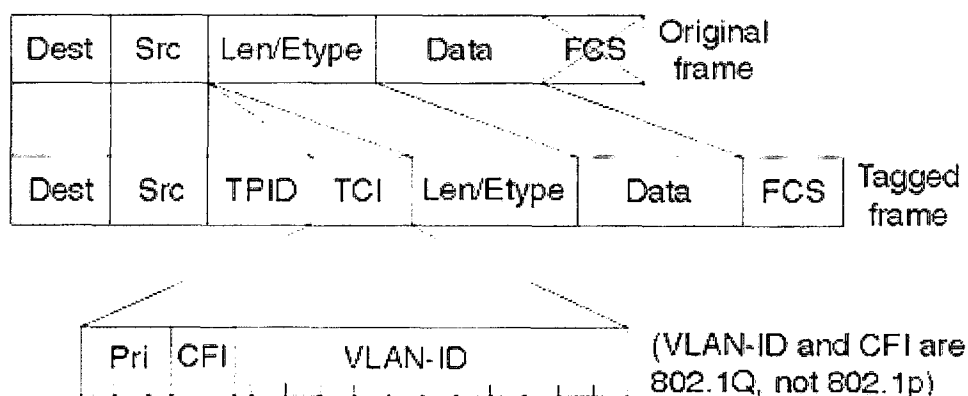


Figure 8. 14 An Ethernet Frame to a Tagged 802.1Q Frame

802.1p provides a way to maintain priority information across LANs. It offers eight priorities from the three 802.1p bits. To support 802.1p, the link layer has to support multiple queues—one for each priority or traffic class. The high-priority traffic is always preferred over lower-priority traffic. A switch preserves the priority values while switching a frame.

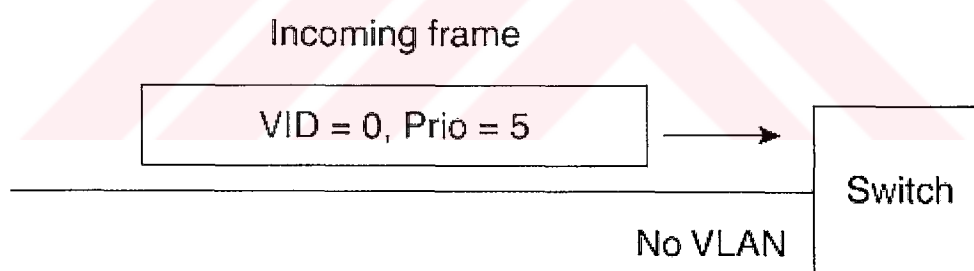


Figure 8. 15 Use of 802.1p in the Absence of VLANs

With the addition of the 4-byte tag introduced by the 802.1Q and 802.1p specifications, an Ethernet frame can now exceed the maximum frame size of 1518 bytes. Hence, IEEE 802.3ac is tasked with modifying the 802.3 standard to extend the maximum frame size from 1518 to 1522 bytes.

IEEE 802.1Q "Standard for Virtual Bridged Local Area Networks" defines a method of establishing VLANs.

CHAPTER NINE

QoS in MPLS-BASED NETWORKS

9 QoS in MPLS-Based Networks

Multiprotocol Label Switching (MPLS) is an Internet Engineering Task Force (IETF) standard for a new switching paradigm that enables packet switching at Layer 2 while using Layer 3 forwarding information. Thus, MPLS combines the high-performance capabilities of Layer 2 switching and the scalability of Layer 3-based forwarding.

At the ingress to the MPLS network, Internet Protocol (IP) precedence information can be copied as class of service (CoS) bits, or can be mapped to set the appropriate MPLS CoS value in the MPLS Layer 2 label. Within the MPLS network, MPLS CoS information is used to provide differentiated services. Hence, MPLS CoS enables end-to-end IP quality of service (QoS) across an MPLS network.

MPLS-based forwarding enables a service provider network to deploy new services, particularly Virtual Private Networks (VPNs) and traffic engineering.

This chapter introduces MPLS and MPLS-based VPNs and discusses their QoS offerings. Traffic engineering is covered in Chapter 10, "MPLS Traffic Engineering."

9.1 MPLS

MPLS is an IETF standard for label-swapping -based forwarding in the presence of routing information. It consists of two principal components: control and forwarding. The control component uses a label distribution protocol to maintain label-forwarding information for all destinations in the MPLS network. The forwarding component

switches packets by swapping labels using the label information carried in the packet and the label-forwarding information maintained by the control component.

MPLS, as the name suggests, works for different network layer protocols. As such, the forwarding component is independent of any network layer protocol. The control component has to support label distribution for different network layer protocols to enable MPLS use with multiple network layer protocols.

9.1.1 Forwarding Component

MPLS packet forwarding occurs by using a label-swapping technique. When a packet carrying a label arrives at a Label Switching Router (LSR), the LSR uses the label as the index in its Label Information Base (LIB). For an incoming label, LIB carries a matching entry with the corresponding outgoing label, interface, and link-level encapsulation information to forward the packet. Based on the information in the LIB, the LSR swaps the incoming label with the outgoing label and transmits the packet on the outgoing interface with the appropriate link-layer encapsulation.

A Label Edge Router (LER) is an edge router in the MPLS cloud. An LER adds an MPLS label to all packets entering the MPLS cloud from non-MPLS interfaces. On the same token, it removes the MPLS label from a packet leaving the MPLS cloud. The forwarding behavior in an MPLS network is depicted in Figure 9.1.

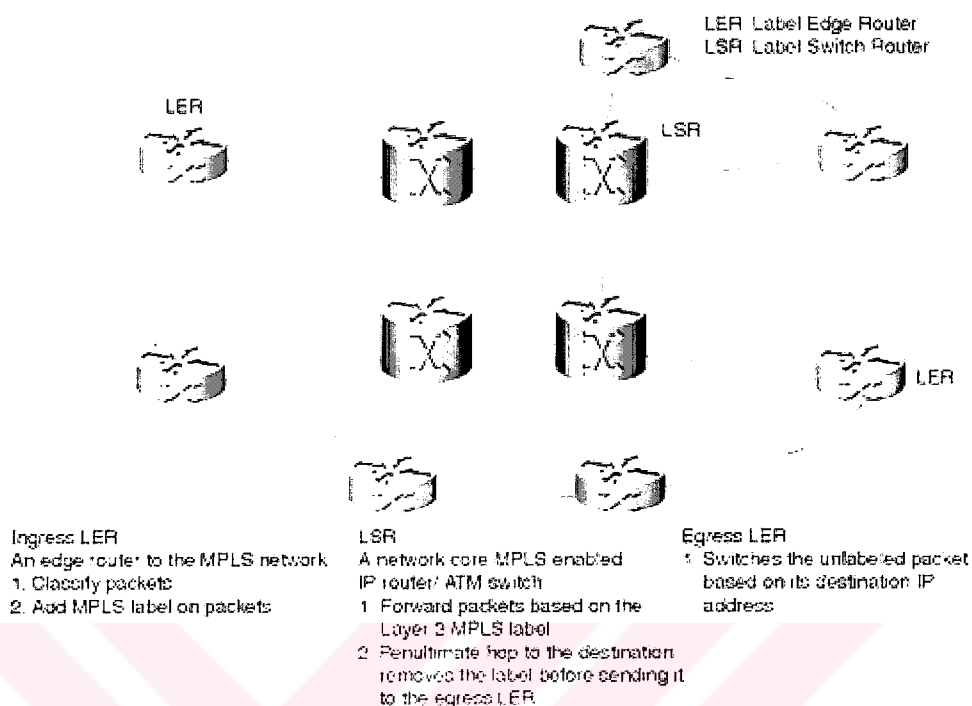


Figure 9.1 MPLS Network

The preceding procedure simplifies a normal IP router's forwarding behavior. A non-MPLS router performs destination-based routing based on the longest match from the entries in the routing-table-based forwarding table. An MPLS router, on the other hand, uses a short label, which comes before the Layer 3 header, to make a forwarding decision based on an exact match of the label in the LIB. As such, the forwarding procedure is simple enough to allow a potential hardware implementation.

9.1.2 Control Component

The control component is responsible for creating label bindings and then distributing the label binding information among LSRs.

Label binding is an association between a label and network layer's reachability information or a single traffic flow, based on the forwarding granularity. On one end, a label can be associated to a group of routes, thereby providing MPLS good-scaling

capabilities. On the other end, a label can be bound to a single application flow, accommodating flexible forwarding functionality. MPLS network operation is shown in Figure 9.2.

