This chapter covers the following exam topics specific to the DQOS and QoS exams:

DQOS Exam Topics

- Explain the reason for classification and marking.
- Explain the difference between classification and marking.
- Explain class of service, IP precedence, and DiffServ code points.
- Configure QoS policy using Modular QoS CLI.
- Explain the role of network-based application recognition (NBAR).
- Classify and mark traffic.

QoS Exam Objectives

- Describe policy-based routing and how it can be used to classify and mark IP packets.
- Configure the policy-based routing mechanism on Cisco routers.
- List other mechanisms that also support classification and marking capabilities (committed access rate, class-based marking).
- Describe the Modular QoS CLI (MQC) concept and its structure.
- Describe Modular QoS CLI classification options.
- Configure the Modular QoS CLI to perform classification.
- Describe network-based application recognition (NBAR).
- Describe Modular QoS CLI policy options.
- Configure the Modular QoS CLI to perform service policies.

CHAPTER 3

Classification and Marking

QoS classification tools categorize packets by examining the contents of the frame, cell, and packet headers; whereas marking tools allow the QoS tool to change the packet headers for easier classification. Many QoS tools rely on a classification function to determine to which traffic the tool applies. To place voice and data traffic in separate queues, for example, you must use some form of classification to differentiate the two types of traffic and place the identified traffic in the proper queue. Marking provides a way for QoS tools to change bits in the packet header to indicate the level of service this packet should receive from other QoS tools. For instance, you can use marking tools to change the marking in voice packets to ensure that a classification tool can differentiate a voice packet from a data packet. Without the marking feature, the frame, packet, or cell remains unchanged.

Marking involves placing a value into one of the small number of well-defined frame, packet, or cell header fields specifically designed for QoS marking. By marking a packet, other QoS functions can perform classification based on the marked field inside a header. Marking simplifies the network's QoS design, it simplifies configuration of other QoS tools, and it reduces the overhead required by each of the other QoS tools to classify the packets.

Although classification and marking tools do not directly affect the bandwidth, delay, jitter, or loss experienced by traffic in the network, classification and marking tools are the building blocks for all other QoS tools. With these tools, all traffic on the network is identified for the next QoS tool to act upon.

The concepts that apply to all classification and marking are covered in the first section of this chapter, including the terminology, fields used, and the meaning behind each of the available marked fields. Following that, each of the classification and marking tools is covered, with example configurations, **show**, and **debug** commands.

"Do I Know This Already?" Quiz Questions

The purpose of the "Do I Know This Already?" quiz is to help you decide whether you really need to read the entire chapter. If you already intend to read the entire chapter, you do not necessarily need to answer these questions now.

The 8-question quiz, derived from the major sections in "Foundation Topics" portion of the chapter, helps you determine how to spend your limited study time.

Table 3-1 outlines the major topics discussed in this chapter and the "Do I Know This Already?" quiz questions that correspond to those topics.

Table 3-1 *"Do I Know This Already?" Foundation Topics Section-to-Question Mapping*

Quizlet Number	Foundation Topics Section Covering These Questions	Questions
1	Classification and Marking Concepts	1 to 4
2	CAR, PBR, and CB Marking	5 to 8
All questions		1 to 8

CAUTION The goal of self-assessment is to gauge your mastery of the topics in this chapter. If you do not know the answer to a question or are only partially sure of the answer, mark this question wrong for purposes of the self-assessment. Giving yourself credit for an answer you correctly guess skews your self-assessment results and might provide you with a false sense of security.

You can find the answers to the "Do I Know This Already?" quiz in Appendix A, "Answers to the 'Do I Know This Already?' Quizzes and Q&A Sections." The suggested choices for your next step are as follows:

- **6 or less overall score**—Read the entire chapter. This includes the "Foundation Topics," the "Foundation Summary," and the "Q&A" sections.
- **7 or 8 overall score**—If you want more review on these topics, skip to the "Foundation Summary" section and then go to the "Q&A" section. Otherwise, move to the next chapter.

