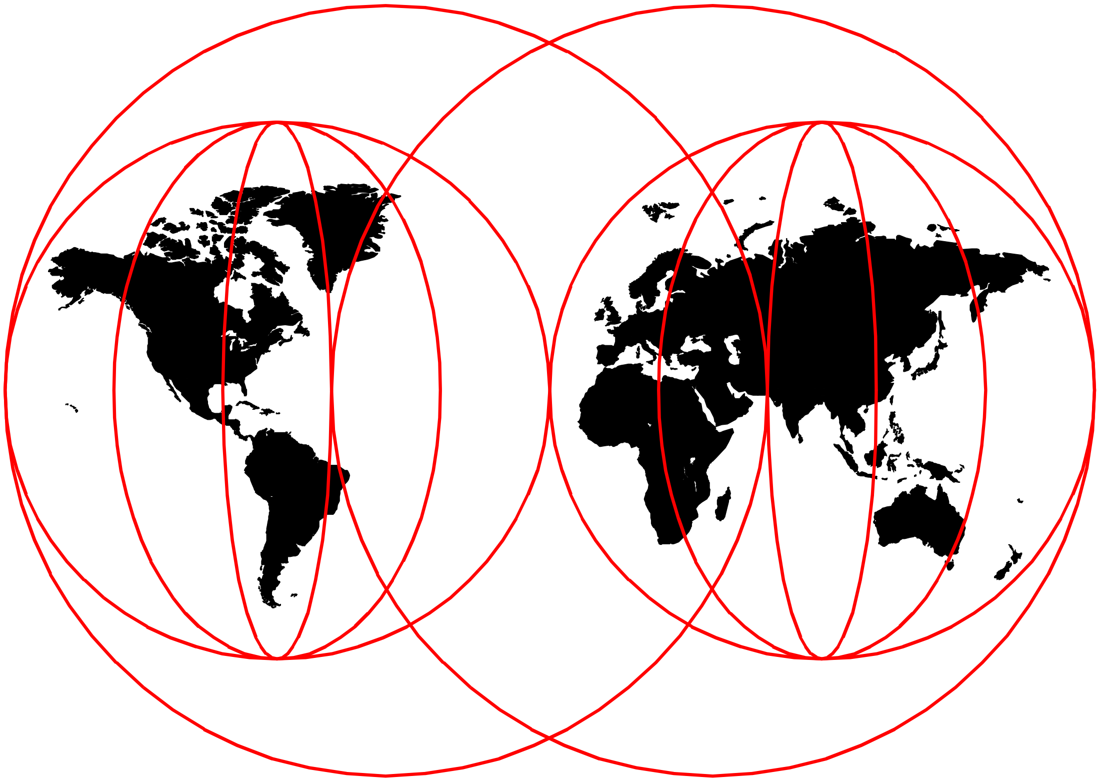
Application-Driven Networking: Class of Service in IP, Ethernet, and ATM Networks

Jonathan Follows, Detlef Straeten



International Technical Support Organization

http://www.redbooks.ibm.com

SG24-5384-00

International Technical Support Organization

SG24-5384-00

Application-Driven Networking: Class of Service in IP, Ethernet, and ATM Networks

December 1999

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix C, “Special notices” on page 139.

First Edition (December 1999)

This edition applies to Version 3.3 of Multiprotocol Access Services, Multiprotocol Routing Services, and Access Integration Services for use with the IBM 2216, 2210, and 2212 routers. It also includes information on Version 3.4 of the same software, in particular when discussing aspects of transporting voice traffic over IP networks.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. HZ8 Building 678
P.O. Box 12195
Research Triangle Park, NC 27709-2195

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Preface

This redbook discusses the implementation of Class of Service in three related networking arenas with IBM products. In each section, the terminology is discussed and then practical examples of the implementation of Class of Service using existing IBM products is shown. This book brings together information on these related subjects in a relatively concise form.

A lot of terms and abbreviations are bandied around today without any great understanding of their meaning. For example, VLAN trunking and 802.1Q are stated as being Good Things for future networks, but there does not seem to be a single source of information that explains what these things are and why they are so good. So this book helps to plug this gap, by explaining the terms and concepts to the interested technical reader, and demonstrates how to implement these features on IBM products.

This book will help you understand the relevance of the latest standards and their applicability to a real network; the implementation examples show IBM networking hardware but the standards are applicable to anyone's hardware.

The book includes discussion of the implementation of emerging standards: IP Differentiated Services and Integrated Services, LAN 802.p/Q priority and VLAN tagging mechanisms, and the combination of the two over ATM leading to MPLS.

The book has been updated to include aspects of the latest V3.4 code release for IBM routers, especially covering enhancements for transporting voice traffic over IP networks.

It also became apparent that this book fits into part of the overall scheme of application-driven networking and is being published in conjunction with *Application-Driven Networking: Concepts and Architecture for Policy-Based Systems*, SG24-5640, which describes the concepts and architecture behind a model which:

- Ensures secure transmission based on application needs
- Prioritizes application traffic based on business need
- Ensures predictable, repeatable application performance

The areas covered in this volume are:

1. IP networks, discussing the implementation and use of Differentiated Services and Integrated Services
2. Ethernet networks, mechanisms such as 802.1p and 802.1Q, their implementation and their use, and with special interest to their use in gigabit Ethernet networks
3. ATM networks, how different types of data traffic can make use of the Quality of Service mechanisms inherent in ATM networks and how imminent developments in the area of tag switching will be of benefit

This book does not cover token-ring networks, other than by reference and by implication. This is because most development and innovation is currently taking place in the Ethernet arena, in some cases to bring Ethernet up to the same capability level that token-ring has had for some time. In addition, it is seen as inevitable that the majority of new networks are being designed around Ethernet standards rather than around token-ring standards. But it's interesting to reflect that switched Ethernet with prioritization and large frame sizes is more expensive than token-ring, which has had these features all along.

The team that wrote this redbook

This redbook was produced by a team of two working at the International Technical Support Organization, Raleigh Center, with invaluable assistance from many others working in Raleigh and around the world.

Jonathan Follows is a networking specialist at the International Technical Support Organization, Raleigh. He writes redbooks on all areas of IBM networking hardware and software, most recently on ATM and related topics. Before joining the ITSO in 1998, he worked as a technical specialist providing sales and marketing support in the United Kingdom and has 15 years' experience in all types of networking. Jonathan read Mathematics at Oxford University, England, and holds a degree in Computing Science from London University, England.

Detlef Straeten is a Certified Consulting I/T Architect in Heidelberg, Germany. He has 10 years' experience in network design, implementation, and operation. He has been working for IBM Global Services for the past seven years in conjunction with IBM's Networking Hardware Division. His areas of expertise include campus high-speed networking technologies and voice/data WAN integration technology. He has participated in prior residencies at IBM Research (Yorktown, U.S.), IBM Networking Hardware Division (La Gaude, France) and IBM ITSO (Raleigh). He is an elected member of the IBM Technical Expert Council, which is a cross-organizational group of technical people in IBM's Central Region in Europe.

Thanks to the following people for their invaluable contributions to this project:

Martin Murhammer, Harri Levanen
International Technical Support Organization, Raleigh Center

Shawn Walsh, Tate Renner, Gail Christensen
International Technical Support Organization, Raleigh Center

Lynda Linney, Peter Russell
Formerly Product and Installation Support Centre, Hursley, U.K.

Chris Blenkhorn
Formerly IBM Networking Systems, U.K.

Colin Bird
IBM Networking Systems, U.K.

Tim Kearby
Formerly International Technical Support Organization, Raleigh Center

Steve Monti
IBM Networking Hardware Division, Raleigh

Robindhra Mangtani
Cable and Wireless, Global Markets, Guildford, UK

Harry Dutton
Formerly International Technical Support Organization, Australia

Comments welcome

Your comments are important to us!

We want our redbooks to be as helpful as possible. Please send us your comments about this or other redbooks in one of the following ways:

- Fax the evaluation form found in “IBM Redbooks evaluation” on page 151 to the fax number shown on the form.
- Use the online evaluation form found at `http://www.redbooks.ibm.com/`
- Send your comments in an Internet note to `redbook@us.ibm.com`

Part 1. Class of service in IP networks

Chapter 1. Overview

The concept of Class of Service when applied to IP networks is one in which different types of IP traffic are treated differently by the network.

SNA architecture defines Class of Service as follows (see *Systems Network Architecture Technical Overview*, GC30-3073):

> A class of service designates the transport network characteristics of a session. It includes such characteristics as security, transmission priority, and bandwidth. The components of a class of service differ between subarea and APPN networks. But the process of defining class of service is an activity that must take place in both types of networks before route selection can take place. During session initiation, the class of service for the session is obtained from the session-initiation request, or derived from a mode name specified in the session-initiation request. The route then selected for the session depends on the class of service for the session.
>
> In an SNA network, different classes of service can be specified, based upon the needs of the end users in the network. End users typically require sessions with widely varying data transmission requirements. A range of classes of service can therefore be provided to accommodate their session requirements. For example, the following classes of service can exist in a network:
>
> A class that provides response times suitable for high-priority interactive sessions
>
> A class that provides response times suitable for low-priority interactive sessions
>
> A class that provides routes that have the best availability
>
> A class suitable for batch processing
>
> A class suitable for high-security transmissions
>
> The classes of service available in a network are identified with alphanumeric names. After the classes of service are named, these names are used to label entries in a class-of-service table. A class-of-service table (CoS table) defines a range of acceptable characteristics and transmission priorities for each class of service in the table.
>
> In subarea networks, the characteristics of a CoS are defined implicitly by the virtual routes that are defined for it. The characteristics of a CoS are the characteristics of the underlying explicit routes to which the virtual routes for that CoS are assigned. In APPN networks, CoS characteristics are defined explicitly with CoS table parameters.
>
> In a typical network, requested sessions have differing data transmission requirements. Inquiry-response sessions usually require faster data transmission and more predictable response times than data-collection sessions. Because sessions for several different kinds of applications can be in progress over a given route, multiple classes of service specifying different transmission priorities should be provided for the route. Classes of service intended for sessions that require rapid response times should be assigned higher transmission priorities than classes of service for sessions for which slower data flow is acceptable.

SNA Class of Service was introduced to handle different networking requirements of different types of traffic over a single SNA network. SNA routers were designed to implement Class of Service, for example, by transmitting different types of data traffic over the same link at different transmission priorities. By implementing Class of Service, SNA networks were extremely successful in being able to make full use of relatively limited network bandwidth but at the same time ensuring that important network traffic was processed efficiently. SNA networks have long been able to transport a mixture of interactive and batch traffic types simultaneously and successfully.

The obvious observation is that there's nothing new with the requirement for Class of Service in IP networks. It is possibly the case with many IP networks that they grew rapidly from a position only a few years ago in which they were being used solely for unimportant traffic, and perhaps it was once possible to dismiss any requirement for differential treatment of IP traffic as unnecessary. Networks were able to merge SNA and IP traffic into a single infrastructure simply by treating all SNA traffic as important and all IP traffic as less important. This distinction can no longer be made; especially with the addition of traffic such as voice and video over IP it is necessary both to distinguish and differentiate between different types of IP traffic and, in cases where SNA is still being transported natively over the same network infrastructure, to define a correspondence between different types of IP Class of Service and the existing SNA CoS.

There are currently two different approaches to implementing Class of Service in IP networks, and both approaches are continuing to evolve at the same time as practical implementations are being developed. The two approaches are known as:

1. Differentiated Services
2. Integrated Services

These two models and examples of their implementation are covered in the next two chapters of this book.

Chapter 2. Differentiated Services

In the context of this book, the term Differentiated Services has two meanings:

1. The general term referring to the ability of network components to differentiate between different types of traffic based on indications in the received frames.
2. The specific term referring to the DiffServ feature implemented in the router code (V3.3 for 2210, 2212, 2216, and similar products) and the related standards defined by the Internet Engineering Task Force (IETF) and those documented in Request for Comments (RFC) documentation.

This chapter will deal with the theory and implementation of methods of differentiation between different traffic types using IBM's router family, which starts with discussion of the general term. Although the examples in this chapter are shown using a 2210 running Multiprotocol Routing Services (MRS) Version 3 code, any reference to version and release levels of code should be interpreted as being applicable to any of the other router platforms: 2212, 2216, and others.

The chapter concludes with mention of the capabilities of TCP/IP running on IBM mainframes to participate in DiffServ networks.

Differentiated Services differs from existing SNA Class of Service mechanisms and from other IP Class of Service mechanisms in that it takes no account of network flows or network sessions. Devices that implement Differentiated Services rules do not need to maintain any state information relating to the existence of these sessions or flows, but make their forwarding decisions solely on the information they receive in each network packet. The intent is to make implementation of Differentiated Services simple, applicable to large and small routers, and to make it scalable across large and complex networks.

The common approach taken by all of the following methods is of a router being a device that transports LAN protocols over wide area links. The assumption is that the bottleneck in the network is the collection of wide area links, and therefore, all the following mechanisms classify and queue data that is to be transmitted over these links. Specifically, the priority queueing mechanisms described below only apply to frame relay and PPP links. All LAN connections are assumed never to be congested, and no form of queueing takes place for traffic destined for these types of link.

Code Release 3.4

Shortly before the publication of this book, IBM announced the release of a new level of Common Code: V3.4. This code level includes many small enhancements but includes significant enhancements to the capability of IBM routers to transport voice traffic over IP networks. Voice over IP will be discussed in its own section; elsewhere significant changes in V3.4 over V3.3 code will be highlighted where appropriate.

2.1 Differentiation by well-known port

TCP and UDP applications identify themselves to the TCP/IP protocol suite by the use of one or more 16-bit port numbers. By common agreement, well-known port numbers fall in the range between 1 and 1023, and "ephemeral" port numbers occupy the remaining number range between 1024 and 65535; routers can inspect IP packets and make decisions based on the port numbers found in them.

2.1.1 Bandwidth Reservation System

The Bandwidth Reservation System (BRS) is the system used in IBM routers to decide which packets to prioritize and which packets to drop when demand (traffic) exceeds supply (throughput) on a network connection. When bandwidth utilization on frame relay or PPP links reaches 100%, BRS determines which traffic to drop. BRS defines traffic or circuit classes to which percentages of the bandwidth are assigned and then defines four priority queues (urgent/high/medium/low) for each defined class. BRS uses a weighted fair queueing algorithm for transmission of packets assigned to a particular class: urgent packets within a given traffic class are always transmitted first and packets are only sent from the other queues when higher-priority queues are empty.

2.1.2 A practical example

Simply identifying the well-known port number in a frame and making decisions based on the port value proved a simple and effective method of prioritizing between different types of IP traffic. Even these relatively simple filters can become quite specific:

- A UDP port filter for UDP port numbers in the range 25 to 29, which assigns the filter to traffic class A with a priority of normal.
- A TCP port filter for TCP port number 50 for IP address 5.5.5.25, which assigns the filter to traffic class B with priority of urgent.

```
local_2210 BRS [i 1] [dlci 531]>assign
Protocol or filter name [IP]? ?
IP
ARP
ASRT
APPN-HPR
SNA/APPN-ISR
TUNNELING-IP
SDLC/BSC-IP
RLOGIN-IP
TELNET-IP
NETBIOS
SNMP-IP
MULTICAST-IP
DLSW-IP
TAG1
TAG2
TAG3
TAG4
TAG5
NETWORK-HPR
HIGH-HPR
MEDIUM-HPR
LOW-HPR
XTP-IP
UDP_TCP1
UDP_TCP2
UDP_TCP3
UDP_TCP4
UDP_TCP5
TOS1
TOS2
TOS3
TOS4
TOS5
Protocol or filter name [IP]? UDP_TCP1
Class name [DEFAULT]? A
Priority <URGENT/HIGH/NORMAL/LOW> [NORMAL]? normal
Frame Relay Discard Eligible <NO/YES> [NO]?
Port Type <UDP/TCP> [UDP]?
Port Range (Low) [1]? 25
Port Range (High) [25]? 29
IP Address [0.0.0.0]?
local_2210 BRS [i 1] [dlci 531]>assign UDP_TCP2
Class name [DEFAULT]? B
Priority <URGENT/HIGH/NORMAL/LOW> [NORMAL]? urgent
Frame Relay Discard Eligible <NO/YES> [NO]?
Port Type <UDP/TCP> [UDP]? tcp
Port Range (Low) [1]? 50
Port Range (High) [50]?
IP Address [0.0.0.0]? 5.5.5.25
```

Figure 1. Configuring TCP/UDP port number filtering

```
local_2210 BRS [i 1] [dlci 531]>list all

BANDWIDTH RESERVATION listing from SRAM
bandwidth reservation is enabled
interface number 1, circuit number 531
maximum queue length 10, minimum queue length 3
total bandwidth allocated 80%
total classes defined (counting one local and one default) 4


class LOCAL has 10% bandwidth allocated
  protocols and filters cannot be assigned to this class.


class DEFAULT has 40% bandwidth allocated
  the following protocols and filters are assigned:
    protocol IP with priority NORMAL is not discard eligible
    protocol ARP with default priority is not discard eligible
    protocol ASRT with default priority is not discard eligible
    protocol APPN-HPR with default priority is not discard eligible
    protocol SNA/APPN-ISR with default priority is not discard eligible
    filter TELNET-IP with priority HIGH is not discard eligible
    filter DLSW-IP with priority HIGH is not discard eligible


class A has 10% bandwidth allocated
  the following protocols and filters are assigned:
    filter UDP_TCP1 with priority NORMAL is not discard eligible
       and represents UDP with port range 25 - 29
       and IP address 0.0.0.0


class B has 20% bandwidth allocated
  the following protocols and filters are assigned:
    filter UDP_TCP2 with priority URGENT is not discard eligible
       and represents TCP with port range 50 - 50
       and IP address 5.5.5.25

assigned tags:

default class is DEFAULT with priority NORMAL
```

Figure 2. Displaying BRS configuration

2.1.3 Advantages and pitfalls

The overwhelming advantage of this approach is one of simplicity. It is very easy to set up differentiation between, say, FTP and Telnet traffic to allow the important interactive Telnet traffic to take priority over batch file transfer (FTP) traffic. IP networks have worked reasonably well for several years by using just this approach.

The disadvantages of this approach include:

- Fragmentation. If IP packets are fragmented, only the first fragment contains the source and destination port number fields, and therefore, all subsequent fragments cannot be given an appropriate priority.
- Encryption. If IP packets are encrypted, specifically in the case of Virtual Private Networks in which encrypted tunnels are established over public IP networks, the original port number information is also encrypted and therefore are not visible to intermediate routers in the public network.

- It is a best effort approach; it is not possible to provide guarantees of minimum throughput or minimum transit delay, for example, but simply that in the case of network congestion one type of traffic will receive favorable treatment over another.

2.2 SNA: data link switching, HPR and TN3270

IP routers were frequently installed to provide an IP network alongside an existing SNA network. The expense of running parallel networks soon led to convergence on a single network, and it was usually seen as easier and more strategic to consolidate SNA and IP traffic on a backbone of IP routers.

One method of transporting SNA traffic over such a network was by using wide area bridging, and in the early days of LAN interconnection this was an approach that was often used. Some care had to be taken when enabling bridging across wide area links, but the danger in this came from the way other protocols (especially IPX) made profligate use of bandwidth.

Another approach has been to use some kind of IP transport mechanism for SNA traffic.

2.2.1 Data link switching

Data link switching (DLSw) was introduced by IBM in 1993 and is documented in RFC 1795 (Version 2 is documented in RFC 2166). DLSw provides a switching mechanism for SNA and NetBIOS traffic by providing an external appearance of a bridged LAN network by transporting frames over a TCP session.

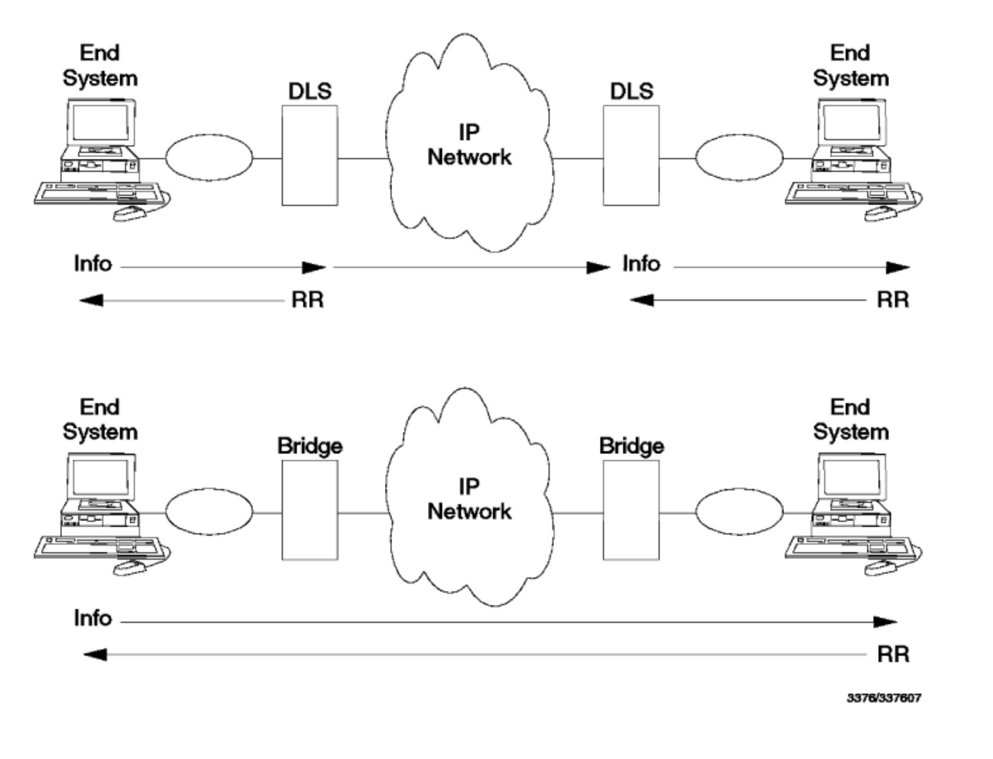


Figure 3. DLSw versus bridging

2.2.2 Enterprise Extender

Enterprise Extender, first introduced in V2.2 of the common router code, provides a UDP transport mechanism for APPN/HPR Network Layer Packets (NLPs). Another name for Enterprise Extender is therefore HPR over IP. By default, APPN/HPR packets with different transmission priorities are mapped to different UDP ports as follows:

Table 1. Mapping of HPR priority to UDP port numbers

APPN/HPR significance	UDP port
LLC Commands (TEST, XID, and others)	12000
Network transmission priority	12001
High transmission priority	12002
Medium transmission priority	12003
Low transmission priority	12004

2.2.3 TN3270

Telnet 3270, or TN3270, is defined in RFCs 1576, 1646, 1647, and 2355. It describes a method of transporting an SNA 3270 data stream over a TCP connection. By default, TN3270 uses the same well-known port number as for Telnet, which is port number 23.

2.2.4 Differentiating between SNA and IP traffic

Many networks that combine SNA and IP traffic require to differentiate in their treatment of the different protocols. SNA is more sensitive to certain types of network delays which can result in broken sessions, and for many people SNA continues to deliver the most mission-critical applications.

Since all three of the previous methods of transporting SNA data encapsulated in some form of IP packets use their own distinct port numbers (DLSw uses 2065 and 2067 by default), differentiation on the basis of port number can be extended to these SNA packets.

Many networks exist today in which SNA traffic is transported by encapsulation in IP packets and shares common wide area links with various sorts of IP traffic. Simple classification and prioritization mechanisms ensure that SNA traffic is not inappropriately hindered by IP file transfer sessions. BRS includes provision for identifying and classifying traffic into the DLSw filter class, and this is just a shortcut method of identifying DLSw by its TCP port number.

```
local_2210 BRS [i 1] [dlci 531]>assign
Protocol or filter name [IP]? dlsw
Class name [DEFAULT]? B
Priority <URGENT/HIGH/NORMAL/LOW> [NORMAL]? high
Frame Relay Discard Eligible <NO/YES> [NO]?
local_2210 BRS [i 1] [dlci 531]>list all

BANDWIDTH RESERVATION listing from SRAM
bandwidth reservation is enabled
interface number 1, circuit number 531
maximum queue length 10, minimum queue length 3
total bandwidth allocated 80%
total classes defined (counting one local and one default) 4


class LOCAL has 10% bandwidth allocated
  protocols and filters cannot be assigned to this class.


class DEFAULT has 40% bandwidth allocated
  the following protocols and filters are assigned:
    protocol IP with priority NORMAL is not discard eligible
    protocol ARP with default priority is not discard eligible
    protocol ASRT with default priority is not discard eligible
    protocol APPN-HPR with default priority is not discard eligible
    protocol SNA/APPN-ISR with default priority is not discard eligible
    filter TELNET-IP with priority HIGH is not discard eligible


class A has 10% bandwidth allocated
  the following protocols and filters are assigned:
    filter UDP_TCP1 with priority NORMAL is not discard eligible
       and represents UDP with port range 25 - 29
       and IP address 0.0.0.0


class B has 20% bandwidth allocated
  the following protocols and filters are assigned:
    filter DLSW-IP with priority HIGH is not discard eligible
    filter UDP_TCP2 with priority URGENT is not discard eligible
       and represents TCP with port range 50 - 50
       and IP address 5.5.5.25

assigned tags:

default class is DEFAULT with priority NORMAL
```

Figure 4. Assigning DLSw to a BRS class

2.2.5 Precedence and type of service

IP Version 4 defines a one-byte field in the IP header[1] as denoting service type. Exactly how this byte is to be interpreted is not consistent, because different attempts have been made at different times to define the bit settings and field lengths contained in this field. This has led to many different terms for this particular byte, such as TOS byte and Precedence byte and, most recently, DS-byte. *The key point is that in any one network, all network devices must treat this byte consistently.* This book will attempt some consistency by referring to this byte as the service type byte as much as possible, but reference to the DS-field or DS-byte should be treated as synonymous.

[1] See Appendix B, “The IP datagram header” on page 137

RFC 791 (and RFC 795) originally defined the first three bits in this field as denoting *precedence*, and the subsequent three bits as denoting *type of service.* RFC 1349 then redefined the service type byte so that four bits were used to denote *type of service*. Now the format of this byte according to this definition looks something like:

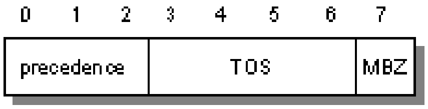


Figure 5. IP - service type

Starting with V3.1 code, IBM routers provide for:

1. The ability to set the precedence field for different types of SNA traffic being transported over IP
2. The ability for the Bandwidth Reservation System to honor the precedence bit settings

This function is provided *in addition to* normal BRS handling of IP traffic according to port numbers, and only applies when BRS cannot identify the port numbers, such as for IP traffic transported over an IP secure tunnel or in a secondary UDP or TCP fragment. This is important because - at this stage - it does not imply an alternative classification and differentiation method because the majority of frames will continue to be handled according to the port numbers contained in them.

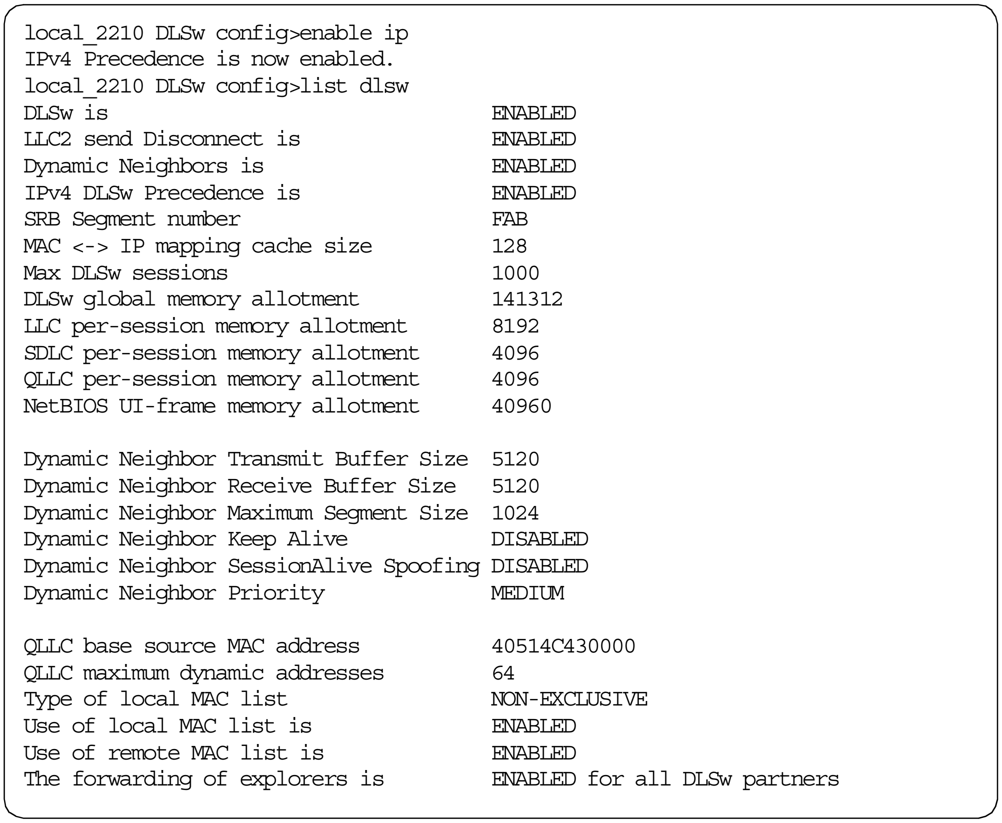
Although RFC 791 also defined the type of service (TOS) bits and gave each bit a specific meaning (to minimize delay/throughput/reliability/monetary cost) there is no support at this code level to set and/or interpret these bits. Later developments have effectively redefined the interpretation of both precedence and TOS bits, so as far as IBM routers are concerned the requirement to interpret the TOS bits according to RFC 791 is not of great significance.

When setting the precedence bits for SNA traffic, IBM routers take the values proposed in RFC 791 and map different types of SNA traffic to particular settings according to the following table:

Table 2. SNA traffic to IP precedence mapping

Precedence bits	RFC 791	SNA traffic category
111	Network control	
110	Internetwork control	HPR Network Transmission Priority
101	Critical	
100	Flash Override	HPR High Priority
011	Flash	DLSw, TN3270, HPR LLC exchanges
010	Immediate	HPR Medium Priority
001	Priority	HPR Low Priority
000	Routine	

Configuration of the use of the precedence bit is extremely simple: the function is enabled as part of the configuration of DLSw, TN3270, or HPR/IP. It is not possible to change the mappings shown in Table 2.



```
local_2210 DLSw config>enable ip
IPv4 Precedence is now enabled.
local_2210 DLSw config>list dlsw
DLSw is                                ENABLED
LLC2 send Disconnect is                ENABLED
Dynamic Neighbors is                   ENABLED
IPv4 DLSw Precedence is                ENABLED
SRB Segment number                     FAB
MAC <-> IP mapping cache size          128
Max DLSw sessions                      1000
DLSw global memory allotment           141312
LLC per-session memory allotment       8192
SDLC per-session memory allotment      4096
QLLC per-session memory allotment      4096
NetBIOS UI-frame memory allotment      40960

Dynamic Neighbor Transmit Buffer Size  5120
Dynamic Neighbor Receive Buffer Size   5120
Dynamic Neighbor Maximum Segment Size  1024
Dynamic Neighbor Keep Alive            DISABLED
Dynamic Neighbor SessionAlive Spoofing DISABLED
Dynamic Neighbor Priority              MEDIUM

QLLC base source MAC address           40514C430000
QLLC maximum dynamic addresses         64
Type of local MAC list                 NON-EXCLUSIVE
Use of local MAC list is               ENABLED
Use of remote MAC list is              ENABLED
The forwarding of explorers is         ENABLED for all DLSw partners
```

Figure 6. Enabling the use of IP precedence for DLSw

BRS will use the precedence bit settings to classify an IP packet if both:

- IPv4 precedence filtering is enabled for BRS using the ACTIVATE-IP-PRECEDENCE-FILTERING command
- The classification cannot be determined from the port number in the usual way because the packet is in a secure IP tunnel or is a secondary TCP or UDP fragment

BRS will classify these packets according to its classification rules for the native format of the frames determined from the mapping table in Table 2 on page 12. For example, an IP packet with a precedence value of 110 will be mapped to the same BRS class and priority as for HPR Network traffic. In the case of traffic with precedence bit settings 011, BRS will map the packet to the first filter from the following list that it has configured:

1. SNA/APPN-ISR
2. DLSw
3. Telnet

```
local_2210 Config>feat brs
Bandwidth Reservation User Configuration
local_2210 BRS Config>?
INTERFACE
LIST
ACTIVATE-IP-PRECEDENCE-FILTERING
DEACTIVATE-IP-PRECEDENCE-FILTERING
EXIT
local_2210 BRS Config>list
Bandwidth Reservation is available for 2 interfaces.

Interface   Type            State
---------   ----            -----
        1   FR              Enabled
        2   PPP             Enabled

BRS IP precedence filtering is activated.
The use of HPR over IP port numbers is disabled.
```

Figure 7. Enabling IP precedence filtering in BRS

In summary, V3.1 code provided an additional ability for identification and differentiation of SNA traffic when transported over an IP network in a secure IP tunnel or in which the TCP or UDP packets containing SNA data are fragmented.

This capability also offers value in a network in which other routers are making more general use of the precedence bit settings - these could be other IBM routers at a later code release or routers from other manufacturers. The ability to mark SNA packets by setting the precedence bits may make it easier for other routers to apply their own differentiation rules to these packets.

2.3 Full support for TOS through access controls and BRS

Interpretation and use of the service type byte in the IPv4 header has changed, and the latest proposal[2] proposes the replacement of the precedence and type of service fields shown in Figure 5 on page 12 with a six-bit Differentiated Services Code Point (DSCP), as shown in the following figure, in which CU denotes "currently unused":

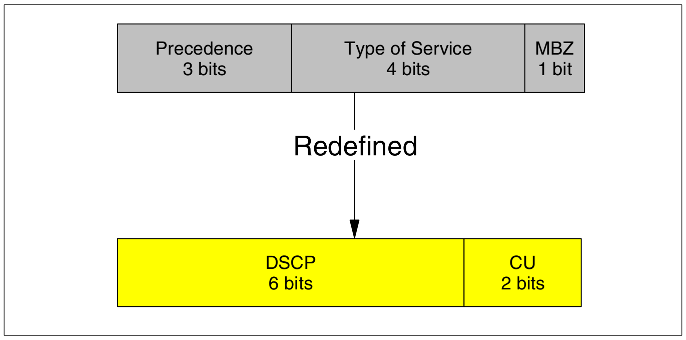


Figure 8. Redefinition of precedence and type of service bits

One significant reason for this change is that this field is now to be treated as an unstructured field; individual bits in the field no longer have meanings of their

[2] RFC 2474, Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers

own. The six bits of the DS Code Point field therefore imply 64 different interpretations, called *per-hop behaviors* (PHBs). A PHB is a description of the externally observable forwarding treatment applied at a Differentiated Services-compliant (DS-compliant) node to each IP packet.

To retain compatibility with existing implementations shown in Table 2 on page 12, DS Code Point values in the form xxx000 (which denotes any value for the first three bits and a value of zero for each of the last three bits) are referred to as *Class Selector Code Points*, and networks that are to be classified as DS-compliant are required to comply with the following rules:

1. There must be at least two different PHBs into which the eight different Class Selector code points are mapped. In other words, routers must have at least two different treatments for IP packets classified according to RFC 791 as shown in the first two columns of Table 2 on page 12.
2. Different PHBs resulting from different Class Selector code points should give packets a probability of timely forwarding that matches the definitions of RFC 791. In other words, if a packet with a lower numerical value of Class Selector code point gets mapped to a different PHB, this packet will be treated less preferentially and will more likely be discarded.
3. PHBs resulting from the two Class Selector code points, 111000 and 110000, must give these packets preferential forwarding treatment in comparison to the PHB resulting from code point 000000. This preserves the common usage of network control and internetwork control classifications in RFC 791 as having preferential treatment over routine traffic.

DS-compliant networks may chose only to implement Class Selector code points, which is little more than a different description of the implementation of priority differentiation according to RFC 791, but emerging standards are defining additional code points for additional traffic types. These can be thought of as an evolution of the TOS bit settings defined in RFC 791 because they extend the meaning of differentiation above simply that of a relative priority mechanism into other different types of treatment for different traffic types. Two RFCs[3] propose PHBs for:

- Expedited forwarding (EF), used to build a low loss, low latency, low jitter, assured bandwidth, end-to-end service through DS domains. Such a service appears to the endpoints like a point-to-point connection or a virtual leased line. This service has also been described as Premium service.
- Assured forwarding (AF), a mechanism for a service provider to offer different levels of forwarding assurances for IP packets received from a customer, in which IP packets are marked with different drop precedence values to indicate the relative order of packet discard during network congestion.
  The standard provides for the forwarding of IP packets in N independent AF classes. Within each AF class, an IP packet is assigned one of M different levels of drop precedence. An IP packet that belongs to an AF class i and has drop precedence j is marked with the AF code point AFij, where 1 <= i <= N and 1 <= j <= M. Currently, four classes (N=4) with three levels of drop precedence in each class (M=3) are defined for general use.

The V3.2 code release for the IBM router platform extends the abilities of routers to be able to:

[3] RFC 2598, An Expedited Forwarding PHB and RFC 2597, Assured Forwarding PHB Group

- Set any bits in the service type byte to any value
- Differentiate between IP packets based on any specified settings of the bits in the service type byte
- Override the normal IP routing table for specific packets by directing them to a manually selected next-hop gateway address - policy-based routing

Implementation of these abilities is shared between two components of the Common Code platform: access controls and Bandwidth Reservation System.

2.3.1 Access controls

IP access controls define rules for classification and processing of individual IP packets based on the following parameters:

- IP source address
- IP destination address
- IP protocol number
- TCP or UDP source port number
- TCP or UDP destination port number
- TCP SYN and ACK bits
- ICMP type and code
- Precedence and type of service filtering

Access controls are defined using a global access control list with two access control lists per interface. The interface access control lists define separate packet filters for incoming and outgoing packets. If all three lists are defined for a particular set of interfaces, a single packet will be processed three times according to the separate access control lists:

1. Access control list for inbound traffic on the receiving interface
2. Global access control list for the router
3. Access control list for outbound traffic on the transmitting interface

Access control rules applied to TOS/Precedence are simply bit mask operations that make no assumptions about the meanings of the bit settings themselves, and no significance should be attached to use of the terms TOS and Precedence when configuring access controls. This terminology is simply because of the original field definitions in RFC 791 (see Figure 5 on page 12).

All the examples given here show the definition of global access controls, but identical rules can be applied to individual packet filters as well.