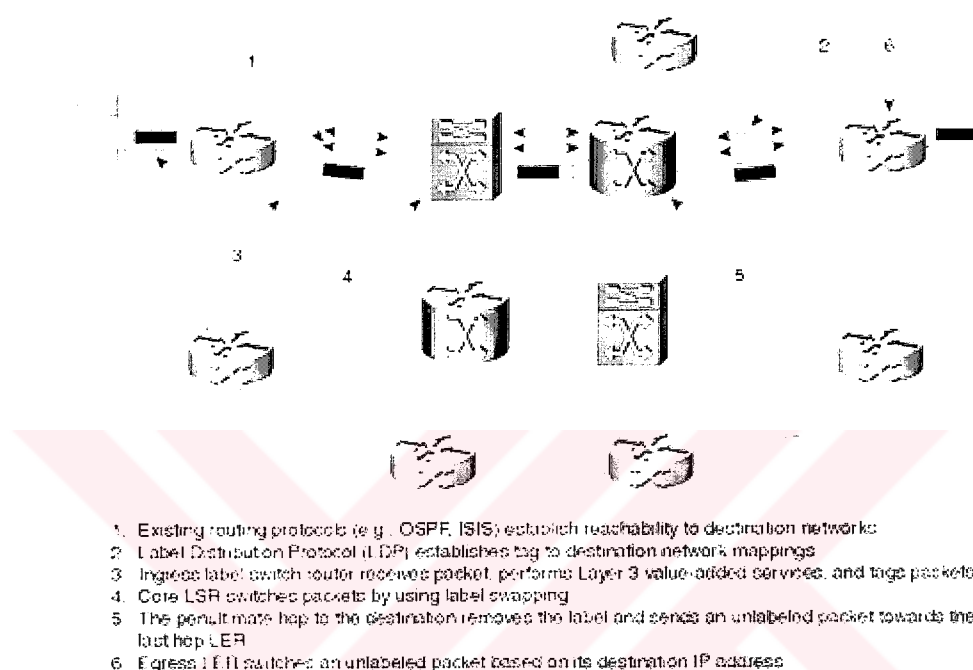


Figure 9.2 MPLS Network Operation

When label binding is based on routing information, MPLS performs destination-based forwarding. Destination-based forwarding is not amenable to more granular and flexible routing policies, however. For cases involving flexible forwarding policies, the label binding might not be based on routing information. MPLS provides flexible forwarding policies at a granularity of a flow or group of flows. You can use this aspect of MPLS to offer a new service called traffic engineering. Traffic engineering is discussed in Chapter 10.

The next section discusses label-binding procedures for achieving destination-based forwarding.

9.1.3 Label Binding for Destination-Based Forwarding

Cisco Express Forwarding (CEF) is the recommended packet switching mechanism for IP networks today. CEF is discussed in Appendix B, "Packet Switching Mechanisms." A CEF table carries forwarding information based on the routing table; as such, it forwards packets on the basis of the destination. MPLS extends the CEF table to accommodate label allocation for each entry. LIB binds each CEF table entry with a label.

MPLS allows three methods for label allocation and distribution:

- Downstream label allocation
- Downstream label allocation on demand
- Upstream label allocation

For all the different types of label allocations, a protocol called Label Distribution Protocol (LDP) is used to distribute labels between routers. Note that the terms "downstream" and "upstream" are used with respect to the direction of the data flow.

9.1.4 Downstream Label Allocation

Downstream label allocation occurs in the direction opposite the actual data flow's direction. The label carried in a packet is generated and bound to a prefix by an LSR at the link's downstream end. As such, each LSR originates labels for its directly connected prefixes, binds them as an incoming label for the prefixes, and distributes the label association to its prefixes to all the upstream routers. An upstream router puts the received label binding as an outgoing label for the prefix in the CEF table and, in turn, creates an incoming label to it and advertises it to a router further upstream.

In independent label distribution mode, each downstream router binds an incoming label for a prefix independently and advertises it as an outgoing label to all its upstream routers. It is not necessary to receive an outgoing label for a prefix before an incoming

label is created and advertised. When a router has both the incoming and outgoing labels for a prefix, it can start switching packets by label swapping.

The other label distribution mode is termed ordered control mode. In this mode, a router waits for the label from its downstream neighbor before sending its label upstream.

9.1.5 Downstream Label Allocation on Demand

This label allocation process is similar to downstream allocation, but it is created on demand by an upstream router. The upstream router identifies the next hop for each prefix from the CEF table and issues a request to the next hop for a label binding for that route. The rest of the allocation process is similar to downstream label allocation.

9.1.6 Upstream Label Allocation

Upstream label allocation occurs in the direction of the actual data flow. The label carried in the data packet's header is generated and bound to the prefix by the LSR at the upstream end of the link. For each CEF entry in an LSR, an outgoing label is allocated and distributed as an incoming label to downstream routers. In this case, incoming labels are allocated to prefixes.

When an LSR has both the incoming and the outgoing labels for a prefix, it can start switching packets carrying a label by using label swapping.

When an LSR creates a binding between an outgoing labels and a route, the switch, in addition to populating its LIB, also updates its CEF with the binding information. This enables the LSR to add labels to previously unlabeled packets it is originating. Table 9.1 compares downstream and upstream label distribution methods.

Table 9.1 Comparison Between Downstream and Upstream Label Distribution

	Downstream Allocation	Upstream Allocation
Direction of Label Allocation	Occurs in the direction opposite the data flow.	Occurs in the direction of the data flow.
Label Allocation and Distribution	Allocates the incoming prefix for all entries in the CEF table and distributes the outgoing label to the upstream routers.	Allocates the outgoing label for all entries in the CEF table and distributes the incoming label to the downstream routers.
Label Distribution Protocol	Allocates outgoing labels.	Distributes incoming labels.
Applicability	Applicable for non-ATM-based IP networks.	Downstream label allocation on demand and upstream label allocation are most useful in Asynchronous Transfer Mode (ATM) networks.

9.1.7 Label Encapsulation

A packet can carry label information in a variety of ways:

- **As a 4-byte label inserted between the Layer 2 and network layer headers**— This applies to Point-to-Point Protocol (PPP) links and Ethernet (all flavors) LANs. A single MPLS label or a label stack (multiple labels) can be carried in this way. Figure 9.3 shows how the label is carried over PPP links and over an Ethernet-type LAN.

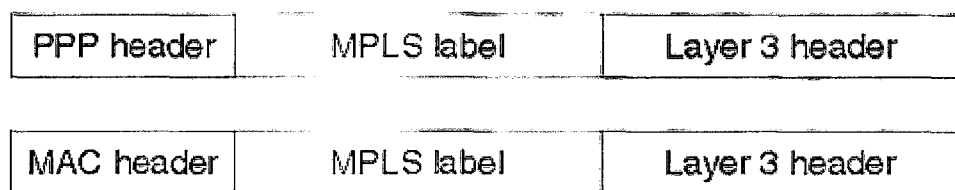


Figure 9.3 MPLS Label in Ethernet and PPP Frame

- **As a part of the Layer 2 header**— This applies to ATM, where the label information is carried in the VPI/VCI fields, as shown in Figure 9.3.

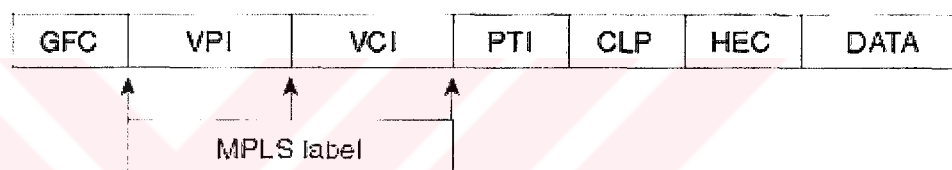


Figure 9.4 MPLS Label Carried in the VPI/VCI Fields in an ATM Header

- **As part of the ATM Adaptation Layer 5 (AAL5) frame before segmentation and reassembly (SAR)**— This occurs in an ATM environment for label information made up of a label stack (multiple MPLS label fields).

An MPLS label field consists of a label header and a 20-bit label. The label header consists of three fields: CoS, S bit, and Time-to-Live (TTL). The 4-byte MPLS label field format is shown in Figure 9.5.

To support the MPLS control component, an ATM switch needs to run routing protocols such as OSPF or IS-IS to peer with the other connected LSRs so that it can obtain IP layer reachability information and populate its CEF table based on it. An ATM LSR might not need to run BGP because, in most cases, it can't be an LER anyway. In addition, it needs to run label distribution protocols such as LDP and Resource Reservation Protocol (RSVP) with traffic engineering modifications (TE-RSVP), discussed in Chapter 10, to distribute the label information to the peer LSRs.

MPLS on an ATM switch might require that the switch maintain several labels associated with a route (or a group of routes with the same next hop). This is necessary to avoid the interleaving of packets arriving from different upstream label switches but sent concurrently to the same next hop. Either the downstream label allocation on demand or the upstream label allocation scheme can be used for the label allocation and LIB maintenance procedures with ATM switches.

9.3 MPLS QoS

QoS is an important component of MPLS. In an MPLS network, QoS information is carried in the label header's MPLS CoS field.

Like IP QoS, MPLS QoS is achieved in two main logical steps, as shown in Table 9.2, and uses the same associated QoS functions. Figure 9.6 depicts the QoS functions used in an MPLS network.

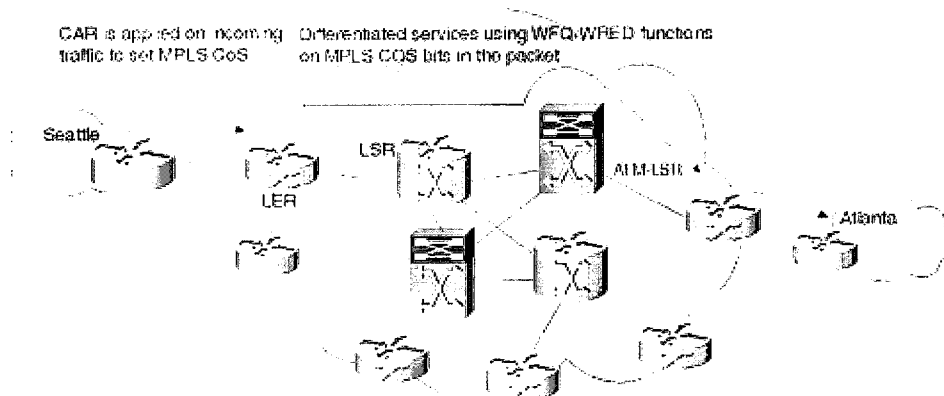


Figure 9. 6 QoS in an MPLS Network

MPLS uses the same IP QoS functions to provide differentiated QoS for traffic within an MPLS network. The only real difference is that MPLS QoS is based on the CoS bits in the MPLS label, whereas IP QoS is based on the IP precedence field in the IP header.