Classification and Marking Concepts Questions

1. Describe the difference between classification and marking.
2. Describe, in general, how a queuing feature could take advantage of the work performed by a classification and marking feature.
3. Which of the following QoS marking fields are carried inside an 802.1Q header: QoS, CoS, DE, ToS byte, User Priority, ToS bits, CLP, Precedence, QoS Group, DSCP, MPLS Experimental, or DS?
4. Which of the following QoS marking fields are carried inside an IP header: QoS, CoS, DE, ToS byte, User Priority, ToS bits, CLP, Precedence, QoS Group, DSCP, MPLS Experimental, or DS?

CAR, PBR, and CB Marking Questions

5 Define the meaning of MQC, and spell out what the acronym stands for.

6 What configuration command lists the marking details when configuring CB marking? What configuration mode must you use to configure the command? What commands must you issue to place the configuration mode user into that mode?

7 What configuration command lists the marking details when configuring CAR? What configuration mode must you use to configure the command? What commands must you issue to place the configuration mode user into that mode?

8 What configuration command lists the classification details when configuring PBR? What configuration mode must you use to configure the command? What commands must you issue to place the configuration mode user into that mode?

Foundation Topics

The contents of the “Foundation Topics” section of this chapter, and most of the rest of the chapters in this book, follow the same overall flow. Each chapter describes a type of category of QoS tool. Each “Foundation Topics” section begins with coverage of the concepts behind these tools. Then, each tool is examined, with coverage of how each tool works like the other tools, and how it works differently than the other tools. So, most of the core concepts are explained in the first part of the chapter; some of the concepts may be explained in the section about a specific tool, however, particularly if the concepts apply only to that tool.

The second part of the chapter covers several classification and marking tools: class-based marking (CB marking), committed access rate (CAR), policy-based routing (PBR), and dial peers. For each tool, the pertinent configuration, **show**, and **debug** commands are also covered.

Classification and Marking Concepts

Most QoS tools classify traffic, which allows for each class of traffic to receive a different level of treatment from other traffic classes. You can use this method to prioritize one type of traffic over another. Classification and marking tools play a large role in this solution. Instead of giving queuing preference, or dropping a packet, or shaping, or policing, or fragmenting, and so on, classification and marking tools change some bits in the packet header. Other QoS tools throughout the network can examine the marked bits to classify packets.

First, this discussion focuses on classification, and then examines marking more closely.

Classification

Almost every QoS tool uses classification to some degree. To put one packet into a different queue than another packet, the IOS must somehow differentiate between the two packets. To perform header compression on Real Time Protocol (RTP) packets, but not on other packets, the IOS must determine which packets have RTP headers. To shape data traffic going into a Frame Relay network, so that the voice traffic gets enough bandwidth, the IOS must differentiate between Voice over IP (VoIP) and data packets. If an IOS QoS feature needs to treat two packets differently, you must use classification.

Because most QoS tools need to differentiate between packets, most QoS tools have classification features. In fact, many of you will already know something about several of the QoS tools described in this book, and you will realize that you already know how to perform classification using some of those tools. For instance, many QoS tools enable you to classify using access-control lists (ACLs). If ACL 101 *permits* a packet, a queuing tool might put the packet into one queue; if ACL 102 permits a packet, it is placed in a second queue; and so on. In one way of thinking, queuing could instead be called *classification and queuing*, because the queuing feature must somehow decide which packets end up in each queue. Similarly, traffic shaping could be called *classification and traffic shaping*, policing could be called *classification and policing*, and so on. Because most QoS tools classify traffic, however, the names of most QoS tools never evolved to mention the classification function of the tool.

Most classification and marking tools, like the other types of QoS tools, generally operate on packets that are entering or exiting an interface. The logic works something like an ACL, but the *action* is marking, as opposed to allowing or denying (dropping) a packet. More generally, classification and marking logic for ingress packets can be described as follows:

- For packets entering an interface, if they match criteria 1, mark a field with a value.
- If the packet was not matched, compare it to criteria 2, and then mark a potentially different field with a potentially different value.
- Keep looking for a match of the packet, until it is matched, or until the classification logic is complete.