2.3.1.1 Enabling access control

IP access control is enabled using the `SET ACCESS-CONTROL ON` command:

```
local_2210 Config>
local_2210 Config>PROTOCOL IP
Internet protocol user configuration
local_2210 IP config>SET ACCESS-CONTROL ON
local_2210 IP config>?

Possible completions:
 ...    LIST
 ...    CHANGE
 ...    DELETE
 ...    DISABLE
 ...    ENABLE
 ...    ADD
 ...    SET
 ...    MOVE
 ...    UPDATE
        EXIT
(you may cycle through these commands by pressing the TAB key)
local_2210 IP config>LIST ACCESS-CONTROLS
Access Control is: enabled
Access Control facility: USER

No access control records in configuration.
```

Figure 9. Enabling access control

2.3.1.2 Setting maximum throughput

Global access control rules are defined using the `ADD ACCESS-CONTROL` command. In this example we are assuming that other routers in the network are still treating the service type byte according to the rules proposed in RFC 791, and we want to set all packets destined for a particular IP address (9.24.104.193 in our example) to be marked for maximum throughput. This means we want to set the value 0100 in the TOS field defined in Figure 5 on page 12.

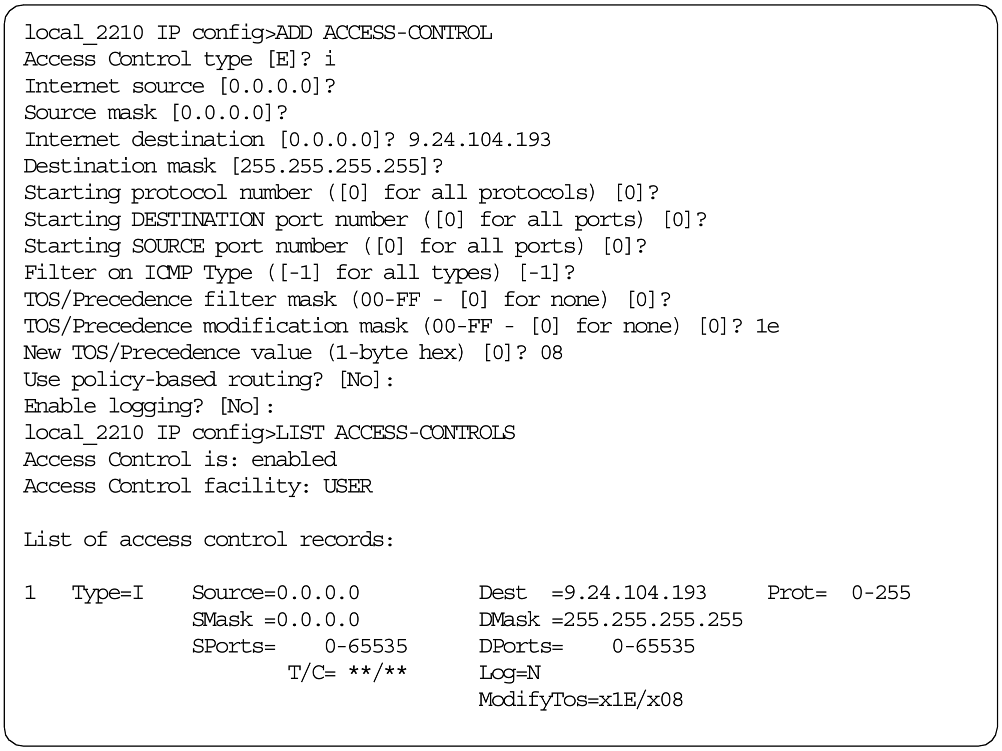
The effect of the following rule on the service type byte can be determined by comparing the modification mask and new precedence values as follows:

Table 3. Result of applying access control for maximum throughput

Field	Hex	Binary
TOS/Precedence modification mask	1E	00011110
New TOS/Precedence value	08	00001000
Resulting bit settings		uuu0100u

Note: The letter u denotes unchanged bit settings from the original settings.

The access control type here is an inclusive rule, which means that matching packets are processed further by the router; an exclusive rule would cause them to be discarded.



```
local_2210 IP config>ADD ACCESS-CONTROL
Access Control type [E]? i
Internet source [0.0.0.0]?
Source mask [0.0.0.0]?
Internet destination [0.0.0.0]? 9.24.104.193
Destination mask [255.255.255.255]?
Starting protocol number ([0] for all protocols) [0]?
Starting DESTINATION port number ([0] for all ports) [0]?
Starting SOURCE port number ([0] for all ports) [0]?
Filter on ICMP Type ([-1] for all types) [-1]?
TOS/Precedence filter mask (00-FF - [0] for none) [0]?
TOS/Precedence modification mask (00-FF - [0] for none) [0]? 1e
New TOS/Precedence value (1-byte hex) [0]? 08
Use policy-based routing? [No]:
Enable logging? [No]:
local_2210 IP config>LIST ACCESS-CONTROLS
Access Control is: enabled
Access Control facility: USER

List of access control records:

1   Type=I     Source=0.0.0.0         Dest  =9.24.104.193     Prot=  0-255
               SMask =0.0.0.0         DMask =255.255.255.255
               SPorts=    0-65535     DPorts=    0-65535
                       T/C= **/**     Log=N
                                      ModifyTos=x1E/x08
```

Figure 10. Setting maximum throughput using access control

2.3.1.3 Setting high priority

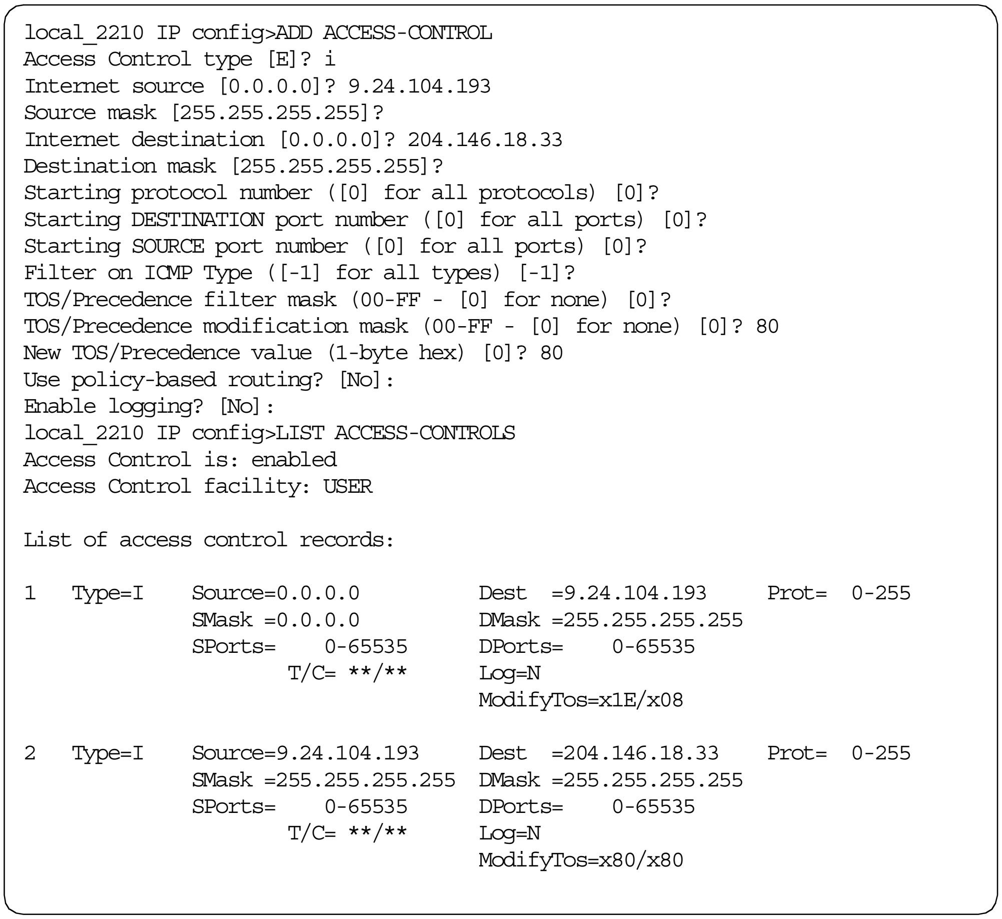
In this second example of access control usage the aim is to set the first bit of the service type byte to 1 for a particular traffic flow. This setting corresponds to setting the first precedence bit according to RFC 791, or to a high priority Class Selector code point according to the more recent definition of RFC 2474. In this particular case we will leave the remaining bits in the service type byte unchanged.

Table 4. Result of applying access control for high priority

Field	Hex	Binary
TOS/Precedence modification mask	80	10000000
New TOS/Precedence value	80	10000000
Resulting bit settings		1uuuuuuu

Note: The letter u denotes unchanged bit settings from the original settings.

In this particular example, the modification to the service type byte is only to be performed for packets originating from IP address 9.24.104.193 and destined for IP address 204.146.18.33.



```
local_2210 IP config>ADD ACCESS-CONTROL
Access Control type [E]? i
Internet source [0.0.0.0]? 9.24.104.193
Source mask [255.255.255.255]?
Internet destination [0.0.0.0]? 204.146.18.33
Destination mask [255.255.255.255]?
Starting protocol number ([0] for all protocols) [0]?
Starting DESTINATION port number ([0] for all ports) [0]?
Starting SOURCE port number ([0] for all ports) [0]?
Filter on ICMP Type ([-1] for all types) [-1]?
TOS/Precedence filter mask (00-FF - [0] for none) [0]?
TOS/Precedence modification mask (00-FF - [0] for none) [0]? 80
New TOS/Precedence value (1-byte hex) [0]? 80
Use policy-based routing? [No]:
Enable logging? [No]:
local_2210 IP config>LIST ACCESS-CONTROLS
Access Control is: enabled
Access Control facility: USER

List of access control records:

1   Type=I    Source=0.0.0.0         Dest  =9.24.104.193     Prot=  0-255
              SMask =0.0.0.0         DMask =255.255.255.255
              SPorts=    0-65535     DPorts=    0-65535
                      T/C= **/**     Log=N
                                     ModifyTos=x1E/x08

2   Type=I    Source=9.24.104.193    Dest  =204.146.18.33    Prot=  0-255
              SMask =255.255.255.255 DMask =255.255.255.255
              SPorts=    0-65535     DPorts=    0-65535
                      T/C= **/**     Log=N
                                     ModifyTos=x80/x80
```

Figure 11. Setting high priority using access control

2.3.1.4 Setting assured forwarding

The final example in this section demonstrates the use of access control to set bits to correspond to one of the latest proposed definitions for differentiated service types - assured forwarding. One reason for implementing such a mechanism would be that of a service provider in offering an “Olympic” service that comprises three service classes of bronze, silver, and gold. Packets within each class are assigned either low, medium or high drop preference. In this particular example we are defining a particular source/destination address pair as defining a flow that falls into the bronze class and with high drop preference[4].

Table 5. Result of applying access control for assured forwarding

Field	Hex	Binary
TOS/Precedence modification mask	fc	11111100
New TOS/Precedence value	38	00111000
Resulting bit settings		001110uu

Note: The letter u denotes unchanged bit settings from the original settings.

[4] Assured forwarding code point AF13 according to the draft standard

```
local_2210 IP config>ADD ACCESS-CONTROL
Access Control type [E]? i
Internet source [0.0.0.0]? 9.24.104.193
Source mask [255.255.255.255]?
Internet destination [0.0.0.0]? 192.31.7.130
Destination mask [255.255.255.255]?
Starting protocol number ([0] for all protocols) [0]?
Starting DESTINATION port number ([0] for all ports) [0]?
Starting SOURCE port number ([0] for all ports) [0]?
Filter on ICMP Type ([-1] for all types) [-1]?
TOS/Precedence filter mask (00-FF - [0] for none) [0]?
TOS/Precedence modification mask (00-FF - [0] for none) [0]? fc
New TOS/Precedence value (1-byte hex) [0]? 38
Use policy-based routing? [No]:
Enable logging? [No]:
local_2210 IP config>LIST ACCESS-CONTROLS
Access Control is: enabled
Access Control facility: USER

List of access control records:

1   Type=I    Source=0.0.0.0          Dest  =9.24.104.193     Prot=  0-255
              SMask =0.0.0.0          DMask =255.255.255.255
              SPorts=    0-65535      DPorts=    0-65535
                     T/C= **/**       Log=N
                                      ModifyTos=x1E/x08

2   Type=I    Source=9.24.104.193     Dest  =204.146.18.33    Prot=  0-255
              SMask =255.255.255.255  DMask =255.255.255.255
              SPorts=    0-65535      DPorts=    0-65535
                     T/C= **/**       Log=N
                                      ModifyTos=x80/x80

3   Type=I    Source=9.24.104.193     Dest  =192.31.7.130     Prot=  0-255
              SMask =255.255.255.255  DMask =255.255.255.255
              SPorts=    0-65535      DPorts=    0-65535
                     T/C= **/**       Log=N
                                      ModifyTos=xFC/x38
```

Figure 12. Setting assured forwarding using access control

2.3.2 Policy-based routing

Policy-based routing refers to another aspect of service differentiation that is not part of RFC 2474.

Routers maintain global routing tables that identify the interface and next-hop destination to be used for each packet that passes through the router. Although this table is usually modified in response to changes in network topology, it still defines a single best route, which is the same route for all packets destined to the same IP address.

Policy-based routing allows the addition of access control rules which modify the next-hop route for packets that conform to the specifications of the access control.

One important use of this will be in a network that uses the public Internet for transport of some of the traffic across the network. For example, a network may provide limited private capacity for important traffic and wish to ensure that other traffic be routed across the Internet.

In the following example policy-based routing has been combined with earlier examples to add an access control that:

- Defines a source/destination IP address pair 1
- Defines a destination port of 23 2
- Sets the DS-field for all matching packets to silver service class with low drop priority 3
- Sends all the matching packets to the next hop 9.37.3.60 rather than to the next hop indicated in the router's routing table 4
- Uses the actual routing table entry only if 9.37.3.60 is unreachable 5

```
local_2210 IP config>ADD ACCESS-CONTROL
Access Control type [E]? i
Internet source [0.0.0.0]? 9.24.104.193 1
Source mask [255.255.255.255]?
Internet destination [0.0.0.0]? 135.145.9.134 1
Destination mask [255.255.255.255]?
Starting protocol number ([0] for all protocols) [0]?
Starting DESTINATION port number ([0] for all ports) [0]? 23 2
Ending DESTINATION port number [23]?
Starting SOURCE port number ([0] for all ports) [0]?
Filter on ICMP Type ([-1] for all types) [-1]?
TOS/Precedence filter mask (00-FF - [0] for none) [0]?
TOS/Precedence modification mask (00-FF - [0] for none) [0]? fc 3
New TOS/Precedence value (1-byte hex) [0]? 48 3
Use policy-based routing? [No]: y
Next hop gateway address []? 9.37.3.60 4
Use default route if next hop gateway unreachable? [Yes]: 5
Enable logging? [No]:
local_2210 IP config>LIST ACCESS-CONTROLS
Access Control is: enabled
Access Control facility: USER

List of access control records:

1   Type=I    Source=0.0.0.0          Dest  =9.24.104.193     Prot=  0-255
              SMask =0.0.0.0          DMask =255.255.255.255
              SPorts=    0-65535      DPorts=    0-65535
                      T/C= **/**      Log=N
                                      ModifyTos=x1E/x08

2   Type=I    Source=9.24.104.193     Dest  =204.146.18.33    Prot=  0-255
              SMask =255.255.255.255  DMask =255.255.255.255
              SPorts=    0-65535      DPorts=    0-65535
                      T/C= **/**      Log=N
                                      ModifyTos=x80/x80

3   Type=I    Source=9.24.104.193     Dest  =192.31.7.130     Prot=  0-255
              SMask =255.255.255.255  DMask =255.255.255.255
              SPorts=    0-65535      DPorts=    0-65535
                      T/C= **/**      Log=N
                                      ModifyTos=xFC/x38

4   Type=I    Source=9.24.104.193     Dest  =135.145.9.134    Prot=  0-255
              SMask =255.255.255.255  DMask =255.255.255.255
              SPorts=    0-65535      DPorts=   23-23
                      T/C= **/**      Log=N
                                      ModifyTos=xFC/x48
              PbrGw=9.37.3.60         UseDefRte=Y
```

Figure 13. Implementing policy-based routing

Important note

If you attempt to configure access control in a similar manner, don't forget to include a default access control rule. For every list that includes at least one access control rule, an inclusive rule must exist for any packets that do not match any of the other access control rules. Failure to do this could effectively lead to the router being removed from the network, and recovery may only be possible from a terminal attached directly to the console port of the router.

Another way of defining access control rules to implement policy-based routing would be to route based on the *received* DS-field. In this case, access control rules are not being used to set DS-field values but are instead being used to make alternative routing decisions based on the values received. The following example shows the configuration commands required to route any bronze packets received to an alternative gateway regardless of source/destination IP address; the intent here is that this alternative gateway could in fact be a connection to the public Internet.

The reason for the choice of mask and filter values chosen is that the bronze service class is defined here as synonymous with Class 1 in the IETF draft proposal, in which all code points have the value 001 in the first three bits.

Table 6. Selecting packets based on DS-field bit settings

Field	Hex	Binary
TOS/Precedence filter mask	E0	11100000
TOS/Precedence start value	20	00100000
TOS/Precedence end value	20	00100000
DS-field value selected by this filter		001xxxxx

Note: The letter x denotes bit positions in the DS-field ignored by this filter.

```
local_2210 IP config>ADD ACCESS-CONTROL
Access Control type [E]? i
Internet source [0.0.0.0]?
Source mask [0.0.0.0]?
Internet destination [0.0.0.0]?
Destination mask [0.0.0.0]?
Starting protocol number ([0] for all protocols) [0]?
Starting DESTINATION port number ([0] for all ports) [0]?
Starting SOURCE port number ([0] for all ports) [0]?
Filter on ICMP Type ([-1] for all types) [-1]?
TOS/Precedence filter mask (00-FF - [0] for none) [0]? e0
TOS/Precedence start value (1-byte hex) [0]? 20
TOS/Precedence end value (1-byte hex) [20]?
TOS/Precedence modification mask (00-FF - [0] for none) [0]?
Use policy-based routing? [No]: y
Next hop gateway address []? 9.37.3.60
Use default route if next hop gateway unreachable? [Yes]:
Enable logging? [No]:
local_2210 IP config>LIST ACCESS-CONTROLS
Access Control is: enabled
Access Control facility: USER

List of access control records:

1   Type=I    Source=0.0.0.0          Dest  =9.24.104.193     Prot=  0-255
              SMask =0.0.0.0          DMask =255.255.255.255
              SPorts=    0-65535      DPorts=    0-65535
                      T/C= **/**      Log=N
                                      ModifyTos=x1E/x08

2   Type=I    Source=9.24.104.193     Dest  =204.146.18.33    Prot=  0-255
              SMask =255.255.255.255  DMask =255.255.255.255
              SPorts=    0-65535      DPorts=    0-65535
                      T/C= **/**      Log=N
                                      ModifyTos=x80/x80

3   Type=I    Source=9.24.104.193     Dest  =192.31.7.130     Prot=  0-255
              SMask =255.255.255.255  DMask =255.255.255.255
              SPorts=    0-65535      DPorts=    0-65535
                      T/C= **/**      Log=N
                                      ModifyTos=xFC/x38

4   Type=I    Source=9.24.104.193     Dest  =135.145.9.134    Prot=  0-255
              SMask =255.255.255.255  DMask =255.255.255.255
              SPorts=    0-65535      DPorts=   23-23
                      T/C= **/**      Log=N
                                      ModifyTos=xFC/x48
              PbrGw=9.37.3.60         UseDefRte=Y

5   Type=I    Source=0.0.0.0          Dest  =0.0.0.0          Prot=  0-255
              SMask =0.0.0.0          DMask =0.0.0.0
              SPorts=    0-65535      DPorts=    0-65535
                      T/C= **/**      Log=N

6   Type=I    Source=0.0.0.0          Dest  =0.0.0.0          Prot=  0-255
              SMask =0.0.0.0          DMask =0.0.0.0
              SPorts=    0-65535      DPorts=    0-65535
                      T/C= **/**      Log=N
              Tos=xE0/x20-x20
              PbrGw=9.37.3.60         UseDefRte=Y
```

Figure 14. Policy-based routing based on DS-field settings

The example shown in Figure 14 also shows that a default access control rule has been added (record number 5). As shown here, since the default rule is positioned higher in the list of access control rules than the newly added rule, the new access control rule will not take effect until the list is reordered, as shown in Figure 15 on page 25.

The configuration steps to add the default access control rule have not been shown here; all that is required is the addition of an inclusive rule that is applicable to all packets received and which does not indicate any special action to be taken. But this step should never be overlooked; the absence of such a rule will cause any packets that do not match any of the other access control rules to be discarded.

```
local_2210 IP config>MOVE ACCESS-CONTROL
Index of control to move [1]? 5
Move record AFTER record number [0]? 6
About to move:

5   Type=I    Source=0.0.0.0          Dest  =0.0.0.0          Prot=  0-255
              SMask =0.0.0.0          DMask =0.0.0.0
              SPorts=    0-65535      DPorts=    0-65535
                      T/C= **/**      Log=N
to be after:

6   Type=I    Source=0.0.0.0          Dest  =0.0.0.0          Prot=  0-255
              SMask =0.0.0.0          DMask =0.0.0.0
              SPorts=    0-65535      DPorts=    0-65535
                      T/C= **/**      Log=N
              Tos=x60/x20-x20
              PbrGw=9.37.3.60         UseDefRte=Y
Are you sure this is what you want to do(Yes or [No]): y
local_2210 IP config>LIST ACCESS-CONTROLS
Access Control is: enabled
Access Control facility: USER

List of access control records:

1   Type=I    Source=0.0.0.0          Dest  =9.24.104.193     Prot=  0-255
              SMask =0.0.0.0          DMask =255.255.255.255
              SPorts=    0-65535      DPorts=    0-65535
                      T/C= **/**      Log=N
                                      ModifyTos=x1E/x08

2   Type=I    Source=9.24.104.193     Dest  =204.146.18.33    Prot=  0-255
              SMask =255.255.255.255  DMask =255.255.255.255
              SPorts=    0-65535      DPorts=    0-65535
                      T/C= **/**      Log=N
                                      ModifyTos=x80/x80

3   Type=I    Source=9.24.104.193     Dest  =192.31.7.130     Prot=  0-255
              SMask =255.255.255.255  DMask =255.255.255.255
              SPorts=    0-65535      DPorts=    0-65535
                      T/C= **/**      Log=N
                                      ModifyTos=xFC/x38

4   Type=I    Source=9.24.104.193     Dest  =135.145.9.134    Prot=  0-255
              SMask =255.255.255.255  DMask =255.255.255.255
              SPorts=    0-65535      DPorts=   23-23
                      T/C= **/**      Log=N
                                      ModifyTos=xFC/x48
              PbrGw=9.37.3.60         UseDefRte=Y

5   Type=I    Source=0.0.0.0          Dest  =0.0.0.0          Prot=  0-255
              SMask =0.0.0.0          DMask =0.0.0.0
              SPorts=    0-65535      DPorts=    0-65535
                      T/C= **/**      Log=N
              Tos=xE0/x20-x20
              PbrGw=9.37.3.60         UseDefRte=Y

6   Type=I    Source=0.0.0.0          Dest  =0.0.0.0          Prot=  0-255
              SMask =0.0.0.0          DMask =0.0.0.0
              SPorts=    0-65535      DPorts=    0-65535
                      T/C= **/**      Log=N
```

Figure 15. Reordering the access control list

2.3.3 Bandwidth Reservation System

The second major component in the V3.2 code release related to Differentiated Services is the ability of the Bandwidth Reservation System (BRS) to honor the service type bit settings.

This enhancement differs from the limited provision to use certain bit settings to denote certain types of SNA traffic in particular circumstances discussed in “Precedence and type of service” on page 11. BRS is now capable of defining filters that categorize traffic based on any setting of the service type byte and to associate this traffic to a particular BRS traffic class and priority.

This filtering capability allows BRS to perform filtering for all traffic sent over a secure tunnel, traffic that is fragmented, or traffic that cannot be identified using the BRS UDP and TCP port number filter support. It allows the TOS bits to be set to any user-defined values rather than having to use the hard-coded values defined for APPN and DLSw. The order in which BRS filters are evaluated puts TOS filters at the top of the list, in other words, if these filters are defined then they will take precedence over any other types of BRS filters.

The following example shows the BRS configuration to categorize the same bronze traffic used in the previous example. This time BRS is being used to assign the traffic to the default traffic class with low priority.

This configuration could well be used in conjunction with the previous example; access control is used to modify where the bronze packets are next sent whereas BRS is used to prioritize transmission on the chosen link.

Note

The sharp-eyed will notice one other difference between Figure 16 on page 27 and Figure 4 on page 11. Figure 16 now shows an entry for "protocol VOFR". This refers to voice over frame relay and has been introduced with V3.3 code, which was the code level being used when capturing the configuration screens for this figure.

BRS also provides an additional class - the superclass. This is a traffic class that can be used for time-critical traffic such as voice traffic; it does not have an associated bandwidth reservation but instead always takes precedence over all other traffic classes. Use the `CREATE-SUPER-CLASS` command to create and name a superclass and then assign traffic to this class in the normal manner. The ability to define such as superclass was introduced in the V3.3 code release.

```
local_2210 BRS [i 1] [dlci 531]>ASSIGN
Protocol or filter name [IP]? tos1
Class name [DEFAULT]?
Priority <URGENT/HIGH/NORMAL/LOW> [NORMAL]? LOW
Frame Relay Discard Eligible <NO/YES> [NO]?
TOS Mask [1-FF] [FF]? e0
TOS Range (Low) [0-FF] [0]? 20
TOS Range (High) [20]?
local_2210 BRS [i 1] [dlci 531]>LIST all

BANDWIDTH RESERVATION listing from SRAM
bandwidth reservation is enabled
interface number 1, circuit number 531
maximum queue length 10, minimum queue length 3
total bandwidth allocated 80%
total traffic classes defined (counting one local and one default) 4


class LOCAL has 10% bandwidth allocated
  protocols and filters cannot be assigned to this class.


class DEFAULT has 40% bandwidth allocated
  the following protocols and filters are assigned:
    protocol IP with priority NORMAL is not discard eligible
    protocol ARP with default priority is not discard eligible
    protocol VOFR with default priority is not discard eligible
    protocol ASRT with default priority is not discard eligible
    protocol APPN-HPR with default priority is not discard eligible
    protocol SNA/APPN-ISR with default priority is not discard eligible
    filter TELNET-IP with priority HIGH is not discard eligible
    filter TOS1 with priority LOW is not discard eligible
       with TOS range x20 - x20 and TOS mask xE0


class A has 10% bandwidth allocated
  the following protocols and filters are assigned:
    filter UDP_TCP1 with priority NORMAL is not discard eligible
       and represents UDP with port range 25 - 29
       and IP address 0.0.0.0


class B has 20% bandwidth allocated
  the following protocols and filters are assigned:
    filter DLSW-IP with priority HIGH is not discard eligible
    filter UDP_TCP2 with priority URGENT is not discard eligible
       and represents TCP with port range 50 - 50
       and IP address 5.5.5.25

assigned tags:

default class is DEFAULT with priority NORMAL
```

Figure 16. Assigning bronze traffic to a BRS traffic class

2.3.4 Summary

The combination of BRS and access control at the V3.2 level of code allows for comprehensive classification of IP packets according to different access control rules and for incorporation of these packets into the existing BRS priority queueing mechanism. It delivers the ability to:

- Filter IP packets based on the service type byte
- Modify the service type byte
- Select routes based on the service type byte
- Assign traffic to BRS classes based on the service type byte

One example of its use would be in an environment in which different data packets were classified as being either gold, silver, or bronze, with bronze packets being routed across the public Internet and the other packets being prioritized appropriately over the same private links.

It makes no statement about *how* different data packets are to be treated, simply that some relative priority can be assigned to each of them and that those with higher priority are less likely to be discarded when there is congestion in the network.

It allows the construction of a data network in which many different types of data flow can be handled at once, with the assumption that those flows categorized as less important will only be transported if there is sufficient capacity in the network. This type of network has been suitable for mission-critical data transport for many years - SNA networks have delivered just this for a long time now - but the challenges of incorporating the transport of different types of voice traffic (over IP as well as over frame relay), video, and other types of traffic mean that this solution is not sufficiently advanced for some of these types of network.

To put the last paragraph another way: had the capabilities now available through BRS and access control features been available some years ago for IP networks, many of the problems encountered with these networks and in particular with the migration of SNA networks onto IP backbones would not have been seen. For today's increasing complex networks, however, even this level of sophistication is no longer always adequate.

2.4 Directories and policies

The latest release of router code, V3.3, has introduced a more generic policy management process in which policies are defined as to how IP traffic is to be managed in a network. Policies may be as simple as filter rules (such as whether to drop or pass a packet) or may extend into complexities of security and Class of Service.

2.4.1 Quis custodet ipsos custodes?

This Latin term translates as "who will guard the guardians?". One of the questions which has to be asked when contemplating a network in which different types of traffic are differentiated and given different treatments is: where in the network are Class of Service determinations made? This is to say, who initially makes a decision to give a particular IP packet a particular CoS classification?

On the face of it, the simplest place for this decision is the end user or user application. This allows users to transmit packets into the network and for the network to treat each of them according to the classification transmitted by the user. In reality, however, each user will tend to think of his or her traffic as being more important than that of other users of the network. Similarly, every application programmer will tend to treat his or her application program with inappropriate importance in the context of all application programs using the network. Ultimately this leads to a situation in which all packets are marked with this highest priority and therefore, the object of Differentiated Services is defeated because every packet now ends up being treated in the same way.

So a second approach is for the priority signaled by the end users in the IP packets that they themselves originated to be ignored completely. As we have already seen, we can construct access control rules for the routers at the periphery of networks (the routers to which the users themselves attach). We need to recognize that all IP endstations are capable of setting any combination of bits in the DS-field when they originate IP packets, and we must therefore define access control rules to override the original user values and substitute values appropriate to the network as a whole.

The problem with the second approach is that although it has the merit of handing control of the network back to the people who run the network, it has traditionally been implemented by configuring each router in the network separately. Although it has been relatively easy to “clone” router configurations (making the configuration of an additional router a matter of copying an existing configuration and modifying a small number of parameters), it is not a good approach in which the requirements of the network are changing frequently.

What is required is a centralized method of implementing rules across the network. This is already available to some extent using network management stations and the Simple Network Management Protocol (SNMP), but in reality even this approach is too cumbersome for our purposes. It may indeed be possible to store all router configurations in one central location and may also be possible to transmit updated configuration information across the network to the individual routers, but it still requires that multiple separate configuration files are managed centrally (with no consistency between them) and may also lead to the practical disadvantage of requiring a complete router reload in order to implement changes.

The approach which has been taken with the latest release of code is consistent with the industry approach: it is not router configurations themselves that are stored centrally but the policies themselves. The policies are stored in a database which is a *lightweight* version of the X.500 database, and Lightweight Directory Access Protocol (LDAP) is used by routers in the network to retrieve policy information from the server.

This means that each router maintains a policy database in its memory, but the policy database comprises the combination of policies defined locally in the router and those that have been read from the server using LDAP. SNMP can still be used to control this environment, but now it does so by allowing the central management station to send a command to a network router which instructs the router to reload its policy database.

V3.4

The V3.4 code level allows a 2212 or 2216 router to store a copy of the information retrieved from the LDAP server on its hard disk and to use that information if the LDAP server is not available for any reason.

The policy feature can initially be viewed as a replacement for classification of packets using access control, but it is part of a much larger picture. It is required for implementation of the Differentiated Services (DiffServ) feature.

2.4.2 A centralized approach

The following figure compares the old approach of configuring each router in the network separately with the newer one providing all the network policy information in a centralized repository. Each router maintains its own policy database when it is operating, but the input to this policy database is information stored in a centralized repository rather than being preconfigured in the routers themselves. Hence if a subset of the routers in the network requires the implementation of identical policy rules, these rules are defined once in the central server and retrieved separately by each of the routers.

Control of the implementation of these policies can be accomplished by a combination of:

- Having the router read policy rules from a centralized server when the router initializes
- Using SNMP to instruct the router to reload its policy database after configuration changes have been made to the centralized rules
- Configuring the routers to reload their policy databases on a regular basis, for example, at midnight each night

All the practical examples shown in this chapter are of policy rules being configured locally on each router. However, the drive behind application-driven networking in a realistic implementation of the same examples would be to have the same rules defined on the centralized policy server and to have the routers retrieve the rules from the server rather than forming part of the local configuration information. Nonetheless, regardless of which method is used to provide policy rules to the routers, the actual interpretation and implementation of the rules is identical in both cases.

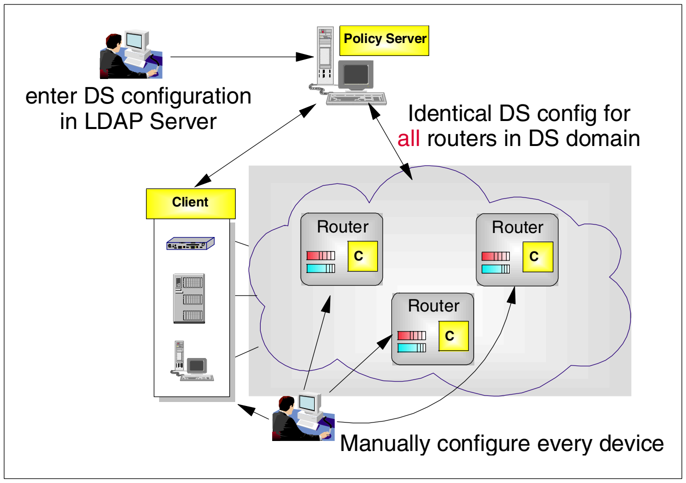

Policy Server
enter DS configuration in LDAP Server
Identical DS config for all routers in DS domain
Client
Router
C
Router
C
Router
C
Manually configure every device

Figure 17. Policies and LDAP

For more information on the architectural model behind IBM's implementation, see:

- *Application-Driven Networking: Concepts and Architecture for Policy-based Systems,* SG24-5640.

2.4.3 Policy database entries and access control

Reference to the policy database is made *after* access control rules have been applied. Note that the policy feature is not specifically enabled, it is automatically active in all routers at the appropriate level of code (Version 3, Release 3 and later); the default configuration (shown in Figure 19 on page 32) has no policy entries and therefore no actions configured, meaning that the default policy feature configuration has no effect on the operation of the router.

The policy database resides in the memory of the routers and comprises the set of policies loaded from the preconfigured local database and the policies retrieved from the LDAP server. The policy database serves as the policy decision point (PDP).

Care needs to be taken if both access control rules and policy database rules cause packets to be modified: for example, packets in which the DS-field has been modified by global access control rules will then be presented for comparison with the policy database, and therefore, the policy database rules should take this into account. It may happen that packets which have been modified by the application of access control rules will subsequently be modified again by the actions resulting from policy database rules.

Access controls are still required for the implementation of what was perhaps unfortunately named policy-based routing (see 2.3.2, "Policy-based routing" on page 20). Figure 28 on page 39 shows that the policy database is only examined once the decision on which outbound port is to be used has been made. If routing decisions are to be made based on the contents of a packet, the implementation of these decisions remains part of the use of access control.

The following figure shows the flow of IP packets through an IBM router and shows that the policy database is only queried once outbound packets have passed (and perhaps been modified) by the output packet filter. The terms input filter and output filter here refer to the packet filters defined for each interface using the `add packet-filter` command and configured using the `update packet-filter` command. IP access controls which are documented in 2.3.1, "Access controls" on page 16 show the implementation of global access control rules which apply to the IP forwarding engine, but access controls can also be added to specific packet filters.

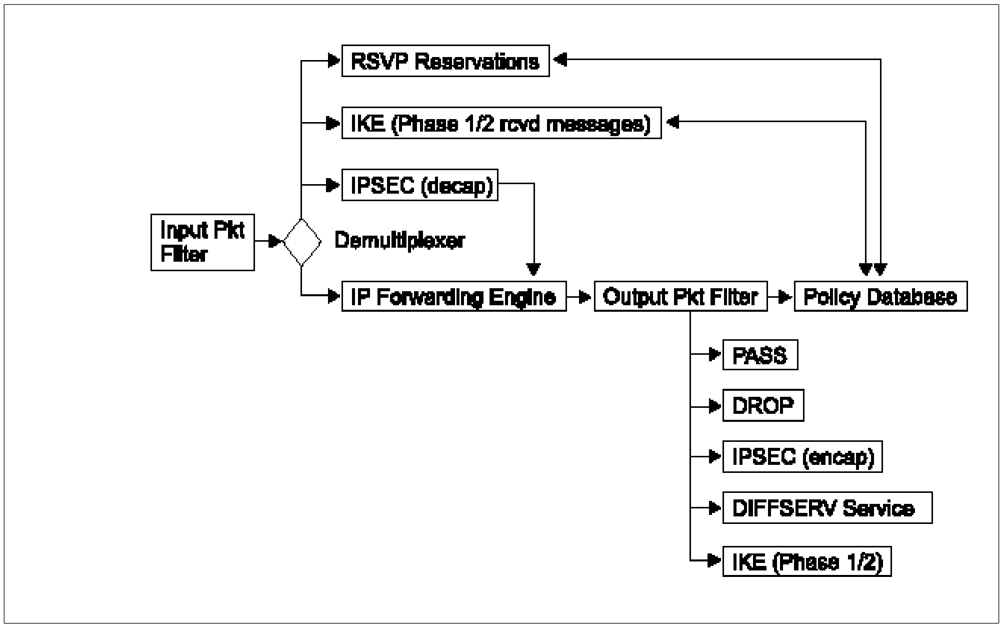


Figure 18. IP packet flow and the policy database

2.4.4 Configuring a local policy database entry

This example configuration shows the steps required to create an entry in a router's local policy database which would have been configured using access control in previous code releases. The configuration steps are more complex because the policy database is more granular than access control rules and more fine-tuning of the network can be achieved using policies, by setting different policy rules for different times of day and night, for example.