On an ATM backbone with an MPLS-enabled ATM switch, the switch can support MPLS CoS in two ways:

Single Label Switched Path (LSP) with Available Bit Rate (ABR) service

Parallel LSPs with Label Bit Rate (LBR) service

Table 9. 2 MPLS QoS

Step	Place of Application	Applicable Functions	QoS	QoS action
1	Ingress (edge) router to the MPLS cloud	Committed Access Rate (CAR)		(Option 1) CAR polices traffic on the ingress router for all incoming IP traffic entering the MPLS cloud. It sets an IP precedence value for traffic according to the traffic profile and policies. The IP packet's IP precedence value is copied

				into the MPLS CoS field.
				(Option 2) CAR polices traffic on the ingress router for all incoming IP traffic entering the MPLS cloud. It sets an MPLS CoS value for traffic according to the traffic profile and contract.
				The precedence value in the IP header is left unchanged end-to-end, unlike Option 1.
2	Entire MPLS network	Weighted Queuing (WFQ)	Fair (WFQ),	Traffic differentiation based on the MPLS CoS field in the MPLS backbone
		Weighted Error (WRED)	Random using the IP QoS functions WFQ and WRED.	

A single LSP using ATM ABR service can be established through LDP. All MPLS traffic uses the same ABR LSP, and the differentiation is made on the ingress routers to the ATM cloud by running WFQ and WRED algorithms on traffic going over an LSP.

Multiple LSPs in parallel can be established through LDP to support traffic with multiple precedence values. Each established LSP is mapped to carry traffic of certain MPLS CoS values. The LSPs use the LBR ATM service. LBR is a new ATM service category that relies on scheduling and discarding in the ATM switch based on WFQ and WRED, respectively, and hence is more appropriate for IP.

When an ATM switch doesn't support MPLS, you can use ATM QoS using the ATM Forum traffic class (Constant Bit Rate [CBR], Variable Bit Rate [VBR], and ABR) and its IP interworking, as discussed in Chapter 8, "Layer 2 QoS: Interworking with IP QoS."

9.4 End-to-End IP QoS

You can set the MPLS CoS bits at the edge of the network to provide traffic classification so that the QoS functions within the network can provide differentiated QoS. As such, to deliver end-to-end IP QoS across a QoS-enabled MPLS network, you map or copy the IP precedence value to the MPLS CoS bits at the edge of the MPLS network. The IP precedence value continues to be used after the packet exits the MPLS network. Table 9.3 shows the various QoS functions for delivering end-to-end IP QoS across an MPLS network.



Table 9.3 End-to-End IP QoS Across an MPLS Network

Step	Place of Application	Type of QoS	QoS Function
1	IP cloud (before entering the MPLS network)	IP QoS	Standard IP QoS policies are followed. At the network boundary, incoming traffic is policed and set with an IP precedence value based on its service level. Differentiated service is based on the precedence value in the IP network.
2	Ingress router to the cloud	IP/MPLS QoS Interworking	Packet's IP precedence value is copied into the MPLS CoS field. Note that the MPLS CoS field can also be set directly based on the traffic profile and service contract.
3	MPLS network	MPLS QoS	Traffic differentiation is based on the MPLS CoS field in the MPLS backbone using the IP QoS functions WFQ and WRED.
4	IP network (after traversing the MPLS network)	IP QoS	IP precedence in the IP header continues to be the basis for traffic differentiation and network QoS.

9.4.1 LER

IP traffic from the Seattle site going to the Atlanta site enters the MPLS network on the LER router.

9.5 MPLS VPN

One important MPLS application is the VPN service. A client with multiple remote sites can connect to a VPN service provider backbone at multiple locations. The VPN backbone offers connectivity among the different client sites. The characteristics of this connectivity make the service provider cloud look like a private network to the client. No communication is allowed between different VPNs on a VPN service provider backbone. Any device at a VPN site can communicate only with a device at a site belonging to the same VPN.

In a service provider network, a Provider Edge router connects to the Customer Edge router at each VPN site. A VPN usually has multiple geographically distributed sites that connect to the service provider's local Provider Edge routers. A VPN site and its associated routes are assigned one or more VPN colors. Each VPN color defines the VPN a site belongs to. A site can communicate with another site connected to the VPN backbone only if it belongs to the same VPN color. A VPN intranet service is provided among all sites connected to the VPN backbone using the same color. A site to one VPN can communicate to a different VPN or to the Internet by using a VPN extranet service. A VPN extranet service can be provided by selectively leaking some external routes of a different VPN or of the Internet into a VPN intranet.

Provider Edge routers hold routing information only for the VPNs to which they are directly connected. Provider Edge routers in the VPN provider network are fully meshed using multiprotocol internal BGP (IBGP) peerings.

Multiple VPNs can use the same IP version 4 (IPv4) addresses. Examples of such addresses are IP private addresses and unregistered IP addresses. The Provider Edge router needs to distinguish among such addresses of different VPNs. To enable this, a new address family called VPN-IPv4 is defined. The VPN-IPv4 address is a 12-byte value; the first eight bytes carry the route distinguisher (RD) value, and the last four bytes consist of the IPv4 address. The RD is used to make the private and unregistered

IPv4 addresses in a VPN network unique in a service provider backbone. The RD consists of a 2-byte autonomous system (AS) number, followed by a 4-byte value that the provider can assign. VPN-IPv4 addresses are treated as a different address family and are carried in BGP by using BGP multiprotocol extensions. In these BGP extensions, label-mapping information is carried as part of the Network Layer Reachability Information (NLRI). The label identifies the output interface connecting to this NLRI.

The extended community attribute is used to carry Route Target (RT) and Source of Origin (SOO) values. A route can be associated with multiple RT values similar to the BGP community attribute that can carry multiple communities for an IP prefix. RT values are used to control route distribution, because a router can decide to accept or reject a route based on its RT value. The SOO value is used to uniquely identify a VPN site. Table 9.4 lists some important MPLS VPN terminology.

Table 9.4 MPLS VPN Terminology

Term	Definition
Customer Edge Router	A customer router that interfaces with a Provider Edge router.
Provider Edge Router	A provider router that interfaces with a Customer Edge router.
Provider Router	A router internal to the provider network. It doesn't have any knowledge of the provisioned VPNs.
VPN Routing and Forwarding (VRF) Instance	A routing and forwarding table associated with one or more directly connected customer sites. A VRF is assigned on the VPN customer interfaces. VPN customer sites sharing the same routing information can be part of the same VRF. A VRF is identified by a name, and it has local significance only.
VPN-IPv4	Includes 64-bit RD and 32-bit IPv4 addresses.

Address	
SOO	Identifies the originating site.
RD	64-bit attribute used to uniquely identify VPN and customer address space in the provider backbone.
RT	64-bit identifier to indicate which routers should receive the route.



9.6 MPLS VPN QoS

QoS is a key component of a VPN service. Similar to IP QoS discussed in Part I of the book, the MPLS VPN QoS can be a differentiated service or a guaranteed service. You can use the same IP QoS functionality discussed in Part I, "IP QoS," to deliver MPLS VPN QoS. The coarse-grained differentiated QoS is provided by use of the CAR, WFQ, and WRED functions, whereas the fine-grained guaranteed service is provided by use of the RSVP protocol.

9.6.1 Differentiated MPLS VPN QoS

This QoS model delivers capabilities that, in some ways, look similar to Committed Information Rate (CIR) in a Frame Relay network.

Each VPN site is offered a CAR and a Committed Delivery Rate (CDR) for each port at which they access the network. In a Frame Relay network, CIR applies to both the incoming and the outgoing traffic from a device connected to the network. In a connectionless IP VPN environment, however, two different rates exist for each port connected to the network offering VPN service:

CAR for all incoming traffic from the site into the VPN service network's access port

CDR for all outgoing traffic from the rest of the VPN network to the site connected through the access port

The traffic on an MPLS VPN-enabled network's access port is said to be committed if the access port's incoming and outgoing traffic falls below the contracted CAR and CDR, respectively. Committed packets are delivered with a probability higher than that of uncommitted traffic.

Because of the connectionless nature of an IP VPN service, you can send packets from any site to any site within a VPN, but you must specify the committed traffic rate for a site's outgoing and incoming traffic separately. Because Frame Relay is connection-oriented, the same traffic rate applies for both ends of the circuit.

To implement CAR and CDR service on an access port, a traffic policing function for both incoming and outgoing traffic is applied. A policing function applies a higher IP precedence value for committed traffic than uncommitted traffic. In the service provider VPN backbone, the WFQ and WRED differentiated QoS functions are applied to deliver committed traffic at a probability higher than uncommitted traffic. Table 9.5 illustrates the QoS functions applied on traffic from one VPN site to the other through an MPLS VPN provider network.

Table 9.5 MPLS VPN QoS Functions

Step	Place of Application	QoS Function
1	Ingress router	<p>(Option 1) CAR polices traffic on the Provider Edge router at the service provider for the incoming traffic from the Customer Edge router. Sets the IP precedence value for traffic according to the traffic profile and contract. The IP packet's IP precedence value is copied into the MPLS CoS field.</p> <p>(Option 2) CAR polices traffic on the Provider Edge router at the service provider for the incoming traffic from the Customer Edge router. Sets the MPLS CoS value for traffic according to the traffic profile and contract.</p>
2	Service provider backbone	Traffic differentiation based on the MPLS CoS field in the MPLS backbone by using the WFQ and WRED IP QoS functions.
3	Egress	CoS field on the MPLS label is copied to the IP precedence

	router	field in the IP header.
4	Egress router	CAR does outbound traffic policing on the Provider Edge router interface connecting to the destination Customer Edge router based on CDR.