NOTE This book uses the following terms describe the data structures used when sending data:

- **Frame**—Bits that include the data link layer header and trailer (for example, Ethernet frame and Frame Relay frame)
- **Cell**—Specifically, an Asynchronous Transfer Mode (ATM) cell
- **Packet**—Bits that include the network layer header, but does not include the data link header (for instance, an IP packet)
- **Segment**—Bits that include the TCP or UDP header, but not the data link or network layer header

The key to evaluating the classification features of different classification and marking tools is to examine what can be matched in packet headers. Because many classification and marking tools can refer to IP ACLs to classify packets, Table 3-2 shows the list of items that you can match with an extended IP ACL. Table 3-3 lists the fields that classification and marking tools can match without use of an ACL. Note that some header fields can be matched by an ACL or directly through some other style of configuration—in those cases, it is typically better to match the field directly, rather than with an ACL.

Table 3-2 *IP Extended ACL Matchable Fields—IOS 12.2*

Field	Comments
Source IP address	A range of source IP addresses can be matched by using a wildcard mask.
Destination IP address	A range of source IP addresses can be matched by using a wildcard mask.

continues

Table 3-2 *IP Extended ACL Matchable Fields—IOS 12.2 (Continued)*

Field	Comments
IP Precedence	Format of command uses names for precedence. The following table lists the decimal value for each name. Name IP precedence value **routine** 0 **priority** 1 **immediate** 2 **flash** 3 **flash-override** 4 **Critic** 5 **internet** 6 **network** 7
IP DSCP	Format of the command allows use of differentiated services code point (DSCP) names, as well as decimal values.
IP ToS	Can check to see whether a single Type of Service (ToS) field bit is toggled on; keywords are **normal** (binary **0000**), **max-reliability** (binary **1000**), **max-throughput** (binary **0100**), **min-delay** (binary **0010**), and **min-monetary-cost** (binary **0001**).
TCP ports	Can check source and destination ports; can also check a range of port numbers, whether a port number is larger or smaller than a single value.
TCP Established	Although not typically useful for QoS classification, ACLs can match all TCP segments after the initial segment used for connection establishment.
UDP	Checks the source and destination ports; can also check a range of port numbers, whether a port number is larger or smaller than a single value.

Table 3-2 *IP Extended ACL Matchable Fields—IOS 12.2 (Continued)*

Field	Comments
ICMP	Checks a larger variety of ICMP messages and code types (for example, echo request and echo reply).
IGMP	Checks for Internet Group Management Protocol (IGMP) message types.

Table 3-3 *Fields* Directly *Matchable by Classification and Marking Tools*

Field	Tool	Comments
Source MAC address	CAR, CB marking	Committed access rate (CAR) uses special "access-rate" ACLs; class-based (CB) marking uses the **match** command.
IP Precedence	CAR, CB marking	CAR uses special "access-rate" ACLs specific to CAR; CB marking uses the **match** command; both can match a subset of values.
MPLS Experimental	CAR, CB marking	CAR uses special "access-rate" ACLs specific to CAR; CB marking uses the **match** command; both can match a subset of values.
CoS	CB marking	Checks incoming ISL/802.1P CoS bits. Can match multiple values.
Destination MAC address	CB marking	Checks for destination MAC address. Can match multiple values.
Input Interface	CB marking	Checks for input interface. Can match multiple values.
IP DSCP	CB marking	Can check for multiple values using multiple **match** commands.
RTP's UDP port-number range	CB marking	RTP uses even-numbered UDP ports from 16,384 to 32,767. This option allows matching a subset of these values, even-numbered ports only, because RTP only uses even-numbered ports.
QoS Group	CB marking	The QoS Group field is used to tag packets internal to a single router.
NBAR protocol types	CB marking	Refer to the "Network Based Application Recognition (NBAR)" section in this chapter for more details.
NBAR Citrix applications	CB marking	NBAR can recognize different types of Citrix applications; CB marking can use NBAR to classify based on these application types.