The first step in the process is to add a new policy entry:

```
remote router *TALK 6

remote router Policy config>EXIT
remote router Config>FEATURE Policy
IP Network Policy configuration
remote router Policy config>LIST ALL

Configured Policies....
No Policies configured

Configured Profiles....
No Profiles configured

Configured Validity Periods
No Policy Valid Periods configured

Configured DiffServ Actions....
No DiffServ Actions configured
remote router Policy config>ADD POLICY
Enter a Name (1-29 characters) for this Policy []? JonathanTest
Enter the priority of this policy (This number is used to
determine the policy to enforce in the event of policy conflicts) [5]?
```

Figure 19. Addition of a new policy entry

Next, a profile has to be associated with this policy entry, which identifies packets to be controlled by the policy by a combination of source/destination IP addresses and the contents of the DS-field (in this case, the same bronze packets selected by the access control rule in Figure 14 on page 23).

```
List of Profiles:
        0: New Profile

Enter number of the profile for this policy [0]? 0
Profile Configuration questions.  Note for Security Policies, the Source
Address and Port Configuration parameters refer to the Local Client Proxy
and the Destination Address and Port Configuration parameters refer to the
Remote Client Proxy
Enter a Name (1-29 characters) for this Profile []? JonathanTest
Source Address Format (1:NetMask, 2:Range, 3:Single Addr) [1]? 3
Enter IPV4 Source Address [0.0.0.0]? 9.24.104.193
Destination Address Format (1:NetMask, 2:Range, 3:Single Addr) [1]? 1
Enter IPV4 Destination Address [0.0.0.0]? 9.0.0.0
Enter IPV4 Destination Mask [255.0.0.0]?

Protocol IDs:
     1)  TCP
     2)  UDP
     3)  All Protocols
     4)  Specify Range

Select the protocol to filter on (1-4) [3]? 3
Enter the Starting value for the Source Port [0]?
Enter the Ending value for the Source Port [65535]?
Enter the Starting value for the Destination Port [0]?
Enter the Ending value for the Destination Port [65535]?
Enter the Mask to be applied to the Received DS-byte [0]? E0
Enter the value to match against after the Mask has
been applied to the Received DS-byte [0]? 20
Limit this profile to specific interface(s)? [No]:
```

Figure 20. Creation of a new profile entry

Now the new profile entry is associated with the new policy entry:

```
Here is the Profile you specified...


Profile Name    = JonathanTest
        sAddr     =   9.24.104.193 :  sPort=    0 : 65535
        dAddr:Mask=        9.0.0.0 : 255.0.0.0      dPort=    0 : 65535
        proto     =              0 : 255
        TOS       =            xE0 : x20
Is this correct? [Yes]:
List of Profiles:
        0: New Profile
        1: JonathanTest

Enter number of the profile for this policy [1]? 1
```

Figure 21. Association of profile entry with new policy entry

A validity period for the policy now needs to be defined; this is a significant additional capability because it allows the definition of different policies for different times of day which can all be stored in the same policy database. In this case the policy is defined as being effective all the time:

```
List of Validity Periods:
        0: New Validity Period

Enter number of the validity period for this policy [0]?
Enter a Name (1-29 characters) for this Policy Valid Profile []? AllTheTime
Enter the lifetime of this policy. Please input the
information in the following format:
                yyyymmddhhmmss:yyyymmddhhmmss OR '*' denotes forever.
 [*]? *
During which months should policies containing this profile
be valid.  Please input any sequence of months by typing in
the first three letters of each month with a space in between
each entry, or type ALL to signify year round.
 [ALL]?
During which days should policies containing this profile
be valid.  Please input any sequence of days by typing in
the first three letters of each day with a space in between
each entry, or type ALL to signify all week
 [ALL]?
Enter the starting time (hh:mm:ss or * denotes all day)
 [*]?
```

Figure 22. Definition of a new validity period

As before, this new validity period is next associated with the new policy definition:

```
Here is the Policy Validity Profile you specified...


Validity Name   = AllTheTime
        Duration  = Forever
        Months    = ALL
        Days      = ALL
        Hours     = All Day
Is this correct? [Yes]:
List of Validity Periods:
        0: New Validity Period
        1: AllTheTime

Enter number of the validity period for this policy [1]? 1
```

Figure 23. Association of new validity period with new policy

In the case of this particular policy definition, we now want to configure a DiffServ action to be associated with this policy. For now, consider this action as the combined ability to specify the output queue for this packet and the ability to modify the DS-byte; in this case we are going to turn the high-order bit on in the same way as was accomplished using access control in Figure 11 on page 19.

```
Do you wish to Map a DiffServ Action to this Policy? [No]: y
DiffServ Actions:
        0: New DiffServ Action

Enter the Number of the DiffServ Action [0]?
Enter a Name (1-29 characters) for this DiffServ Action []? JonathanTest
Enter the permission level for packets matching this DiffServ
Action (1. Permit, 2. Deny) [2]? 1
List of DiffServ Queues:
   1) Premium
   2) Assured/BE
Enter the Queue Number(1-2) for outgoing packets matching
this DiffServ Action [2]? 2
How do you want to specify the bandwidth allocated to this service?
Enter absolute kbps(1) or percentage of output bandwidth(2) [2]? 2
Enter the percentage of output bandwidth allocated to this service [10]? 5

Transmitted DS-byte mask [0]? 80
Transmitted DS-byte modify value [0]? 80
```

Figure 24. Specifying DiffServ action

V3.4

V3.4 code offers preconfigured DiffServ policies: EF, AF11, AF21, AF31, AF41. All of these pre-configured policies *modify* the DS-byte according to the code points described in the draft standard and allocate percentages of the output bandwidth (EF - 19%, AF11 - 15%, AF21 - 10%, AF31 - 10%, AF41 - 5%). If these defaults are not suitable, the user still has the option to configure his or her own policy as before, and this approach should be used if the policy is required not to modify the DS-byte in any way.

Again, just like before, this new DiffServ action has to be associated with the new policy:

```
Here is the DiffServ Action you specified...


DiffServ Name   = JonathanTest                   Type =Permit
        TOS mask:modify=x80:x80
        Queue:BwShare  =Assured    : 5 %
Is this correct? [Yes]:
DiffServ Actions:
        0: New DiffServ Action
        1: JonathanTest

Enter the Number of the DiffServ Action [1]? 1
Policy Enabled/Disabled (1. Enabled, 2. Disabled) [1]?
```

Figure 25. Associating the new DiffServ action with the new policy

The final step in the configuration process is to confirm that all is correct:

```
Here is the Policy you specified...


Policy Name     = JonathanTest
        State:Priority =Enabled    : 5
        Profile        =JonathanTest
        Valid Period   =AllTheTime
        DiffServ Action=JonathanTest
Is this correct? [Yes]:
You must enable and configure DiffServ in feature DS before
QOS can be ensured for this policy
remote router Policy config>
```

Figure 26. Confirmation of the new policy definition

As a final confirmation, the output from the `LIST ALL` command is now shown:

```
remote router Policy config>LIST ALL

Configured Policies....

Policy Name     = JonathanTest
        State:Priority =Enabled    : 5
        Profile        =JonathanTest
        Valid Period   =AllTheTime
        DiffServ Action=JonathanTest

Configured Profiles....

Profile Name    = JonathanTest
        sAddr     =   9.24.104.193 :  sPort=    0 : 65535
        dAddr:Mask=        9.0.0.0 : 255.0.0.0       dPort=    0 : 65535
        proto     =              0 : 255
        TOS       =            xE0 : x20

Configured Validity Periods

Validity Name   = AllTheTime
        Duration  = Forever
        Months    = ALL
        Days      = ALL
        Hours     = All Day

Configured DiffServ Actions....

DiffServ Name   = JonathanTest                  Type =Permit
        TOS mask:modify=x80:x80
        Queue:BwShare  =Assured    : 5 %
```

Figure 27. Listing all configured policies

The use of policies allows the definition of consistent actions across a network, especially in combination with the centralized storage and retrieval of policy information from an LDAP server. Policy definitions are a much more powerful method of differentiating between different types of traffic than were available before and - in the context of Class of Service - have powerful interactions with DiffServ and RSVP functions in the router.

2.5 The Differentiated Services feature

The Differentiated Services (DiffServ) feature is provided with the V3.3 release of router code. It builds on many of the concepts of RFC 2474 and of the current IETF draft documents: it is in fact implementing a *premium* router queue for traffic defined by the expedited forwarding PHB. An obvious application for this is in the transport of traffic such as voice traffic (voice traffic transported over IP in fact, because DiffServ is only applicable to IP traffic).

DiffServ is implemented on frame relay and PPP links and can only be used instead of BRS; the two cannot be enabled at the same time on the same interface. Watch out if you are configuring using the command line, because no warning is issued if BRS and DiffServ are both configured for the same interface; the actual result is that DiffServ will always be disabled and BRS will operate as configured.

V3.4

V3.4 code introduces support for DiffServ on multilink PPP interfaces, and this is very important if voice traffic is to be transported over low bandwidth PPP links.

DiffServ requires policies to be defined in the policy database in order to classify traffic and, in particular, to be able to identify the traffic which should be handled by the premium queue and the traffic which should be handled by the assured forwarding queue. Traffic assigned to this premium queue is guaranteed bandwidth and low delays, and by default the premium queue is assigned 20% of the bandwidth, of which 95% can be allocated to specific streams or flows by the policy database. In the default case, this means that the sum of the bandwidth requirements of policies which are to use the premium queue must not exceed 19% of the output bandwidth. If insufficient bandwidth remains on the premium queue, traffic streams which request more than the available bandwidth will be handled as best effort traffic. By default, best effort traffic is given a minimum output bandwidth percentage of 10% and control traffic is given 5%.

Table 7 shows how the default bandwidth allocations lead to slightly under 50% of the output bandwidth actually being available for allocation by policies which define the use of the assured forwarding/best effort (AF/BE) queue. A shared buffer pool is defined as 20% of this queue by default; the shared queue is available for use by any streams but will not be assigned to any specific streams. The actual percentages of the total queue available for explicit allocation are shown under the heading *MaxQos %* in Figure 30 on page 40.

Table 7. Default DiffServ bandwidth allocation

Default	Queue	Allocation	Percentage of output bandwidth
20%	Expedited Forwarding	QoS Allocation (default 95% of EF)	19%
		Shared Buffers	1%

Default	Queue	Allocation		Percentage of output bandwidth
80%	Assured Forwarding / Best Effort	Shared Buffers		16%
		QoS Allocation (default 80% of AF/BE)	Control Traffic	5%
			Best Effort Traffic	10%
			Assured Forwarding Traffic	49%

Traffic flows are aggregated into streams by use of the policy database and information is stored in the DiffServ cache. Resources allocated to specific streams will be returned to the system when streams become idle - when no packets have been sent on a stream for some time.

Traffic requiring expedited forwarding (EF) will be *policed* to ensure that it does not exceed the configured rate; excess traffic will be dropped before being put on the priority queue. The policing uses a l*eaky bucket* algorithm to allow short bursts of traffic but also to ensure that the average delivery rate over time does not exceed the configured rate. Assured forwarding (AF) and best effort (BE) traffic can be sent in excess of the configured rate and will be forwarded if idle resources exist.

V3.4

V3.4 allows customization of the parameters controlling the leaky bucket: both the rate at which tokens are inserted into the bucket and the depth of the bucket itself can be modified. This is of importance when dealing with voice traffic over IP over low bandwidth PPP links using RTP header compression.

V3.4 also allows policing of the AF/BE queue, in which packets in excess of the rate configured in the DiffServ policy are marked as red or yellow packets by setting medium or high drop precedence. Packets with low drop precedence are therefore referred to as green packets. This marking allows subsequent DiffServ routers to drop red and yellow packets in preference to dropping green packets under congestion conditions.

The DiffServ scheduler takes traffic from both queues (EF and AF/BE); unlike BRS it uses a self-clocked fair queuing algorithm which is a variant of weighted fair queueing. By default, the weights for the two queues are set to 90% for the premium queue and to 10% for the AF/BE queue; this means that the scheduler checks the two queues in this ratio. This ratio is a means of delivering low delay to traffic on the premium queue; since traffic destined for this queue is policed to ensure that it does not exceed its defined rate it also means that this ratio will not lead to excess EF traffic preventing the transmission of traffic from the AF/BE queue.

All these components are shown in the following diagram:

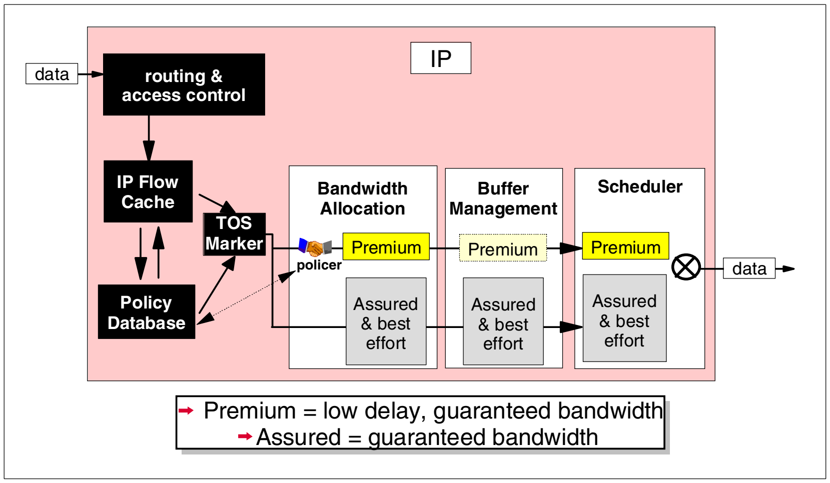


Figure 28. Components of Diffserv

The key difference between EF and AF traffic is that although both are guaranteed bandwidth, only EF traffic is guaranteed low delays as well.

The key difference between AF and BE traffic is that the former is identified as a stream and is allocated a fixed amount of the AF/BE bandwidth by entries in the policy database. Traffic for which the policy database does not specify any DiffServ action and traffic for which no match is found in the policy database both get treated as best effort which means there is no guarantee of bandwidth or low delay.

2.5.1 Implementation of DiffServ

Most of the effort in implementing DiffServ is not through configuring the DiffServ feature in the router. All that needs to be done with this feature is that DiffServ needs to be enabled globally and then on each desired interface.

The following example shows DiffServ enabled on a frame relay interface with all the default parameters being taken. If any of the default parameters are modified, consider the following points:

- The premium queue size should not be made too large, because this could cause a high queueing delay, nor too small, because this would make it impossible to buffer small bursts. For example, 25 kB (25,000 bytes or 200,000 bits) means a potential queueing delay of 133 milliseconds and 2 kB means the inability to buffer a 2-packet burst of 1500-byte packets.
- The assured queue size should simply be large enough to cater for simultaneous bursts from several flows although it should also take into account the actual amount of memory available in the router.
- Allow appropriate bandwidth for best effort traffic - the default of 10% (1 below) may not be sufficient if a lot of traffic falls into this category.

```
remote router Config>FEATURE DS
Differential Services Config
remote router DS Config>ENABLE DS
DiffServ enabled
remote router DS Config>LIST ALL

System Parameters:

        DiffServ:                ENABLED
        Packet_size:             550
        Min BE Alloc (%):        10 1
        Min CTL Alloc (%):       5
        Number_of_Q:             2

DiffServ Interface parameters record is not found
```

Figure 29. Enabling DiffServ on a router

```
remote router DS Config>SET INTERFACE
Enter Diffserv Interface number [0]? 1
Set Premium Queue Bandwidth (%) (1 - 99) [20]?
    Assured Queue Bandwidth (%) = 80
Configure Advanced setting (y/n)? [No]:
Accept input (y/n)? [Yes]:
remote router DS Config>LIST ALL

System Parameters:

        DiffServ:                ENABLED
        Packet_size:             550
        Min BE Alloc (%):        10 1
        Min CTL Alloc (%):       5
        Number_of_Q:             2

-------------------------------------------------------------------------------
                        ---------- Premium -------    --------- Assured --------
Net If      Status  NumQ Bwdth  Wght  OutBuf  MaxQos Bwdth  Wght  OutBuf  MaxQos
Num                        (%)   (%)  (bytes)    (%)   (%)   (%)  (bytes)    (%)
-------------------------------------------------------------------------------
1   FR     Enabled     2    20    90    5500      95    80    10   27500      80
```

Figure 30. Enabling DiffServ on an interface

Most of the work necessary to enable DiffServ is in the creation of policies which identify streams and assign appropriate DiffServ actions. The defined action (see Figure 24 on page 35) is where the appropriate queue and bandwidth percentages are defined.

When defining policies, consider the following:

- It is unlikely that more than two streams or DiffServ policy actions will be defined assigning traffic to the premium queue. And even if there is more than one, ensure that the total bandwidth requirements do not exceed the available bandwidth, which is the MaxQos percentage of the bandwidth - in the default case above, 95% of 20% or 19% of the bandwidth. Don't define two streams each with a bandwidth of 10%, because this will mean that one of them (the second to be used) will be treated as best effort.
- Do not exceed 50% (in the default case) of the available bandwidth for assured forwarding traffic; see Table 7 on page 37 for a breakdown of how much

bandwidth is available to be assigned to this type of traffic (the calculation here is the displayed MaxQos percentage of the available bandwidth minus the reservations for best effort and control traffic). Bandwidth can only be assured for this sort of traffic if the policy definitions do not overcommit bandwidth; there is no automated mechanism for detecting this.

- The relationship between percentage of output bandwidth and percentage of output buffers allocated is not immediately obvious. By default, the AF/BE bandwidth is 80% of the total and the bandwidth available for allocation is 80% of that, or 64% of the total. Therefore, if buffers are allocated proportionally, it follows that the available number of output buffers corresponds to 64% of the output bandwidth. In turn, then, a stream which actually uses 40% of the output bandwidth will use 40/64 of the available output buffers, which is in fact 62.5% of them. The number of buffers available for allocation can be determined by taking the MaxQos percentage of the OutBuf number in Figure 30, which by default comes to 22,000 buffers of 550 bytes each.
- Remember that all other traffic will be treated as best effort traffic. In the absence of any other traffic, best effort traffic can use 100% of the output bandwidth; by default, all best effort traffic is only guaranteed 10% of the output bandwidth.
- Changing the service type byte (the TOS marker in Figure 28 on page 39) is an action performed to mark packets for subsequent treatment by other network devices. There are various implications from this:
  - Routers at the edge of a network can use this to override any values originating from endstations and - in conjunction with a central LDAP server - can do this in a consistent manner across the network. One implementation might be to define policies for edge routers to mark packets and different policies for core routers to honor the markings. The policies would be stored centrally, with the routers being configured to retrieve and implement the different policies.
  - The value received in a packet by the router’s policy engine may be different from when the router initially received the packet; other router components such as IP access control may have modified the byte already.
  - Once a packet has been treated by a policy and allocated a corresponding DiffServ action, the value in the service type field has no further significance to a router. DiffServ will queue the packet provided that it has available buffers. The scheduler will subsequently schedule the packet for onward transmission, but it does this without reference to the service type field. Whether the DiffServ action causes the service type byte to be changed or not has no bearing on subsequent actions by DiffServ in the same router.
  - Using V3.4 code, it is possible to specify that the AF/BE queue will be policed. This means that packets in excess of the rate defined in the corresponding DiffServ policy will be marked as yellow or red packets[5] - in fact they will have their service type byte changed to show medium or high drop precedence respectively. This has no effect on the router that performs the marking: all packets will be queued on the AF/BE queue in first-in, first-out (FIFO) manner and *any* packet will be discarded if the queue is full. Subsequent routers in the network which receive these yellow and red packets can make more intelligent decisions, allowing these to be discarded before green packets.

[5] See RFC 2697, A Single Rate Three Color Marker and RFC 2698, A Two Rate Three Color Marker

- V3.4 code provides some predefined DiffServ actions which can be used to map a flow to a particular EF or AF queue. In addition, if you chose to define your own DiffServ action, predefined AF classes are provided (AF1, AF2, AF3, AF4). These classes should only be used in conjunction with the AF queue policer, because these predefined DiffServ actions set an incomplete subset of the service type byte for the outgoing packet. The AF policer is required to set appropriate drop precedence bits; without it the packet will end up with a bit setting in the service type byte which does not correspond to any of the defined Class Selector code points for an AF PHB.
- It is not possible to define a policy that modifies the service type byte but that does not map the packet to an EF or AF queue. It is possible to define a DiffServ action which assigns the packet to the AF queue but with zero bandwidth allocated to it; if this is done it is effectively the same as a policy which drops all packets.

2.5.2 Configuring policies for DiffServ

This example shows how DiffServ can be applied to routing in the core of a network such as the one shown in the following figure. It shows a network in which the edge routers classify IP packets according to the latest draft standards.

The core routers route this bronze traffic out across the public Internet using a secure tunnel; there is no guarantee on the delivery of this sort of traffic. The core routers also define gold traffic with assured forwarding of 40% of the output bandwidth on the wide area links and silver traffic to get 10% of the output bandwidth. Expedited forwarding is being used for other traffic, such as voice, although this is not shown here.

The edge routers in this picture may be providing a connection to an external organization, so this network could be a very simple model of a service provider’s network. Again, leaving out for now the aspects of security/encryption, the edge routers can classify traffic according to rules based on how much money the end user is prepared to pay for the service.

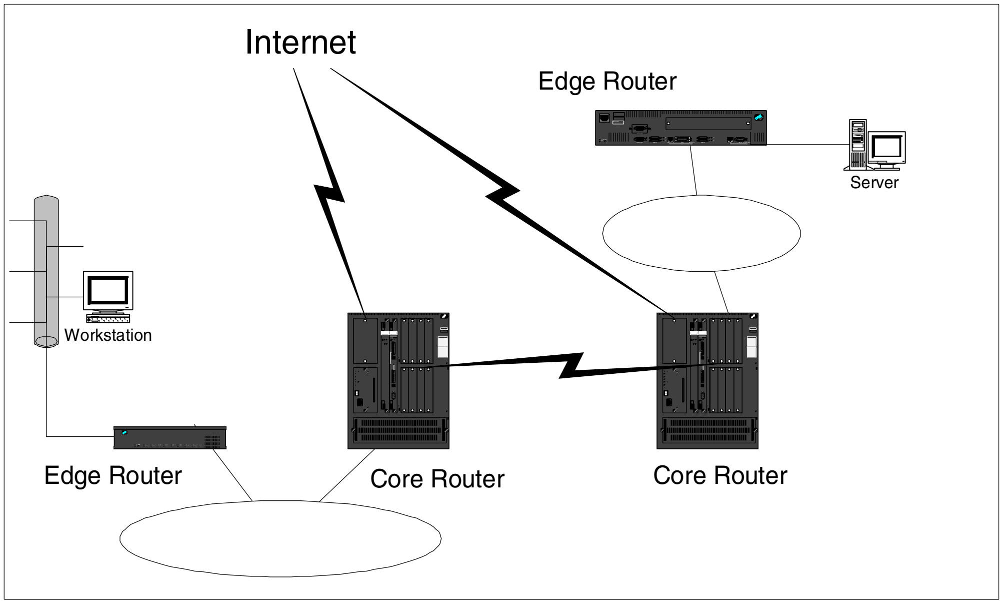


Figure 31. Bronze, silver, and gold services

All that the policy database entries in the core routers need to do is identify packets and assign the appropriate DiffServ actions. The same mask value, E0, should be used in both cases and packets with the value of 40 (representing bit setting 010xxxxx) should be treated as silver packets and those with value of 60 (representing bit setting 011xxxxx) should be treated as gold packets.

In addition, the core routers implement IP access control rules to identify bronze packets (mask value E0, actual value 20) and route these packets across the Internet.

The edge routers also implement a policy engine, but in this case it is used for the classification of packets based on a policy traffic profile. The classification policy may change frequently (for example, a bronze customer might decide to pay more money) and therefore, it makes sense to store these and other policy database entries in a central policy database. All the routers in the network could be configured to update their policy database entries automatically from the server at a certain time each night. Additional policies might be defined so that, for example, SNMP traffic is treated as gold traffic regardless of its origin.

As an example, here is the list of policies configured in the core routers to deliver gold and silver service:

```
remote router Policy config>LIST ALL

Configured Policies....

Policy Name     = GoldService
        State:Priority =Enabled    : 4
        Profile        =GoldService
        Valid Period   =AllTheTime
        DiffServ Action=GoldService

Policy Name     = SilverService
        State:Priority =Enabled    : 4
        Profile        =SilverService
        Valid Period   =AllTheTime
        DiffServ Action=SilverService
```

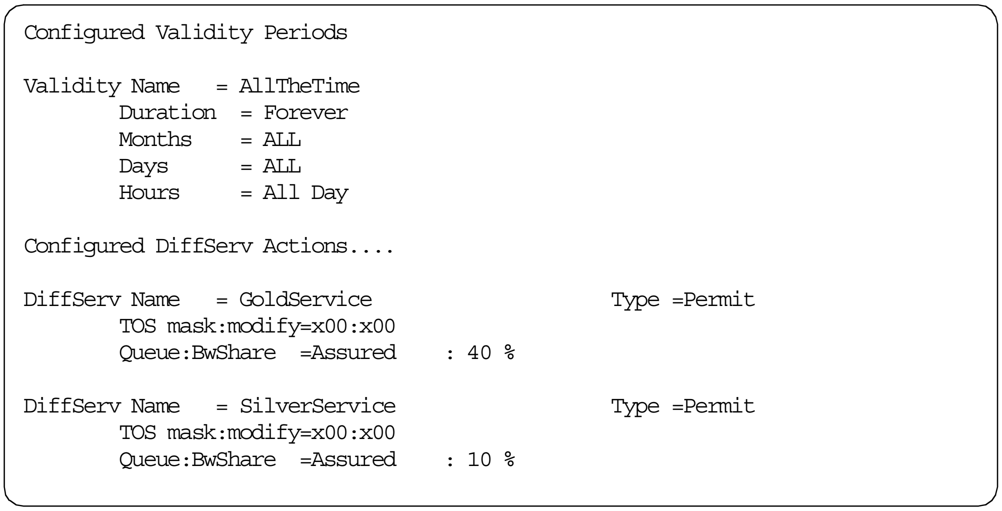
Figure 32. Policies configured in core routers

```
Configured Profiles....

Profile Name    = GoldService
        sAddr:Mask=        0.0.0.0 : 0.0.0.0         sPort=    0 : 65535
        dAddr:Mask=        0.0.0.0 : 0.0.0.0         dPort=    0 : 65535
        proto     =              0 : 255
        TOS       =            xE0 : x60

Profile Name    = SilverService
        sAddr:Mask=        0.0.0.0 : 0.0.0.0         sPort=    0 : 65535
        dAddr:Mask=        0.0.0.0 : 0.0.0.0         dPort=    0 : 65535
        proto     =              0 : 255
        TOS       =            xE0 : x40
```

Figure 33. Profiles configured in core routers



```
Configured Validity Periods

Validity Name   = AllTheTime
        Duration  = Forever
        Months    = ALL
        Days      = ALL
        Hours     = All Day

Configured DiffServ Actions....

DiffServ Name   = GoldService                   Type =Permit
        TOS mask:modify=x00:x00
        Queue:BwShare  =Assured    : 40 %

DiffServ Name   = SilverService                 Type =Permit
        TOS mask:modify=x00:x00
        Queue:BwShare  =Assured    : 10 %
```

Figure 34. Validity periods and DiffServ actions configured in core routers

Neither of the two additional DiffServ actions modifies the service type byte; they are simply used to allocate assured bandwidth based on the settings of this byte in the packets they receive.

2.6 Considerations for transporting voice traffic over IP networks

Although IBM routers are no longer to be developed to provide the capability of transporting directly attached voice traffic over IP networks (voice over IP, or VoIP), the increasing use of other devices (H.323 terminals and gateways) in existing IP networks means that IBM routers will increasingly be required to transport voice traffic intermixed with other IP traffic.

Anyone who introduces VoIP traffic into an existing IP network essentially needs to redesign the existing network, because the requirements of voice traffic are so different from the requirements of data traffic that significant changes need to be made to the network in order to transport both types of traffic simultaneously and satisfactorily.

The first change that should be made is that all IBM routers that are to transport voice traffic over low bandwidth (56 kbps or 64 kbps) wide area links (PPP or frame relay) should be upgraded to the V3.4 code level. All the remaining discussion in this section of the book refers to this level of code.

2.6.1 Voice over IP basics

This is not the place for a tutorial on transporting voice traffic over IP networks, but some basic assumptions which derive from current practice and implementations include:

- A single voice call requires four IP sessions:
  1. Q.931 signaling in H.225 over TCP/IP port 1720
  2. H.245 control channel over TCP/IP over a negotiated port number
  3. H.245 audio channel over RTP/UDP/IP over an even-numbered port
  4. H.245 call control channel over RTCP/UDP/IP over an odd-numbered port

The RTCP control channel always uses a port number one greater than the RTP audio channel itself.

- Voice traffic is transported using the Real Time Protocol[6] (RTP), which means that each packet contains an IP header, a UDP header, and an RTP header.
- Although H.323 terminals are required to support G.711 voice encoding (a 64 kbps data stream), any implementations that attempt to transport voice traffic over relatively low bandwidth (56 kbps or 64 kbps) WAN links are likely to use G.729 or similar voice-encoding schemes. G.729 typically results in a 20-byte voice packet payload which requires data transmission speeds of 8 kbps; the RTP/UDP/IP headers add another 40 bytes to each packet. Without modification, G.729 VoIP streams each require 24 kbps bandwidth.
- Transmission of voice and data traffic over the same low bandwidth link requires that the data traffic be fragmented. Consider a router that has no voice traffic to transmit, and therefore begins to transmit a 2,000-byte data packet over a 64 kbps link. Immediately after the router starts the data transmission a 60-byte voice packet arrives. Even if the voice packet is transmitted immediately after the data packet, it will still have to wait for a quarter of a second for the data packet to be transmitted. This delay (250 milliseconds) is already close to the maximum acceptable delay (around 400 milliseconds) for the total end-to-end delay for the voice traffic across the entire network, and two such delays will together obviously exceed the acceptable total delay. Data traffic must be split into smaller pieces for transmission over low-speed links, which will then allow the more time-critical voice packets to be interspersed between the data fragments.
- Voice traffic should neither be encrypted nor compressed on a link, even if data traffic on the same link is either encrypted, compressed, or both. Compression will be detrimental because the voice traffic will already have been compressed at the endpoints (CODEC in H.323 terminals); if any encryption of voice traffic is required then this will be performed at the endpoints rather than at the link level on specific links.

2.6.2 PPP considerations

To transport both voice and data traffic over relatively low bandwidth PPP links, the following points should be noted:

1. Prioritization: Voice traffic should either use the DiffServ EF premium queue mechanism or the BRS superclass.
   Policies which identify voice traffic need to use both the service type byte and/or the UDP port numbers; even though voice traffic uses RTP, the RTP header follows the UDP header in the IP packet and therefore both BRS and DiffServ simply view these packets as UDP packets.
   To put this another way: neither BRS nor DiffServ provides a simple method of identifying RTP traffic; this traffic needs to be identified by falling into a particular range of UDP port numbers and/or having a particular setting of the service type byte. There is no fixed UDP port number, but it is almost certain that the range of acceptable port numbers will have been defined for the network when VoIP was initially configured; H.225 registers a default pair of port numbers (5004 and 5005) and the recommendation is that RTP use ports numbered greater than 5000.

[6] See RFC 1889 RTP, A Transport Protocol for Real-Time Applications

2. Fragmentation and interleaving: Multilink PPP should be used even where a single physical PPP link is used, because it allows large data packets to be fragmented for transmission over the link and also allows voice packets to be interleaved between the data fragments. Standard PPP can fragment but it can not interleave. A minimum fragment size should be defined so that voice packets are never fragmented although large data packets are. Voice packets will only be inserted between fragments of data packets if the voice packets are first of all allocated to the DiffServ EF premium queue or the BRS superclass.
3. Encryption and compression: PPP will not encrypt or compress data taken from either the DiffServ EF premium queue or the BRS superclass. Prior to code Release V3.4, routers which received unencrypted or uncompressed packets over a link otherwise defined for encryption or compression would discard the unencrypted or uncompressed packets, so there is a real requirement for V3.4 at both ends of the link.
4. RTP header compression[7] effectively overcomes the overhead of IP/UDP/RTP headers on each voice packet, allowing the transport of G.729 (8 kbps) traffic at approximately 10 kbps on a PPP link. This form of compression has to be set explicitly (`SET IPCP` command). If DiffServ is being used, the leaky bucket parameters for the EF premium queue must be modified, because the default values assume the same bandwidth for traffic arriving into the queue as allocated to the outbound link. So, for example, if three voice channels are to be allowed on a 64 kbps link, DiffServ must be customized to allow 9,000 bytes per second (9,000 Bps=72 kbps=3x24 kbps) as the rate for the leaky bucket algorithm but must also only allocate 30 kbps (3x10 kbps or approximately 50% of the output bandwidth) for transmission buffer allocation. Watch out for possible confusion caused because some parameters are specified in bytes per second (Bps) whereas others need to be specified in bits per second (bps). Also note that if the percentage of output bandwidth allocated to the EF premium queue is changed from the default value (20%), then it is necessary to choose the *Configure Advanced setting* option to change the EGRESS BufSize for the queue in proportion; this option is found using the `SET INTERFACE` command at the `DS Config>` prompt.
   RTP header compression is badly named; the fundamental problem of using RTP to transport voice traffic is that the typical packet header is twice the size of the typical voice packet being transported. What RTP header compression actually does is removes the IP/UDP/RTP headers from almost all packets and replaces the header information with a minimal header which can be used by the recipient to reconstruct the full header. So, for example, the RTP sequence number field only needs to be sent if it differs in an unusual manner from the preceding packet - for example, if it does not contain a value 1 greater than the preceding packet's sequence number field. The full IP/UDP/RTP header need only be sent periodically - by default only once every 256 packets in IBM's implementation, which is also the recommendation in RFC 2509.
   Note also that some implementations of RTP header compression do not follow the proposals of the RFCs mentioned above and therefore are proprietary to a particular router manufacturer.

Although this discussion has concentrated on using DiffServ as a method of identifying and favoring voice traffic over data traffic, BRS could be used instead. The only significant difference is that with voice traffic being assigned to a BRS

[7] See RFC 2507, IP Header Compression, RFC 2508, Compressing IP/UDP/RTP Headers for Low-Speed Serial Links and RFC 2509, IP Header Compression over PPP

superclass instead of to the DiffServ EF premium queue, excess voice traffic will affect data traffic, ultimately to the extent of voice traffic completely blocking all data traffic. This is not possible if DiffServ is used because the EF policer will cause voice packets in excess of the configured rate to be dropped. Either of these two possibilities could be preferable depending on the exact needs of the network being designed: DiffServ allows more control, but if the ability to accommodate excess voice traffic is important, BRS might be more appropriate.

2.6.3 Frame relay considerations

One option for transporting voice traffic across frame relay networks is to transport the voice traffic over frame relay without IP encapsulation (voice over frame relay - VoFR). This is outside the realm of DiffServ, and indeed BRS must be used to differentiate and prioritize between VoFR traffic and data traffic using IP and other protocols. Many of the considerations for transporting VoFR also apply when VoIP traffic is being transmitted over low-speed frame relay interfaces, with one major disadvantage: frame relay does not provide RTP header compression and therefore every packet must be transmitted with the significant overhead of the IP/UDP/RTP header of 40 bytes.

Similar considerations apply to frame relay interfaces as for PPP interfaces:

1. Prioritization: Even in a pure IP environment (in which IP voice and data traffic is being transmitted over frame relay links) DiffServ will not be used; this is an implementation restriction caused by DiffServ’s inability to work in conjunction with frame relay FRF.12 fragmentation. Voice traffic should be assigned to a BRS superclass because otherwise it will be delayed by data traffic.
2. Fragmentation: FRF.12 fragmentation should be set on the interface using the `ENABLE FRAGMENTATION` command. The default fragmentation type *UNI/NNI Fragmentation* should be changed to *End-to-end Fragmentation*. Voice traffic will now be interleaved with data fragments provided that the voice traffic is defined either in a superclass or as urgent in a particular traffic class. Do not assign voice traffic to its own BRS traffic class because, if so, the voice traffic will be allocated a percentage of the bandwidth which will be allocated on a round-robin basis with other data traffic classes. Note also that FRF.12 cannot be used in conjunction with frame relay SVCs.
3. Encryption and compression: Unlike for PPP interfaces (in which all traffic taken from a BRS superclass or the DiffServ EF premium queue is never compressed or encrypted), access control filters are required to identify voice traffic and explicitly disable compression and encryption. Compression should be enabled on the frame relay interface itself, so the access control should be created (which identifies voice traffic as being UDP traffic with a range of port numbers appropriate to the network’s implementation of VoIP) and then modified using the `ENABLE COMPRESSION-BYPASS` and `ENABLE ENCRYPTION-BYPASS` commands.
   One similarity with PPP remains: the router at the receiving end of the frame relay link must be able to accept uncompressed and unencrypted packets on a link otherwise transmitting compressed and/or encrypted data packets; in the IBM implementation this requires V3.4 code in both routers at each end of the link.
4. Header compression is not possible when transporting VoIP traffic over frame relay links using IBM routers[8]. This alone may dictate against implementing

VoIP across low bandwidth frame relay links. Two other options may warrant consideration:

1. Use VoFR, support for which has been available since the V3.3 code release. Direct voice encapsulation in frame relay incurs a much lower encapsulation overhead than VoIP (approximately 25% for VoFR compared with 200% for VoIP - G.729 produces 20-byte packets, the VoFR overhead is 5 bytes and the VoIP overhead is 40 bytes).
2. Use the Frame Relay Frame Handler (FRFH) support in IBM routers, formally introduced with V3.4 code but in fact available as an earlier PTF to V3.3 code. This support allows downstream VoFR devices to treat the IBM router as a frame relay switch; this allows the IBM router to have a single frame relay WAN connection and combine voice traffic switched from downstream VoFR devices with multiprotocol data traffic originating from the router's LAN interfaces.

In addition, frame relay PVCs which transport voice traffic require careful analysis and tuning of the related frame relay parameters B_c, CIR, and T_c: the burst size, committed information rate, and burst interval respectively (remember that B_c = CIR x T_c). In particular, T_c must be set to a significantly lower value than the default value of 1. If the burst interval is left at 1 second, then a router may send a burst of traffic which adds up to the amount specified in the committed information rate and then have to wait for the expiration of the remainder of the burst interval before sending more traffic. The default value of 1 second will almost certainly impose unacceptable delays on voice traffic, and a reasonable setting for transporting voice over frame relay might instead be the lowest possible value of 30 milliseconds (T_c=0.03). Increasing the burst interval above this value effectively increases the delay associated with the frame relay network hop. As the burst interval reduces, however, the size of the fragments required for FRF.12 fragmentation also reduces, which increases the number of fragments into which data packets will be split. This is increasingly inefficient in terms of bandwidth and processor utilization. A likely range of acceptable values is for T_c to be between 0.03 and 0.06. In any case, once a value for T_c has been determined, the value for the fragment size has to be derived, in part based on the number of simultaneous calls that are to be supported over the frame relay PVC. The calculations for VoIP over frame relay are essentially the same as for VoFR, with the only difference being because of the significant difference in packet sizes already mentioned above. See Appendix A, "Sample calculations for frame relay parameters" on page 135 for some initial thoughts on how to calculate fragment sizes.

Having determined appropriate values for various frame relay parameters, the router must also be configured to honor these settings, using the `ENABLE CIR-MONITOR` command. Before the introduction of voice traffic to the network, the decision may well have been taken to allow the frame relay interfaces to send traffic into the frame relay network at rates greater than the committed information rate and allowing the frame relay network itself to discard packets under congestion. This may well be acceptable in a pure data network, mainly because the applications and protocols being used are capable of retransmitting discarded data packets. Although voice traffic is actually able to tolerate the loss of a percentage of the voice packets, there is no way in which voice traffic can be retransmitted. To avoid significant degradation of voice traffic, the devices which

[8] In fairness: there is no standards-based method for RTP header compression across frame relay, but some manufacturers (Cisco, for example) implement a proprietary mechanism for this.

send voice traffic into the frame relay network must police themselves to ensure that they do not exceed the committed information rate, which is essentially a contract by the frame relay network guaranteeing transmission of traffic at this rate but no higher.

If nothing else, this section should serve to emphasize the point that adding voice traffic to an existing data network will require careful analysis and redesign of the existing network.

2.7 Protocols other than IP

It's worth restating that both DiffServ and the policy feature are applicable only to IP traffic, and that access controls are actually IP access controls. These new features may be essential for implementation of new IP-based networks, but they may not be best suited to some existing multiprotocol networks.