The service provider for the VPN customer provides the CAR and CDR services for each access port, as shown in Figure 9.7. The VPN service provider provisions its network to deliver its access ports' CAR and CDR rates. Any traffic over the committed rates is dropped at a higher probability over the committed traffic.

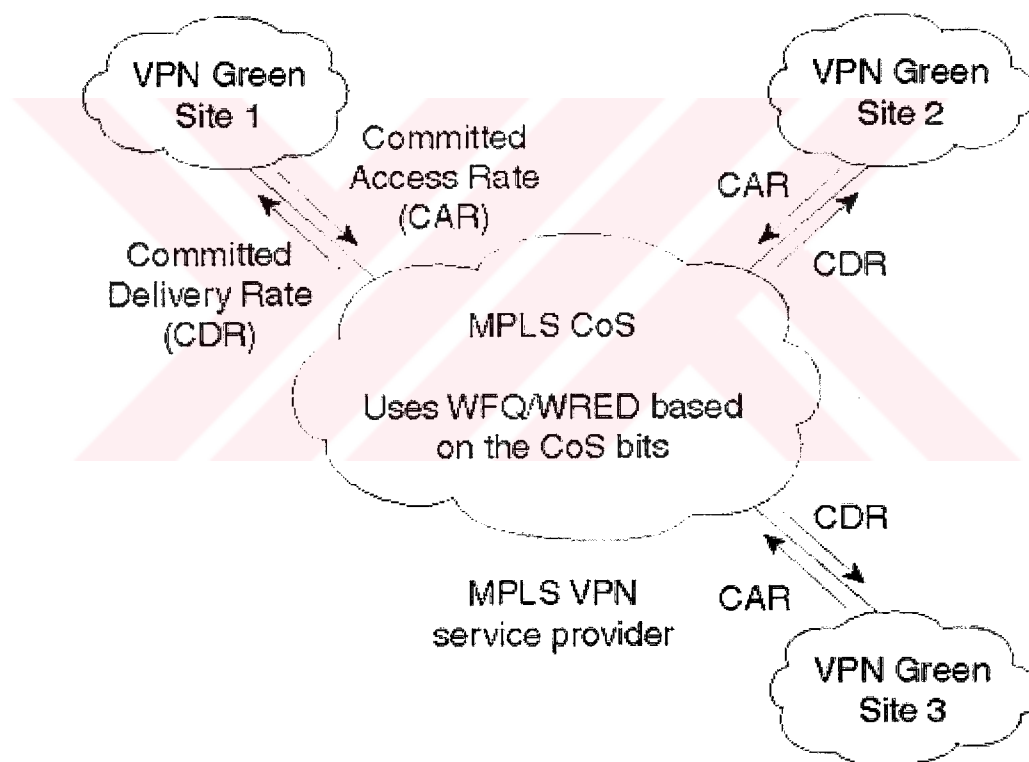


Figure 9. 7 MPLS VPN Differentiated Services (Diff-Serv) QoS

Committed packets in a properly provisioned MPLS VPN network are delivered with a high probability and provide the same service level as CIR in Frame Relay networks.

9.6.2 Guaranteed QoS

Guaranteed QoS requires the use of RSVP end-to-end along the path, from the source to the destination at the VPN sites connected by the VPN service provider backbone. The extent and level of guaranteed QoS depends on which part of the network makes explicit reservations through RSVP PATH messages. The three levels of guaranteed QoS deployment are discussed next. Figure 9.8 depicts these three options, which vary primarily on the QoS offered from the VPN service provider.

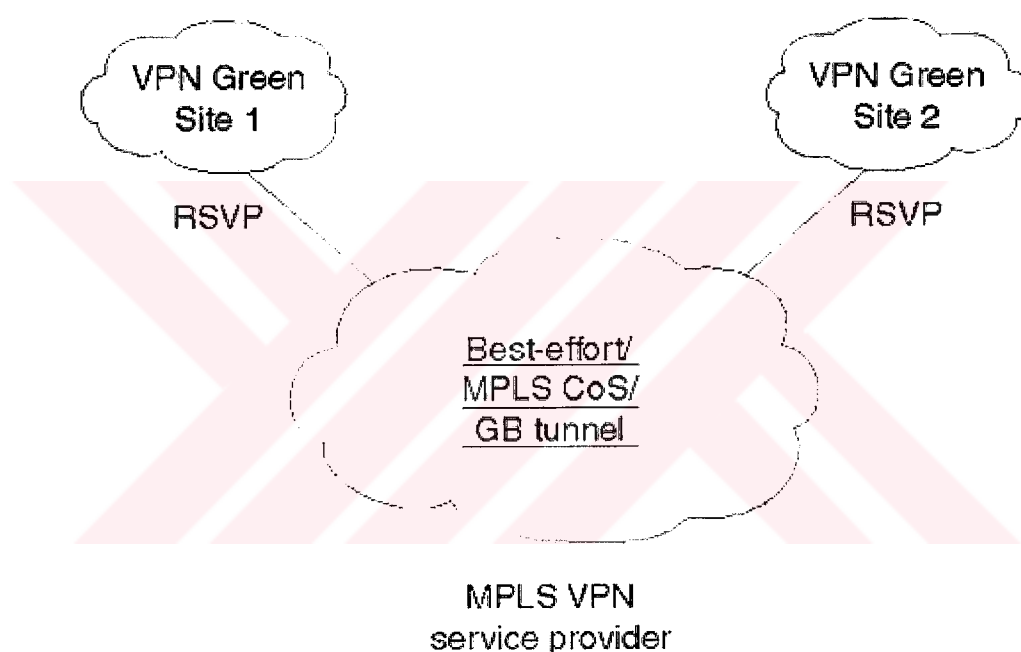


Figure 9. 8 Guaranteed MPLS VPN QoS

9.6.3 RSVP at VPN Sites Only

RSVP reservations are made only on the nodes in the VPN sites. A VPN service provider just passes any RSVP packet as it would any normal IP data packet.

This enables a customer to control resource allocation within the customer's sites to specific applications but has no effect on how the customer's traffic is handled in the service provider network.

9.6.4 RSVP at VPN Sites and Diff-Serv Across the Service Provider Backbone

RSVP reservations are made only on the nodes in the VPN sites. At the ingress to the VPN service provider, the guaranteed traffic is marked with a high MPLS CoS value, such that the guaranteed traffic is delivered across the MPLS VPN service provider with a high degree of probability across its network by IP QoS functions such as WFQ and WRED. The MPLS VPN service provider passes any RSVP packet as any normal IP data packet.

Thus, reserved traffic receives better service without the service provider having to keep any per-customer reservation state in the provider network.

9.6.5 End-to-End Guaranteed Bandwidth

Guaranteed bandwidth (GB) tunnels are used to carry traffic that requires resource reservations across a service provider backbone. The basis for a GB tunnel project is to mark a fraction of guaranteed bandwidth's queue weight as occupied.

RSVP reservations are made end-to-end from the source to the destination across the VPN service provider. Within the service provider network, these RSVP messages are carried along the GB tunnel as normal data packets.

CHAPTER TEN

MPLS TRAFFIC ENGINEERING

10 MPLS Traffic Engineering

Traffic engineering (TE) provides the capability to specify an explicit path for certain traffic flows to take. Internet Protocol (IP) traffic is routed on a hop-by-hop basis and follows a path that has a lowest cumulative Layer 3 metric to the traffic destination. The path the IP traffic takes might not be optimal, because it depends on static link metric information without any knowledge of the available network resources or the requirements of the traffic that needs to be carried on that path. (Keshav,1997)

Chapter 9, "QoS in MPLS-Based Networks,"_discusses Multiprotocol Label Switching (MPLS) label allocation and distribution for switching packets so that they follow the destination-based routing path. This chapter focuses on MPLS TE. In MPLS TE, a Label Switched Path (LSP) is established for carrying traffic along an explicit traffic-engineered path, which can be different from the normal destination-based routing path. Resource Reservation Protocol (RSVP) with TE modifications (TE-RSVP) is used as a signaling protocol for setting up the LSP.

For a service provider, the biggest challenge in deploying end-to-end quality of service (QoS) is the inability to determine the exact path the IP packets take. Any one router in the routed path does not predetermine the path the IP traffic takes; rather, it is a result of hop-by-hop routing decisions. IP routing behaves this way because IP is a connectionless protocol, unlike telephone networks, Frame Relay networks, or Asynchronous Transfer Mode (ATM) networks, which are connection-oriented. In Routing by Resource Reservation (RRR), MPLS circuit capabilities are exploited for IP

TE. Hence, MPLS LSP-based tunnels are used for TE, and labels are used to provide forwarding along an explicit path different from the one resulting from destination-based forwarding.

Presently, the MPLS TE solution is limited to a single routing domain. Interior Gateway Protocols (IGPs) need extensions to support TE. At this time, only the Open Shortest Path First (OSPF) and Intermediate System-to-Intermediate System (IS-IS) protocols support extensions for TE.

10.1 The Layer 2 Overlay Model

Historically, Layer 2 overlay networks are used to engineer traffic and manage bandwidth. Layer 3 (IP) sees a logical full mesh across the Layer 2 (ATM or Frame Relay) cloud. Each router has a logical Layer 3 connection to every other router connected to the Layer 2 cloud.