continues

Table 3-3 *Fields* Directly *Matchable by Classification and Marking Tools (Continued)*

Field	Tool	Comments
Host name and URL string	CB marking	NBAR can also match URL strings, including the host name, using regular expressions. CB marking can use NBAR to match these strings for classification.
Outgoing Interface	Policy-based routing (PBR)	Checks the routing table and finds all valid routes for the packet; matches based on the outgoing interface.
Next-Hop	PBR	Similar to the outgoing interface, but it checks the next-hop routers' IP addresses.
Metric	PBR	Checks the routing table entry for this packet, and compares the metric value to match the packet.
Route type	PBR	Checks the routing table, looking at the source of the routing table entry that matches the packet.
Dial Peer	Dial peers	Based on the dial peer and used to connect a VoIP call.

These two tables can be a little intimidating, especially for those of you studying to pass the QoS exams! Rest assured, however, that this book covers approximately 30 QoS tools, and the exams typically have no more than 60 questions on QoS. So, statistically speaking, the exams simply do not have enough questions to ask you about every little item that a tool could use for classification. I just included these tables, and others like them, for reference.

Marking

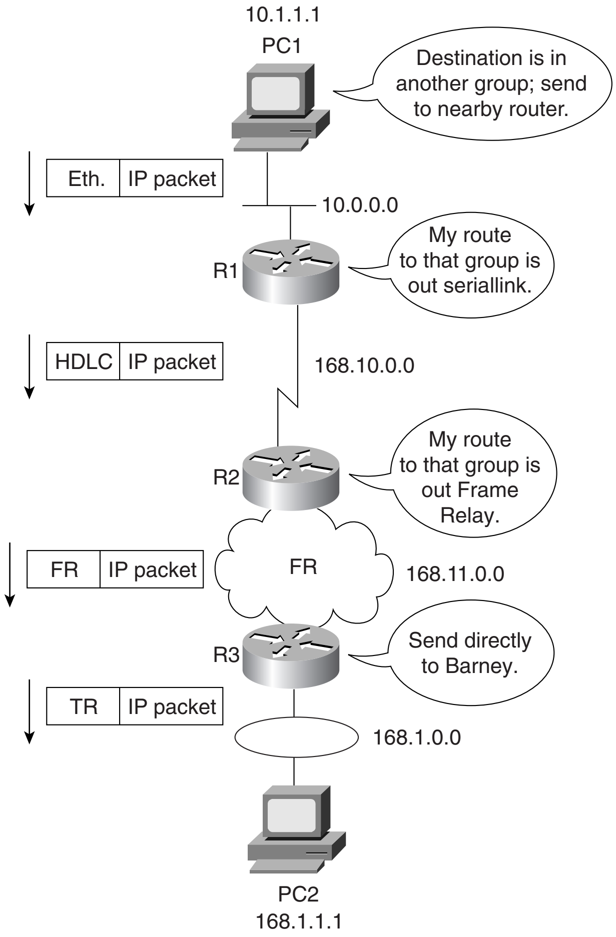
Marking involves setting some bits inside a data link or network layer header, with the goal of letting other devices' QoS tools classify based on the marked values. You can mark a wide variety of fields, and each has a particular purpose. Some fields are more widely used, and some are less widely used. Some marking options make sense for all devices on the LAN, whereas others only when using specific hardware platforms. Marking at the WAN is possible, too.

The following sections list the header fields that you can use for marking, along with explanations of when it is most useful to use that particular field. Recommendations follow these sections as to when to use classification and marking.

IP Header QoS Fields: Precedence and DSCP

The two most popular marking fields for QoS are the IP Precedence and IP DSCP fields that were introduced in Chapter 2, “QoS Tools and Architectures.” QoS tools frequently use these two fields in part because the IP packet header exists from endpoint to endpoint in a network, as shown in Figure 3-1.