The Bandwidth Reservation System, on the other hand, and for all its limitations, remains as a method of prioritizing many different types of traffic using different protocols over a single link. Take, for example, a single PPP link which is transporting IP, IPX, and APPN traffic at the same time; BRS can still play a powerful role in differentiating between types of traffic, and the power of BRS is that it can define priority relationships between different types of traffic using different network protocols. HPR high priority traffic and IP Telnet traffic could be grouped together in a priority relationship and given priority over IP FTP and IPX traffic, which in turn could be given a higher priority than SNA batch traffic.

BRS may be a valid approach; an alternative approach could be to remove protocols other than IP from the backbone of the network. SNA traffic, for example, can effectively be removed from the backbone by the use of any or all of DLSw, TN3270, and Enterprise Extender (HPR over IP). In all of these the data transmitted over wide area links would be as IP frames, which can therefore be mapped to a stream and classified and handled by DiffServ.

If NetBIOS is being transported over NetBEUI, this traffic may be moved to IP over the backbone either by the use of DLSw or, in the case of Microsoft Windows networks, by the use of NetBIOS over IP using WINS and Microsoft's implementation of RFCs 1000/1001.

Note that it is not possible to run BRS and DiffServ on separate frame relay circuits which are on the same physical interface, so even if different VCs are defined for IP and SNA traffic they must both be handled by the same differentiation and queueing mechanism.

2.8 CS for OS/390: service policy agent and LDAP server

Communications Server for OS/390 (CS for OS/390) is IBM's implementation of the TCP/IP and SNA stacks for its mainframe computers. Network administrators can use the OS/390 UNIX service policy agent (PAGENT) to define service-level policies for their users.

The service policy agent was first made available in the V2R7 release of CS for OS/390.

OS/390 is sometimes used as a router: a single OS/390 system may have multiple IP addresses and may be configured to route between different interfaces. It is important to note that the policy agent does not apply to packets which use this path; it only applies to TCP/IP applications which originate packets themselves.

PAGENT retrieves rules either from a local policy configuration file (`/etc/pagent.conf`) or from an LDAP server. The LDAP server may be elsewhere in the network, but it can also be implemented on the OS/390 system itself.

PAGENT uses the policy rules to set the service type byte values for outgoing frames and defines a local transmission priority value which can be used by certain types of network interfaces.

2.8.1 Outgoing TOS

Policy rules define service categories which specify outgoing TOS definitions. Although much of the CS for OS/390 documentation refers to the use of the service type field in terms of the original RFC 791 definitions, all the actual definitions allow the setting of any of the bits in the service type byte without making assumptions about the meaning of any particular bit.

A service category definition might look like the following:

```
ServiceCategories              networkcontrol
{
OutgoingTOS                   11100000
}
```

Next, service policy rules are used to assign traffic to a service category:

```
ServicePolicyRules            routed
{
ProtocolNumber                UDP
SourcePortRange               520
ServiceReference              networkcontrol
}
```

Service policy rules can be defined for incoming traffic, outgoing traffic, or both; the default is to apply the rules to outgoing traffic only.

CS for OS/390 ships samples of these configuration statements in `/usr/lpp/tcpip/samples/pagent.conf`.

2.8.2 Local transmission priority

The service policy agent also allows the derivation of a local transmission priority value from the settings of the service type byte. This local transmission priority is only used today by QDIO device types, the only current implementation of which is the Gigabit Ethernet Open Systems Adapter (LINK IPAQGNET on an MPCIPA device). QDIO supports four priority levels, given numerical values between 1 and 4, with 4 being the lowest priority.

The default mapping of the service type byte to QDIO priority values is given in the following table:

Table 8. Default mapping of service type to QDIO priority

IP service type	QDIO priority
00000000	4
00100000	4
01000000	3
01100000	2
10000000	1
10100000	1
11000000	1
11100000	1

Again, these default values assume an interpretation consistent with RFC 791. CS for OS/390 allows these values to be changed, either for all interfaces or for specific interfaces. For example, for a specific interface:

```
SetSubnetPrioTosMask
{
Subnetaddr                  9.11.12.13
SubnetTosMask               11100000
PriorityTosMapping          1 00000000
PriorityTosMapping          1 00100000
PriorityTosMapping          2 01000000
PriorityTosMapping          3 01100000
PriorityTosMapping          4 11100000
}
```

This local transmission priority value is only used by the TCP/IP stack itself and by the I/O devices which support it: QDIO causes IP traffic to be queued into four separate traffic queues prior to transmission based on this value.

2.8.3 Other service types

The policy agent can also enforce other policy rules, including those limiting the application of the rules to specific times and days of the week, plus the following:

MaxRate — An integer value representing the maximum rate in kbps (thousands of bits per second) allowed for traffic in this service class. This only applies to TCP connections. If a non-zero value is specified, each TCP connection mapped to this service category will have its transmission rate limited to this value.
If this value is set very low (1, for example), the observed throughput may in fact be much greater than this; this is because a minimum rate of *packet size divided by round-trip time* is also observed. So, for 1524-byte packets and a 10 ms round-trip time, a rate of about 1,200 kbps (152,400 bytes per second) will actually be observed.
Another way of looking at this: whatever the value for

MaxRate, CS for OS/390 will not go any further than by setting the TCP window size to its minimum value of 1.

MinRate An integer value representing the minimum rate allowed for traffic in this service class, again only applicable to TCP connections. Provided there is traffic and provided that the network is not congested, the TCP/IP stack will maintain this minimum throughput for any TCP connection mapped to this service category.

MaxConnections An integer value representing the maximum number of end-to-end TCP connections at any instant in time. This rule may be useful for limiting the number of simultaneous connections to a Web server, for example.

Service policy rules can also be defined with a specific policy scope which causes the rule to be applied only to traffic which has been specifically reserved using RSVP. See 3.4.4, “RSVP and S/390 host systems” on page 72 for specific information about the implementation of RSVP on the OS/390 platform.

2.8.4 OS/390 LDAP Server

The OS/390 LDAP Server does not form part of CS for OS/390, it is instead shipped as part of the OS/390 Security Server product. It has been available since OS/390 V2R5.

LDAPSRV is the LDAP Server that runs on OS/390. The name of the program in the hierarchical file system (HFS) is *slapd* to be consistent with other UNIX implementations. You can use it to provide a directory service of your very own. Your directory can contain just about anything you want to put in it.

Some of LDAPSRV's more interesting features and capabilities include:

Robust database OS/390 LDAPSRV comes with an RDBM backend database based on DB2.

Multiple instances LDAPSRV can be configured to serve multiple databases at the same time. This means that a single LDAPSRV server can respond to requests for many logically different portions of the LDAP tree.

Access control LDAPSRV provides a rich and powerful access control facility, allowing you to control access to the information in your database or databases. You can control access to entries based on LDAP authentication information, including users and groups. Access control is configurable down to sets of attributes within entries.

Threads LDAPSRV is threaded for high performance. A single multithreaded LDAPSRV process handles all incoming requests, reducing the amount of system overhead required.

Replication LDAPSRV can be configured to maintain replica copies of its database. This master/slave replication scheme is vital in high-volume environments where a single LDAPSRV just does not provide the necessary availability or reliability.

Configuration LDAPSRV is highly configurable through a single configuration file which allows you to change just about everything you would ever want to change. Configuration options have reasonable defaults, making your job much easier.

Secure communications LDAPSRV can be configured to encrypt data to and from LDAP clients. It has a variety of ciphers for encryption to choose from, all of which provide server authentication through the use of X.509 certificates.

PAGENT can be configured to read its configuration information from an LDAP server in addition to the configuration information in the local configuration file by the use of the ReadFromDirectory statement:

```
ReadFromDirectory
{
LDAP_Server           ldapserver.itso.ral.ibm.com
LDAP_Port             9000
Base                  o=ibm,c=us
LDAP_SelectedTag      mvs03a
}
```

If no value for LDAP_Server is specified, the local host address (127.0.0.1) is used. The default value for LDAP_Port is 389.

Chapter 3. Integrated Services

All the discussion in the previous chapter has built on a model which can be described as one that implements a best effort delivery service of IP. This is the model which has been the basis for the underlying design of IP networks and stretches back at least 20 years to the research projects which preceded the IP networks and the Internet we know today.

Integrated Services (IntServ) introduces a new model which changes this fundamental model but attempts to do so by extending the original architecture. This allows components of Integrated Services to be added to existing networks.

Integrated Services addresses the need to handle traffic in which the time of delivery of packets across the network is critically important, and does so by managing the per-packet delay by setting bounds on the minimum and maximum delays. It defines two sorts of service[1]:

1. Guaranteed service, in which the network guarantees an upper bound on delay across the network, allowing applications to rely on this value. Another way of looking at this approach is that this maximum delay value is based on the worst case assumptions of the behavior of the network for this type of traffic.
2. Predictive service, in which an upper bound on delay across the network is provided which is fairly reliable but not guaranteed. Looking at this another way, the upper bound on delay is calculated not by the worst case assumptions across the network but instead based on realistic and conservative assumptions of the behavior of the network.

The reason for offering both types of service is that the first is expensive - any network that guarantees a maximum value for delay must reserve resources across the network which would otherwise be available for use by other users. For many applications it is less expensive and more efficient to relax the service requirements from perfectly to fairly reliable bounds.

The comparison with all the Differentiated Services approaches discussed in the previous chapter is that no guarantees of any sort on network delay are made with these. Packets may be classified in such a way as to indicate that they should be handled in ways which minimize delay through the network, but the way each packet is handled by each node in the network offers no guarantees. It may well be the case that in a well-designed network, especially using facilities such as priority queueing using the DiffServ feature, delay-critical traffic can be handled perfectly well enough. The danger here is that a Differentiated Services network which delivers this type of service well enough today may not perform adequately in the future simply by the addition of traffic load to the network; Integrated Services provides a mechanism for users of the network to request *and be guaranteed* maximum delay bounds, and even the predictive service category offers a fairly reliable guarantee which compares well with none at all from a Differentiated Services implementation.

[1] See RFC 1633, Integrated Services in the Internet Architecture: An Overview

3.1 Guaranteed service and controlled load

More recent definitions of Integrated Services rename one of the service models and define:

1. RFC 2212: Specification of Guaranteed Quality of Service
2. RFC 2211: Specification of the Controlled-Load Network Element Service

The controlled load model is supported beginning with V3.3 of IBM router code, the two key aspects of which are that:

- A very high percentage of transmitted packets will be successfully delivered by the network to the receiving end nodes.
- The transit delay experienced by a very high percentage of the delivered packets will not greatly exceed the minimum transit delay experienced by any successfully delivered packet.

Clients requesting controlled load service must provide an estimation of the data traffic they will generate; in return, the service ensures that network element resources adequate to process traffic falling within this descriptive envelope will be available to the client.

3.2 Resource Reservation Protocol

Resource Reservation Protocol (RSVP) is defined in RFC 2205 and is the signaling protocol used by Integrated Services to set up and control resource reservations across the network. RSVP signals per-flow resource requirements to network elements using parameters defined by the Integrated Services model. There are two practical network models to consider:

1. A network in which all hosts and routers support RSVP. Hosts use RSVP to request resources to be reserved across the network. Routers enable RSVP on specific interfaces and define the bandwidth available for reservation on the interfaces. At the moment, only a small number of hosts generate RSVP signaling, and even in the future many applications and hosts will never do this.
2. A network in which hosts do not support RSVP. Routers can be configured to reserve resources on behalf of these hosts.

RSVP messages are identified by the use of the protocol field in the IPv4 header, which is the 10th byte of the header, set to the value 46 (decimal).

RSVP requests resources in one direction, between a sender and a receiver.

RSVP reservation is initiated by a sender sending PATH messages towards the receivers of a proposed data traffic flow. PATH messages include information on the traffic characteristics (TSPEC) of the sender's data flow. The PATH messages are addressed by the sender to the IP address of the destination of the proposed flow; the PATH message is required to use the alert option field in the IP header. The alert option is defined in RFC 2113[2], which is understood by routers in the network path between sender and receiver as meaning "routers should examine this packet more closely" and provides a mechanism for routers to intercept

[2] RFC 2113, IP Router Alert Option

packets that are not explicitly addressed to them. This mechanism is used to accomplish the following:

1. PATH messages are forwarded using the same route as IP data packets because routers forward them using their existing routing table entries.
2. PATH messages include the IP address of the interface through which the PATH message is sent.
3. Each router intercepting a PATH message creates path state information for the sender that includes at least the unicast IP address of the previous hop node along with a session identifier (destination IP address, IP protocol, destination port).
4. Each router which intercepts a PATH request modifies it before forwarding it, at the very least by updating the value of the IP address of the interface over which the PATH request is to be forwarded.
5. Routers which do not implement the alert option (and, therefore, do not implement RSVP) simply forward the PATH message to the next-hop router. This means that it is not a requirement that all routers implement RSVP in a network which uses the Integrated Services model.

When the receiver receives a PATH message it then sends a RESV message back through the network to reserve the resources requested. Rather than addressing the RESV request to the IP address of the sender of the PATH request, the RESV message is addressed to the unicast address of the last hop which forwarded the PATH request. This ensures that the RESV traverses *exactly* the same reverse path over which the original PATH request was sent; if the RESV were instead addressed to the IP address of the sender of the PATH request it could take a totally different reverse route.

The RESV request is the request which actually causes resources to be reserved.

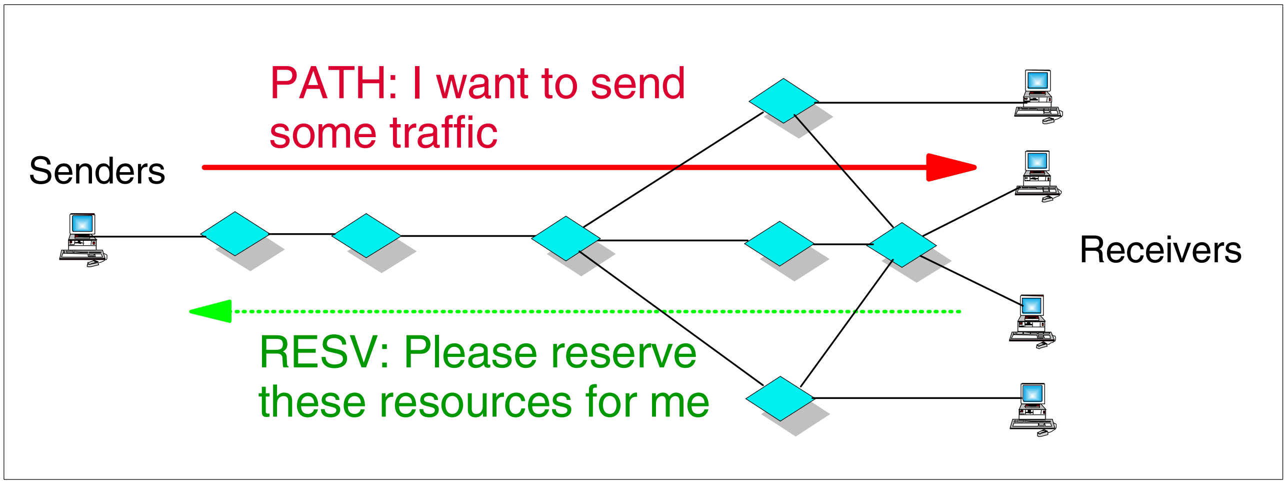

PATH: I want to send some traffic
Senders
Receivers
RESV: Please reserve these resources for me

Figure 35. PATH and RESV

This simple model has limitations, the most important of which is that receipt of a PATH request by a receiver says nothing about the state of the network over which the PATH request has passed, and in particular offers no guarantees that any type of resource reservation is possible. The information in the PATH request simply contains the type of service requested by the sender. The receiver constructs an RESV packet requesting the reservation of resources and sends this packet back along the reverse route, and each node in the path can then accept or reject the reservation request.

RSVP supports an enhanced model called One Pass With Advertising, or OPWA. This adds an advertisement object (ADSPEC) to the PATH requests which is used to collect information on the state of network resources as the PATH request is routed through the network. The initial ADSPEC information is provided either by the requesting application program or the requesting host as an indication of the types of resource reservation requests that will be made. This ADSPEC information is then updated as the PATH request flows through the network and is ultimately made available to the receiver, and perhaps even to applications running on the receiver. The receiver can then use this information to tailor its RESV request so that it will actually succeed. The sort of information provided to the receiver using OPWA includes information such as:

- Maximum bandwidth available for reservation across the network path
- MTU size available for reservation across the network path
- Whether or not there is a non-RSVP hop along the path, and hence at least one point of the path is providing best effort service
- Whether or not a specific IntServ service is implemented at every hop along the path; for example, whether Controlled Load and/or Guaranteed Service are provided

The information provided by OPWA should not be regarded as a guarantee of resource availability; resources which were available for reservation at the time of the PATH request may no longer be available when the RESV request arrives. However, OPWA is a lot better than nothing.

Figure 36 shows an example of OPWA in which the receiver knows not to issue a reservation request for bandwidth greater than 0.5 Mbps. Without OPWA, the receiver might have attempted to reserve more bandwidth than actually available across the network and the reservation request would have been rejected.

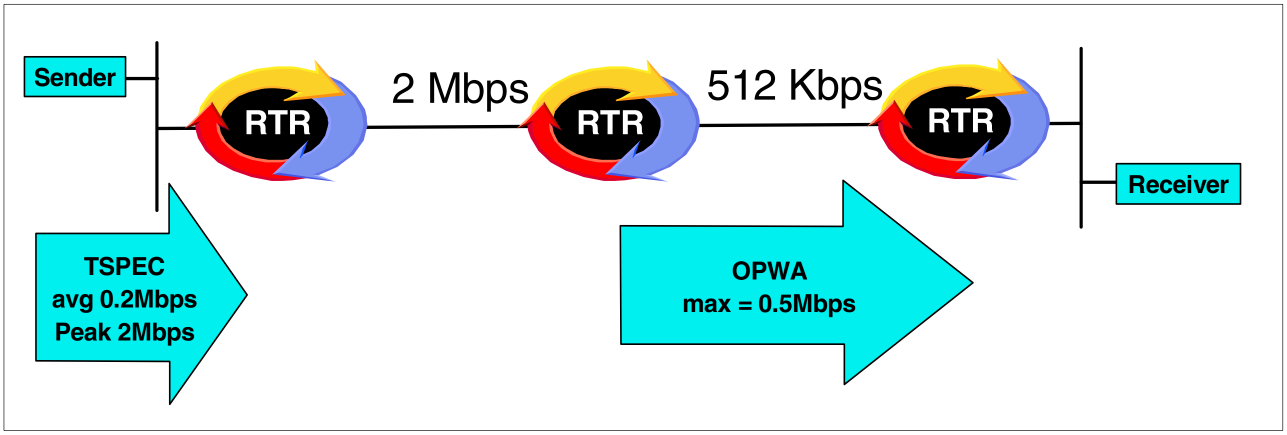


Figure 36. OPWA flow

When a router receives a RESV request it takes several actions:

1. It passes the request to its admission control function to determine whether or not there are sufficient resources to implement the reservation request.

2. It passes the request to its policy control function to determine whether policy rules allow the user to make the reservation request.
3. If both checks succeed, it sets parameters in the packet classifier and packet scheduler functions to implement the requested reservation.
4. If both checks succeed, it provides a confirmation of the successful reservation request to the originator of the request if required.
5. If the reservation request fails, it returns an error message to the appropriate receiver (originator of the RESV request).
6. It forwards the reservation request to the next upstream node in the path.

Figure 37 shows the relationship of all these components in hosts and in routers:

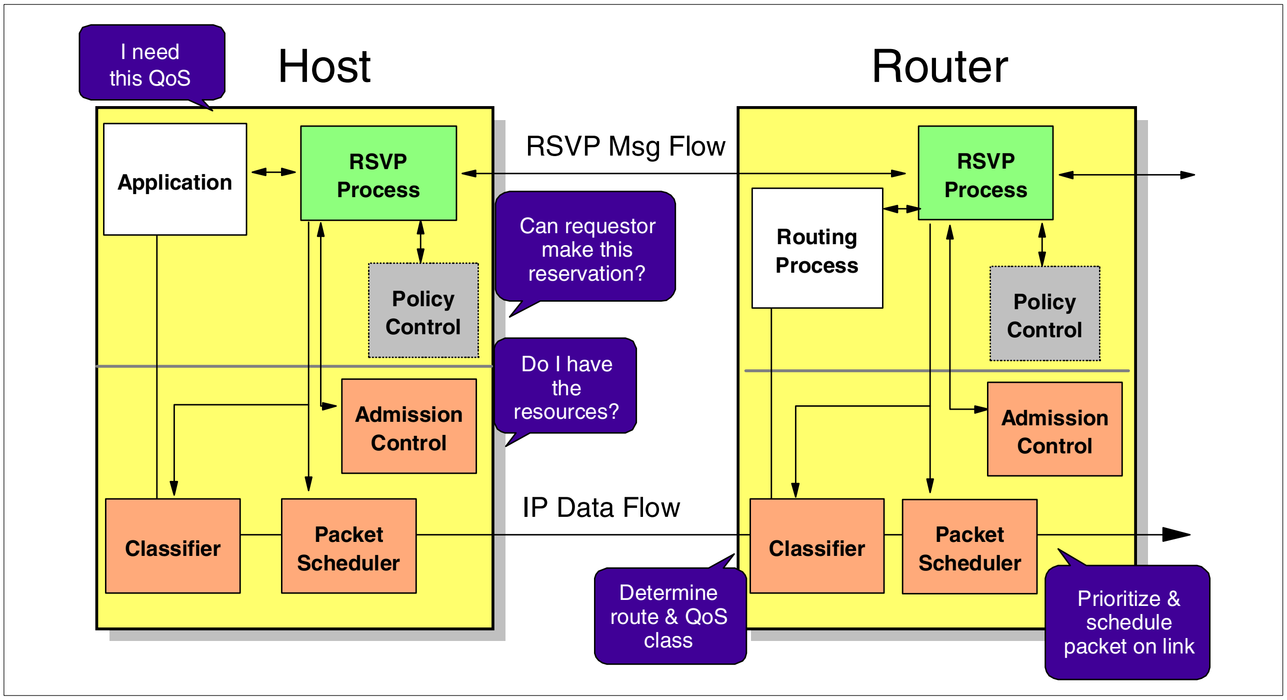


Figure 37. IntServ implementation model

Two final points on the design of RSVP:

1. The implementation of RSVP can be more intricate than even this summary implies; the standard includes methods for merging flows, making shared reservations and includes the ability to cater for multicast applications.
2. PATH and RESV messages are called idempotent messages in RFC 2205. What this means is that they can be used repeatedly to reserve resources for a single flow. This is important when dynamic routing updates are considered. Resource reservations are made along a specific route across a network between sender and receiver which corresponds to the routing table entries of the network devices at the time the PATH message is sent. Subsequent changes in network topology can mean that the actual traffic route changes. RSVP takes a soft state approach to managing the reservation state in hosts and routers by refreshing the reserved path state periodically by sending PATH and RESV messages. Subsequent PATH messages can flow on a new route across the network, and each RSVP router can determine whether the PATH

and RSVP messages it receives are for an existing resource reservation or for a new resource reservation. For those routers no longer in the routed path between sender and receiver, reservation state is deleted if no PATH/RESV messages are received to refresh the reservation before the expiration of a cleanup time-out interval. By default, nodes randomly set their refresh timers between 15 and 45 seconds; the cleanup timer is then set to 157½ seconds to allow up to two successive refresh messages to be lost without deleting the state (which would in fact mean a worst case of 135 seconds between successive refresh messages).

Further observations related to RSVP states should be made:

- RSVP state is dynamic in the sense that hosts can change the parameters in the reservation request (to reserve more bandwidth, for example) simply by sending revised PATH/RESV messages. This implies that routers will recognize the change in an existing reservation and will establish the updated reservation state as long as resources are available and that the new reservation request remains inside the allowed parameters for this particular flow.
- RSVP tear down messages remove the path or reservation states across a network immediately; it is recommended that all end hosts send such requests as soon as an application finishes. The alternative is that the network retains the reservation until the expiration of the relevant times, which can mean the reservation will be preserved unnecessarily (possibly preventing other reservation requests from succeeding) for several minutes.
  In addition to being issued by end hosts, tear down messages may be issued by a router as the result of service preemption[3]. One practical example is in the implementation of a reservation priority scheme in which certain reservation requests have priority over others. A router may decide to grant a high-priority reservation request by destroying an existing reservation with lower priority. In this case the router will issue tear down requests for the lower-priority reservation and send ResvErr messages to the originator of the original RESV reservation request.
- The refresh timer values mentioned earlier can be different for each reservation state. PATH and RESV messages include the refresh period R between the generation of successive refreshes; each node should calculate a time value for the local lifetime of the reservation for each reservation state based on the received value of R. The actual refresh timer will be set to a random value between 0.5R and 1.5R to avoid network disruption by flooding the network with multiple simultaneous refresh messages for different reservations.
- The soft state approach is required because RSVP is not turning IP into a connection-oriented protocol. IP traffic still flows according to the routers' IP routing tables, and changes in network topology can mean that the route of a particular flow through the network can change. The reservations in the network must also be changed when this happens. In normal operation, however, flows will remain on one route and the PATH/RESV refresh messages will duplicate existing reservation states in the routers through which they flow. A question which does not appear to be addressed by the standards is: what actions do routers take when they determine that a PATH or a RESV message is one that refreshes an existing state? In

[3] RFC 2205, Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification, September 1997

particular, do the routers need to refer to the policy database each time? In IBM's implementation, it is fortunate that the policy database is implemented on the same machine as the routing engine, and so what happens is that the policy database is queried even for these refresh messages so that the question, “Can the requester make this reservation?” is asked each time. It may well be that a particular reservation is only allowed to be made between certain hours, therefore an existing reservation can end up being torn down following a refresh message and a query of the policy database. But if the policy database is implemented remotely from the routing engine this may prove an unacceptable overhead and is an indication of the scalability problem in implementing RSVP in networks.

3.3 RSVP and IBM's packet scheduler

The implementation of Integrated Services using RSVP on IBM routers differs from BRS and DiffServ because it can be used over many more types of network interface. The link types that RSVP supports include:

- ATM point-to-point SVC.
- PPP links. RSVP supports PPP over all supported link types, such as V.35,T1/E1, and ISDN, that are established on a permanent basis. Links that are used in dial-on-demand, WAN restoral, short-hold mode, or load-balancing configurations should not be used for RSVP.
- Frame relay PVC. As with PPP, all supported link types will support RSVP, but only links that implement a permanent connection should be used for RSVP. Links that are used in dial-on-demand, WAN restoral, short-hold mode, or load-balancing configurations should not be used with RSVP.
- Frame relay SVC. This is supported in the same way as frame relay PVC; that is, RSVP cannot set up separate DLCIs for QoS traffic but will use part of the default DLCI for QoS bandwidth allocation.
- HSSI
- All LAN links:
  - Ethernet
  - Token-ring
  - FDDI
  - Fast Ethernet

  Note: For shared media networks such as LANs, other methods, such as traffic engineering, are needed to coordinate the sharing of LAN bandwidth. RSVP controls the bandwidth usage of one particular router, but does not coordinate the usage of the LAN bandwidth by multiple routers and hosts. In other words, each router is not aware of the total reserved bandwidth over a shared-medium LAN caused by other routers making reservations over the same shared bandwidth.
- X.25 and ESCON/390. Supported in the same way as PPP or frame relay PVC. RSVP cannot set up separate VCs for QoS traffic; it uses part of the default VC for QoS bandwidth allocation.

RSVP is disabled on PPP or frame relay links configured to use the Bandwidth Reservation System.

3.3.1 Virtual Circuit Resource Manager

The Virtual Circuit Resource Manager (VCRM) is a feature of IBM routers which is automatically enabled when RSVP is configured. There are no configuration parameters for VCRM, but its current status can be displayed.

VCRM determines whether enough bandwidth is available to satisfy a resolution request and creates the connection for the data flow over the physical interface.

If the interface is an ATM interface, VCRM attempts to set up a separate SVC for each RSVP reservation request, and the RSVP reservation is deemed to be successful if the SVC setup succeeds. VCRM then reserves an appropriate number of buffers and sends the actual data packets over the appropriate SVC. The SVC established for the RSVP flow uses the RFC 1483 format. The SVC used for the transmission of best effort traffic (in other words, all traffic which doesn’t form part of an RSVP reservation request) will be established as usual for the connection and can use either RFC 1483 or LANE.

For all other interface types, VCRM schedules all packets on a given outbound link to prioritize packets forming part of flows reserved by RSVP and other packets forming best effort traffic appropriately. This scheduling algorithm is a credit-based scheme which ensures that the RSVP flows are allocated an appropriate percentage of the output bandwidth and that all other traffic uses unreserved and unused bandwidth.

See 3.5.2, “Coexistence” on page 73 for further discussion on how RSVP interoperates with DiffServ.

3.4 RSVP router configuration example

Note

Many 2210 code loads do not include RSVP. For example, no code loads for the 2210-127/128 models include RSVP and only one code load for the 2210-12T/12E models includes RSVP. If you are thinking of configuring RSVP on any of the lower-end 2210 models, first ensure that a suitable code load that matches the memory installed in your 2210 is available.

The following example is taken from an ITSO laboratory exercise in which a pair of 2210 routers were configured to use RSVP over a wide area link.The link between the two routers is PPP at 1 Mbps speed connected via null modem.

There are two possible types of RSVP configuration in our network:

1. One in which all hosts in the network support RSVP, so that although RSVP needs to be enabled in the routers, no specific configuration for the host systems is required in the routers themselves. RSVP-enabled hosts will generate the PATH and RESV messages and the routers will forward these messages and act upon them.
   In practice, the routers will either define local configuration information or, preferably, retrieve it from a central repository database even in this environment, configuring parameters such as the maximum percentage of bandwidth which specific or general flows can reserve. Since we did not have

any RSVP-enabled host applications, we are not showing this sort of configuration here.

2. One in which hosts cannot support RSVP. In the example shown here, the routers themselves are configured with information about how to handle specific traffic flows originating from the non-RSVP-capable host systems.

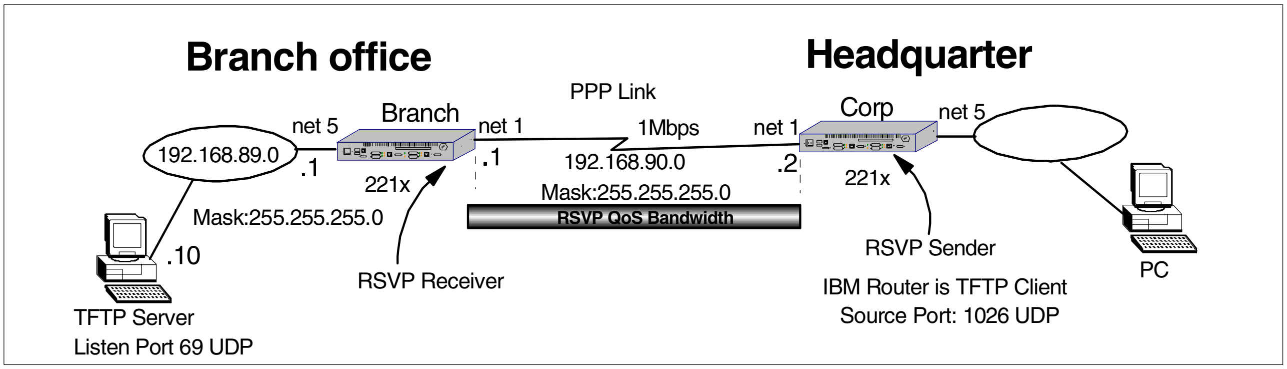


Figure 38. RSVP network diagram

The configuration examples shown below show the configuration of both routers for a TFTP data transfer flow:

- From the right-hand router itself (Corp), IP address 192.168.90.2
- To the TFTP server 192.168.89.10
- From source UDP port 1026
- To destination UDP port 69
- With peak rate 3,000 bytes per second
- With average rate 2,500 bytes per second

The reason for using TFTP in this example is because it is an application whose source and destination port numbers are known in advance (1026 on the client, 69 on the server). One of the problems with configuring RSVP for non-RSVP-aware host applications is that the applications may use varying port numbers for the traffic flows: if FTP were used, for example, we would not know in advance the port number which will be chosen by the FTP client.

3.4.1 Headquarter router (Corp) configuration steps

In this example, both routers are IBM 2210 routers.

1. Configure WAN and LAN interfaces.
   Configure the hostname as Corp. Then configure the PPP interface 1 for V.35 DTE with external clock speed 1 Mbps. In our lab, the WAN PPP link was connected via a null modem. The token-ring medium is STP with 16 Mbps ring speed.

```
Config (only)>SET HOSTNAME Corp
Host name updated successfully
Config (only)>NETWORK 1
Point-to-Point user configuration
Corp PPP 1 Config>SET HDLC CABLE V35 DTE
Corp PPP 1 Config>SET HDLC CLOCKING EXTERNAL
Must also set the line speed to a valid value
Line speed (2400 to 6312000) [0]? 1024000
Corp PPP 1 Config>EXIT
Config (only)>NETWORK 5
Token-Ring interface configuration
Corp TKR Config [5]>MEDIA SHIELDED
Corp TKR Config [5]>SPEED 16
Corp TKR Config [5]>EXIT
```

Figure 39. Configuring network interfaces

2. Configure the IP addresses per the network diagram shown in Figure 38 on page 63. RIP is also enabled in this example.

```
Config (only)>PROTOCOL IP
Internet protocol user configuration
Corp IP config>ADD ADDRESS 1 192.168.90.2 255.255.255.0
Corp IP config>ADD ADDRESS 5 192.168.91.1 255.255.255.0 1
Corp IP config>ENABLE RIP
Corp IP config>EXIT
Config (only)>RESTART y
```

Figure 40. Adding IP addresses to interfaces

1 This interface is not used in this example: our RSVP traffic originates in the router itself.

3. Enable RSVP.
   Enable RSVP in the router using the talk 6 `enable rsvp` command. RSVP can be enabled only on interfaces that are configured for IP. This `enable rsvp` command sets the RSVP router parameters to default values, including 0 as the default bandwidth on the interfaces.

```
Corp *TALK 6
Gateway user configuration
Corp Config>PROTOCOL RSVP
Resource ReSerVation Protocol config console
Corp RSVP Config>ENABLE RSVP
RSVP enabled.
take effect immediately?(Yes or [No]): yes
starting RSVP...

Corp RSVP Config>
```

Figure 41. Enabling RSVP globally

4. Enable RSVP on each interface.
   This command enables RSVP and sets reservable bandwidth on each interface. If the reservable bandwidth for a particular interface is 0 (as is the case for each interface by default), RSVP reservations cannot be made over that interface. Normally the reserved bandwidth should be a small portion of the total link's bandwidth: less than 30% of the total bandwidth is

recommended. You can also use the `set bandwidth` command later to change the bandwidth setting.

```
Corp RSVP Config>ENABLE Interface
Interface [0]? 1
Creating RSVP i/f record...
Set Link Reservable Bandwidth (bits) [0]? 64000

Interface enabled.
  To take effect immediately, use talk-5 RSVP's 'reset interface'

Corp RSVP Config>ENABLE Interface
Interface [0]? 5
Creating RSVP i/f record...
Set Link Reservable Bandwidth (bits) [0]? 64000

Interface enabled.
  To take effect immediately, use talk-5 RSVP's 'reset interface'
Corp RSVP Config>
```

Figure 42. Enabling RSVP on specific interfaces

Use the talk 5 `reset interface` command if you want RSVP to take effect immediately on this interface.

5. Enable One-Pass With Advertising (OPWA)
   OPWA is optional; see the description in 3.2, “Resource Reservation Protocol” on page 56. Use the command `enable opwa` and the interface number for each interface over which OPWA is to be enabled. Be sure to enable RSVP over the interface before you enable OPWA.

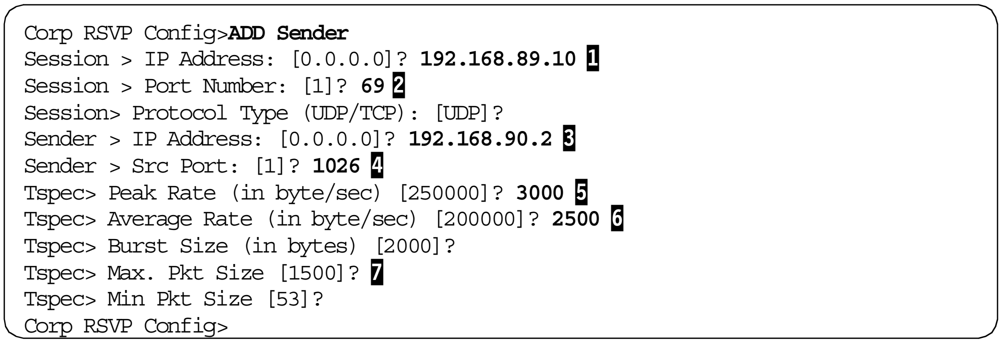
```
Corp RSVP Config>ENABLE OPWA
Interface [0]? 1
Controlled Load installed on interface 1
take effect immediately?(Yes or [No]): yes
Corp RSVP Config>ENABLE OPWA
Interface [0]? 5
Controlled Load installed on interface 5
take effect immediately?(Yes or [No]): yes
Corp RSVP Config>
```

Figure 43. Enabling OPWA (optional)

Note: After completing this step, you can activate RSVP by using the talk 5 `reset rsvp` or the `reset interface` command or restarting the router. At this point, RSVP-enabled applications in hosts that are connected to the router will establish RSVP traffic flows and sessions dynamically.

6. Add Sender
   When there is a host application that is not enabled for RSVP and that sends/receives packets to/from a known IP address and port number, a static sender and receiver can be configured to cause the router to generate RSVP signaling for that flow.

In our example, The Corp router is configured as an RSVP sender using the `add sender` command.



```
Corp RSVP Config>ADD Sender
Session > IP Address: [0.0.0.0]? 192.168.89.10 1
Session > Port Number: [1]? 69 2
Session> Protocol Type (UDP/TCP): [UDP]?
Sender > IP Address: [0.0.0.0]? 192.168.90.2 3
Sender > Src Port: [1]? 1026 4
Tspec> Peak Rate (in byte/sec) [250000]? 3000 5
Tspec> Average Rate (in byte/sec) [200000]? 2500 6
Tspec> Burst Size (in bytes) [2000]?
Tspec> Max. Pkt Size [1500]? 7
Tspec> Min Pkt Size [53]?
Corp RSVP Config>
```

Figure 44. Defining a static sender/receiver for a flow

Notes:

1 In our example, the tftp traffic flow is unicast. The session IP address is the unicast address of the receiver (tftp server) of the IP traffic flow. If the traffic flow were multicast, the session IP address would be the multicast address of the destination of the IP traffic flow.

2 The session port number is the destination port number: in our example, it is the port number used by the receiving tftp server.

3 The sender IP address is the address of the originator of the flow, in this case a tftp file transfer originating from the router itself.

4 The sender source port is tftp client's source port UDP 1026. IBM router's built-in tftp client always uses UDP port 1026.