Figure 10.1 shows a physical and logical Layer 3 view of such overlay models. The physical model shows Layer 3 devices—routers connected through Layer 2 (Frame Relay or ATM) switches. The switches in the network help define the exact physical path taken by any Layer 3 traffic entering the Frame Relay or the ATM cloud.

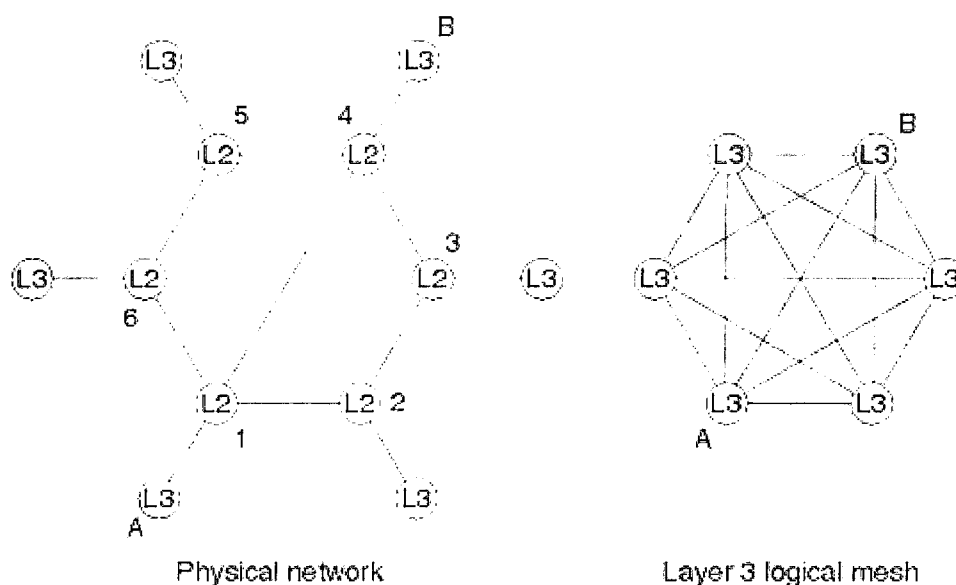


Figure 10.1 Layer 2 Overlay Model for TE

For the traffic from Router A to Router B, for example, the Layer 2 cloud offers three physical paths: A->1->4->B, A->1->2->3->4->B, and A->1->6->5->4->B. The actual path the IP traffic takes, however, is determined by the path predetermined by the Layer 2 switches in the network. The use of the explicit Layer 2 transit layer gives you exact control over how traffic uses the available bandwidth in ways not currently possible by adjusting the Layer 3 IP routing metrics.

Large mesh networks mean extra infrastructure costs. They might also cause scalability concerns for the underlying IGP routing protocol, such as OSPF and IS-IS, as the normal IGP flooding mechanism is inefficient in large mesh environments.

10.2RRR

Routing protocols such as OSPF and IS-IS route traffic using the information on the network topology and the link metrics. In addition to the information supplied by a

routing protocol, RRR routes an IP packet taking into consideration its traffic class, the traffic class' resource requirements, and the available network resources.

Figure 10.2 shows two paths from San Francisco to New York in a service provider network—one through Dallas and another through Chicago. The service provider noticed that the traffic from San Francisco to New York usually takes the San Francisco->Chicago->New York path. This path becomes heavily congested during a certain period of the day, however. It also was noted that during this period of congestion on the path from San Francisco to New York through Chicago, the path from San Francisco to New York through Dallas is heavily underutilized. The need is to engineer the traffic such that it is routed across a network by best utilizing all the available network resources. In this case, all the traffic between San Francisco and New York is getting poor performance because the network cannot use the available alternate path between these two sites during a certain period of the day. This scenario typifies the application and the need for TE. Case Study 10-1 discusses a more detailed scenario to illustrate the workings of MPLS TE.

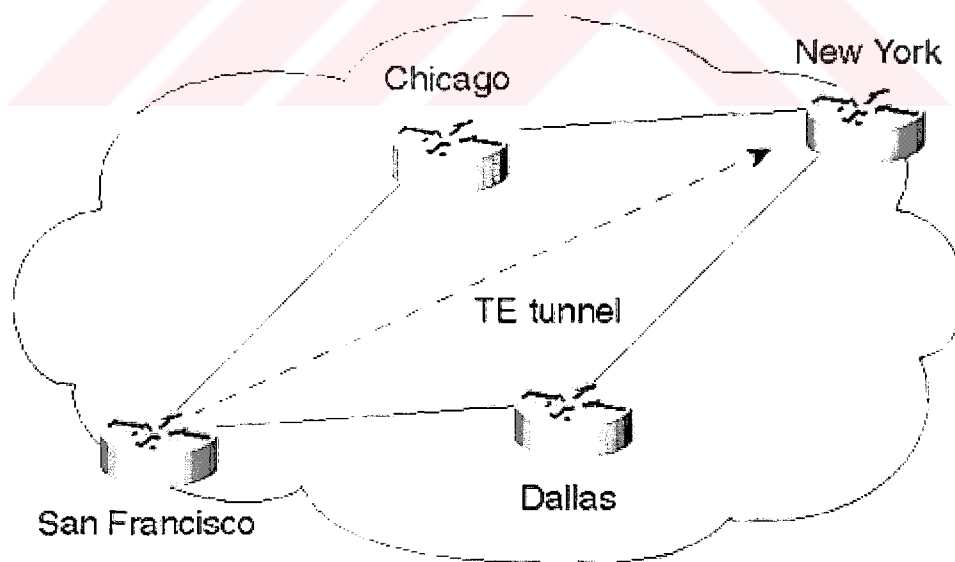


Figure 10. 2 TE Tunnel from the San Francisco Router to the New York Router

After some traffic analysis, it was clear that if all New York-bound traffic from San Francisco is carried along the path through Dallas rather than Chicago during its period of congestion, both the paths will be optimally utilized. Therefore, a TE tunnel is established between San Francisco and New York. It is called a tunnel because the path taken by the traffic is predetermined at the San Francisco router and not by a hop-by-hop routing decision. Normally, the TE tunnel takes the path through Chicago. During the period of congestion at Chicago, however, the TE path changes to the path through Dallas. In this case, TE resulted in optimal utilization of the available network resources while avoiding points of network congestion. You can set up the TE path to change back to the path through Chicago after sufficient network resources along that path become available. (Keshav,1997)

RRR TE requires the user to define the traffic trunk, the resource requirements and policies on the traffic tunnel, and the computation or specification of the explicit path the TE tunnel will take. In this example, the traffic trunk is New York-bound traffic from San Francisco, the resource requirements are the TE tunnel's bandwidth and other policies, and the explicit path is the path from San Francisco to New York through Dallas.

An RRR operational model is shown in Figure 10.3. It depicts the various operational functional blocks of the TE in a flowchart format. The following sections discuss each function in detail.

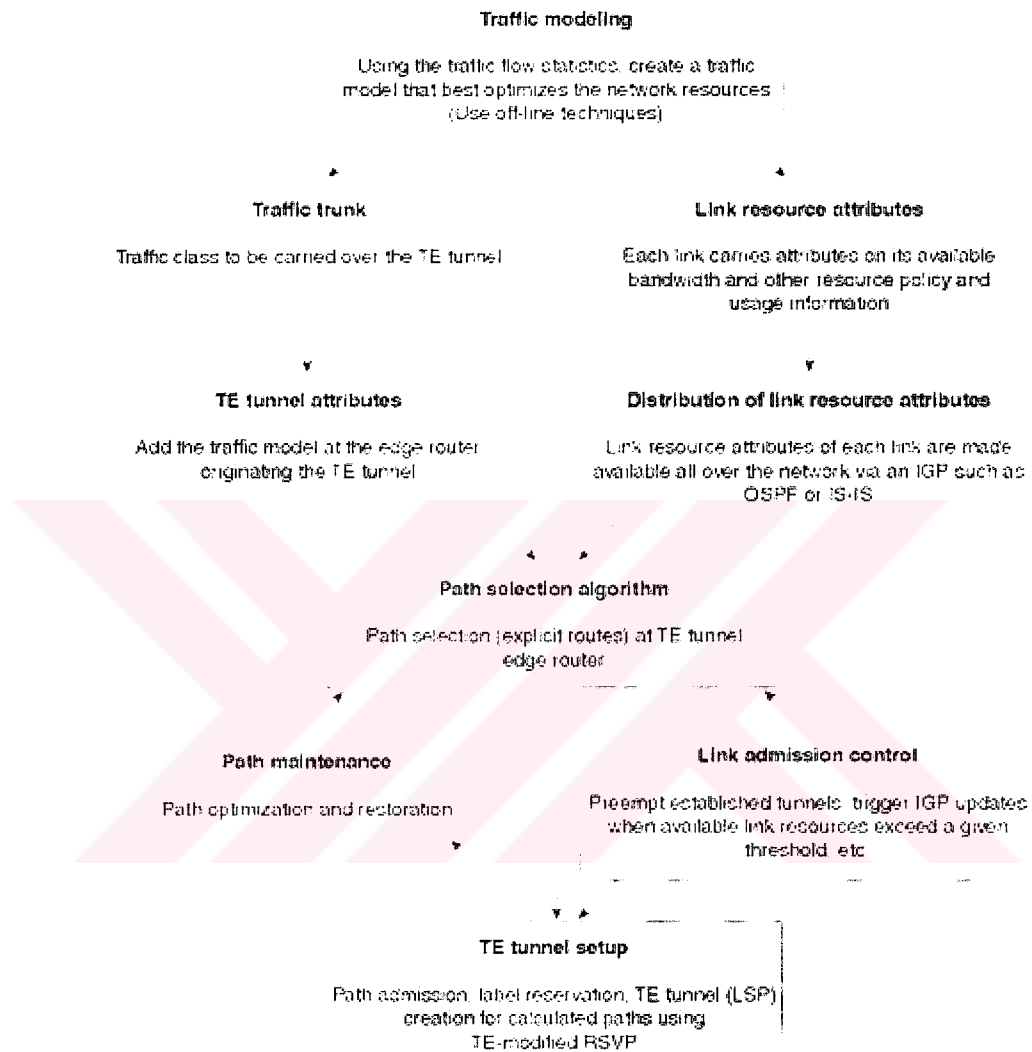


Figure 10.3 Block Diagram of TE Operation

10.3 TE Trunk Definition

The TE trunk defines the class of packets carried on a TE tunnel. This policy is local to the head-end router originating the TE tunnel setup.