5 Be careful. This parameter is specified in *bytes* per second, not bits per second.

6 Using this value (2500 byte/sec. = 20 kbps), bandwidth will be reserved.

7 Remember that if this maximum packet size is greater than the MTU of any link on the path between the sender and the receiver, the RSVP request will be rejected and no reservation will be made.

7. The `list all` command is now used to show and verify the values configured in the previous steps:

```
Corp RSVP Config>LIST ALL
Software Version:
RSVP Control: IBM RSVP Router Release 1.0 (RFC 2205)
RSVP Configuration:
RSVP Status:                    Enabled
Maximum RSVP Msg Size:          1500 (bytes)
Refresh Interval:               30 (sec)
Allowed Successive Msg Loss:    3 (frame)
Flow Life-Time:                 158 (sec)
Refresh Slew Max:               30 (percent)
Total system reservable b/w:    4294967 (kbps)
RSVP Interfaces:
If      IP address   RSVP-enabled  Encaps.  max_res_bw SRAM_rec
1       192.168.90.2    Y          IP       64000        1
5       192.168.91.1    Y          IP       64000        2
OPWA configuration:
Network OPWA    CTL-LOAD
1       Y       Y
5       Y       Y
Following senders/receivers are defined in SRAM:
Rec.No   Type     DestAddr      Dest Port  Protocol   Src Addr    Src Port
1    Sender(PATH) 192.168.89.10   69        17       192.168.90.2    1026
                                        [r=2500 b=2000 p=3000 m=53 M=1500]
Corp RSVP Config>EXIT
```

Figure 45. Showing the full RSVP configuration

8. For this example, we also chose to log all RSVP messages to the 2210's Event Logging System (ELS).

```
Corp *TALK 5
CGW Operator Console
Corp +
Corp +EVENT
Event Logging System user console
Corp ELS>NODISPLAY SUBSYSTEM all all
Complete
Corp ELS>DISPLAY SUBSYSTEM rsvp all all
Corp ELS>EXIT
Corp +
```

Figure 46. Sending RSVP messages to ELS

3.4.2 Branch office router configuration

The branch office router is also an IBM 2210 router.

The first five configuration steps are virtually identical to those for the Corp router and will not be shown in detail here:

1. Set up for the V.35 and token-ring interface
2. Configure the IP addresses per the network diagram shown in Figure 38 on page 63.
3. Enable RSVP.
4. Enable RSVP on individual interfaces.
5. Enable One-Pass With Advertising (OPWA).

6. Add Receiver.
   The branch office router is now going to act on behalf of the application which is the recipient of the defined flow. This means that the branch office router has to generate the RESV reservation request, and it does so based on a combination of values received in the PATH message sent from the sending application and from values configured in the router.
   In our example the branch router is configured to act as an RSVP receiver using the `add receiver` command.

```
Branch RSVP Config>ADD Receiver
RESV requestor IP Address: [0.0.0.0]? 192.168.89.10 1
Session > IP Address: [192.168.89.10]? 2
Session > Port Number: [1]? 69
Session> Protocol Type (UDP/TCP): [UDP]?
Style> (WF, FF, SE): [FF]? 3
Need confirmation?(Yes or [No]):
Service Type : CTL-LOAD
Tspec> Peak Rate (in byte/sec) [250000]? 3000
Tspec> Average Rate (in byte/sec) [200000]? 2500
Tspec> Burst Size (in bytes) [2000]?
Tspec> Max. Pkt Size [1500]? 4
Tspec> Min Pkt Size [53]?
Sender > IP Address: [0.0.0.0]? 192.168.90.2
Sender > Src Port: [1]? 1026
Branch RSVP Config>
```

Figure 47. Configuring an RSVP receiver

Notes:

1 The requestor IP address is the address of the host application on whose behalf this router is to act and issue RESV requests: the IP address of the tftp server.

2 The session IP address, session port number and session protocol type parameters must match the parameters configured in the RSVP sender (in Figure 44 on page 66). The receiver and not the sender determines what bandwidth the routers along the path will attempt to establish on each link.

3 Fixed-Filter style reservation does not share resources with others. For multicast application, WF or SE is recommended.

4 The receiver can request a different value for maximum packet size than the sender, in which case the value requested by the receiver will be the one used for the reservation request. OPWA will modify the value actually requested if a link with a lower MTU size is discovered along the path.

7. Again, the `list all` command is used to verify the configuration:

```
Branch RSVP Config>LIST ALL
Software Version:
RSVP Control: IBM RSVP Router Release 1.0 (RFC 2205)
RSVP Configuration:
RSVP Status:                    Enabled
Maximum RSVP Msg Size:          1500 (bytes)
Refresh Interval:               30 (sec)
Allowed Successive Msg Loss:    3 (frame)
Flow Life-Time:                 158 (sec)
Refresh Slew Max:               30 (percent)
Total system reservable b/w:    4294967 (kbps)
RSVP Interfaces:
If      IP address   RSVP-enabled  Encaps.  max_res_bw SRAM_rec
1       192.168.90.1    Y          IP       64000        1
5       192.168.89.1    Y          IP       64000        2
OPWA configuration:
Network OPWA    CTL-LOAD
1       Y       Y
5       Y       Y
Following senders/receivers are defined in SRAM:
Rec.No   Type     DestAddr      Dest Port  Protocol  Src Addr       Src Port
1    Receiv(RESV) 192.168.89.10   69         17      192.168.90.2    1026
                             [r=2500 b=2000 p=3000 m=53 M=1500 sty=FF cfm=N]
Branch RSVP Config>EXIT
```

Figure 48. Verifying the RSVP configuration

8. Set up ELS to record RSVP messages on this router as well, if there is a desire to verify the operation of RSVP using ELS (see Figure 50 on page 70).

3.4.3 Monitoring RSVP

1. In our example, we use the 2210’s own tftp client to generate traffic which matches the RSVP reservation parameters:

```
Corp *TALK 6
Gateway user configuration
Corp Config>BOOT
TFTP Boot/dump configuration
Corp Boot config>TFTP PUT
local filename [CONFIG]?
remote host [0.0.0.0]? 192.168.89.10
host filename [C0A85A02.cfg]? RSVP_test.cfg
TFTP transfer of CONFIG complete, size=65606 status: OK
Corp Boot config>
```

Figure 49. Generating traffic

2. Assuming that all RSVP messages have been sent to the Event Logging System, the following messages can be displayed on the branch router:

```
Branch *TALK 2

RSVP.076: Forward QoS pkt from 192.168.90.2 to 192.168.89.10 prot 17 rt-code=0
RSVP.076: Forward QoS pkt from 192.168.90.2 to 192.168.89.10 prot 17 rt-code=0
RSVP.076: Forward QoS pkt from 192.168.90.2 to 192.168.89.10 prot 17 rt-code=0
RSVP.076: Forward QoS pkt from 192.168.90.2 to 192.168.89.10 prot 17 rt-code=0
RSVP.076: Forward QoS pkt from 192.168.90.2 to 192.168.89.10 prot 17 rt-code=0
RSVP.076: Forward QoS pkt from 192.168.90.2 to 192.168.89.10 prot 17 rt-code=0
```

Figure 50. ELS messages demonstrating RSVP traffic flow

3. Use the `list interface` command to show the current status of interfaces that are using RSVP. The `bwCtrl` state designates a link that is under RSVP bandwidth control; bandwidth can be reserved on this interface for RSVP QoS. The `notCnf` state indicates a link that is not configured for RSVP. The `up` state indicates that a link is configured for RSVP, but the bandwidth is under the control of a link-level QoS function (such as the Differentiated Services feature).

```
Branch RSVP>LIST INTERFACE
RSVP Interfaces:
If      IP address      b/w(K)  res'able  curr-res     state
1/PPP   192.168.90.1    1024    64        0      Kbps  bwCtrl
5/TKR   192.168.89.1    16000   64        20     Kbps  bwCtrl
```

Figure 51. RSVP interfaces on the branch router

```
Corp RSVP>LIST INTERFACE
RSVP Interfaces:
If      IP address      b/w(K)  res'able  curr-res     state
1/PPP   192.168.90.2    1024    64        20     Kbps  bwCtrl
5/TKR   192.168.91.1    16000   64        0      Kbps  bwCtrl
```

Figure 52. RSVP interfaces on the central router

4. Other displays can be used to show the advertisement spec (adspec) of all flows, the status of the QoS flow entries in the RSVP packet classifier, the active RSVP traffic flows, the RSVP senders, reservations, and sessions.

```
Branch RSVP>SHOW ADSPEC
To (Session)    From          Prot DPrt  SPrt #hop  B/W   MTU OPWA-ok
----------------------------------------------------------------------
192.168.89.10   192.168.90.2  UDP  69    1026  1   5500   1500   N
```

```
Corp RSVP>SHOW ADSPEC
To (Session)   From          Prot DPrt  SPrt #hop  B/W     MTU OPWA-ok
----------------------------------------------------------------------
192.168.89.10  192.168.90.2  UDP  69    1026  0  Unknown    1500  N
```

```
Branch RSVP>SHOW CLASSIFIER
        ====== RSVP Packet Classifier Entries ======
 hkey src            dest           pro gsi     oif nexthop       vcc-hndl m q
 793  192.168.90.2   192.168.89.10   17 4020045  5  192.168.89.10        0  0 Q
```

```
Corp RSVP>SHOW CLASSIFIER
        ====== RSVP Packet Classifier Entries ======
 hkey src            dest           pro gsi     oif nexthop      vcc-hndl m q
 793  192.168.90.2   192.168.89.10  17 4020045  1  192.168.90.1         0 0 Q
```

```
Branch RSVP>SHOW RSVP FLOWS
Number of flows:        1
Num To (Session)    From            Prot DPrt  SPrt In-If Out-If Rsvd Nhop's
----------------------------------------------------------------------------
1   192.168.89.10   192.168.90.2    UDP  69    1026  1     5      Y    1
```

```
Corp RSVP>SHOW RSVP FLOWS
Number of flows:        1
Num To (Session)    From            Prot DPrt  SPrt In-If Out-If Rsvd Nhop's
----------------------------------------------------------------------------
1   192.168.89.10   192.168.90.2    UDP  69    1026  6     1      Y    1
```

```
Branch RSVP>SHOW RSVP SENDERS
Number of RSVP senders :        1
TO              From           Prot DPrt  SPrt  PHOP           PLIH  If FH Lfe

------------------------------------------------------------------------------
192.168.89.10   192.168.90.2    UDP 69    1026  192.168.90.2    0     1  Y -1
```

```
Corp RSVP>SHOW RSVP SENDERS
Number of RSVP senders :        1
TO              From           Prot DPrt  SPrt  PHOP         PLIH   If  FH Lfe

------------------------------------------------------------------------------
192.168.89.10   192.168.90.2    UDP 69    1026  192.168.90.2  510000 6   Y -1
```

```
Branch RSVP>SHOW RSVP RESERVATIONS
Number of senders:      1
To              From            Prot DPrt  SPrt  Style NHOP          If RH Life
------------------------------------------------------------------------------
192.168.89.10   192.168.90.2    UDP  69    1026  FF    192.168.89.10  5  Y 180
```

```
Corp RSVP>SHOW RSVP RESERVATIONS
Number of senders:      1
To               From            Prot DPrt  SPrt  Style NHOP          If  RH Life
--------------------------------------------------------------------------------
192.168.89.10   192.168.90.2    UDP  69    1026  FF    192.168.90.1    1   Y 180
```

```
Branch RSVP>SHOW RSVP SESSIONS
Number of sessions: 1
Num  To              DPort Prot  Style  Num-senders   Refresh-time-slot
-------------------------------------------------------------------------
1    192.168.89.10   69    UDP   FF     1             16
```

```
Corp RSVP>SHOW RSVP SESSIONS
Number of sessions: 1
Num  To              DPort Prot  Style  Num-senders   Refresh-time-slot
------------------------------------------------------------------------
1    192.168.89.10   69    UDP   FF     1             0
```

Figure 53. Various RSVP displays on both routers

3.4.4 RSVP and S/390 host systems

A prototype version of CS for OS/390 support for RSVP is available in conjunction with CS for OS/390 V2R7 (generally available in March 1999) at:

http://www.software.ibm.com/enetwork/commserver/downloads/demos/csos390.html

CS for OS/390 V2R8 (available in September 1999) includes enhancements to the service policy agent (see 2.8, “CS for OS/390: service policy agent and LDAP server” on page 49) to add support for RSVP. This release also adds support for a policy API which allows applications to make policy queries; in this case the support is for the RSVPD user (daemon) which queries RSVP policies to admit or deny incoming and/or outgoing reservation requests. So the RSVPD daemon can either change reservation parameters or deny a reservation altogether based on parameters defined in the service policy agent.

3.5 Practical considerations: IntServ and DiffServ

The biggest problem with the concepts behind the Integrated Services model and the implementation capabilities offered using RSVP is that too much faith has been placed in them. IntServ offers a method of reserving resources for particular types of network flow, which in turn means that it may enable the more efficient transport of certain types of traffic across IP networks. What IntServ does not offer, however, is a general method for the efficient and appropriate transport of all sorts of IP traffic. IntServ and RSVP must be used carefully and appropriately. In many large networks, DiffServ offers a more appropriate, efficient, and scalable model.

3.5.1 Realistic implementation

The IntServ model of resource reservation across the network using RSVP has practical difficulties of implementation and mean that this approach is not always the most appropriate for all types of traffic and all types of network. RFC 2208[4] has been created by the IETF to attempt to document the sorts of network environment for which RSVP is appropriate.

A particular concern covers the aspect of scalability of RSVP networks:

> The resource requirements (processing and storage) for running RSVP on a router increase proportionally with the number of separate sessions (i.e., RSVP reservations). Thus, supporting numerous small reservations on a high-bandwidth link may easily overly tax the routers and is inadvisable. Furthermore, implementing the packet classification and scheduling capabilities currently used to provide differentiated services for reserved flows may be very difficult for some router products or on some of their high-speed interfaces (e.g. OC-3 and above).
>
> These scaling issues imply that it will generally not be appropriate to deploy RSVP on high-bandwidth backbones at the present time. Looking forward, the operators of such backbones will probably not choose to naively implement RSVP for each separate stream. Rather, techniques are being developed that will, at the “edge” of the backbone, aggregate together the streams that require special treatment. Within the backbone, various less costly approaches would then be used to set aside resources for the aggregate as a whole, as a way of meeting end-to-end requirements of individual flows.

The document concludes:

> Given the current form of the RSVP specifications, multimedia applications to be run within an intranet are likely to be the first to benefit from RSVP. SNA/DLSW is another “application” considered likely to benefit. Within the single or small number of related administrative domains of an intranet, scalability, security and access policy will be more manageable than in the global Internet, and risk will be more controllable. Use of RSVP and supporting components for small numbers of flows within a single Internet Service Provider is similar to an intranet use.
>
> General issues of scalability, security and policy control... are the subjects of active research and development, as are a number of topics beyond this applicability statement, such as third-party setup of either reservations or differentiated service.

If users of the network are themselves making resource reservation requests, it is vital to be able to ensure that these requests can be policed and controlled. As it stands at the writing of this redbook, very few applications and hosts are capable of RSVP signaling. This position will change over time, not least because Microsoft is developing support for RSVP in the new Windows 2000 platform, although it is not clear how many applications will take advantage of this support even then.

3.5.2 Coexistence

An increasingly popular model is one in which IntServ and DiffServ networks coexist; RSVP is used to make resource reservations across peripheral networks;

[4] RFC 2208, Resource ReSerVation Protocol (RSVP), Version 1 Applicability Statement, Some Guidelines on Deployment

but because of the issues of scalability and controllability in the core networks, DiffServ is used here. In this model, edge routers mark traffic arriving over IntServ flows with appropriate DiffServ settings in the service type field in the IP header before transmitting them across the DiffServ core of the network:

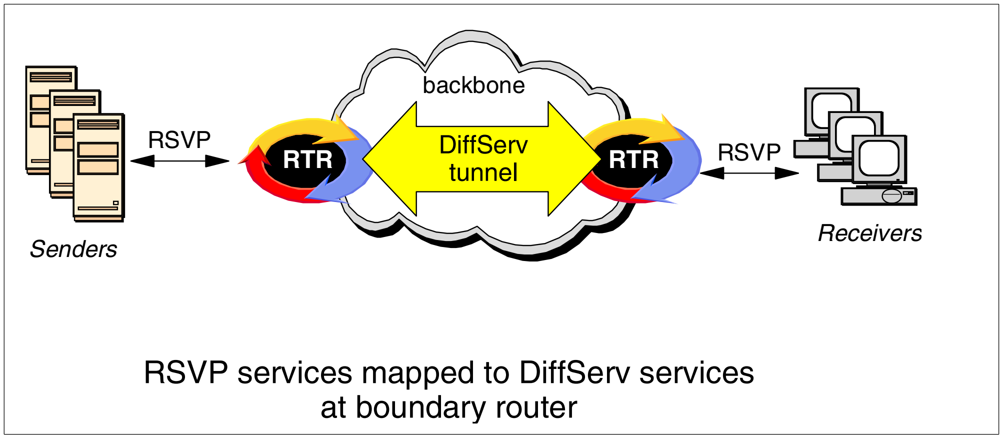

backbone
RSVP
RTR
DiffServ tunnel
RTR
RSVP
Senders
Receivers
RSVP services mapped to DiffServ services at boundary router

Figure 54. Coexistence of IntServ and DiffServ networks

In an IBM router in which RSVP reservations are made over frame relay or PPP interfaces, policy rules can be defined so that a DiffServ action is associated with each flow. Although BRS and RSVP cannot both be enabled on one of these interfaces, DiffServ and RSVP can.

If RSVP and DiffServ are to be used on the same outbound interface, the following observations can be made:

- If the interface leads to a non-RSVP-capable network, it is still possible to establish RSVP reservations across this network provided that both endpoints of the proposed flow do support RSVP. As described in 3.2, “Resource Reservation Protocol” on page 56, PATH requests are ignored by non-RSVP-capable routers, so from the perspective of the RSVP reservation, the non-RSVP-capable network looks like a single router-router hop.
- The existence of non-RSVP-capable routers can usually be determined by RSVP-capable routers. A router receiving a PATH message can usually establish that the previous RSVP router that forwarded the PATH message is not in fact its immediate upstream neighbor router by comparing TTL values in the IP header of the PATH message with those contained in the RESV message itself. (Other methods are sometimes required, occasionally requiring manual configuration in the boundary router.) Information in the form of a NonRSVP flag is added to the ADSPEC information to make the ultimate receiver of the PATH request aware that the reservation path traverses one or more non-RSVP-capable routers.
- In a router in which DiffServ is enabled on an interface, VCRM no longer controls the scheduling of the outbound packets but instead passes all packets to the DiffServ scheduler. Care must then be taken to match the DiffServ and RSVP configurations for that router. For example, take a case in which 30% of the output bandwidth of a router has been reserved using RSVP. If all of these flows are then mapped to the DiffServ expedited forwarding queue (see 2.5, “The Differentiated Services feature” on page 37), then the default allocation

of 19% of the output bandwidth to expedited forwarding traffic on this interface will need to be changed.

- Although an end-to-end reservation may be in place across the entire network, no end-to-end guarantee can be made for transit time or delay across the network. As noted above, the situation is communicated to the host which is ultimately going to issue the RESV reservation request, and therefore, policy decisions on whether or not to issue a reservation request can be made by the receiver.
- It is not currently possible to tunnel RSVP requests across an intervening network: the PATH/RESV messages are not hidden from the intervening network. If two RSVP-capable networks are connected with a network whose components are not enabled for RSVP, there is no problem because the RSVP setup messages are ignored by the intervening network. If, however, the intervening network's routers also have RSVP enabled, these routers are compelled to respond to all PATH and RSVP requests that flow through them. This could be a problem if the intervening network is configured to deny RSVP requests originating from other networks, as might well be the case.

Chapter 4. Summary

Early implementations of routers offered a stateless method to differentiate treatments of IP packets based on easily detectable differences between IP packets. Stateless here means that each different IP packet is treated separately and uniquely and is not dependent on the action of a preceding packet. An early candidate as a discriminating indicator was the UDP or TCP port number, because different types of IP traffic tends to use predefined well-known ports, so that for example, SNA traffic encapsulated in IP packets using data link switching could be given priority over other types of IP traffic throughout the network.

Differentiated Services offers another stateless approach to differentiation between different types of IP traffic by defining a consistent interpretation of a byte in the IP header. Every device that receives an IP packet can examine this byte, modify this byte, and take other actions dependent on the content of this byte.

The latest interpretation of this service type byte treats it as a six-bit numerical value and assigns different interpretations to each of the 64 different values.

Earlier interpretations of this service type byte treated it as a three-bit numerical priority indication (therefore giving potentially eight different priority values) and four bit settings indicating type of service indicators.

In practice, little use was ever made of the type of service indicators.

Consistency means that all network components are able to derive the same interpretation from a specific bit setting, and some confusion has come about because of the changing standards which have apparently reinterpreted the contents of this byte.

In reality, there is little cause for alarm and the two main standards (RFC 791 and RFC 2474) are not necessarily incompatible because:

- Routers interpreting packets according to the newer standard are required to offer a consistent, downward-compatible interpretation of the first three bits, formerly used to indicate priority. So a network implemented according to the newer standard can cater for devices which mark packets according to the old standard.
- The code points defined in the new Differentiated Services architecture still use the first three bits of the field to discriminate between the major forwarding classes, and hence if network routers are implementing the older standard they will end up prioritizing between these major classes even though they may not implement additional DiffServ features such as the provision of an expedited queue.

It should also be noted that all IBM implementation of access controls and DiffServ actually treat the service type byte as a collection of eight bits so that any interpretation can be placed on any of the bit settings. Some confusion arises because of the terminology used: often terms such as TOS and priority are still used in configuration programs and reference material. Ultimately, the only requirement is that any implementation of DiffServ is consistent across the network, and this can always be achieved if routers do not force a specific meaning on any bit setting in the byte. The attraction of the latest RFC is that it describes a convention which is useful when discussing Differentiated Services and may be necessary for interoperability between manufacturers' implementations. IBM's implementation of Differentiated Services allows compliance with the conventions outlined in both old and new RFCs without forcing compliance.

Integrated Services offers a different model which is going to be more difficult to implement. It seems to present a connection-oriented model for IP traffic, but it is built on top of the connectionless infrastructure provided by IP routers and has to retain the flexibility offered by dynamic routing protocol updates enabled by RIP and OSPF in today's networks. The danger with the Integrated Services model is that it is not scalable in the same way as Differentiated Services; it is unlikely that all networks are going to have RSVP enabled and it is extremely unlikely that RSVP and resource reservations offer a solution to all types of network traffic. Short-lived flows, such as many of those between Web browsers and Web servers will not benefit from this approach. On the other hand, some special types of network flows may only be possible with the use of RSVP, and although many application hosts are today incapable of requesting reservations through the network using RSVP signaling, this will change shortly.

Part 2. Class of service in Ethernet networks

Chapter 5. Overview

When computers started to connect to each other over local area networks (LANs), various standards emerged defining the physical methods for these connections. IBM developed token-ring, but Ethernet has grown in popularity, for the primary reason it has been good enough. Ethernet is simple and cheap, and has over 80% of the market in terms of installed LAN ports.

One comparison which explains the direction the marketplace has taken is a comparison of the cost per user and the cost per megabit per second over three types of LAN infrastructure:

Table 9. LAN infrastructure relative costs

Infrastructure type	Cost per user	Cost per Mbps
10 Mbps shared Ethernet	$61	$245
16 Mbps shared token-ring	$471	$126
10/100 Mbps switched Ethernet	$108	$2

One interpretation of this table is that Ethernet grew rapidly in environments in which the cost per user was important, such as in branch office locations. Token-ring remained strong in data centres in which high volumes of data traffic were seen. Newer Ethernet technology is now replacing token-ring even in these environments.

One of the reasons for the relatively low cost of Ethernet is its simplicity, and for many years the additional complexity and expense of token-ring was not appropriate to the majority of environments in which Ethernet was installed. This situation is now changing rapidly.

One of the fundamental components of token-ring architecture which has not been present in Ethernet is the concept of priority. Token-ring stations have the capability of transmitting frames at a specific access priority, and this priority information is retained in the token-ring frame itself. A similar implementation is also available for FDDI LANs.

As stated above, Ethernet has been good enough for many data networks, but now that LANs are being used for the transmission of voice and video traffic as well as data traffic, the absence of a priority queueing and priority signaling mechanism for Ethernet is a serious one. Without the implementation of the standards described in this part of the book, Ethernet is only going to be suitable for the transmission of today’s network traffic by overproviding network bandwidth.

One example of the different types of traffic which could be present on a single LAN defines the following seven traffic types:

1. Network control traffic, vital to the support of the network itself
2. Voice traffic, requiring a delay of less than 10 milliseconds through the network
3. Video traffic, requiring a delay of less than 100 milliseconds
4. Controlled load, defining applications which benefit from the definition of a flow through the network

5. Excellent effort, in which customers who pay more get a better service from the network
6. Best effort, which is how LAN traffic as we know it is currently handled
7. Background, or other traffic which should be allowed to use the network without affecting the use of the network by other users and applications

The following chapters will describe three standards or proposals, a subset of which can be described as:

802.1p A standard which provides the ability of LAN bridges to provide expedited traffic capabilities that support the transmission of time-critical information in a LAN environment

802.1Q A standard which provides the ability of all LAN media to include user priority information as part of the MAC frame

Jumbo frames A proposal for a standard allowing the transmission of MAC frames of up to 9022 bytes over full-duplex Gigabit Ethernet connections

It is important to note that the first two standards are applicable to all LAN media types, and not restricted specifically to Ethernet LANs[1]. It is specifically with Ethernet, though, that the real benefits are to be gained, and therefore most mention of these standards is seen today in the context of Ethernet devices. The IEEE 802.1 standards are common specifications applicable to “Overview and Architecture, and Interworking (including bridging and Virtual bridged LAN (VLAN))”. In the context of today’s LAN switches, adherence to 802.1 standards refers to the methods these switches use to forward layer-2 packets; in their simplest form, LAN switches are multiport LAN bridges.

[1] IEEE 802 LANs include ISO/IEC 8802-3 (CSMA/CD), ISO/IEC 8802-4 (Token Bus), ISO/IEC 8802-5 (token-ring), ISO/IEC 8802-6 (DQDB), ISO/IEC 8802-9 (IS-LAN), IEEE Std. 802.11-1997 (Wireless), ISO/IEC DIS 8802-12 (Demand Priority) and ISO 9314-2 (FDDI) LANs.

Chapter 6. 802.1p

Although IEEE 802.1p is often quoted as being the standard to which devices conform, it is in fact now part of the 802.1D standard named:

> Information Technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Common specifications - Part 3: Media Access Control (MAC) Bridges (Incorporating IEEE P802.1p: Traffic Class Expediting and Dynamic Multicast Filtering)

In general terms, 802.1D is the standard which determines how LAN bridges process frames and how bridges forward frames between their LAN ports. In the current environment, this equates to how LAN switches perform the same task, and therefore, the 802.1p additions to the standard define how these devices honor Class of Service indications received in LAN frames and how these devices apply priority discriminations between different frames. This is a process which takes place at the layer-2 level, and is therefore independent of the higher-layer network protocol in the frame, and in particular is unaffected by IP Class of Service mechanisms described earlier in this book.

IEEE nomenclature is the reason for using a lowercase p in 802.1p: 802.1p is not itself an IEEE standard but is part of the 802.1D standard. (802.1Q is a full standard in its own right, and therefore merits an uppercase Q.)

6.1 Traffic class expediting

Expedited traffic is defined as traffic that requires preferential treatment as a consequence of jitter, latency, or throughput constraints, or as a consequence of management policy. A bridge or a switch[1] is a device with multiple LAN ports which forwards frames between its ports. In its simplest form, the process of relaying MAC frames between bridge ports can be thought of as a combination of:

- Frame reception, in which a frame is received on a port
- Forwarding, in which decisions are made as to which ports, if any, are to be used for forwarding the frame
- Frame transmission, in which a frame is transmitted over a port onto another LAN segment

The forwarding process provides one or more transmission queues for every port in the bridge; the essence of the ability of a bridge to support expedited classes of traffic over a given port is that a bridge can support more than one queue (or traffic class) for the port.

6.1.1 Frame reception

When a frame is received on a bridge port, the user priority associated with the frame is regenerated by the bridge. If the LAN medium is one which has the ability to contain user priority in the MAC frame, such as token-ring, it would be usual for the regenerated user priority to take the value directly from the MAC frame. For Ethernet, however, no priority information is contained in the MAC frame, and therefore, the bridge will define the regenerated user priority value based on the configuration of the bridge itself, which will normally comprise a

[1] For the remainder of this section the term bridge should be taken to mean bridge or switch. The standard refers to the bridging operation of a device which will probably be called a LAN switch.

default priority value for the port on which the frame is received. The bridge may be configured to assign different default user priority values to different LAN ports, and this is a simple method of prioritizing between different LAN devices which are not themselves capable of signaling a user priority value.

6.1.2 Frame forwarding

The frame forwarding process decides whether to discard the received frame or to forward it on one or more ports; one input to this decision process is the filtering database. In an Ethernet transparent bridge, MAC addresses are learned and associated with bridge ports, and this information is stored in the filtering database and used to determine on which port (if any) the received frame should be transmitted.

The frame forwarding process may also provide more than one transmission queue for a given bridge port; it is not necessary to provide eight transmission queues for a bridge port, nor is it necessary to provide the same number of transmission queues for each port.

Each transmission queue is assigned a traffic class value in the range 0 to N-1, where N is the number of queues associated with a given outbound port. The user priority value determined by the frame reception process is then mapped to a traffic class value. The 802.1p standard provides a recommended mapping table that maps the received user priority value to a traffic class value dependent on the number of transmission queues implemented; it has the characteristic that if four or more queues are implemented, received frames which carry the default user priority of 0 are not mapped to the lowest-priority outbound queue.

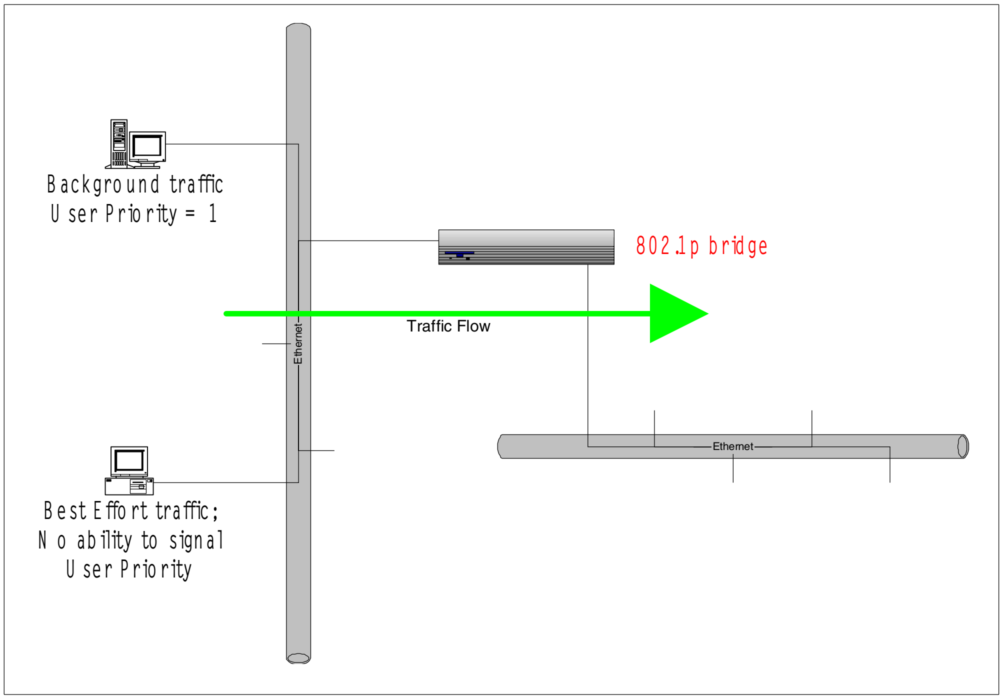


Figure 55. Preferential treatment for default user priority

Consider the figure above, in which a LAN device which has no ability to signal a user priority value over Ethernet is connected to the same LAN segment as a server that is performing a bulk file transfer and is capable of signaling an appropriate user priority (leaving aside, for now, how this signaling is actually possible). If the bridge implements 802.1p with at least four outbound transmission queues, traffic from the LAN device will be in fact prioritized over the batch transmission, which is desirable. This is because the signaled Background Traffic priority indication maps to a user priority value of 1, which in turn gets mapped to traffic class 0; the default user priority of 0 gets mapped either to a priority class of 1 or 2.

The defined traffic types and acronyms are shown in the following tables, which show how different traffic types are mapped into different numbers of outbound transmission queues:

Table 10. Traffic type acronyms

User priority	Acronym	Traffic type
1	BK	Background
2	-	Spare
0 (Default)	BE	Best Effort
3	EE	Excellent Effort
4	CL	Controlled Load
5	VI	Video
6	VO	Voice
7	NC	Network Control

Table 11. Defining traffic types

<table>
<tr><th>Number of queues</th><th colspan="8">Defining traffic type (in increasing priority order from left to right)</th></tr>
<tr><td>1</td><td colspan="8">BE</td></tr>
<tr><td>2</td><td colspan="4">BE</td><td colspan="4">VO</td></tr>
<tr><td>3</td><td colspan="4">BE</td><td colspan="2">CL</td><td colspan="2">VO</td></tr>
<tr><td>4</td><td colspan="2">BK</td><td colspan="2">BE</td><td colspan="2">CL</td><td colspan="2">VO</td></tr>
<tr><td>5</td><td colspan="2">BK</td><td colspan="2">BE</td><td>CL</td><td>VI</td><td colspan="2">VO</td></tr>
<tr><td>6</td><td colspan="2">BK</td><td>BE</td><td>EE</td><td>CL</td><td>VI</td><td colspan="2">VO</td></tr>
<tr><td>7</td><td colspan="2">BK</td><td>BE</td><td>EE</td><td>CL</td><td>VI</td><td>VO</td><td>NC</td></tr>
<tr><td>8</td><td>BK</td><td>-</td><td>BE</td><td>EE</td><td>CL</td><td>VI</td><td>VO</td><td>NC</td></tr>
</table>

6.1.3 Frame transmission

The 802.1p standard defines a default algorithm in which frames are only selected for transmission if all queues corresponding to numerically higher values of traffic class are empty at the time the selection is made. This algorithm has the advantage of simplicity, but the standard does not preclude the use of an alternative algorithm provided that some basic rules on frame ordering are followed. This would allow the implementation of some kind of fair queueing algorithm for the selection of frames for transmission, for example.

Once a frame has been selected for transmission its user priority needs to be mapped to the access priority on the outbound port. It is especially important to note that in the case of Ethernet (without 802.1Q) this outbound access priority will be 0 for all frames. This mapping priority is defined for all 802.2 LAN types and is not modifiable in any way.

In the case of token-ring LANs, outbound access priority will take a value between 0 and 6. This access priority is used to reserve a token of the required priority and therefore gain transmission priority over lower-priority users on the ring. This process is a standard component of token-ring architecture. The fact that an access priority value of 7 is not used means that seven and not eight different access priority values can be contained in token-ring MAC frames; since these priority values are used to regenerate the user priority values in subsequent bridges this means that - without the application of the 802.1Q standard - token-ring LANs are capable of signaling seven different values of priority across the entirety of the bridged network. A priority value of 7, higher than the highest user priority value of 6, has to be reserved for certain MAC frames such as the Active Monitor Present frame. Because user priority values of both 6 and 7 are actually mapped to an access priority value of 6, this leads to a situation similar to the last row of Figure 11 on page 85 but with Voice and Network Control traffic both mapped to the same priority.

6.1.4 Reality check

Although a switch is a special type of bridge, it tends to be the case that LAN switches are designed to operate at full media speed and to act in cut-through mode wherever possible. In this mode, frames are transmitted on eligible outbound ports before the frame reception process has completed. In such an environment, the concept of outbound priority queues has no meaning, since no traffic waits for transmission and needs to be queued behind other traffic. In a campus LAN environment in which a switch connects similar LAN segments, for example, a workgroup switch used to connect multiple 10 Mbps shared Ethernet segments, much of the 802.1p standard will not have great significance. Where it starts to make sense is an environment in which queues build up, for example, in an environment in which many user LANs connect to a single backbone LAN. But even here, the network may be designed with a higher-speed backbone, for example, 100 Mbps Ethernet, and again frames may not need to be queued.

In the case of LAN frames that contain user priority information, including both token-ring and Ethernet-plus-802.1Q, cut-through switching is only possible if the LAN frame checksum (FCS) field is not modified. This is simply because the switch does not check the FCS value, because it is retransmitting the LAN frame on an outbound port before it has even received the FCS value in the LAN frame on the inbound port. Bridges must recalculate the FCS if any data in the frame is changed; this includes the user priority indication. Any LAN switch that is to operate in cut-through mode, therefore, will neither queue traffic on outbound ports nor will it modify the user priority information in the received frame.

6.2 Dynamic multicast filtering

To step back for a moment, it should be observed that both the following features are optional components of the 802.1D standard:

- The ability to support expedited traffic on any port

- The support of extended filtering services

Conformance with 802.1p might imply the ability to support both of these features, but because 802.1p is not a formal standard it would be sensible to examine any claims to support it carefully. Bridges can conform to the actual 802.1D standard and support one, both, or neither of the above features. For example, the announcement letter for the 8275-217/225 correctly states that it supports “part” of 802.1p; and in fact this means that it provides support for the extended filtering services option of the 802.1D standard.

Support of extended filtering services implies that bridges (and endstations) dynamically register and deregister group membership information using the Generic Attribute Registration Protocol (GARP) Multicast Registration Protocol (GMRP). These groups in turn lead to the provision of filtering of frames by bridges so that frames addressed to a given group are forwarded only on those LAN segments that are required in order to reach the members of that group.

GMRP serves to register a group MAC address, which in turn causes a bridge’s filtering database to be updated to show which LAN ports should be used for transmission of MAC frames destined to the group address.

Extended filtering services, GARP and GMRP are not directly related to the issue of Class of Service but they allow the implementation of services across the LAN which may in turn require the implementation of expedited traffic capabilities. For example, the ability to define multicast groups across a network may in turn make it feasible to transmit streamed voice and video across the network to members of the multicast group, but only if the LAN bridges are capable of discriminating and prioritizing between different traffic types.

Chapter 7. 802.1Q

The 802.1Q standard's full title is "IEEE Standards for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks". It defines an architecture for virtual bridged LANs in which all 802 LAN types have the capability of:

- End-to-end signaling of user priority information regardless of the intrinsic ability of the underlying MAC protocols to signal user priority information
- Provision of Virtual LAN (VLAN) services, including the definition of frame formats used to represent VLAN identification information

The 802.1Q defines that this additional information is carried in the MAC frame in the format of a tag. The tag is an additional header which is inserted in the MAC frame immediately following the destination and source MAC address fields, and following the routing information if this is present. When applied to Ethernet LANs, the tag header is four bytes in length, comprising:

- A two-byte Tag Protocol Identifier (TPI) containing the value 81-00
- A two-byte Tag Control Information (TCI) field which in turn comprises:
  - Three bits of user priority information
  - A single bit Canonical Format Indicator (CFI)
  - Twelve bits denoting the VLAN Identifier (VID)

See Table 13 on page 96 for more details on the layout of a complete Ethernet frame which includes these two tag fields.

7.1 Tagged, untagged and priority-tagged frames

Frames which do not contain a tag header are known as untagged frames.

Frames which include a tag header which carries a null VLAN ID are known as priority-tagged frames.

VLAN-tagged frames are defined as frames containing *both* priority information *and* a non-null VLAN ID.

802.1Q requires the association of a specific VLAN ID with each of its ports, known as the Port VLAN Identifier (PVID). The PVID provides a VID value for untagged and priority-tagged frames received on a given port. This PVID means that all frames received in VLAN-aware bridges are assigned a non-null VLAN ID. The 802.1Q standard describes a system for classifying untagged and priority-tagged frames as belonging to a particular VLAN based on parameters associated with the receiving port; proprietary extensions to the standard allow these frames to be classified based on the data content of the frame instead - based on MAC address or layer-3 protocol, for example. In other words, 802.1Q defines port-based VLANs, but protocol-based VLANs or MAC address-based VLANs are not precluded by specific implementations of extensions to the standard.

When a bridge transmits frames, it must transmit them either in the untagged or VLAN-tagged format. A bridge cannot transmit priority-tagged frames; if the frame is tagged it must contain a non-null VID in its tag header.

Each bridge port can be set to:

- Admit only VLAN-tagged frames
- Admit all frames

The default value is to admit all frames, otherwise all untagged or priority-tagged frames that are received will be discarded.

7.1.1 Access, trunk and hybrid Links

A trunk link is a LAN segment on which all devices must be VLAN aware. Conversely, an access link is a LAN segment on which all frames carry no VLAN identification. A hybrid link is one which carries both VLAN-tagged frames and other (untagged or priority-tagged) frames simultaneously.

In its simplest incarnation, VLAN-aware bridges are connected by a trunk link. Traffic over this link is VLAN-tagged and - if there is contention for the link - user priority ensures that traffic is prioritized appropriately across the link. Note that in the following figure, because user devices on both access links are using Ethernet and are not using VLAN-tagged frames, user priority is actually determined by the VLAN-aware bridges based on their configuration information.

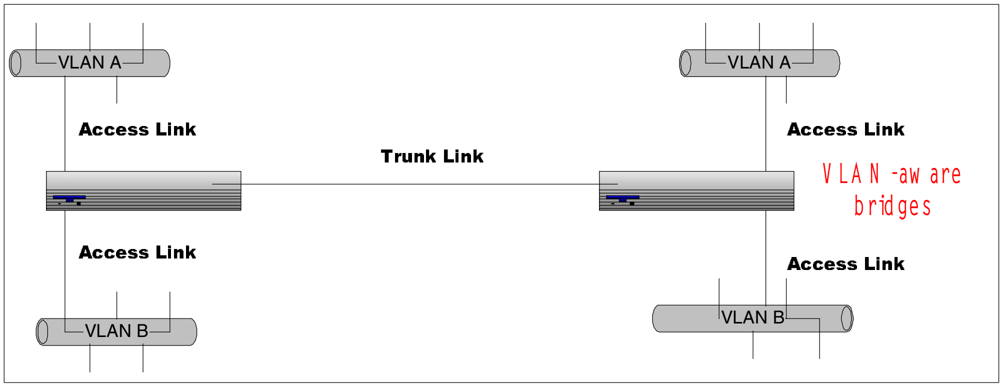


Figure 56. Port-based VLANs and trunk links

For any given VLAN, all frames transmitted by a given bridge on a given hybrid link must be tagged in the same way on that link. In the following figure, all the frames for VLAN A are VLAN-tagged on the hybrid link whereas all the frames for VLAN B are untagged on the same link.

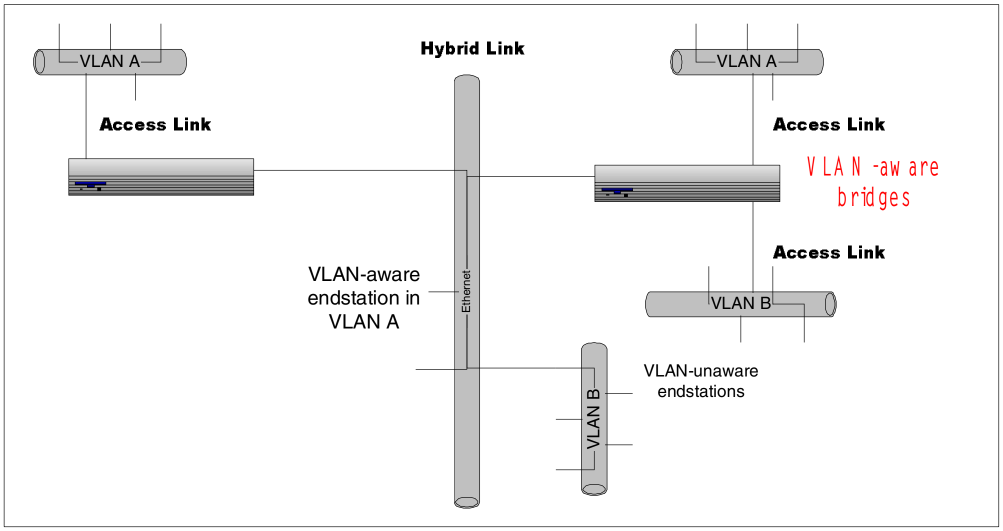


Figure 57. Hybrid links and access links

7.2 User priority information

The user priority of a frame received by a bridge is now defined by the following rules:

1. If the received frame is a tagged frame or a priority-tagged frame, the value contained in the tag header is used. Otherwise;
2. The value of the priority information received in the MAC frame is used, optionally modified by the User Priority Regeneration Table.

The user priority information is then used by the forwarding process of the bridge, assuming that it supports the independent 802.1p standard defined in Chapter 6, “802.1p” on page 83.

When the frame is subsequently transmitted on an outbound LAN port, the access priority determined by 802.1p is used if appropriate. In addition, if the bridge transmits the frame in tagged format then the user priority information will be inserted in the tag header of the transmitted frame.

In its simplest form, where a bridge supports VLAN-tagged frames on ingress and egress ports, this means that the user priority information contained in the received tagged frame will be preserved across the bridge and therefore across the entire 802.1Q-capable network.

In its next simplest form, where a bridge only supports a single LAN media type but converts between untagged frames on one link and tagged frames on another link, user priority information may be included in MAC frames arriving on the access link and therefore included in the forwarded frame (but not if the inbound link is an Ethernet link). Otherwise this information will only be included in the outbound frame if the bridge is capable of defining rules to generate user priority values for these frames.

To signal user priority information across an Ethernet network, endstations and bridges must all support 802.1Q. There is no other standard mechanism for signaling priority across an Ethernet network.

7.3 VLAN services

The 802.1Q standard defines a 12-bit VLAN Identifier (VID) to identify the VLAN to which the frame containing the tag header belongs. The VID is considered an unsigned binary number; 0 is used to represent the null VID and 1 is used to represent the default port VID (PVID). This leaves the remaining number space between 2 and FFF available for assignment as VID values, which allows the assignment of up to 4094 unique VID values. Devices do not have to support as many as 4094 concurrently active VLANs, nor do they have to support the allocation of VIDs across the entire 2-FFF number space.

7.3.1 Filtering database

VLAN information is stored in the filtering database of the bridge. There are four different processes for storing VLAN information in the filtering database:

1. Static VLAN registration entries are created by explicit configuration of the bridge.
2. Dynamic VLAN registration entries are created, updated, and removed by the Generic Attribute Registration Protocol (GARP) VLAN Registration Protocol (GVRP). GVRP provides a mechanism for VLAN-aware endstations to register their membership of VLANs across a network.
3. Static filtering entries represent static information for individual and group MAC addresses and are used to allow administrative control of frame forwarding to particular destination addresses.
4. Dynamic filtering entries are created and updated by the bridge’s normal learning process by inspecting received frames and associating their VID with the port on which they are received.

The VLAN information which is stored in the filtering database comprises the VLAN identifier and a port map showing which outbound bridge ports are appropriate for each VLAN.

7.3.2 Member set and untagged set of bridge ports

For each VLAN, the information from the static and dynamic VLAN registration entries in the filtering database are used to define:

1. The member set of ports through which members of the VLAN can currently be reached
2. The untagged set of ports through which members of the VLAN can be reached but through which frames must be transmitted without tag headers

7.4 Progress of a frame through an 802.1Q bridge

802.1Q defines additional processes defined for bridges that apply to VLANs:

7.4.1 Frame reception

All frames received on a bridge port are assigned a VID value. If the frame did not originally contain a VID and the port was not set to “admit only VLAN-tagged frames”, then the bridge has to use an algorithm to assign a VID value.The default algorithm is to use the Port VID (PVID) value configured for that port, but alternative algorithms may be used.

7.4.2 Frame filtering

The bridge filtering process now uses the VID information in addition to the normal destination MAC address information to determine which ports should be used for forwarding the frame.

7.4.3 Frame forwarding

Frames are queued for transmission on outbound ports, and if multiple transmission queues are provided a queueing mechanism based on the user priority of the frame will be used (as in 802.1p). The frame will either be transmitted as a VLAN-tagged frame or as an untagged frame. The choice is made based on the port’s membership of either the member set or the untagged set.

7.5 Default configuration of 802.1Q bridges

Since 802.1Q alters the format of LAN frames, it is imperative that devices that do not understand the altered format do not receive tagged frames. The default configuration of 802.1Q bridges addresses this in the sense that such a bridge can be inserted in an existing network without any special preconfiguration.

The default configuration of an 802.1Q bridge is so that it:

- Admits all frame types on all ports and therefore will not discard untagged or priority-tagged frames
- Sets the port VID for all ports to the value of the default PVID, 1
- Contains a static VLAN registration entry for the VLAN corresponding to the default PVID which specifies both:
  - Fixed registration, which means that this registration cannot be overridden by dynamic VLAN registration entries as a result of GVRP
  - Untagged frame forwarding over all ports of the bridge

7.5.1 Modification of the default configuration

If more than one VLAN-capable device is in the network, some modification of the default configuration will be necessary.

One common modification will be to use 802.1Q for the transmission of user priority information across the network but not to implement VLANs. To transmit user priority information over Ethernet networks, tagged frames must be used. As discussed earlier, bridges cannot transmit priority-tagged frames, that is, tagged frames containing the null VID.

The following figure shows a simple network in which all the transmitting devices are VLAN aware but which are only using 802.1Q in order to transmit user priority information across the entire network:

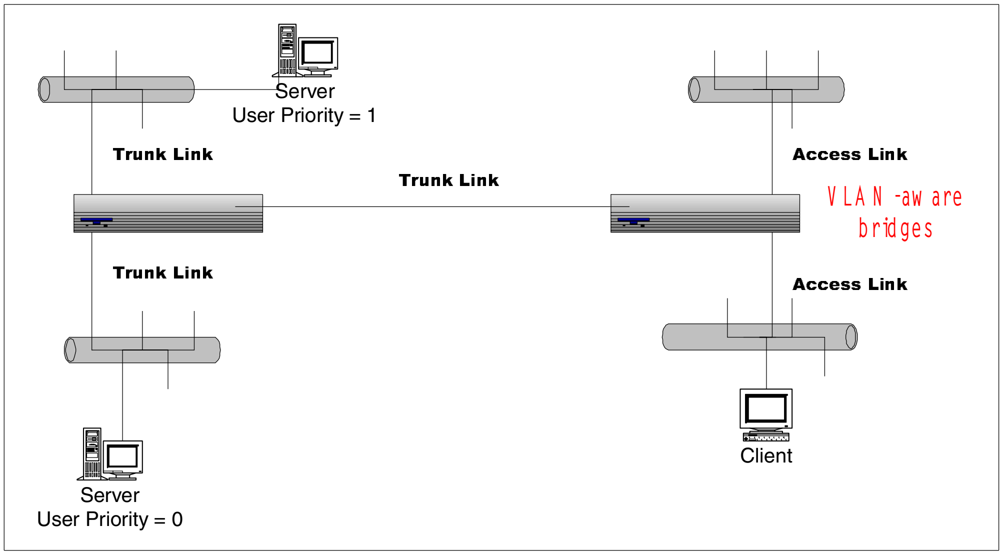


Figure 58. Use of 802.1Q to transport user priority indications

If the default configuration is modified by changing the static registration entry for the default PVID to use VLAN-tagged frames on designated trunk ports, then all frames over these links will flow with the same VID but will now transport user priority values. In Figure 58, traffic from the two left-hand servers to any number of clients on the right-hand side of the network will be treated differently in the left-hand bridge. The two servers can transmit priority-tagged frames which contain appropriate user priority values but with the null VID in the tag header. The bridge will use the priority information to queue the outbound traffic and will retain the priority information in the frames transmitted over the link to the right-hand bridge, with the difference that these frames will now contain a non-null VID value. The right-hand bridge will also queue these frames appropriately before transmitting them to the client; in this network these frames will finally be transmitted as untagged frames over the final access link.

As soon as VLANs are defined in the network, extra care and consideration have to be given to the configuration of bridges in the network. Configuration has to be consistent across the network, and essentially requires that the tagging behavior of all VLAN-aware bridges needs to be the same for any given VLAN.

7.6 LAN types other than Ethernet

A considerable proportion of the 802.1Q standard is devoted to defining how the standard is implemented on LAN types other than Ethernet. We do not intend to do justice to this work here but make the following observations:

- VLAN tagging is the part of the standard which is equally applicable to all LAN types, since any other methods of indicating membership of specific VLANs are proprietary.
- User priority information and VIDs can be transported across mixed-media networks provided that all bridges support the 802.1Q standard.

- The Canonical Format Indicator (CFI) in the tag header is used in different ways:

  1. In token-ring frames it is used to signal the bit order of the address information in the frame; when set to 1 it indicates that all MAC address information is in noncanonical format, which is the default or native format used for token-ring LANs. Otherwise, when set to 0, it indicates that canonical format is being used.

  2. In Ethernet frames it is used to signal the presence or absence of a Routing Information Field (RIF). This capability allows traffic which originates in and is destined for a source-routed environment to transit as VLAN-tagged traffic across a non source-routed environment.

- The total combination of possibilities of frame transformations encompasses:

  - Eight different combinations of LAN services:

    1. Ethernet Version 2 or LLC
    2. Canonical or noncanonical format
    3. Frames containing source-routing information or those that are bridged transparently

  - Two different basic LAN types:

    4. 802.3/Ethernet
    5. Token-ring/FDDI

  - Two different VLAN environments:

    6. Untagged
    7. Tagged

  These combinations lead to a theoretical possible 96 different one-way bridging functions between the 32 possible frame/encapsulation formats. Fortunately these only all come into play in bridges that connect to multiple LAN types simultaneously. Most of today's LAN switches only connect to a single type of LAN medium. Furthermore, it's likely that the majority of these Ethernet devices will either not be required to support source-routing or will simply be incapable of it, and will similarly only require to support canonical address formats. So the practical number of combinations reduces drastically for any specific implementation of the 802.1Q standard.

7.7 Ethernet frame sizes

Ethernet frames are made up of several components:

Table 12. Traditional Ethernet frame format

Field length (bytes)	Field description	Field contents
7	*Preamble*	*56 bits of 10101...*
1	*Delimiter*	*10101011*
6	Destination MAC Address	Standard allows 2 bytes as well
6	Source MAC Address	Also can be 2 bytes in length
2	Length/type	Length of Ethernet frame
n	Data	

Field length (bytes)	Field description	Field contents
p	Pad	
4	Frame Checksum	

The length of an Ethernet frame is defined as the number of bytes between the start of the destination MAC address field and the end of the frame checksum field; in other words, not including the preamble and delimiter fields (which is the reason for showing these two fields in italics in these tables). The Ethernet standards dictate that:

1. Frames must not be smaller than 64 bytes in length, and the pad field must be used to increase the size of Ethernet frames which would otherwise be too small.
2. Frames must not be larger than 1518 bytes in length, which in turn defines a maximum length for the data field itself of 1500 bytes.

802.1Q adds a four-byte frame to the existing Ethernet frame, resulting in a total frame layout for tagged frames as follows:

Table 13. Tagged Ethernet frame format

Field length (bytes)	Field description	Field contents
7	*Preamble*	*56 bits of 10101...*
1	*Delimiter*	*10101011*
6	Destination MAC Address	Standard allows 2 bytes as well
6	Source MAC Address	Also can be 2 bytes in length
2	Length/type	Length of Ethernet frame
2	Tag Protocol Identifier	8100 in hexadecimal
2	Tag Control Information	User Priority/CFI/VID
n	Data	
p	Pad	
4	Frame Checksum	

The 802.1Q does not mandate any changes to the existing rules for minimum and maximum frame sizes, but the following considerations and standards should be taken into account.

7.7.1 Minimum frame size

802.1Q defines a variation on the rule for minimum frame size in the case of tagged frames. Implementations are allowed to choose between two options for tagged frames:

1. Strict conformance with the letter of the 802.3 standard by imposing a minimum frame size of 64 bytes for tagged frames.
2. The addition of bytes in the pad field of the tagged frame to ensure that such frames have a minimum size of 68 bytes.

The reason for the second approach proposed above is that it allows for direct conversion between tagged and untagged frames without the need for the addition or the removal of pad bytes from the frame. This may lead to a more efficient operation of the bridge.

7.7.2 Maximum frame size

Again, 802.1Q imposes no requirement on the increase in the maximum frame size for tagged frames. The implication of making no change in the maximum allowed frame size, however, is that the maximum size of the user data field in a tagged Ethernet frame is reduced by four bytes to 1496 bytes. In turn, this means that legitimate untagged frames of greater than 1514 bytes in length (in which the user data field is greater than 1496 bytes in length) cannot be converted to tagged frames because the length of these tagged frames would now exceed 1518 bytes. This restriction may prove unacceptable to Ethernet endstations, or may at least require the reconfiguration of many existing such devices to prevent them transmitting frames which can no longer be processed by the network.

To circumvent this problem, IEEE standard 802.3ac changes the maximum frame size for tagged frames to 1522 bytes. Devices which conform to this standard can now take untagged Ethernet frames with a full 1500-byte data payload and convert these frames to a tagged frame with the same data payload.

It would be normal to expect to see that Ethernet devices which conform to the 802.1Q standard also conform to the 802.3ac standard. Indeed, although it is technically incorrect, it is now normal usage to refer to the increased 1522-byte maximum frame size as being part of the 802.1Q standard and therefore as part of any implementation which complies with the standard.

7.7.3 Frames containing source-routing information

If the tag’s CFI indicates that source-routing information is included, this information is included as an Embedded Source-routing Information Field (E-RIF) and is present between the end of the frame’s Length/type field and the start of the frame’s user data field. The E-RIF is between two and 30 bytes in length. No change is made to the Ethernet frame’s maximum allowed frame size to take the E-RIF into account, so the presence of the E-RIF field can mean that the maximum allowed user data field is as low as 1468 bytes.

In a network in which two source-routing environments are connected over an Ethernet network that supports 802.1Q tagging to carry routing information across the network, this will probably result in the token-ring endstations reverting to a maximum user data size of 516 bytes, which is the smallest maximum frame size allowed according to the 802.2 standard. This is also the only code point value defined for token-ring networks lower than 1500.

Most implementations of 802.1Q will be concerned either with networks comprised solely of Ethernet segments or with networks in which even if there are portions which use source-routing, the routing information does not need to be transported across the Ethernet portion of the network. In either of these cases, user data of up to 1500 bytes can be transported across the Ethernet networks provided that all devices which use trunk links also support 802.3ac.

7.8 802.1Q implemented in endstations

So far we have talked about how the 802.1Q standard applies to bridges, and therefore to LAN switches. Table 15 on page 107 shows the current range of IBM products which implement 802.1Q, and this list also includes LAN adapters for endstations of various types.

The following example shows a case in which a server must implement 802.1Q in order for port-based VLANs to be defined correctly. The example is one in which three different LAN segments connect to a single switch; the three different user segments comprise devices which do not implement 802.1Q and therefore the switch implements port-based VLANs to separate the traffic. The reason for using the switch is to provide connectivity to a server, and the requirement is to provide access to the server from all three separate VLANs and at the same time using VLANs to isolate users on one VLAN from users on another VLAN:

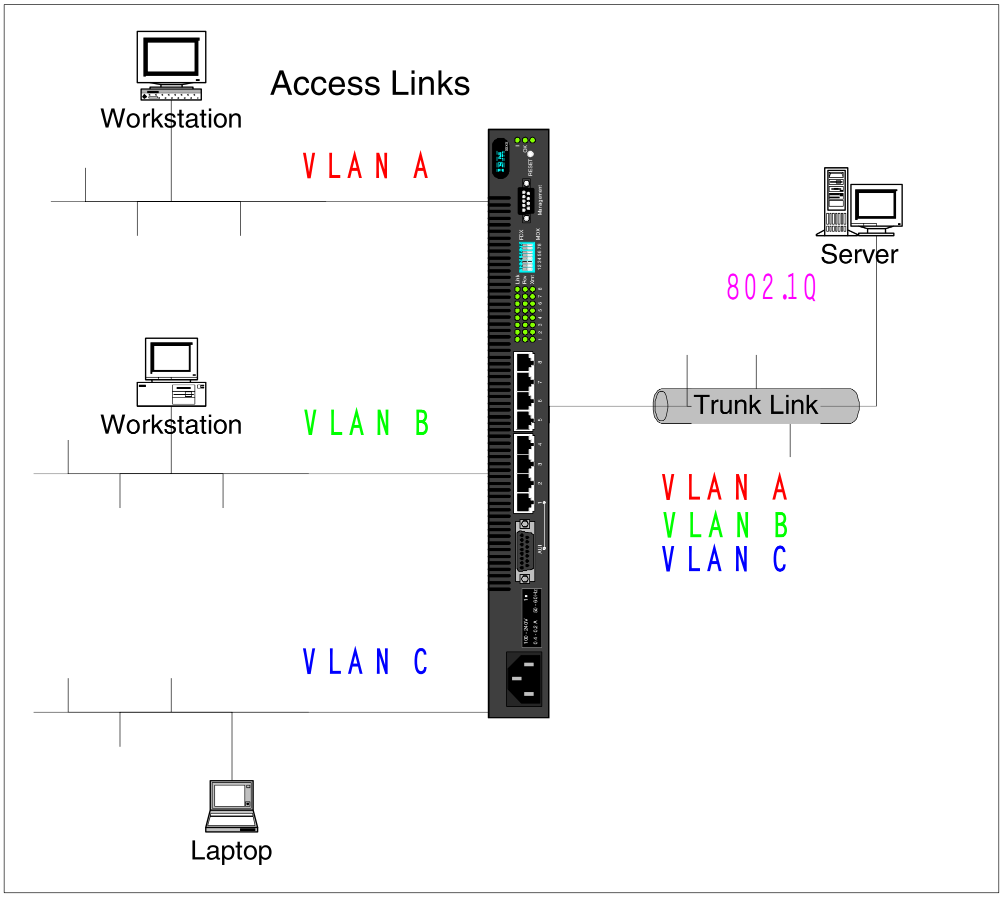


Figure 59. An example of port-based VLANs

If the switch in Figure 59 implements port-based VLANs according to 802.1Q then the server must also implement 802.1Q in order to implement the network as shown. The server will need to provide a mechanism for tagging frames which it sends over the trunk link (so that, for example, frames destined for the laptop computer on VLAN C are tagged with this information).

The reason for this follows a consideration of how the switch is to treat packets received from the server. If these are untagged, then the port-based VLAN rules in the switch mean that all these frames will be given the same VID. If, for example, this is for VLAN A, then the server will only be able to send frames to devices on VLAN A and it will be unable to send frames to devices on VLANs B or C.

There are three ways to implement this network using VLANs:

1. Connect the server to the switch with three separate connections, each of them defined in a different VLAN. All ports on the switch send frames in untagged format. This solution is simple to implement because it maintains the strict separation of VLANs inside the switch but requires no implementation of 802.1Q outside the switch. It comes with the price of additional complexity in the server.

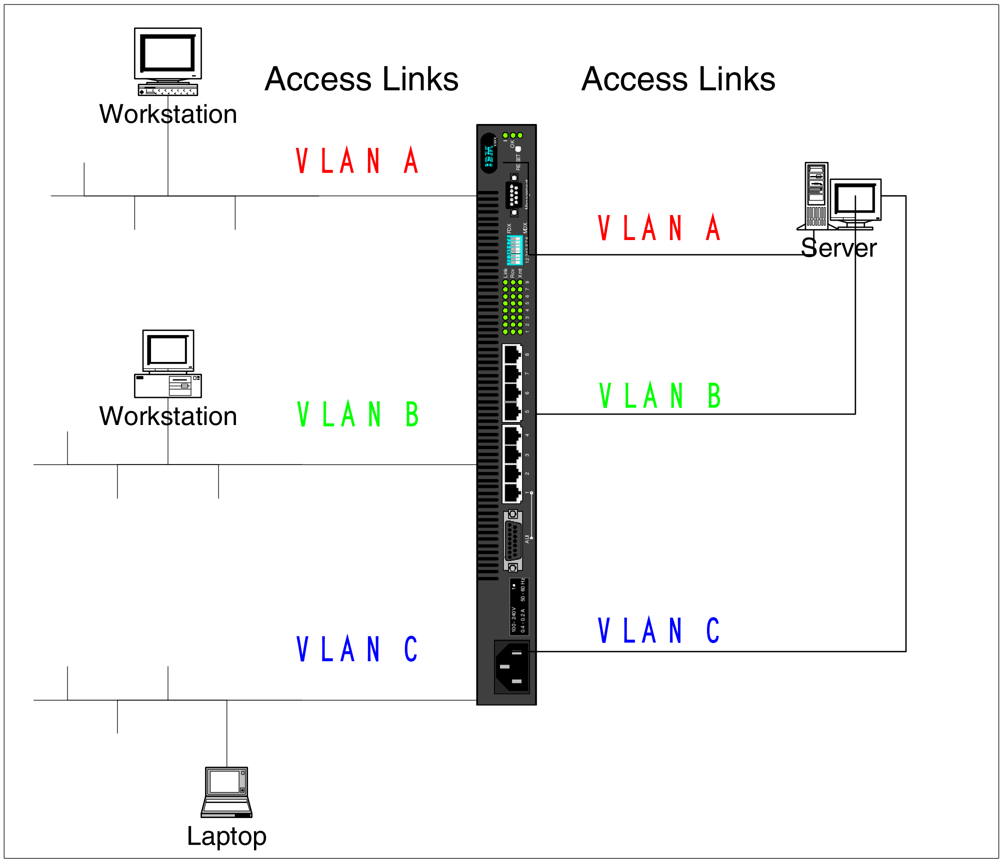


Figure 60. Port-based VLANs inside a switch

2. Connect the server to the switch over a single link provided that 802.1Q is enabled on the server and configure both the switch and the server to send frames over this connection in VLAN-tagged format. Frames received from the server will contain the correct VID. This description corresponds to the one shown in Figure 59 on page 98.
3. Implement a different policy of VLAN classification. An example would be one based on MAC addresses instead of LAN ports. In our example, if the server is capable of implementing multiple MAC addresses on a single LAN connection then a VLAN classification policy based on the source MAC address in the received frame would achieve our purpose. Port-based VLANs A, B, and C continue to be defined for the user ports as before. Port D has to be defined so

that frames destined for the server are transmitted over it (this could be achieved by putting port D in the member set and the untagged set of all three VLANs) and that frames received over this port are given a VLAN ID appropriate to the source MAC address received.

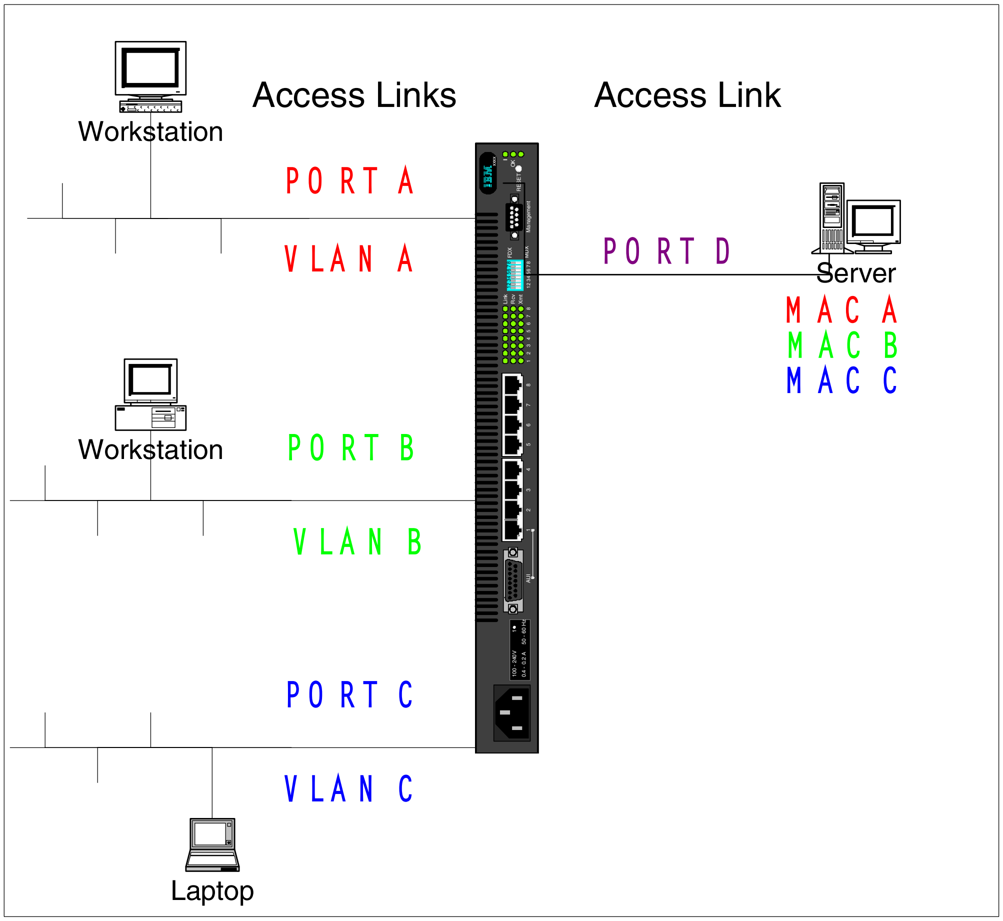


Figure 61. Use of other VLAN styles (a hypothetical example)

The VLAN classification rules proposed for Figure 61 start to get somewhat complex: essentially they have to allow frames to flow between the server port and any of the other ports but not between two non-server ports.This is a hypothetical example of what could be done; 802.1Q does not prevent the implementation of rules such as these. Implementation of anything other than port-based VLAN classification rules is a proprietary extension to the 802.1Q standard, which does not mean it is bad, but it does mean that every implementation of such rules is likely to be different.

Chapter 8. Gigabit Ethernet and jumbo frames

Gigabit Ethernet is defined by the 802.3z standard which defines the additions to the basic Ethernet standard required for transmission of frames at the rate of 1,000 Mbps. All of the considerations of the previous two chapters apply to Gigabit Ethernet except for the potential modifications to the minimum and maximum frame size rules.

8.1 Gigabit Ethernet minimum frame size

Gigabit Ethernet conforms to the rules of IEEE 802.3 and can be implemented on a CSMA/CD shared network in the same way as 10 Mbps and 100 Mbps Ethernet networks. Devices sharing the network must be able to detect when other devices are using the network and are only able to transmit when the common network medium is inactive. Transmission of frames as short as 64 bytes in length at Gigabit Ethernet speeds does not allow for reliable detection of this transmission by competing devices, and therefore, 802.1z requires that frames shorter than 512 bytes in length are increased to 512 bytes by the addition of an extension field, which follows the frame checksum, as shown in the following table:

Table 14. 802.3z Gigabit Ethernet frame format

Field length (bytes)	Field description	Field contents
7	*Preamble*	*56 bits of 10101...*
1	*Delimiter*	*10101011*
6	Destination MAC Address	Standard allows 2 bytes as well
6	Source MAC Address	Also can be 2 bytes in length
2	Length/type	Length of Ethernet frame
2	Tag Protocol Identifier	8100 in hexadecimal
2	Tag Control Information	User Priority/CFI/VID
n	Data	
p	Pad	
4	Frame Checksum	
e	Extension	Required to extend frame size to 512 bytes for half-duplex CSMA/CD operation

This table assumes that 802.1Q tags are being used.

To reduce the overhead of adding relatively large amounts of otherwise unnecessary data to small data frames, 802.1z also allows an individual station to use burst mode in which it can send up to 8 kB of additional data in additional frames following the first frame and in which the subsequent frames do not require the addition of the extension field.

This discussion of minimum frame size is somewhat academic, because it only applies to the use of Gigabit Ethernet as a shared medium LAN. All practical implementations of Gigabit Ethernet implement it as a point-to-point protocol used between pairs of switches or between switches and user devices. When implemented as a point-to-point link, the connection is implemented as full duplex; there are no collisions to detect and therefore the minimum frame size reverts to the 64 byte minimum frame size required in all other Ethernet implementations. It should be observed that if Gigabit Ethernet were to be implemented as a shared medium LAN, the minimum frame size requirement effectively reduces its throughput to that of 100 Mbps Ethernet anyway.

8.2 Gigabit Ethernet maximum frame size

Gigabit Ethernet conforms with standards previously discussed, and therefore, the maximum frame size will be one of 1522 bytes if 802.1Q and 802.3ac standards are supported. Alteon Networks, a manufacturer of Gigabit Ethernet adapters and device drivers, has proposed the implementation of a larger maximum frame size of 9022 bytes for use on full-duplex connections. This proposal is also known as the proposal for jumbo frame sizes. This would allow a data packet of 9000 bytes to be carried in each Gigabit Ethernet frame.

The reason behind this proposal is that because every frame in a network incurs approximately the same assembly and transmission overhead, a given amount of data can be transmitted through the network faster and more efficiently if fewer individual frames are used, and therefore, the largest possible frame size should be used wherever possible.

Although this is not yet even a draft standard, IBM makes use of Alteon hardware and software in some of its products and therefore, it also happens that some of the IBM Gigabit Ethernet implementations support jumbo frames.

To make use of these larger frames, all devices between and including both layer-2 partners must support the ability to transport jumbo frames. It is necessary for the devices which originate traffic to do so using these frame sizes. In a network such as the one shown in Figure 56 on page 90 in which Ethernet LAN segments are connected over a full-duplex trunk link, even if this link is a Gigabit Ethernet link that supports jumbo frames there will never be any frames greater than 1522 bytes in length on this link. Nor could jumbo frames be used by any device in Figure 57 on page 91 since there are no full-duplex links in this network.

The following figure shows a network in which jumbo frames can be used. It requires that the client and server both support jumbo frames, that both the client and server are connected to their local switch over a full-duplex Gigabit Ethernet link, that the switches themselves support jumbo frames and that the link between the two switches is itself a full-duplex Gigabit Ethernet connection.

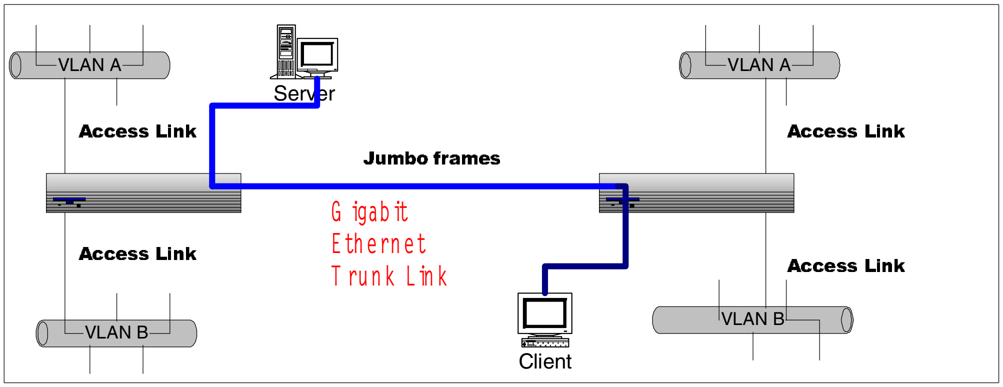


Figure 62. A network which uses jumbo frames

Chapter 9. Summary

The two standards 802.1p and 802.1Q are related but different; between them they implement a layer-2 mechanism for transmitting and respecting priority information across an Ethernet network. Now that these standards have emerged, their implementation is effectively required in all new Ethernet LAN switches and related products, or in software upgrades to existing products. In addition, the 802.1Q standard specifies how LANs can be partitioned into multiple virtual LAN segments.

9.1 The relationship between 802.1p and 802.1Q

The two standards cover different aspects of common LAN standards but with a large overlap which can best be summarized in the following diagram:

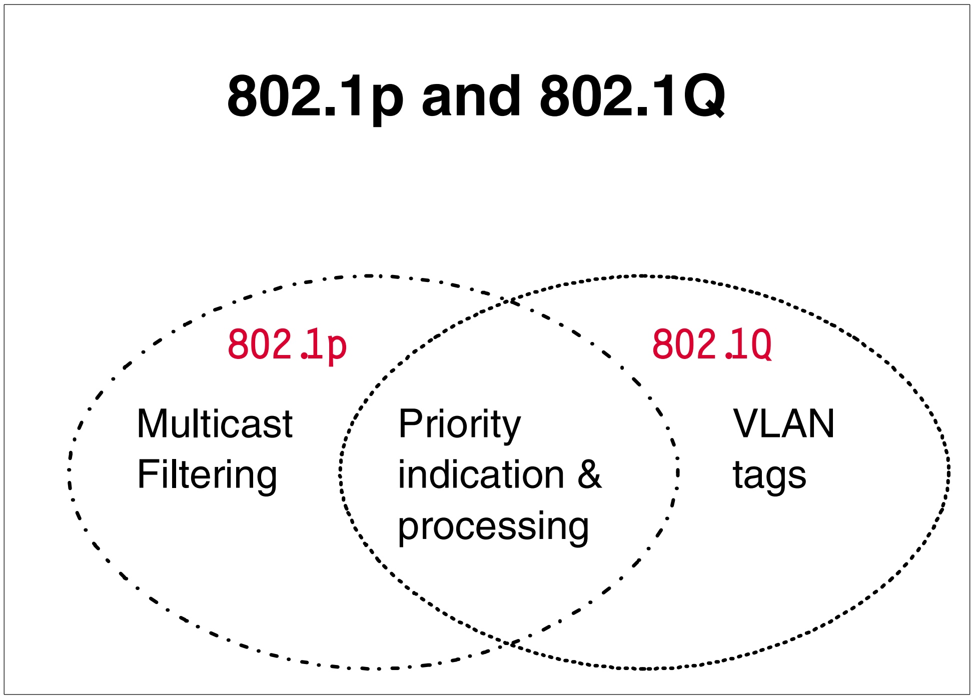


Figure 63. Initial comparison and overlap between standards

In theory this could imply that VLAN tags be implemented in bridges or switches without implementing the bridges or switches to provide expedited traffic capabilities, in other words, 802.1Q without 802.1p. More likely is that a bridge would implement 802.1p without the ability to handle VLAN tags, which means that traffic can be prioritized inside a given bridge but that no traffic prioritization indication can be carried in Ethernet frames.