As discussed in Chapter 3, "Network Boundary Traffic Conditioners: Packet Classifier, Marker, and Traffic Rate Management," a traffic class is defined flexibly based on the Transmission Control Protocol/Internet Protocol (TCP/IP) traffic headers. A traffic class can be based on a single parameter, such as the IP destination or the MPLS Class of Service (CoS) field, or on a number of parameters, such as all File Transfer Protocol (FTP) traffic going from a certain sender to a specific destination. In RRR, all packets of a traffic class take a specified defined or dynamically determined common path across a network. For this reason, RRR traffic classes are also termed traffic trunks.

10.4TE Tunnel Attributes

A TE tunnel is given attributes to describe the traffic trunk's requirements and to specify various administrative policies. This section discusses the various tunnel attributes.

10.4.1 Bandwidth

The bandwidth attribute shows the end-to-end bandwidth required by a TE tunnel. You can define it based on the requirements of the traffic class being carried within the TE tunnel.

10.4.2 Setup and Holding Priorities

The setup and holding priorities are used for admission control. Holding priority determines priority for holding a resource, whereas setup priority determines priority for taking a resource.

When resources are in contention, a new tunnel with a high setup priority can preempt all established tunnels in the path with a holding priority less than the new tunnel's setup priority. An established TE tunnel with the highest holding priority cannot

be preempted. Table 10.1 shows the implications of the low and high values for TE tunnel setup and holding priorities.

Table 10. 1 Implications of the Low and High Values for the TE Tunnel Setup and Holding Priorities

	High Value	Low Value
Setup Priority	Likely to preempt established TE tunnels (preemptor).	Less likely to preempt established TE tunnels (nonpreemptor).
Holding Priority	Less likely to be preempted by a newly established TE tunnel (non-preemptable).	Likely to be preempted by a newly established TE tunnel (preemptable).

10.4.3 Resource Class Affinity

The resource class affinity attribute provides a means to apply path selection policy by administratively including or excluding specific links in the network. This resource class affinity attribute consists of a 32-bit resource affinity attribute and a 32-bit resource class mask. The resource affinity attribute indicates whether to include or exclude a specific link in the path computation process. Each link carries a resource class attribute, which defaults to 0x00000000 unless it is explicitly specified. The resource class mask shows the interesting bits of the resource class link attribute. The resource class, the resource class mask, and the resource class affinity attributes are related to each other as follows:

$$\text{Resource Class} \& \text{Resource Class Mask} = \text{Resource Class Affinity}$$

where

& indicates a bit-wise logical AND operation and

= indicates a bit-wise logical equality.

Table 10.2 tabulates the link inclusion or exclusion policy in a TE tunnel path selection based on the resource attributes.

Table 10. 2 Policy on Including or Excluding a Link in a TE Tunnel Path Selection

Resource Class Attribute of a Link	Resource Class Affinity of a TE Tunnel	Policy on Including or Excluding the Link in a Possible TE Tunnel Path
	Resource Class Mask	Resource Affinity
1	1	1
0	1	0
1 or 0	0	0

10.4.4 Path Selection Order

Path selection order specifies the order in which an edge route selects explicit paths for TE tunnels. The explicit path is a source route specified as a sequence of IP addresses that is either administratively specified or dynamically computed based on the shortest path meeting the constraints. Path selection order shows the order in which the administratively specified paths or the dynamically derived path is used to establish a TE tunnel.

10.4.5 Adaptability

The adaptability attribute specifies whether an existing TE tunnel needs to be reoptimized when a path better than the current TE tunnel path comes up. Reoptimization is discussed later in this chapter.

10.4.6 Resilience

The resilience attribute specifies the desired behavior if the current TE tunnel path no longer exists. This typically occurs due to network failures or preemption. Restoration of a TE tunnel when the current path doesn't work is addressed later in this chapter.

10.5 Link Resource Attributes

All links in an RRR network are described by attributes showing the available resource information and link usage policy for RRR path computation.

10.5.1 Available Bandwidth

The available bandwidth attribute describes the amount of bandwidth available at each setup priority. The available bandwidth might not be the actual available bandwidth. In certain situations, a network operator can choose to oversubscribe a link by assigning it to a value that is higher than its actual bandwidth.

10.5.2 Resource Class

The resource class attribute colors the link. As was discussed earlier in this chapter, a tunnel decides to include or exclude a link in its path selection computation based on the link's resource class attribute and its own resource class affinity attribute.

10.6 Distribution of Link Resource Information

An important underlying requirement for TE is the distribution of local link resource information throughout the RRR network. The link resource attributes are flooded across the network using extensions of the OSPF and IS-IS link-state routing protocols.

Existing link layer routing protocols flood the metric information for all the links in the network, along with the network topology information, to calculate the shortest path to a destination. In an RRR network, a routing protocol is extended to flood the resource attributes in addition to the metric information for each link.

Resource attribute flooding is independent of metric information. Link resource information flooding happens when a link state changes, the link's resource class changes, or the amount of available bandwidth crosses a preconfigured threshold. User-configurable timers control flooding frequency. (Keshav,1997)

10.7 Path Selection Policy

A TE path needs to obey both the link constraints along a path and the TE tunnel requirements. The TE path computation takes the following steps:

1. Considers the following attributes and information:
 - TE tunnel attributes specified on the head-end router originating this tunnel.
 - Link resource attributes from the entire network. They are learned through the IGP.
 - Network topology information from the IGP.
 - The TE tunnel's current path if the TE tunnel is being reoptimized.
2. Prunes links with insufficient bandwidth or fail resource policy.
3. Runs a separate Shortest Path First (SPF) algorithm to compute the shortest (minimum metric) path on the IGP protocol's link state database after removing any pruned links.

This instance of the SPF algorithm is specific to the TE path in question and is different from the SPF algorithm a router uses to build its routing table based on the entire link-state database.

An explicit path for the TE tunnel is computed from the SPF run. The computed explicit path is expressed as a sequence of router IP addresses. Upon request to establish a TE tunnel, an explicit path is used in establishing the TE tunnel based on the path selection order.

10.8 TE Tunnel Setup

TE-RSVP is used to signal TE tunnels. It uses the same original RSVP messages as the generic signaling protocol, with certain modifications and extensions to support this new application. TE-RSVP helps build an explicitly routed LSP to establish a TE tunnel.

TE-RSVP adds two important capabilities that enable it to build an Explicitly Routed Label Switched Path (ER-LSP): a way to bind labels to RSVP flows and explicitly route RSVP messages. On an ER-LSP-based TE tunnel, the only sender to the TE tunnel is the LSP's first node, and the only destination is the LSP's last node. All intermediate nodes in the LSP do normal label switching based on the incoming label. The first node in the LSP initiates ER-LSP creation. The first node in the LSP is also referred to as the head-end router.

The head-end router initiates a TE tunnel setup by sending an RSVP PATH message to the tunnel destination IP address with a Source Route Object (SRO) specifying the explicit route. The SRO contains a list of IP addresses with a pointer pointing to the next hop in the list. All nodes in the network forward the PATH message to the next-hop address based on the SRO. They also add the SRO to their path state block. When the destination receives the PATH message, it recognizes that it needs to set up an ER-LSP based on the Label Request Object (LRO) present in the PATH message and generates an RSVP reservation request (RESV) message for the session. In the RSVP RESV message that it sends toward the sender, the destination also creates a label and sends it as the LABEL object. A node receiving the RESV message uses the label to send all traffic over that path. It also creates a new label and sends it as a LABEL object in the RSVP RESV message to the next node toward the sender. This is the label the node expects for all incoming traffic on this path.

An ER-LSP is formed and the TE tunnel is established as a result of these operations. Note that no resource reservations are necessary if the traffic being carried is best-effort traffic. When resources need to be allocated to an ER-LSP, the normal RSVP objects,

Tspec and Rspec, are used for this purpose. The sender Tspec in the PATH message is used to define the traffic being sent over the path. The RSVP PATH message destination uses this information to construct appropriate receiver Tspecs and Rspecs used for resource allocation at each node in the ER-LSP.

10.9 Link Admission Control

Link admission control decides which TE tunnels can have resources. It performs the following functions:

- **Determines resource availability**— It determines if resources are available.
- **Tears tunnels when required**— It must tear down existing tunnels when new tunnels with a high setup priority preempt existing tunnels with a lower holding priority.
- **Maintains local accounting**— It maintains local accounting to keep track of resource utilization.
- **Triggers IGP updates**— It triggers IGP flooding when local accounting information shows that the available resources exceeded the configured thresholds.