In reality, the overlap between the two standards is such that it makes most sense to implement both of them in a given bridge or switch rather than just one of them.

However, some devices may not implement the entirety of both standards. And, to confuse matters further, some components of 802.1p are extended by 802.1Q: 802.1Q extends the priority handling aspects of 802.1p to make use of the ability of the VLAN frame format to carry user priority information end to end across any set of concatenated underlying MACs. According to this definition, conformance with the priority handling aspects of 802.1p is implicit in the 802.1Q standard.

A better approach, therefore, is to consider 802.1Q as an extension of 802.1p, but remember that some components of 802.1p, and therefore, by implication of 802.1Q. are optional to any device conforming to either standard. This leads to the more accurate picture:

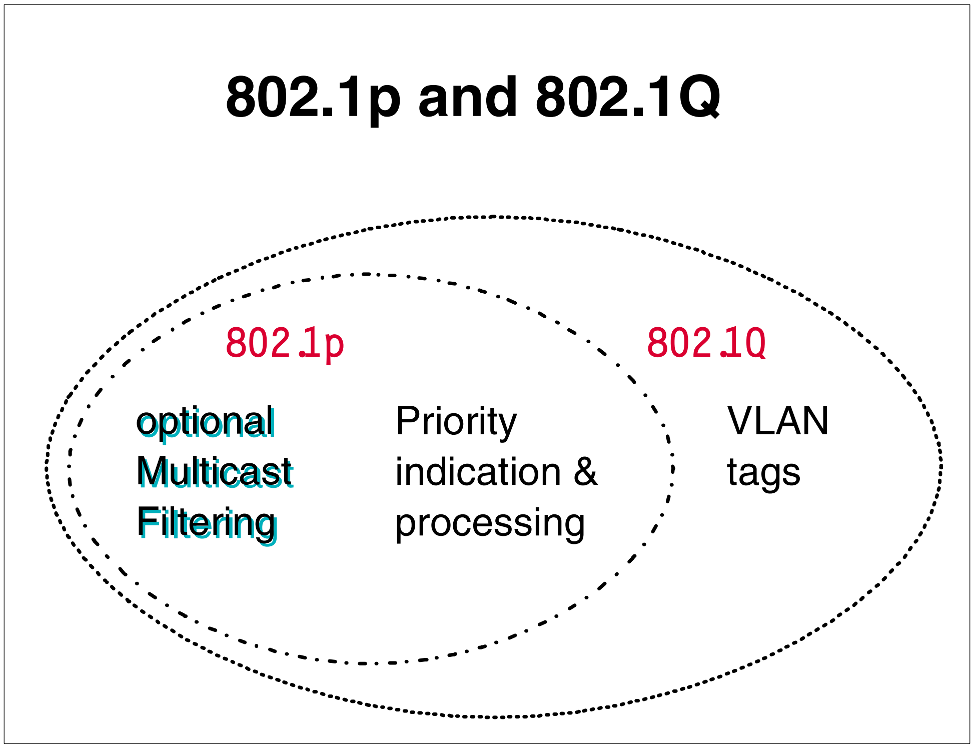

802.1p and 802.1Q
802.1p
802.1Q
optional Multicast Filtering
Priority indication & processing
VLAN tags

Figure 64. Better comparison and overlap between standards

This is still a simplistic view, but it will do for now; although the VLAN specification of 802.1Q is independent of 802.1p, 802.1Q makes use of many of the elements contained in the 802.1p specification.

For many LAN switch implementations, although it seems convenient to remember the p of 802.1p as meaning priority, in fact the standard only refers to the priority processing internal to the bridge or switch itself. Nowadays many LAN switches operate in cut-through mode and neither modify nor need to make use of the priority indication in the LAN frames themselves. The real significance of priority in a layer-2 network is the ability to transport the user priority indication across the entirety of the network, so that devices that do require use of it may do so. 802.1Q is vital to enable this capability for Ethernet networks.

9.2 IBM Ethernet devices

Table 15 shows the extent of implementation of the standards discussed in this section of the book across IBM’s portfolio of Ethernet devices. It shows the position for products which are current or recently withdrawn as of Wednesday, 8. December 1999. It is not exhaustive, although Ethernet products not shown which are not newer than this date probably do not support any of the standards tabulated below.

Table 15. IBM Ethernet devices: compliance with standards

Model	Description	802.1p	802.1Q	Jumbo frames
8371	Multilayer Switch	N	N	N/A
8265-MLS	Multilayer Switch blade	N	N	N/A
8242	10 Mbps Desktop Hub	N	N	N/A
8245	10/100 Stackable Hub	N	N	N/A
8275-113	10 Mbps Desktop Switch [1]	N	N	N/A
8275-217/225	10 Mbps Workgroup Switch	Y [2]	Y	N/A
8275-324	Fast Ethernet Desktop Switch	N	N	N/A
8275-318/322 /326	Fast Ethernet Workgroup Switch	N	N	N [3]
8275-416	High Performance Workgroup Switch	Y [2] [12]	Y [12]	N [3]
8271-524	LAN Switch [4]	N	N	N/A
8271-612/624 /712	LAN Switch [13]	N	N	N/A
8271-Exx/Fxx	LAN Switch [13]	Y [5]	Y [5]	N [3]
826x-5x12	8271 ATM/LAN Switch Module [9]	N	N	N/A
8260-6x12	8271 LAN Switch Module [10]	N	N	N/A
8260 Ethernet	All Ethernet bridge and switch modules for the 8260 [13]	N	N	N/A
34L1201	10/100 EtherJet PCI Management Adapter	Y	Y	N/A
34L0301	Netfinity Gigabit Ethernet SX Adapter	Y	Y	N
34L0901	Netfinity 10/100 Ethernet Adapter	N	N	N/A
8273	Ethernet RouteSwitch [6]	N	N	N/A
8274	LAN RouteSwitch [13]	N	Y [8]	N
8276	Ethernet RoutePort [7]	N	N	N/A
8277	Ethernet RouteSwitch [13]	N	Y [8]	N
OSA	S/390 Open Systems Adapter Gigabit Ethernet	N	N	Y [11]
RS/6000	Gigabit Ethernet - SX PCI Adapter	N	N	Y

Notes:

[1] Withdrawn from marketing May 24, 1999.

[2] Supports static group address configuration and unknown multicast configuration for the dynamic multicast filtering component of 802.1p only; does not support traffic class expediting.

3 Gigabit Ethernet uplink module; it is not possible to implement jumbo frames on a single uplink.

4 Withdrawn from marketing March 31, 1999.

5 Requires Version 2.1 software.

6 Withdrawn from marketing May 15, 1998.

7 Withdrawn from marketing May 18, 1998.

8 Fast Ethernet Mammoth modules and GRS Fast Ethernet (ESX) and Gigabit Ethernet (GSX) modules on the 8274 only; requires Nways RouteSwitch Software Program (NRSP) 3.2 except for 8274-GRS, which requires NRSP 3.4.

9 8271 modules for the 8265 are withdrawn from marketing June 25, 1999 but still remain available for the 8260 beyond this date.

10 These modules have no integrated ATM UFC and are only supported on the 8260.

11 DIX V2 (RFC 894) encapsulation only; does not support 802.3 encapsulation or VLAN tagging for jumbo frames.

12 Code release 1.1, available August 30, 1999.

13 To be withdrawn from marketing March 31, 2000.

Part 3. Class of service in ATM networks

Chapter 10. Overview

Asynchronous Transfer Mode (ATM) has been designed as a compromise: it is designed as a technology which is capable of transferring information at a very high rate but is not designed specifically to handle one type of network traffic. It does not have the best design for handling voice efficiently; it does not handle data as effectively as frame relay does and it does not handle high error rates well. However, ATM is different from most preceding technologies in that it will normally handle all these types of traffic well and in an integrated manner.

ATM is a switched technology in the same way as X.25 and frame relay: users of the ATM network request a connection (a switched virtual circuit, SVC) or use a preconfigured connection (a permanent virtual circuit, PVC) to another user of the ATM network. In the case of SVC connections, which are likely to form the vast majority of connections over an ATM network (like X.25, unlike frame relay today), the user-network interface (UNI) allows the user to add many quality of service parameters to the setup request, requesting that the network provide a connection with a specified maximum transit delay or maximum variation between cell deliveries, for example.

At first glance, this implies a nice picture in which the ATM network is accepting requests from many users of the network and is transporting different types of electronic information (voice, video, data) with different class of service specifications. Everyone is happy.

There are two problems with this picture:

1. How does the network provide the requested class of service? ATM defines five service categories (constant bit rate, real-time variable bit rate, non-real-time variable bit rate, unspecified bit rate, available bit rate), but much hard work and controversy have gone into the design of the interfaces and the networks themselves in order to be able to provide these service categories. Traffic management includes the following functions:
   - Connection admission control: can a connection request be accepted or should it be rejected?
   - Feedback controls: how is traffic through the network regulated based on the changing state of the network?
   - Usage parameter control: is the traffic transmitted by the user in accordance with the agreed contract, and if not, what actions are to be taken to protect the network?
   - Cell loss priority control: is the cell loss priority bit significant for a particular flow, and if packets are so marked, what action should be taken with them under differing circumstances?
   - Traffic shaping
   - Network resource management
   - Frame discard: if a network element needs to discard ATM cells, it is often more efficient to discard at the frame level - the service data unit presented over the UNI - rather than ATM cells. For example, a 2 kB block of data will be transmitted over ATM as 43 different cells; if a node discards just one of these cells then the network is going to have to transmit all 43 again, so it makes sense - if possible - to select cells coming from the same frame if there is a need to discard cells. But how can this be done?

- ABR flow control: ABR offers the user unused network bandwidth; the network needs to identify what bandwidth is unused rapidly in order to offer the best possible ABR service

All these issues need to be resolved by the designers of ATM networks. To take one example: how do networks detect congestion, and what action do they then take to control it? There is no standard for this and different equipment from different suppliers will take different approaches. Taking just frame discard, mentioned above, the cost of providing the ability to discard frames rather than just cells is that the network equipment must keep track of frames as well as cells: the IBM 8260 and 8265 switches (among others) implement this, but other equipment doesn't.

2. How do the users of the network actually request a particular type of service from the network? Even assuming that all the traffic management implementation issues have been solved, that we have an ATM network which is capable of providing distinct service categories and types of service, and that we have a well-agreed UNI which allows users to request specific services from the network, the model often founders because most users of ATM networks are not capable of making full use of these interfaces.
   The reality is that most data networking users of ATM networks do not use ATM directly but pass through intermediate layers which both mask the complexity of ATM from the end users but also reduce the ability of the users to make specific requests of the ATM network. Two main examples are:

   1. Router-based networks in which layer-3 (predominantly IP) routers use ATM as a high-speed point-to-point connection mechanism. In its simplest incarnation, the ATM connection between a pair of routers is a single PVC, and therefore, all traffic between the routers is treated identically by the ATM network. The routers themselves may implement queueing based on DiffServ and IntServ models, but once the data is transmitted over the ATM UNI it is treated in the same manner. Very limited discrimination may be possible if the router sets the cell loss priority (CLP) bit for some flows. An enhancement to this model is the provision of multiple parallel PVCs between routers with different ATM service provisions; this is a somewhat cumbersome mechanism because it requires preconfiguration of the PVCs in the ATM network itself.
   2. LAN Emulation (LANE) in which the ATM network emulates a layer-2 Ethernet or token-ring network. Although LANE sets up ATM SVCs when required for data transfer, the initial LANE implementation provided no ability to request different ATM service types for different LANE SVCs: all LANE traffic was treated as best effort traffic by the ATM network.

   Some exceptions exist; SNA provides mechanisms for its applications to set up ATM SVCs and therefore signal their specific ATM requirements. But the rest of this section will discuss current and emerging standards for the provision of ATM class of service to the majority of today's users of ATM.

Chapter 11. Mapping IP to ATM QoS

This chapter deals briefly with the use of ATM networks by layer-3 routers and their ability to map existing IP class of service mechanisms to ATM quality of service provisions. Although the mechanisms discussed are somewhat messy to implement, they provide a stepping-stone in the path to MPLS, which is discussed in more detail in Chapter 13, “MPLS” on page 123.

11.1 ATM as a high speed link

Increasing use is made today of ATM as an alternative or replacement to frame relay as a high-speed point-to-point transport mechanism between layer-3 routers. Frame relay has already replaced leased point-to-point lines in many countries, notably the United States, because of its lower cost and greater flexibility, but most frame relay networks are based on permanent virtual circuits (PVCs) which are logical point-to-point connections defined by the service provider and defined to provide a certain guaranteed bandwidth (committed information rate, CIR).

Frame relay is optimized for the transport of data traffic, even though increasing use is made of frame relay to transport voice traffic, and one reason is that it provides the ability to transport large data frames. ATM, on the other hand, transports everything as 53-byte cells, and at low speeds this is highly inefficient for data traffic.

At speeds greater than T1/E1 (1,544 Mbps/2,044 Mbps), however, the efficiencies of hardware switching by ATM start to overcome the inefficiencies of chopping large data packets into small cells, and therefore, network devices such as routers often offer ATM interfaces instead of frame relay interfaces for higher speed connections.

IBM routers such as the 2210 and 2216 have ATM adapters which allow them to connect to campus ATM switches (8260 and 8265) at 25 Mbps or at 155 Mbps; the ATM switches themselves have higher speed campus connections (622 Mbps today, 2,488 Mbps tomorrow) and also provide ATM WAN connections which can connect to ATM at speeds as low as T1/E1.

Public network service providers also use ATM: many if not all of them have built very high-speed backbone networks using ATM. Even though these providers continue to offer public frame relay services to customers, the data transported across the backbone is converted to ATM cells, although this process is invisible to the end users. Having built an ATM backbone, it is relatively easy for public service providers to offer a public ATM service, so that although the market for frame relay services still continues to grow, ATM is a realistic alternative for high-speed logical point-to-point connection based on ATM PVCs provided by the service provider in place of frame relay PVCs.

The connections between routers could also be implemented using SVCs. As a general rule, this is not done: the model is simpler where the service provider determines the characteristics of the virtual circuit, and the model assumes that routers will require permanent connections over ATM in order to exchange topology information via routing protocols such as RIP and OSPF.

Treating an ATM VC between two routers as a single point-to-point connection is a very simple model. Class of service differentiation can be made by the transmitting router simply by allocating output bandwidth according to DiffServ and IntServ specifications. This may, however, be an expensive approach, because in order to guarantee delivery of premium services over the ATM connection, the ATM VC must be able to provide and guarantee a specific bandwidth to the edge routers. An extreme approach would be to provide a constant bit rate (CBR) service between two routers; a more realistic approach of using an ABR (available bit rate) or UBR (unspecified bit rate) PVC might mean that the router will make queueing decisions based on the availability of bandwidth which is not always actually available.

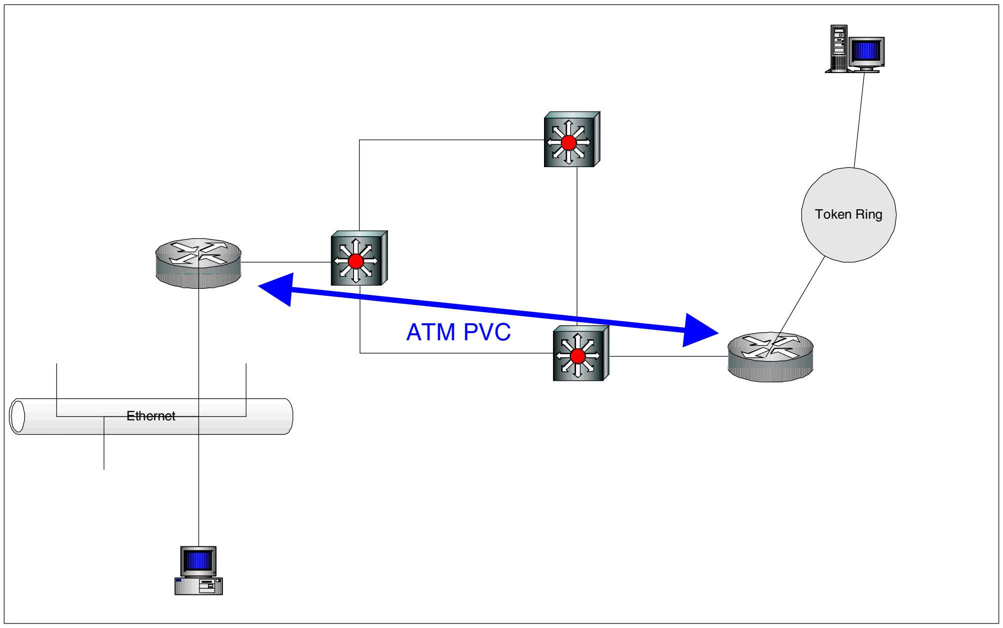


Figure 65. Two routers connected over a single ATM PVC

11.2 Multiple VCs between routers

An extension to the one in the previous section is adopted by some manufacturers, although not by IBM. The model changes to one in which multiple parallel VC connections exist between the same pair of routers, and that different VCs in this bundle provide different types of ATM service. The VCs are then matched to existing router differentiation mechanisms, such as DiffServ and IntServ, so that in addition to providing different output queues for different types of traffic, different traffic types are also allocated to different VCs.

There is no reason why SVCs could not be used for these additional ATM connections; ease of definition and implementation restrictions actually mean these are often required to be PVC connections.

The following diagram is a copy of Figure 28 on page 39, except that rather than scheduling all traffic over a single outbound link we have mapped the premium queue onto one ATM virtual circuit and the remaining traffic over another VC.

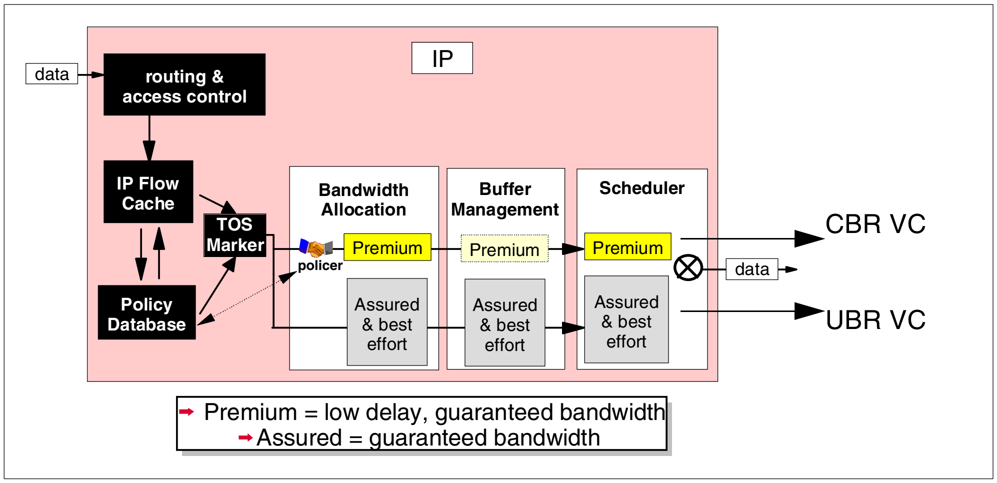


Figure 66. DiffServ mapping to different ATM PVCs

This approach is one which may enable us to save money (if we are subscribing to a public ATM service) or avoid over-commitment of resources in our own private ATM network because it allows the allocation of expensive resources (such as the ATM CBR virtual circuit) to match our requirements more accurately.

There are no agreed standards for the use of multiple VCs between routers, nor are there likely to be. Implementations of designs such as this one are presented as implementation-specific value add approaches by router manufacturers.

In the longer term, this approach is probably just a stepping-stone to more sophisticated and flexible approaches to the classification, marking, and transmission of different types of data across ATM. Section 13.6, “MPLS and DiffServ and ATM” on page 128 describes a draft proposal for such an approach.

11.3 RSVP and ATM

Unlike the implementation of DiffServ, which can only be implemented on PPP and frame relay links using IBM routers, IntServ using RSVP can be implemented on many more interface types (see 3.3, “RSVP and IBM’s packet scheduler” on page 61).

RSVP reservations translate into separate ATM SVCs for the transmission of the data flow across the ATM network. When the RESV reservation request is received by a router for a reservation across an ATM link, and assuming that the router’s policy control function allows the reservation request to be made, the router will attempt to establish a unique ATM SVC to the next hop router across the ATM network.

The SVC always uses RFC 1483 format; the router-router connection used for all other traffic can use either RFC 1483 (Classical IP) or LANE.

A practical application of RSVP in an ATM environment may well be where a public ATM service with limited bandwidth is being used; perhaps we are talking about bandwidth of the order of 2 Mbps. If a PPP or frame relay link were being used, DiffServ or BRS could prioritize traffic for transmission over the link. One application of RSVP could be for specific important traffic flows, for example for SNA traffic encapsulated in IP using DLSw.

Chapter 12. Quality of service using LAN emulation

LAN Emulation over ATM (LANE) was introduced in 1995[1] as a method of emulating the services of existing LANs across an ATM network. LANE enables endstations to connect to the ATM network in such a way that software applications continue to interact as if they were connected over a traditional LAN - Ethernet or token-ring. LANE defines a service which emulates the MAC service, and includes the following characteristics:

- Connectionless service, in which stations connected over LANE are able to send data without previously having to establish connections.
- Multicast service, in which multicast MAC addresses (such as broadcast, group or functional addresses) are supported.
- MAC interfaces in ATM stations, allowing existing applications to access an ATM network using existing protocol stacks (such as APPN, NetBIOS, IPX, IP) as if they were running over traditional LANs.
- The ability to configure one or more emulated LANs, which act as logically independent groupings of LAN stations.
- Interconnection with existing LANs using bridging methods, allowing connectivity both from ATM stations to LAN stations as well as LAN stations to LAN stations across ATM.

LAN Emulation Version 2 (LANE V2) was introduced in 1997[2] and supersedes the original specification. It provides additional capabilities, including:

- LLC multiplexing for VCC sharing
- Support for ABR and other quality of service specifications through an expanded interface
- Enhanced multicast support
- Support for Multiprotocol Over ATM (MPOA)

This chapter will concentrate on the ability of LANE clients and LANE servers to provide differing Quality of Service connections across the ATM network.

12.1 LANE Version 1

LANE establishes a data direct virtual channel connection (VCC) as a bidirectional point-to-point connection between two LANE clients for the purpose of exchanging unicast data traffic. LANE provides a LAN Emulation Server (LES) as a mechanism for establishing a relationship between ATM addresses and emulated MAC addresses.

It is not the intent of this book to describe the workings of LANE in detail; in summary, however, the process by which ATM connections are established and emulated LAN traffic flows according to the original LANE specification is:

1. A LAN Emulation Client (LEC) joins a particular emulated LAN.
2. The LEC informs the LAN Emulation Server (LES) of the individual MAC addresses and source-route bridge descriptors that it represents.

[1] The ATM Forum, LAN Emulation Over ATM Version (LANE 1.0) 1.0 Specification, af-lane-0021.0000, January 1995
[2] The ATM Forum, LAN Emulation Over ATM Version 2, af-lane-0084.0000, July 1997

3. When an application wishes to send data to another MAC address, the LEC resolves the ATM address which represents the other MAC address.
4. The LEC encapsulates the data in an AAL-5 frame and transmits it over ATM.

12.1.1 ATM call setup with LANE V1

The LEC communicates with the ATM network over a private user-network interface (UNI). This interface defines how users of the ATM network can set up calls (known as connections or virtual circuits) across the network to other users of the network.

LANE V1 provides for a single virtual channel connection (VCC) to be set up for the transmission of all unicast data between two LECs across an ATM network. In particular, this connection is established according to the UNI specification by use of a setup message in which:

- AAL-5 is specified, with a maximum service data unit (SDU) size of up to 18,240 bytes and with null SSCS.
- The ATM User Cell Rate (UNI 3.0) or ATM Traffic Descriptor (UNI 3.1) indicates a best effort connection: this indicates that the connection is in the unspecified bit rate (UBR) category and that no guarantees are made by the network with respect to the cell loss ratio (CLR) or the cell transfer delay (CTD).
- The standard recommends using Service Type X (user defined) although Service Type C (connection-oriented data) can be used instead.
- The setup message *must* include a QoS parameter; LANE V1 specifies the use of Service Class 0, which means that no specific quality of service is provided by the network for these connections.

12.2 LANE Version 2

In LANE V2, a LEC is permitted to establish multiple data direct VCCs for the same unicast LAN MAC address destination. Any LEC indicates its willingness to receive these multiple VCCs by registering this capability with the LES. This registration indicates which ATM service categories are supported for a particular LAN destination.

LANE V1 always used the unspecified bit rate (UBR) service category for transmission of data. LANE V2 allows the use of some or all of:

- Constant bit rate (CBR), typically used for a constant stream of bits at a predefined constant rate with short transit delay and low jitter, perhaps used by voice, video or circuit emulation.
- Real-time variable bit rate (rt-VBR), which provides the low transit delay of CBR but for a variable data rate, such as compressed video or voice with silence suppression.
- Non-real-time variable bit rate (nrt-VBR), which provides a guaranteed delivery service in which transit delay and jitter are less important than with rt-VBR; it may be used for unidirectional transmission of encoded voice and video.
- Available bit rate (ABR), which offers a guaranteed delivery service with minimal cell loss but possibly with a widely varying throughput rate, described

as the economical support of applications with vague requirements for throughputs and delays.

Additional data direct VCCs will be established by higher protocol layers instructing the LEC to establish connections with particular quality of service requirements. The specific requirements are communicated using UNI setup messages, and many (but not all) of these are only applicable to UNI 4.0 networks. These signaling elements include:

- ATM Traffic Descriptor
- Alternative ATM Traffic Descriptor (UNI 4.0 only)
- Minimum Acceptable ATM Traffic Descriptor (UNI 4.0 only)
- Broadband Bearer Capability
- Extended QoS Parameters (UNI 4.0 only)
- QoS Parameter
- End-to-End Transmit Delay (UNI 4.0 only)
- ABR Setup Parameters (UNI 4.0 only)
- ABR Additional Parameters (UNI 4.0 only)

When no higher level specifies a specific quality of service capability to the LEC, a default set of parameters must be used for the data direct VCC, which is the use of QoS Class 0 as defined for all such VCCs in LANE V1.

12.2.1 ATM call setup with LANE V2

Some of the parameters which can now be included in the call setup request by the LEC include:

- Peak cell rate (PCR)
- Cell delay variation tolerance (CDVT), an upper bound on the "clumping" effect of the difference between arrival times of consecutive cells
- Sustainable cell rate (SCR)
- Maximum burst size (MBS)
- Minimum cell rate (MCR)
- Cell delay variation (CDV)
- Maximum cell transfer delay (maxCTD)
- Cell loss ratio (CLR)

The use of some or all of these parameters in specifying a particular ATM service category is shown in the following table:

Table 16. ATM service category attributes

<table>
<tr><th>Attribute</th><th>CBR</th><th>rt-VBR</th><th>nrt-VBR</th><th>UBR</th><th>ABR</th></tr>
<tr><td>PCR and CDVT</td><td colspan="5">specified</td></tr>
<tr><td>SCR, MBS, CDVT</td><td>n/a</td><td colspan="2">specified</td><td colspan="2">n/a</td></tr>
<tr><td>MCR</td><td colspan="4">n/a</td><td>spec.</td></tr>
<tr><td>peak-to-peak CDV</td><td colspan="2">specified</td><td colspan="3">unspecified</td></tr>
<tr><td>maxCTD</td><td colspan="2">specified</td><td colspan="3">unspecified</td></tr>
<tr><td>CLR</td><td colspan="3">specified</td><td>unspec.</td><td>?</td></tr>
</table>

LANE V2 continues to specify that VCCs should be set up using AAL-5 with null SSCS.

12.2.2 LANE QoS and IBM routers and ATM edge devices

IBM LANE clients in IBM routers and other ATM edge devices such as the 8371 Multilayer Ethernet Switch allow configuration of ATM QoS parameters for the LANE client. The actual QoS parameters used for a LANE data direct VCC can be based on a combination of the QoS parameters configured in both LANE clients (sending and receiving) and in the LAN Emulation Configuration Server (LECS).

The feature known as Configurable QoS for LAN Emulation allows the specification of six ATM parameters which control the traffic characteristics of data direct VCCs established by the LAN Emulation Client. These parameters can be configured for an individual LAN Emulation Client, an ATM Interface or an Emulated LAN:

1. Maximum reserved bandwidth
2. Traffic type (best effort or reserved bandwidth)
3. Peak cell rate
4. Sustained cell rate
5. Maximum burst size
6. QoS class

Two other options are provided:

1. The ability to negotiate QoS parameters between those configured in the IBM router and those provided by an IBM MSS Lan Emulation Server.
2. The ability to accept QoS parameters from a LAN Emulation Configuration Server (LECS) and to make these parameters override the locally configured parameters.

This feature does not provide the capability for a single LEC on the router to establish multiple data direct VCCs to a single destination. All router-router traffic transmitted over a single LEC-LEC connection will use the same data direct VCC.

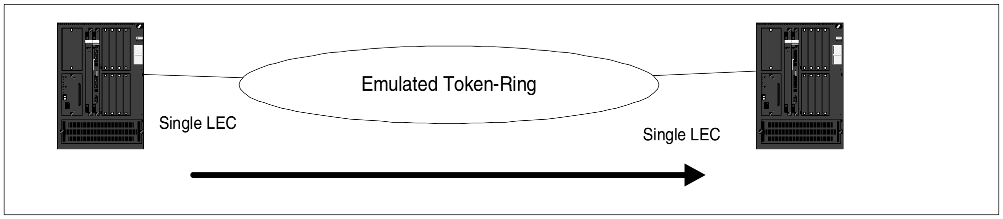


Figure 67. Router-router single LANE connection

To differentiate between the treatment of traffic across the ATM network, a possible approach would be to use more than one LEC on each router. Each LEC has a different IP address associated with it. By using global access controls in the router, different types of traffic can be directed to different next-hop IP addresses, which are associated with the different LECs. The LECs, in turn, are defined with different QoS parameters, so that the separate SVCs set up across the ATM network between the routers have different traffic characteristics associated with them. So, for example, batch traffic could be directed over a best effort connection and interactive traffic directed over a reserved bandwidth connection, as shown in the following figure:

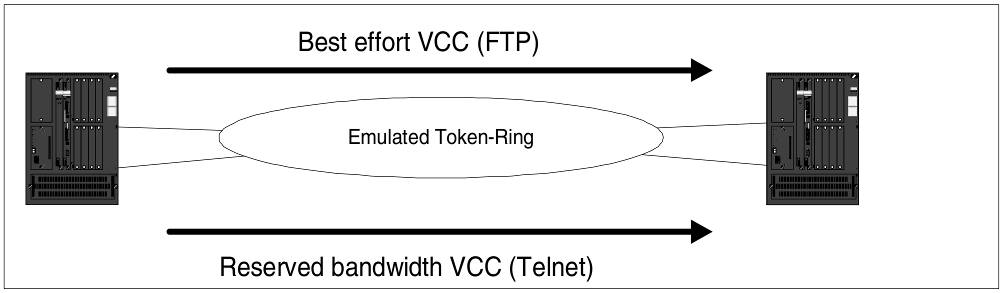


Figure 68. Router interconnection using multiple LEC instances.

12.2.3 LANE V2 and 802.1p

LANE is a layer-2 mechanism: it transports data across ATM between emulated MAC addresses. The previous section discussed the layer-2 priority mechanism defined in 802.1p/802.1D and implemented in LAN bridges and switches.

In addition to describing the behavior of LAN bridges in the implementation of different priority queues for different types of outbound traffic on a port, 802.1p also refers to the native priority mechanism which exists today in token-ring networks. This allows LAN frames to be transmitted at a specific priority, which allows high priority frames transmitted by one device to gain priority over lower priority frames transmitted by another device.

If a bridge (or switch) implements 802.1p and has an ATM uplink which uses LANE, LANE V2 allows the bridge to use multiple SVCs for the transmission of layer-2 frames. Rather than transmitting all outbound traffic for a given destination over a single SVC, as must be the case with LANE V1, traffic in different outbound priority queues can be mapped to different SVCs with different ATM service qualities.

There is no standard describing this implementation: it is proprietary, and is not currently implemented by IBM. 3Com appears to have an implementation along these lines. It allows cells which form part of a layer-2 packet to be treated differently from cells forming part of another layer-2 packet inside the ATM network.

LANE is designed to use ATM SVCs, so this approach differs from the layer-3 approach of multiple VCs (usually PVCs) for the transport of different types of IP packets described in Chapter 11, “Mapping IP to ATM QoS” on page 113.

The requirement is that the device to which multiple SVCs are established supports LANE V2 and indicates its willingness to accept different categories of SVCs.

There is one additional issue: the regeneration of layer-2 priority (see 6.1.1, “Frame reception” on page 83). Even though a bridge may implement 802.1p and may be receiving transmissions over multiple SVCs from another 802.1p bridge, LANE V2 provides no mechanism for indicating the ATM service category of received frames to LANE clients, and therefore, unless the LANE client code implements special code to differentiate between received packets, the 802.1p bridge code will receive all packets over all LANE V2 connections with the same priority indication. LANE hides the essentials of ATM from its users. So the receiving bridge cannot regenerate the 802.1p priority directly from the frames it receives. Even token-ring LAN emulation does not emulate token-ring priority values.

The following figure shows the relationship between the components. The LANE component in the sending device can take information from the outbound 802.1p bridge port to determine what type of SVC to use for the transmission of the layer-2 packet. The LANE component in the receiving device receives no indication of which SVC was actually used, and therefore, cannot provide signaling to the 802.1p bridge to indicate the layer-2 priority of the frame.

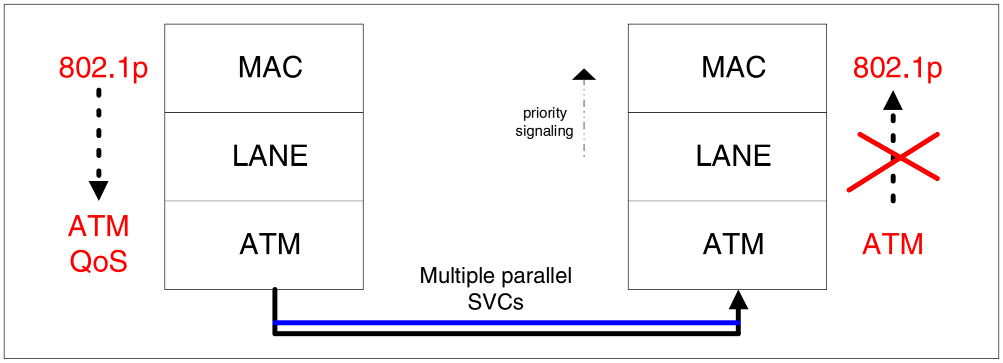


Figure 69. Relationship between LANE and 802.1p

12.2.4 LANE V2 and 802.1Q/802.3ac

LANE V2 provides for the transmission of the 802.1Q tag header by allowing a maximum frame size of 1580 bytes for Ethernet frames which include a tag header. This frame size was not supported in LANE V1. LANE V2 now supports the following pre-defined maximum frame sizes:

Table 17. LANE V2 defined maximum data frame sizes

Nonmultiplexed maximum AAL5 SDU (bytes)	Nonmultiplexed maximum number of ATM cells	Usage
1516	32	Ethernet/802.3
1580	34	802.1p/Q/802.3ac
4544	95	Token-ring 4 Mbps
9234	193	RFC 1626
18190	380	Token-ring 16 Mbps

This support is included in IBM’s MSS LES/BUS implementation of LANE V2, and provides a mechanism for two LANE clients to exchange layer-2 priority indications using 802.1Q VLAN tags. None of IBM’s current LAN emulation clients support the transmission of 802.1Q tags over emulated Ethernet LANs, however. In addition, some components of IBM’s MSS (BUS Police, for example) do not understand the 802.1Q frame format and require modification before they will support the new frame format.

Chapter 13. MPLS

Multiprotocol Label Switching (MPLS) is an emerging standard for IP switching. It is designed to allow existing IP networks to improve in performance and scalability. It is not restricted to ATM networks alone, but is especially suited to them, and its first implementation is likely to be across ATM backbones. The MPLS framework document[1] states that although MPLS core technologies *must* be general with respect to data link technologies, specific optimization methods for particular media types *may* be considered. Likewise, the model is not restricted to the IP protocol alone, although again it is likely that the majority of implementations will be for IP, and the initial implementations are aimed at IP.

In addition to improving the speed and reducing the cost of networks, MPLS also interacts with other class of service mechanisms, and specifically with the IntServ and DiffServ IP mechanisms discussed earlier.

MPLS is an advanced form of packet forwarding which replaces the existing forwarding process based on the best match of destination address with a label swapping model. In the conventional model, each router in the network examines each IP packet and makes a forwarding decision based on the destination IP address of the packet; using MPLS the core routers in the network use a label as an index into a forwarding table. The label can be thought of as a shorthand for the packet header itself. Another term which comes up is FEC, which stands for Forwarding Equivalence Class; the MPLS label is the encoded value of the FEC.

In comparison with an ATM core network, on the other hand, non-MPLS ATM switches do exactly that: they switch ATM cells between interfaces, but they know nothing about the traffic contained in the ATM cells. MPLS-enabled switches also switch ATM cells (at layer-2) but also associate different characteristics with different labels (the VPI/VCI settings in the ATM cells themselves). MPLS brings the performance characteristics of layer-2 network with the connectivity and network services of layer-3 networks.

An ATM label switching router does not reassemble IP frames when switching between two ATM interfaces. The (top-most) label for any received ATM cell can be inferred from the VPI/VCI values in the ATM cell itself, and provided this indicates that the cell is to be retransmitted over another ATM interface, the switch will do so immediately. If the ATM label switching router also has, say, an Ethernet interface, then it will perform frame switching between the ATM and Ethernet interfaces. This means that it will reassemble the ATM cells into a single IP frame but will then switch this frame onto the Ethernet interface based on the MPLS information derived from the MPLS label and will not need to examine the layer-3 IP information in the frame.

MPLS networks comprise the following elements, and are shown in Figure 70 on page 124:

- Label Edge Routers (LERs), which are located at the boundaries of the MPLS network. In an ATM network, LERs may well be the routers placed at the edge of the ATM network. LERs apply labels to packets for transmission across the MPLS network; traffic from multiple sources to the same destination (egress LER) can share labels.

[1] A Framework for Multiprotocol Label Switching, draft-ietf-mpls-framework-02.txt

- Label Switch Routers (LSRs), which switch IP packets or cells based on the labels found in them. LSRs can also support full layer-3 routing functions for unlabeled packets.
- Label Switch Paths (LSPs), which define the path taken through the MPLS network for a given label. In an ATM network, the LSP is equivalent to an ATM VC.
- Label Distribution Protocol (LDP), which is used to distribute label information between LSRs and LERs. As well as defining a new protocol, existing protocols such as BGP and RSVP have been extended to allow label distribution to be “piggybacked” on them.