10.10 TE Path Maintenance

TE path maintenance performs path reoptimization and restoration functions. These operations are carried out after the TE tunnel is established. Path reoptimization describes the desired behavior in case a better potential TE path comes up after a TE path has already been established. With path reoptimization, a router should look for opportunities to reoptimize an existing TE path. It is indicated to the router by the TE tunnel's adaptability attribute.

Path restoration describes how a TE tunnel is restored when the current path doesn't work. The TE tunnel's resilience attribute describes this behavior.

10.11 TE-RSVP

TE-RSVP extends the available RSVP protocol to support LSP path signaling. TE-RSVP uses RSVP's available signaling messages, making certain extensions to support TE. Some important extensions include the following:

- **Label reservation support**— To use RSVP for LSP tunnel signaling, RSVP needs to support label reservations and installation. Unlike normal RSVP flows, TE-RSVP uses RSVP for label reservations for flows without any bandwidth reservations. A new type of FlowSpec object is added for this purpose. TE-RSVP also manages labels to reserve labels for flows.
- **Source routing support**— LSP tunnels use explicit source routing. Explicit source routing is implemented in RSVP by introducing a new object, SRO.
- **RSVP host support**— In TE-RSVP, RSVP PATH and RESV messages are originated by the network head-end routers. This is unlike the original RSVP, in which RSVP PATH and RESV messages are generated by applications in end-hosts.

Hence, TE-RSVP requires RSVP host support in routers.

- **Support for identification of the ER-LSP-based TE tunnel**— New types of Filter_Spec and Sender_Template objects are used to carry the tunnel identifier. The Session Object is also allowed to carry a null IP protocol number because an LSP tunnel is likely to carry IP packets of many different protocol numbers.
- **Support for new reservation removal algorithm**— A new RSVP message, RESV Tear Confirm, is added. This message is added to reliably tear down an established TE tunnel.

A summary of the RSVP objects that were added or modified to support TE is tabulated in Table 10.3.

Table 10.3 New or Modified RSVP Objects for TE and Their Functions

RSVP Object	RSVP Message	Purpose
Label	RESV	Performs label distribution.
Label Request	PATH	Used to request label allocation.
Source Route	PATH	Specifies the explicit source route.
Record Route	PATH, RESV	Used for diagnosis. This object is used to record the path taken by the RSVP message.
Session Attribute	PATH	Specifies the holding priority and setup priority.
Session	PATH	Can carry a null IP protocol number.
Sender_Template	PATH	Sender_Template and
Filter_Spec	RESV	Filter_Spec can carry a tunnel identifier to enable ER-LSP identification.

10.12 IGP Routing Protocol Extensions

The OSPF and IS-IS interior routing protocols have been extended to distribute link resource attributes along with IP reachability information. IS-IS uses a new tuple, consisting of a Type, a Length, and a Value commonly known as TLV, and OSPF uses Opaque Link State Advertisement (LSA) to carry the link resource information. Note that the existing dynamics of IS-IS and OSPF remain the same. Only now, OSPF and IS-IS send the current constraint information whenever it needs to flood the IP reachability information. TE link admission control on a router requests reflooding of the constraint information when it sees significant changes in the available resource information, as determined by the configured threshold values.

10.12.1 IS-IS Modifications

The IS reachability TLV is extended to carry the new data for link resource information. The extended IS reachability TLV is TLV type 22. Within this TLV, various sub-TLVs are used to carry the link attributes for TE.

10.12.2 OSPF Modifications

Because the baseline OSPF Router LSA is essentially nonextensible, OSPF extensions for TE use the Opaque LSA. Three types of Opaque LSAs exist, each having a different flooding scope. OSPF extensions for TE use only Type 10 LSAs, which have a flooding scope of an area.

The new Type 10 Opaque LSA for TE is called the TE LSA. This LSA describes routers and point-to-point links (similar to a Router LSA). For TE purposes, the existing Network LSA suffices for describing multiaccess links, so no additional LSA is defined for this purpose.

10.13 TE Approaches

The example discussed in the beginning of this chapter typifies one approach to TE—engineering traffic around points of congestion. This approach to scope TE to only a few paths might not work when other traffic in the network is carried without TE, however.

A commonly recommended approach for TE uses the full mesh of LSP-based TE tunnels between all the edge routers in a service provider network. Generally, these edge routers are customer Points of Presence (POP) routers, or routers with peer connections to other providers.

CHAPTER ELEVEN

APPLICATION of VoIP ON D.E.U'S DATA NETWORK

11 Application of VOIP on DEU's Data Network

When we begin thinking about consolidating Dokuz Eylül University's (DEU) voice and data networks into a single multiservice network, the initial application we considered is toll-bypass.

Toll-bypass enables businesses to send their intra-office voice and fax calls over their existing TCP/IP network. By moving this traffic off the Public Switched Telephone Network (PSTN), DEU can immediately save on local and long-distance charges by using extra bandwidth on their data network without losing existing functionality.

In fact, some businesses like DEU with plenty of intra-office calling, both domestic and international, is thought to have a Return On Investment (ROI) in as little as three to six months.

11.1 Integration of Networks

To integrate DEU's network , in an effort to reduce costs and maintain functionality ; we laid out the following points for our discussions :

- It wanted to work with existing PBX's from different vendors to provide an end-to-end solution
- It wanted to provide technology to its internal and external customers, but it didn't want to be on the bleeding edge in terms of technology risk.
- It wanted to avoid forklift-upgrades to its existing infrastructure. A forklift-upgrade occurs when most or all of a company's existing network hardware and software needs to be replaced with newer hardware and software. This is not only expensive in terms of capital expenditure, but it also involves physical visits to each site and disruptions of the existing network functionality.
- It wanted the new multiservice network to be cost-effective and expense-reducing.

We determined that Cisco Systems' current multiservice offerings could provide an end-to-end solution that would meet all its stated needs.

11.2 Configuration Issues

We decided not to change the current network topology but to give it a flexibility to be used with VoIP. Existing network is shown in figure 11.1:

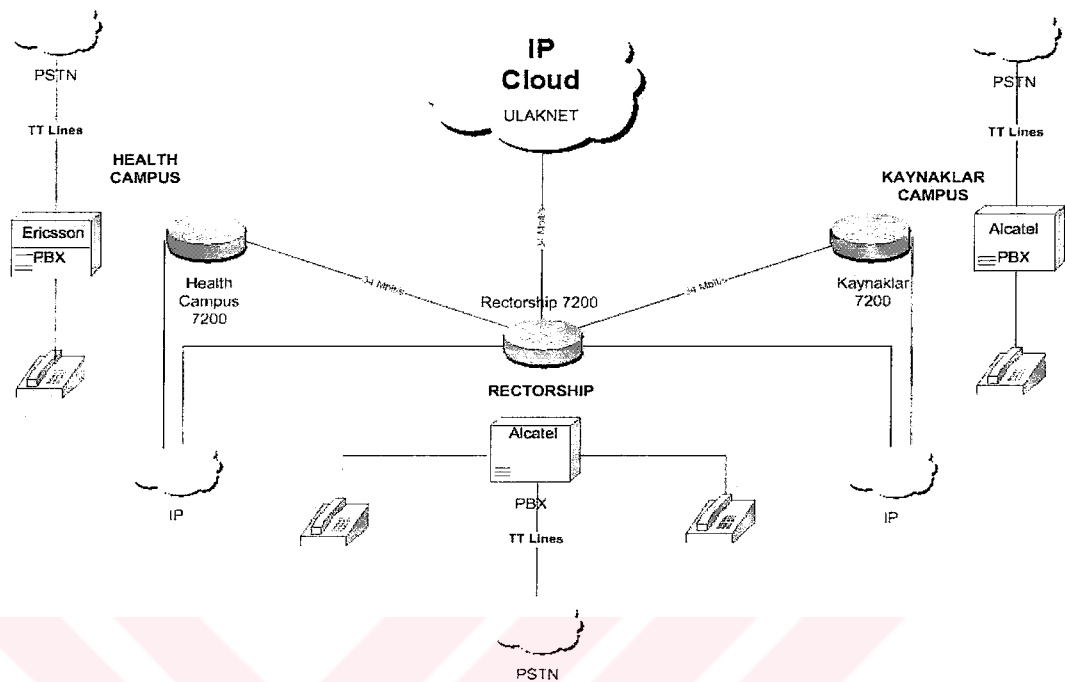


Figure 11. 1 DEU's Existing Network

In existing network;

- Three Cisco 7200 Routers that will be used to carry voice and data traffic
- In each campus , PBX from different vendors is used for voice traffic
- Routers are connected via 34 Mbit/s leased lines
- In the rectorship building ,the connection to “Ulaknet” is made with a 34 Mbit/s leased line

As shown in the figure , there are two networks exist between the campuses. The first one is the PSTN traffic network , in other words voice network and and the second one is the data network.

To integrate these networks we have to combine Routers and PBXs.

11.2.1 Routers Compatibility

In order to use the routers as VoIP gateways, we found the PA-VXA/VXB/VXC voice port adaptors for the Cisco 7200/7400/7500 router platforms.

These voice adapters are able to combine T1/E1 connectivity and onboard digital signal processor (DSP) resources to provide unparalleled flexibility and power in directly supporting voice services on these gateways. These port adaptors are capable of supporting either T1 or E1 interfaces and depending on the selected model can support up to 60 simultaneous High Complexity (HC) or 120 simultaneous Medium Complexity (MC) codec algorithm voice calls.

In addition, and depending on the selected model, it is also possible to use the onboard DSPs as a DSP farm to provide voice services to port adaptors such as the PA-MCX-nTE1 series of products which can support voice telephony interfaces but have no direct DSP resources of their own.

11.2.2 PBX Integration

Since there are PBXs in each campus, we need to integrate PBXs' to the VoIP network.

Many private branch exchanges (PBXs) use E1 trunks running CAS as the main interface to the public switched telephone network (PSTN), and to connect to external peripherals such as voice-mail or interactive voice response (IVR) systems.

Since we have a voice-capable Cisco router equipped with the E1 Drop and Insert (D&I) Voice port adapter; selected time slots on one port of a router to be transparently connected to selected time slots on another port of another router.

A E1 trunk consists of 31 individual 64 Kb channels multiplexed together. E1 frame structure allows samples of each time slot to be sent in a repeating pattern. The timing (clocking) on a E1 trunk is embedded in the bit stream with the timing referenced to a central clock source (generally the telco). Because the clocking between E1s is synchronized, it is possible to take (drop) the bits that represent particular time slots on one E1 and insert them into other time slot positions on a different E1.



11.3 Proposed Configuration

After we have chosen the necessary equipment to be used for VoIP networks, we made the proposed configuration as in the Figure 11.2.

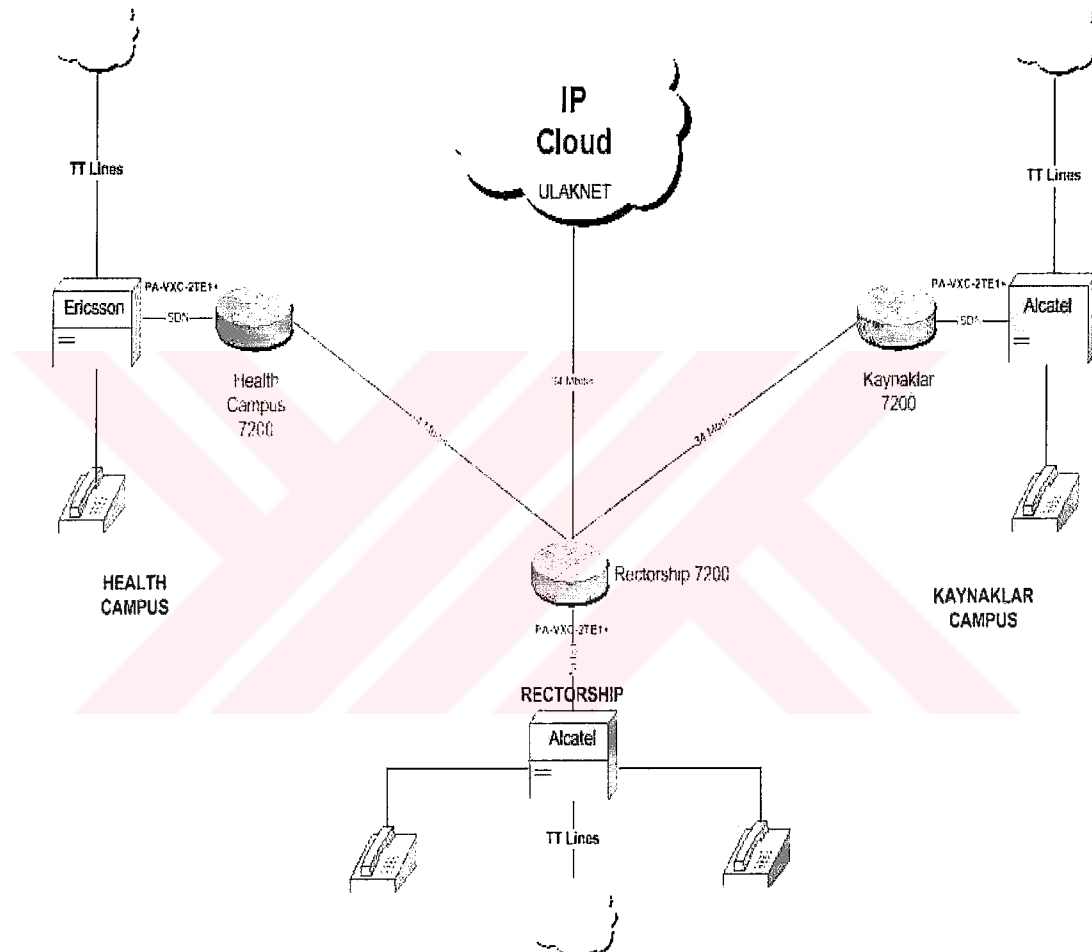


Figure 11. 2 DEU's Proposed Network

The highlights of the solution include the following:

- Leveraged D.E.U.'s existing data network, which was made up of 7200 series routers. 7200 series are modular routers/VoIP gateways. They provide more than 120 LAN and WAN interfaces, from async to optical carrier 3 (OC-3) ATM, as

well as analog and digital voice interfaces such as T1/E1, Foreign Exchange Station (FXS), FXO, and receive and transmit (E&M). Both routers share the same network modules, so stocking, sparing, and consistency across the family of products is maintained.

- Based on open-standard, H.323 protocols.
- Provided an integration path that utilized D.E.U.'s existing data network and PBX equipment.
- Required neither extensive reconfiguration of existing data and voice equipment, nor a forklift-upgrade of any equipment.
- Enabled the D.E.U IT group to continue utilizing the expertise of both its voice and data support staff.
- Interoperable with other multiservice technologies that Cisco Systems offers, including Cisco IP phones and IP PBXs, as well as H.323-based applications, such as Click-2-Dial and Netmeeting.

After modifying PBXs routing tables and making appropriate dialing plan, any employee can call another office by using VoIP.

The most important outcome expected to be immediate cost savings by moving D.E.U's intraoffice voice and fax calls onto its TCP/IP data network. D.E.U will eliminate the leased lines as well as reduced its long-distance charges associated with those calls.

The other outcomes to be , if needed , D.E.U is capable of replacing small-office key-systems with Cisco IP phones and to reduce lease costs when the key-system leases expired. Capability to integrate both Cisco IP phones and existing voice equipment with multiservice applications such as Netmeeting or Intel ProShare video-conferencing using H.323.

11.4 About Quality of Service

After setting up proposed network configuration and numbering plan, we are ready to make VoIP calls between different locations of DEU. But, do all customers need the same quality of service? The answer of this question is the main issue on what we can do about QoS.

For example, there may be a group of users which usually utilize the network only for sending e-mail, applications like MSN, small file transfers or playing Mp3's from the server. These type of applications absolutely don't need a broad bandwidth.

On the other hand, there are users like academicians which come together through video conferencing or there may be doctors which make medical consultation with their colleagues in another location and need a broad bandwidth for synchronous voice and data (MR, Ultrasound, Cardiography,...) transfer.

So, to give different Service Quality to different users (Which we called as "Differentiated Services") we should apply QoS methodologies to our new configuration and manage the resources. We can define certain different service levels between campuses. We can reserve different bandwidth sections to users or user groups.

We can also prioritize some users as "VIP" to give a priority on their traffic. We don't let them wait in a queue or we can drop other packets to reserve a place for "important" packets.

This application would give us the opportunity to test and experience most of the QoS functions like bandwidth effect, packet delay and jitter, packet loss, resource allocation, congestion avoidance, packet drop, etc...

This application would give DEU the opportunity to integrate and upgrade the existing network to an advanced VoIP network and use next generation services like Video Conferencing or Visual IP Telephony with a little cost. On the other hand, the university would save a considerable amount by canceling some of the leased lines used for classical telephony network.



CONCLUSIONS

“YOU CAN’T ALWAYS GET WHAT YOU WANT !”

The famous lyrics above summarizes the need for QoS in IP networks very well. Specially the global internet today offers no guarantees for end-to-end connection quality. That’s why we can’t use the internet for voice traffic efficiently.

In last few years, growing demand for data traffic –specially in GSM networks- drove corporate and individual users to change their classical circuit-switched systems into packet switched IP networks. The advantages behind this are;

- Convergence of separate network types in one network.
- Strategic applications for mixed media.
- Reduced costs because of one infrastructure to maintain.
- Aggregated Bandwidth.
- PBX and Trunk costing.

Data traffic grows faster, for example in GSM networks, classical circuit switched voice traffic grows 1,5 times per year whereas GPRS traffic grows 4 times per year. This indicates that the trend goes towards packet switched networks.

As deeply focused in this thesis, most important point about converged IP networks, specially VoIP networks, is Quality of Service. If a service provider could not

offer differentiated services to the customers and give desired QoS, the satisfaction goes down and the subscription won't continue.

Like corporate users, D.E.U. also can't stand far from convergence of the networks. With the application given in last chapter, D.E.U. would connect the PBX nodes together via data network, use the latest technologie, give different service levels to different user groups, and save costs. But the point not to be forgotten is, this application absolutely needs the QoS parameters to be set.

" This is the Internet, amigo. You should be grateful for what you can get and ask not what the network can do for you."

Ben Teitelbaum , VoIP Workshop, TAMU, College Station, Texas, April 2002

REFERENCES

- S. Shenker, C. Partridge, R. Guerin, Andrew S. Tanenbaum.(1999).Specification of Guaranteed Quality of Service.
- J. Postel (1981). Internet Protocol Specification.
- J. Nagle.(1984.) Congestion Control in IP/TCP Internetworks.
- W. Stevens.(1997). TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms.
- S. Keshav.(1997). An Engineering Approach to Computer Networking.
- A. Mankin.(1997).Resource Reservation Protocol (RSVP) Version 1 Applicability Statement, Some Guidelines on Deployment
- A. Romanow and S. Floyd.(1995).Dynamics of TCP Traffic over ATM Networks
- R. Coltun.(1998).The OSPF Opaque LSA Option