It is important to note that LSRs and LERs build their routing databases using existing routing protocols such as OSPF: MPLS fits on top of existing networks.

Additional requirements for MPLS include:

- MPLS must simplify packet forwarding to reduce the cost and increase performance of the network.
- MPLS must be compatible with the Integrated Services model, including RSVP.
- MPLS switches must be able to coexist with non-MPLS switches in the same switched network.

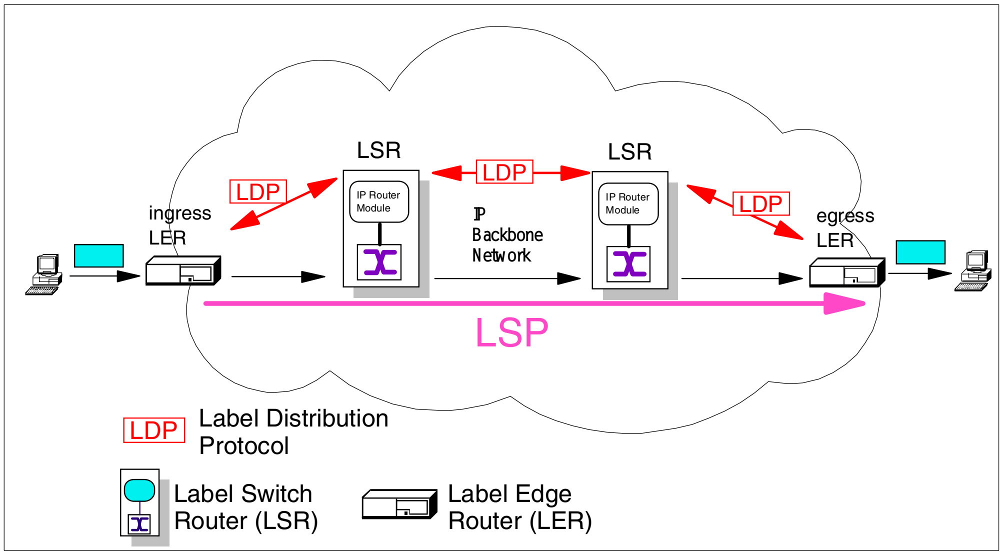


Figure 70. MPLS network model

In ATM networks, the label can actually be the VCI/VPI combination in the ATM header. Likewise, in frame relay networks, the Data Link Control Indicator (DLCI) in the frame relay header can be used. In other network types such as Ethernet, FDDI, PPP or token-ring, the MPLS label has to be added to the existing frame between the layer-2 header (DLL header) and layer-3 data elements of the frame.

13.1 MPLS compared with a router-based core network

The advantages of using MPLS in a network based around layer-3 routers include:

- Simplified forwarding, which may allow simpler routers to be built to perform IP forwarding in the core of a network.
- Explicit routing, in which MPLS allows the label to specify an explicit route through the network, allowing the LER to specify different routes through the network for different traffic types to the same destination. For example:
  - Traffic engineering, in which paths through the network are chosen to balance the traffic load on the components of the network.
  - QoS routing, in which the route chosen for a particular stream is chosen in response to the quality of service required for that stream.
- Labeling at the edge of the network, in which LERs classify and label each packet in an efficient manner. Of course, this is not dissimilar to applying DiffServ classifications at the edge of a network as described in Chapter 2, “Differentiated Services” on page 5.

13.2 MPLS compared with an ATM switch-based core network

If an ATM network which implements MPLS LSR functions in each of the core switches is compared with a network in which ATM is simply used as a means of interconnecting routers, MPLS offers the following benefits:

- Scalability in that the number of logical links between the network components reduces. In a basic ATM network, each edge router requires a connection to every other edge router. With MPLS, edge routers need to connect to just one core LSR.
- Elimination of the need to implement NHRP and on-demand cut-through SVCs and the associated problem of latency. As described in “MPLS label assignment” on page 127, MPLS labels can be defined prior to any data traffic flowing.

13.2.1 MPLS, MPOA, and NHRP

In the specific context of Class of Service, there are some specific comments and comparisons which can be made:

- Multiprotocol over ATM (MPOA) is used to establish shortcuts between layer-2 devices (bridges) which are connected over ATM. Section 12.2.3, “LANE V2 and 802.1p” on page 121 shows how different layer-2 priorities *could* be mapped to separate ATM SVCs; IBM’s implementation only allows the specification of ATM QoS parameters for a single SVC which is used for all traffic using an MPOA shortcut. For more information about MPOA, see:
  - *Layer 3 Switching Using MSS and MSS Release 2.2 Enhancements*, SG24-5311.
- Next Hop Resolution Protocol (NHRP) is used to establish shortcuts between layer-3 devices (routers) which are connected over ATM. Again, a single SVC is used for all traffic using this shortcut; although specific ATM QoS parameters can be requested for this SVC, ATM does not really play a role in differentiating between different IP traffic flows - the edge routers themselves can implement DiffServ and IntServ in order to accomplish this. For more information about NHRP, see:
  - *MSS Release 2.1, Including the MSS Client and Domain Client*, SG24-5231.

- MPLS provides an integrated environment in which edge routers set up *flows* across the network; different flows can be used to transport different types of traffic and the ATM switches in the network will treat each flow appropriately: mapping is possible between IP service differentiation in the LAN environment and the MPLS/ATM environment (see 13.6, “MPLS and DiffServ and ATM” on page 128).

Another view of the differences here is that traffic arriving at an ingress point into an ATM network is mapped to one of the following:

- A data-direct ATM SVC (LANE, Classical IP, MARS)
- A shortcut ATM SVC (NHRP or MPOA)
- A label path (MPLS)

13.3 MPLS traffic granularity

Individual MPLS labels can be used to aggregate several traffic flows; all packets with the same incoming label must be forwarded by core routers out of the same port(s) and with the same encapsulation(s) and with the same next-hop label (if any). Examples of unicast traffic types which can be grouped by a single label include:

PQ	Port Quadruples: with the same IP source address prefix (subnetwork), destination address prefix, TTL, IP protocol, and TCP/UDP source/destination ports
PQT	Port Quadruples with TOS: similar to PQ but with a restriction that packets also have the same service type byte (see Figure 5 on page 12).
HP	Host Pairs: with the same IP absolute source and destination address
NP	Network Pairs: with the same IP source and destination address prefixes
DN	Destination Network: with the same IP destination address prefix
ER	Egress Router: with the same egress router ID (for OSPF, for example)
NAS	Next-hop AS: with the same next-hop AS number (for BGP)
DAS	Destination AS: with the same destination AS number (for BGP)

The basic forwarding operation of a label switch router consists of looking up the incoming label to determine the outgoing label, encapsulation, port, and any additional information such as a particular queue or some other quality of service treatment. To provide a class of service discrimination at each label switch router, two approaches are possible:

1. Expansion of the MPLS header to include some kind of class of service field. This would then be read and acted upon by each LSR. The disadvantage of this approach is that it may contribute to the MPLS header being too large.
2. Propagate class of service information when the label is assigned, and then use different labels (PQT above, for example) for flows requiring different treatment by LSRs.

One additional point is that the next hop is determined by reading the Next Hop Label Forwarding Entry (NHLFE) in the LSR based on the value of the received label. This next hop is not necessarily the same next hop if MPLS had not been in use - in other words, MPLS overrides the next hop determined from the LSR's own routing table.

13.4 MPLS label assignment

Three approaches have been proposed for the assignment of labels by LERs:

1. Topology-based ([manufacturer: Cisco] TAG switching, [IBM] ARIS, IP Navigator)
2. Request-based (RSVP)
3. Data traffic-based (CSR [Toshiba], [Ipsilon])

Actual implementations of MPLS may combine these methods: for example, using the topology-based approach for best effort traffic and the request-based approach in order to support RSVP.

13.4.1 Topology-driven label assignment

The LER assigns labels in response to control protocols such as OSPF and BGP. When the LER updates its routing tables in response to OSPF or BGP updates it will assign a label to each of the entries. The labels therefore have the same granularity as the routes advertised by the routing protocols. One immediate advantage of this approach is that labels exist prior to traffic arising, and therefore there is no latency in setting up labels in response to traffic flows: if the packet can be routed it can immediately be assigned a label.

13.4.2 Request-driven label assignment

Labels are assigned in response to request-based control traffic such as RSVP requests. Again, labels are assigned prior to the flow of data traffic but requires applications to make use of RSVP in order to get a label assigned to their traffic flows.

13.4.3 Traffic-driven label assignment

The arrival of data traffic at an LER triggers label assignment and distribution. This requires high-performance packet classification capabilities in the LER and imposes a latency overhead on data flows during label assignment and distribution.

13.5 MPLS on ATM switches

There are three proposed approaches for implementation of MPLS on ATM switches:

1. Implement MPLS by removing all the traditional ATM switch capabilities and instead implementing devices as MPLS LSRs. This is the approach taken by Ipsilon.

2. Implement MPLS by adding MPLS to existing ATM Forum functions, but isolate the two functions from each other. This is referred to as Ships in the night or SIN. It allows a single device to act simultaneously as an MPLS LSR and as an ATM switch, but there is no interaction between the two functions.

3. Implement MPLS by integrating MPLS with the existing ATM Forum functions so that MPLS can use the existing functions to set up SVCs as needed. This

approach requires the specification of procedures for the use of SVCs by MPLS and for the association of labels with them.

13.6 MPLS and DiffServ and ATM

MPLS provides a means of identifying and transporting a flow (probably of IP data) across a network (probably ATM). The most current definition of DiffServ (see Figure 8 on page 14) defines 64 potential different per-hop behaviors (PHBs) which can be indicated by the contents of the service type byte in the IP header.

MPLS is used specifically because we do not want to have to examine the IP header in each LSR. So we need to provide a different mechanism for differentiating between IP packets across the MPLS ATM network. Remember that ATM transports 53-byte cells and that MPLS provides a mechanism for the treatment of each cell; if we needed to examine the IP header we would have to reassemble each cell into its IP packet at each ATM switch.

Following are three approaches which suggest themselves:

1. The inclusion of the service type byte or similar class of service information in the MPLS header. This would increase the size of the header (and remember that this header has to be included in each ATM cell, so this is a significant concern) and would increase the processing load on the LSR although it would save the need for the LSR to examine the actual payload IP datagram.
2. The inclusion of some kind of class of service indication in the label distribution protocol, and the differentiation of traffic by using different labels for different classes of service. In a network in which labels are created in response to traffic flows this might be possible; in a network in which labels are created prior to traffic flows existing and based on routing information this could lead to the unacceptable overhead of having to set up 64 different labels for each routing table entry.
3. A more restricted number of labels coupled with additional ATM signaling.

The third approach is the one taken by the IETF draft named “MPLS Support of Differentiated Services by ATM LSRs and Frame Relay LSRs”, `draft-ietf-mpls-diff-ext-00.txt`. It proposes a relatively simple mapping between EF/AF classes and labels coupled with the use of the ATM cell loss priority (CLP) bit to denote packets with a greater drop priority.

For a single Forwarding Equivalence Class (FEC) defined in the edge router (which could be equivalent to a routing table entry using the tag switching or ARIS models), there would normally be a single label and a single label switched path (LSP) across the MPLS network for all traffic. The IETF draft defines separate PHB Forwarding Classes (PFCs) for each AF class and for the EF class and then proposes that a separate label (meaning a separate LSP, and hence a separate VC) be used for each (PFC,FEC) combination.

In the context of our Olympic service of gold, silver, and bronze service provided by three AF classes, this would result in one LSP being defined for each AF class. A separate LSP would be used for EF traffic.

The CLP bit would be set off (0) for EF traffic and for any AF traffic with a drop priority of 1; it would be set on (1) for any AF traffic with a drop priority of 2 or 3.

For a full implementation of DiffServ in MPLS over ATM using this approach, five PFC values have to be used to create different LSPs for a single FEC:

Table 18. DiffServ EF/AF to MPLS PFC Correspondence

EF	AF1 bronze	AF2 silver	AF3 gold	AF4	Drop priority
PFC=0 (EFC) CLP=0	PFC=1 (AFC1) CLP=0	PFC=2 (AFC2) CLP=0	PFC=3 (AFC3) CLP=0	PFC=4 (AFC4) CLP=0	DP1
	PFC=1 (AFC1) CLP=1	PFC=2 (AFC2) CLP=1	PFC=3 (AFC3) CLP=1	PFC=4 (AFC4) CLP=1	DP2
	PFC=1 (AFC1) CLP=1	PFC=2 (AFC2) CLP=1	PFC=3 (AFC3) CLP=1	PFC=4 (AFC4) CLP=1	DP3

The PFC values are included in the LDP requests sent across the MPLS network when labels are initially created. The CLP values, on the other hand, are inserted in the frames by the edge LER when they are transmitted over the appropriate ATM circuit and form no part of the label or the label signaling process.

Each LSR in the core of the network should take the PFC value and set up appropriate scheduling behavior for the associated LSP and label. The PFC values are contained in LDP messages which are an extension to the basic format. Presumably LSRs which do not understand these messages simply allocate identical default characteristics for all the separate label switched paths.

The end result is that each ATM MPLS core label switching router will be able to differentiate between up to five different DiffServ classes and between two different levels of drop priority simply by using the MPLS label and the CLP bit.

A similar approach can be adopted if MPLS is implemented over frame relay; in this case frame relay's discard eligible (DE) bit is used in place of ATM's CLP bit.

Chapter 14. Summary

ATM offers quality of service metrics: the ability to set up calls with specific bounds for parameters such as cell loss and cell delay. The issue with today's networks is in how other class of service parameters (with more vague definitions such as high, medium, and low priority) can be mapped to these ATM parameters. Of particular interest is a method of mapping the IP Differentiated Services and Integrated Services approaches to ATM quality of service mechanisms, and some variety of MPLS is seen as the best approach here. MPLS implementations are only now emerging, so other mechanisms have already been used by different manufacturers prior to this standards-based approach.

14.1 APPN and ATM

Having dismissed SNA somewhat in the rest of Part 3, "Class of service in ATM networks" , it is appropriate to return to a discussion of how APPN can use ATM networks.

APPN defines a mechanism for a native ATM interface. In other words, APPN nodes can set up SVCs or use existing PVCs for the transmission of APPN traffic. The architecture does in fact require that High Performance Routing (HPR) be used; the earlier Intermediate Session Routing (ISR) flavour of APPN is not supported, nor is subarea SNA.

Because APPN is using ATM directly, and not being shielded from knowledge of ATM (as is the case with LANE, for example), there is a direct correspondence between ATM VCs and APPN transmission groups (TGs). Each APPN TG corresponds to an ATM VC, and the VC can be established with any combination of quality of service parameters, either:

1. If a PVC is used, by configuration of the network equipment which provides the PVC.
2. If an SVC is used, and if the SVC is dedicated to APPN/HPR traffic, a wide range of ATM QoS parameters can be specified when the APPN node requests the establishment of the SVC.

In a parallel with the case of a pair of routers interconnected using ATM discussed in Chapter 11, "Mapping IP to ATM QoS" on page 113, two APPN nodes (network nodes or end nodes) may use either a single VC for all traffic or may choose to use multiple parallel VCs. Both of these fit very well with the APPN architecture.

14.1.1 Single VC between nodes

The following figure shows a connection between a 2216 network node and a mainframe end node established over ATM. The connection between the 2216 and the mainframe is using a single VC and is using HPR. The connections to the two end nodes over token-ring may use HPR or may use ISR. The 3270 connects to the 2216 using LLC2. The 2216 must implement dependent LU requester (DLUR).

APPN traffic queued for transmission over the single ATM VC will be queued according to APPN transmission priority. In particular, lower-priority traffic can be overtaken by higher-priority traffic in the 2216; this is a fundamental requirement of APPN architecture when a network node routes APPN traffic between its interfaces.

In other words, there is no need to provide additional mechanisms such as DiffServ for APPN: the base APPN architecture already contains the equivalent mechanism which is implemented by all APPN nodes. SNA sessions are established with a particular transmission priority, and APPN network nodes honor the transmission priority. In the case of HPR, transmission priority is indicated in each separate HPR packet. In the case of ISR, network nodes participate in session establishment and remember the priority assigned to each session setup through them.

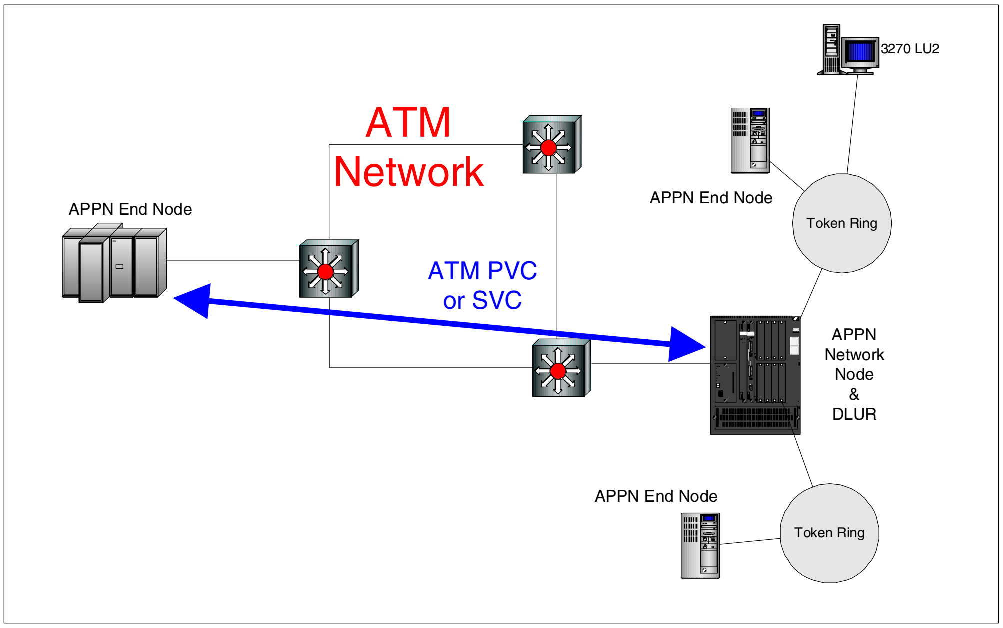


Figure 71. APPN connectivity over ATM

14.1.2 Multiple VCs between nodes

APPN provides four different transmission priorities: network, high, medium, and low. If the previous network is modified to provide multiple parallel VCs between the 2216 network node and mainframe end node, these VCs will appear as parallel TGs to APPN.

Again, this requires no exceptional treatment by APPN; each VC will be assigned particular ATM characteristics and will be associated with APPN characteristics. APPN route selection will choose the best TG to use for any one session. For example, a CBR VC could be used for network and high-priority traffic, with a separate UBR VC being used for remaining traffic.

Which of these two models is actually chosen probably depends on the trade-off between simplicity and cost: a single VC is easier to implement but it may cost less to use multiple VCs under certain circumstances. To restate the earlier point: even if a single VC is used, APPN will always prioritize transmission of traffic over this link based on the transmission priority determined when the SNA session was established.

14.2 General summary

The provision of class of service in ATM networks depends on the network model being used - the way in which ATM is viewed. There are three different approaches to using ATM for the transport of data network traffic, and therefore, three different approaches to the provision of class of service in the ATM environment:

1. If ATM is viewed as emulating an existing LAN environment, some of the existing layer-2 mechanisms can be mapped to the ATM environment - 802.1Q tags can be transported over ATM (using LANE V2 with some possible restrictions or limitations). Multiple parallel connections between a pair of nodes can be established, each connection having its own ATM QoS parameters, and different traffic types can be mapped to different connections.
2. If ATM is viewed as a fast WAN technology as a means of interconnecting routers, existing layer-3 mechanisms (DiffServ and IntServ) can be used. Again, different parallel connections can be made between routers (in some cases PVCs, in others, SVCs) and different mechanisms can be used to map different types of IP traffic to different VCs.
3. If ATM is viewed as a combination of network layer routing and high-speed data link layer forwarding, MPLS seems to provide benefits to both views, and increases price/performance and scalability of the network. MPLS is very much work in progress and actual MPLS implementations are only now emerging.

Appendix A. Sample calculations for frame relay parameters

This appendix is provided as an addition, an elaboration and an explanation of some of the calculations which have to be performed when configuring a frame relay network to be able to transport voice traffic in addition to data traffic.

Consider first of all an environment in which voice calls are being transported over frame relay without encapsulating them in IP: we have referred to this before as VoFR.

If we want to mix voice and data traffic over a 64 kbps[1] PVC and allow for the transport of up to three voice calls simultaneously we need to ensure that the appropriate number of voice packets can be transmitted during the T_c time period and that data traffic can be fragmented to fill the remainder of the time slot.

For this example, consider voice streams which are encoded at a rate of 9.6 kbps per call. This results in a 25-byte voice frame every 15 milliseconds (once the frame relay header is taken into account) and an effective bandwidth requirement of 13.333 kbps for each voice call. Thus voice can use up to 40 kbps of the 64 kbps bandwidth, restricting data traffic to the remaining 24 kbps at worst.

If T_c is set to 0.03, the minimum value possible (which has to be accomplished by setting B_c to 1920, in fact, and remembering that $B_c = T_c \times CIR$), then in every T_c seconds we will expect at maximum 3 x 25 x 2 bytes of voice traffic. This equates to 150 bytes, or 1,200 bits. Taking this away from B_c we can see that 720 bits of data traffic or 90 bytes can also be transmitted in this time interval. Remember that the transmitting device will police its transmission rate and will not exceed CIR in order to avoid having voice packets discarded by the frame relay network. Since the data traffic will have a six-byte frame relay header, the fragment size should be set to 84 bytes (90 - 6). Under these circumstances, we can therefore transmit six 25-byte voice packets and one 90-byte data packet in one time interval.

If we were to make the fragment size larger than 84 bytes under these circumstances then we run the risk of having voice traffic block data traffic completely. If we are handling voice traffic at the maximum rate of three simultaneous calls, and assuming for now that we always have voice traffic ready to be transmitted, transmission of a data fragment larger than 90 bytes will not be allowed because this would exceed the allowed number of bytes in the given time interval. More likely, though, would be the case where we can no longer transmit a voice packet because too much data traffic has already been transmitted in the time interval: either case is to be avoided where possible.

If we increase T_c to 0.06 (by setting B_c to 3840), the calculation changes. Now in the longer time interval we have the ability to send 180 bytes of data traffic, and so the fragment size should now be set to 174 bytes. This is more efficient for transporting data traffic but increases the potential delay to voice packets. Assume that at the beginning of the time interval that no voice packets are available for transmission but that data traffic is available for transmission, and furthermore that the frame relay access rate (and hence the speed at which the line is clocked) is 2 Mbps. The router will send 21 data fragments of 180 bytes each and not exceed the committed transmission rate in the time interval. It will

[1] For the purpose of these calculations we have made the assumption that 64 kbps means 64,000 bits per second and that a byte is exactly eight bits.

take approximately 15 milliseconds to transmit this volume of data traffic at the given access rate. It can then send no more data fragments during this time interval without exceeding the committed rate. Assume that immediately after this data traffic has been transmitted, three 25-byte voice packets are received. To avoid exceeding the contracted rate, only two of these packets can now be transmitted. Furthermore, if another three voice packets arrive after a further 15 milliseconds, none of these can be transmitted during this time interval, but at this point we are only halfway through the 60 millisecond time slot. Hence the trade-off: increasing T_c is more efficient for data traffic but will increase the average delay on the transmission of voice traffic. This does not claim to be a rigorous statistical analysis of the relationship between the T_c value and the average voice delay but some approximate relationship should be clear from the discussion above.

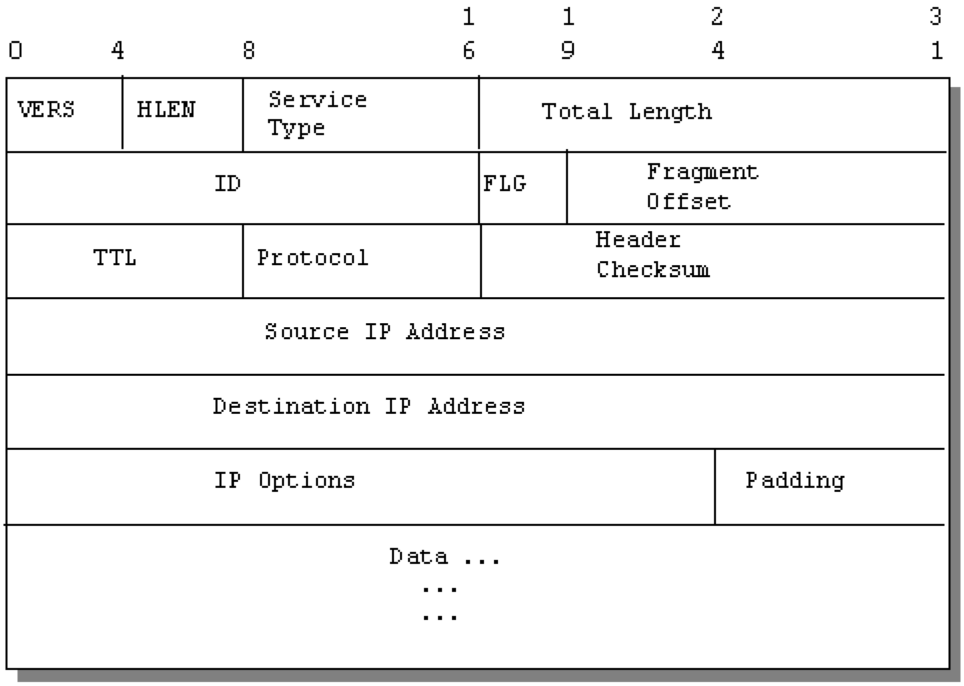
Now consider an example in which voice traffic is initially encapsulated inside IP before being transmitted over frame relay. Assume[2] that each voice circuit is going to require bandwidth for the transmission of 66 bytes of data every 15 milliseconds (20 bytes of voice data, 40 bytes of IP/UDP/RTP headers, 6 bytes of frame relay headers). This approximates to a bandwidth requirement of 35 kbps for each voice call, so we have to assume we can only support a single voice call over a 64 kbps frame relay voice/data circuit.

Performing similar calculations as before, T_c=0.03 leads to a data fragment size of 100 bytes and T_c=0.06 leads to a data fragment size of 208 bytes.

[2] The exact voice transmission rate depends on the characteristics of the CODEC being used, so the values here should be taken as indicative rather than definitive.

Appendix B. The IP datagram header

The IPv4 datagram header is at least 20 bytes in length:



```
                          1     1        2            3
0     4      8            6     9        4            1
+-----+------+------------+-----+--------------------+
|VERS | HLEN | Service    |      Total Length        |
|     |      | Type       |                          |
+-----+------+------------+-----+--------------------+
|           ID            |FLG  |   Fragment         |
|                         |     |   Offset           |
+------------+------------+-----+--------------------+
|    TTL     | Protocol   |      Header              |
|            |            |      Checksum            |
+------------+------------+--------------------------+
|              Source IP Address                     |
+----------------------------------------------------+
|            Destination IP Address                  |
+---------------------------------------+------------+
|            IP Options                 | Padding    |
+---------------------------------------+------------+
|                    Data ...                        |
|                      ...                           |
|                      ...                           |
+----------------------------------------------------+
```

Figure 72. The IPv4 datagram header

For more information on the format and content of IP datagrams, see:

- *TCP/IP Tutorial and Technical Overview*, GG24-3376

Appendix C. Special notices

This publication is intended to help anyone responsible for the design and implementation of networks to understand the jargon, buzzwords and acronyms and understand how networks can be designed and implemented to be better suited to the needs of their users.The information in this publication is not intended as the specification of any programming interfaces that are provided by any of the products described in the book. See the PUBLICATIONS section of the IBM Programming Announcement for IBM routers (2210, 2212, and 2216) and CS for OS/390 for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The information about non-IBM ("vendor") products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

Any performance data contained in this document was determined in a controlled environment, and therefore, the results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

APPN
DB2
EtherJet
Netfinity
OS/390
S/390
System/390
400

AS/400
ESCON
IBM
Nways
RS/6000
SP
XT

The following terms are trademarks of other companies:

Tivoli, Manage. Anything. Anywhere.,The Power To Manage., Anything. Anywhere.,TME, NetView, Cross-Site, Tivoli Ready, Tivoli Certified, Planet Tivoli, and Tivoli Enterprise are trademarks or registered trademarks of Tivoli Systems Inc., an IBM company, in the United States, other countries, or both. In Denmark, Tivoli is a trademark licensed from Kjøbenhavns Sommer - Tivoli A/S.

C-bus is a trademark of Corollary, Inc. in the United States and/or other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and/or other countries.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States and/or other countries.

PC Direct is a trademark of Ziff Communications Company in the United States and/or other countries and is used by IBM Corporation under license.

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States and/or other countries.

UNIX is a registered trademark in the United States and other countries licensed exclusively through The Open Group.

SET and the SET logo are trademarks owned by SET Secure Electronic Transaction LLC.

Other company, product, and service names may be trademarks or service marks of others.

Appendix D. Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

D.1 IBM Redbooks publications

For information on ordering these publications see “How to get IBM Redbooks” on page 143.

- *Application-Driven Networking: Concepts and Architecture for Policy-Based Systems*, SG24-5640
- *MSS Release 2.1, Including the MSS Client and Domain Client*, SG24-5231
- *Layer 3 Switching Using MSS and MSS Release 2.2 Enhancements*, SG24-5311
- *IP Network Design Guide*, SG24-2580
- *TCP/IP Tutorial and Technical Overview*, GG24-3376

D.2 IBM Redbooks collections

Redbooks are also available on the following CD-ROMs. Click the CD-ROMs button at `http://www.redbooks.ibm.com/` for information about all the CD-ROMs offered, updates and formats.

CD-ROM Title	**Collection Kit Number**
System/390 Redbooks Collection	SK2T-2177
Networking and Systems Management Redbooks Collection	SK2T-6022
Transaction Processing and Data Management Redbooks Collection	SK2T-8038
Lotus Redbooks Collection	SK2T-8039
Tivoli Redbooks Collection	SK2T-8044
AS/400 Redbooks Collection	SK2T-2849
Netfinity Hardware and Software Redbooks Collection	SK2T-8046
RS/6000 Redbooks Collection (BkMgr Format)	SK2T-8040
RS/6000 Redbooks Collection (PDF Format)	SK2T-8043
Application Development Redbooks Collection	SK2T-8037
IBM Enterprise Storage and Systems Management Solutions	SK3T-3694

D.3 Other resources

These publications are also relevant as further information sources:

- *SNA Technical Overview*, GC30-3073
- *LAN Emulation Over ATM Version 1.0, January 1995,* available from:
  The ATM Forum Worldwide Headquarters
  2570 West El Camino Real, Suite 304
  Mountain View, CA 94040-1313, USA
- *LAN Emulation over ATM Version 2, July 1997,* available from:
  The ATM Forum Worldwide Headquarters
  2570 West El Camino Real, Suite 304
  Mountain View, CA 94040-1313, USA

- *Information Technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Common specifications - Part 3: Media Access Control (MAC) Bridges (Incorporating IEEE P802.1p: Traffic Class Expediting and Dynamic Multicast Filtering*), available from:
  Institute of Electrical and Electronics Engineers, Inc.
  345 East 47th Street
  New York, NY 10017, USA
- *IEEE Standards for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks,* available from:
  Institute of Electrical and Electronics Engineers, Inc.
  345 East 47th Street
  New York, NY 10017, USA

D.4 Referenced Web sites

These Web sites are also relevant as further information sources:

- `http://www.ietf.org` Internet Engineering Task Force
- `http://www.networking.ibm.com` IBM networking home page
- http://www.redbooks.ibm.com IBM Redbooks home page
- http://w3.itso.ibm.com IBM intranet for ITSO home page
- http://w3.ibm.com IBM intranet home page
- http://www.elink.ibmlink.ibm.com/pbl/pbl for ordering IBM Redbooks
- `http://www.software.ibm.com/enetwork/commserver/downloads/demos/csos390.html`
  Communications Server for OS/390 technology demonstrations

How to get IBM Redbooks

This section explains how both customers and IBM employees can find out about IBM Redbooks, redpieces, and CD-ROMs. A form for ordering books and CD-ROMs by fax or e-mail is also provided.

- **Redbooks Web Site** http://www.redbooks.ibm.com/

  Search for, view, download, or order hardcopy/CD-ROM Redbooks from the Redbooks Web site. Also read redpieces and download additional materials (code samples or diskette/CD-ROM images) from this Redbooks site.

  Redpieces are Redbooks in progress; not all Redbooks become redpieces and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

- **E-mail Orders**

  Send orders by e-mail including information from the IBM Redbooks fax order form to:

	e-mail address
In United States	usib6fpl@ibmmail.com
Outside North America	Contact information is in the “How to Order” section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl

- **Telephone Orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	Country coordinator phone number is in the “How to Order” section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl

- **Fax Orders**

United States (toll free)	1-800-445-9269
Canada	1-403-267-4455
Outside North America	Fax phone number is in the “How to Order” section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl

This information was current at the time of publication, but is continually subject to change. The latest information may be found at the Redbooks Web site.

IBM Intranet for Employees

IBM employees may register for information on workshops, residencies, and Redbooks by accessing the IBM Intranet Web site at http://w3.itso.ibm.com/ and clicking the ITSO Mailing List button. Look in the Materials repository for workshops, presentations, papers, and Web pages developed and written by the ITSO technical professionals; click the Additional Materials button. Employees may access MyNews at http://w3.ibm.com/ for redbook, residency, and workshop announcements.

IBM Redbooks fax order form

Please send me the following:

Title	**Order Number**	**Quantity**

First name Last name

Company

Address

City Postal code Country

Telephone number Telefax number VAT number

□ Invoice to customer number

□ Credit card number

Credit card expiration date Card issued to Signature

We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.

List of abbreviations

AAL	ATM Adaption Layer
ABR	available bit rate
ADSPEC	Advertisement Specification (RSVP)
AF	Assured Forwarding
APPN	Advanced Peer-to-peer Networking
ATM	asynchronous transfer mode
BGP	Border Gateway Protocol
BRS	Bandwidth Reservation System
BUS	Broadcast and Unknown Server
CBR	constant bit rate
CDV	cell delay variation
CDVT	cell delay variation tolerance
CFI	Canonical Format Indicator
CIR	committed information rate
CLP	cell loss priority
CLR	cell loss ratio
CODEC	Coder Decoder
COS	Class of Service
CS for OS/390	Communications Server for OS/390
CSMA/CD	Carrier Sense Multiple Access with Collision Detection
CSR	Cell Switch Router (Toshiba)
CTD	cell transfer delay
DE	discard eligible (frame relay)
DiffServ	Differentiated Services
DIX	DEC, Intel, Xerox
DLCI	Data Link Control Identifier (frame relay)
DLL	Data Link Layer
DLSw	data link switching
DLUR	dependent LU requester
DS	Differentiated Services
DTE	Data Transmission Equipment
EF	Expedited Forwarding
ELS	Event Logging Subsystem
E-RIF	Embedded Source-routing Information Field
ESCON	Enterprise Systems Connection
FDDI	Fibre Distributed Data Interface
FEC	Forwarding Equivalence Class
FRFH	Frame Relay Frame Handler
GARP	Generic Attribute Registration Protocol
GMRP	GARP Multicast Registration Protocol
GVRP	GARP VLAN Registration Protocol
HFS	hierarchical file system
HPR	High Performance Routing (APPN)
HSSI	High Speed Serial Interface
IBM	International Business Machines Corporation
ICMP	Internet Control Message Protocol
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IntServ	Integrated Services
IP	Internet Protocol
IPX	Internetwork Packet Exchange
ISR	Intermediate Session Routing (APPN)
ITSO	International Technical Support Organization
LAN	local area network
LANE	LAN Emulation
LDAP	Lightweight Directory Access Protocol
LDP	Label Distribution Protocol
LEC	LAN Emulation Client
LECS	LAN Emulation Configuration Server
LER	Label Edge Router
LES	LAN Emulation Server
LSP	label switch path
LSR	label switch router

LU	logical unit (SNA)
MAC	media access control
MARS	Multicast Address Resolution Server
MBS	maximum burst size
MCR	minimum cell rate
MPOA	Multiprotocol Over ATM
MPLS	Multiprotocol Label Switching
MSS	Multiprotocol Switched Services
MTU	maximum transmission unit
NetBIOS	Network Basic Input/Output System
NHLFE	Next Hop Label Forwarding Entry
NHRP	Next Hop Resolution Protocol
NLP	Network Layer Packet
NNI	Network-to-Network Interface
OPWA	One Path With Advertising
OS/390	Operating System/390
OSPF	Open Shortest Path First
PAGENT	Policy Agent
PATH	RSVP Path discovery message
PCR	peak cell rate
PDP	policy decision point
PFC	PHB Forwarding Class
PHB	per-hop behavior
PTF	Program Temporary Fix
PVC	permanent virtual circuit
PVID	Port VLAN Identifier
QDIO	Queued Direct I/O
QoS	Quality of Service
RESV	RSVP Reservation Request
RFC	Request for Comments
RIF	Routing Information Field
RIP	Routing Information Protocol
RSVP	Resource Reservation Protocol
RSVPD	RSVP Daemon
RTCP	Real Time Control Protocol
RTP	Real Time Protocol
S/390	System/390
SCR	sustainable cell rate
SDU	service data unit
SIN	ships in the night
SNA	Systems Network Architecture
SNMP	Simple Network Management Protocol
SSCS	Service-Specific Convergence Sublayer
STP	Shielded Twisted Pair
SVC	switched virtual circuit
TCP	Transmission Control Protocol
TFTP	Trivial File Transfer Protocol
TG	transmission group (APPN)
TN3270	Telnet 3270
TOS	type of service
TCI	Tag Control Information
TPI	Tag Protocol Identifier
TSPEC	Traffic Specification (RSVP)
TTL	time to live
UBR	unspecified bit rate
UDP	User Datagram Protocol
UNI	user-network interface
VC	virtual circuit
VCC	virtual circuit connection
VCI	Virtual Channel Identifier
VCRM	Virtual Circuit Reservation Manager
VID	VLAN Identifier
VLAN	Virtual LAN
VoFR	voice over frame relay
VoIP	voice over IP
VPI	Virtual Path Identifier
WAN	wide area network

Index

IBM Redbooks evaluation

Application-Driven Networking: Class of Service in IP, Ethernet, and ATM Networks
SG24-5384-00

Your feedback is very important to help us maintain the quality of IBM Redbooks. **Please complete this questionnaire and return it using one of the following methods:**

- Use the online evaluation form found at http://www.redbooks.ibm.com/
- Fax this form to: USA International Access Code + 1 914 432 8264
- Send your comments in an Internet note to redbook@us.ibm.com

Which of the following best describes you?
_ Customer _ Business Partner _ Solution Developer _ IBM employee
_ None of the above

Please rate your overall satisfaction with this book using the scale:
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)

Overall Satisfaction __________

Please answer the following questions:

Was this redbook published in time for your needs? Yes___ No___

If no, please explain:

What other Redbooks would you like to see published?

Comments/Suggestions: (THANK YOU FOR YOUR FEEDBACK!)

SG24-5384-00

Printed in the U.S.A.

Application-Driven Networking: Class of Service in IP, Ethernet, and ATM Networks

SG24-5384-00