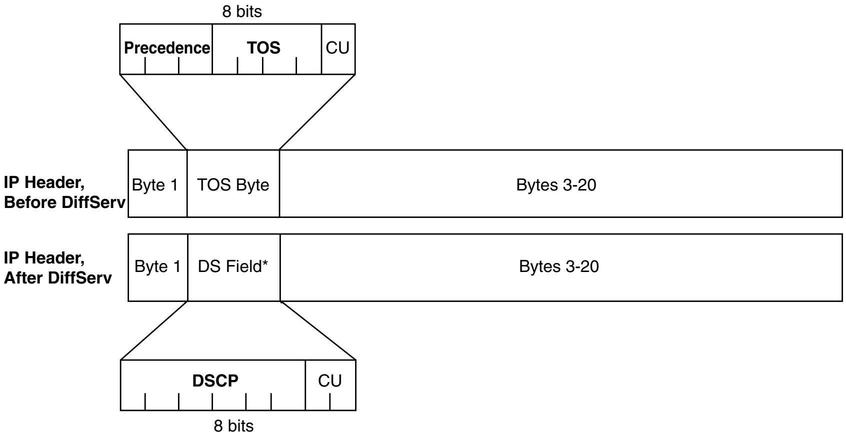
Figure 3-1 *Headers Used During Typical Packet Flow*



As seen in Figure 3-1, the IP packet en route to host PC2 stays intact throughout the network, whereas the data-link headers only exist while a frame is crossing between a host and a router, or between routers.

Figure 3-2 outlines the two marking fields and their positions inside an IP header.

Figure 3-2 *IP Precedence and IP DSCP Fields*



You can mark the Precedence and DSCP fields with any valid binary value of either 3 or 6 bits, respectively. Chapter 2 contains detailed discussion of the recommended values used in these two fields. Briefly, Precedence field values should grow in importance, and in QoS behavior, as the number gets higher. DSCP differs in that several per-hop behavior (PHB) RFCs define suggested DSCP values for which the larger number does not always get a better QoS treatment.

Table 3-4 lists the IP precedence and DSCP values, and their names, for review. Note that not all DSCP values are listed; only the DSCP values suggested by the DiffServ RFCs are listed in the table. QoS tools that are capable of setting DSCP can set any of the actual 64 values.

Table 3-4 *IP Precedence and DSCP—Popular Values and Names*

Field and Value (Decimal)	Binary Value	Name	Defined by This RFC
Precedence 0	**000**	**routine**	791
Precedence 1	**001**	**priority**	791
Precedence 2	**010**	**immediate**	791
Precedence 3	**011**	**flash**	791
Precedence 4	**100**	**flash override**	791
Precedence 5	**101**	**critic**	791
Precedence 6	**110**	**internetwork control**	791
Precedence 7	**111**	**network control**	791
DSCP 0	**000**000	**best effort** or **default**	2475

Table 3-4 *IP Precedence and DSCP—Popular Values and Names (Continued)*

Field and Value (Decimal)	Binary Value	Name	Defined by This RFC
DSCP 8	**001**000	CS1	2475
DSCP 16	**010**000	CS2	2475
DSCP 24	**011**000	CS3	2475
DSCP 32	**100**000	CS4	2475
DSCP 40	**101**000	CS5	2475
DSCP 48	**110**000	CS6	2475
DSCP 56	**111**000	CS7	2475
DSCP 10	**001**010	AF11	2597
DSCP 12	**001**100	AF12	2597
DSCP 14	**001**110	AF13	2597
DSCP 18	**010**010	AF21	2597
DSCP 20	**010**100	AF22	2597
DSCP 22	**010**110	AF23	2597
DSCP 26	**011**010	AF31	2597
DSCP 28	**011**100	AF32	2597
DSCP 30	**011**110	AF33	2597
DSCP 34	**100**010	AF41	2597
DSCP 36	**100**100	AF42	2597
DSCP 38	**100**110	AF43	2597
DSCP 46	**101**110	EF	2598

CS = Class Selector
AF = Assured Forwarding
EF = Expedited Forwarding

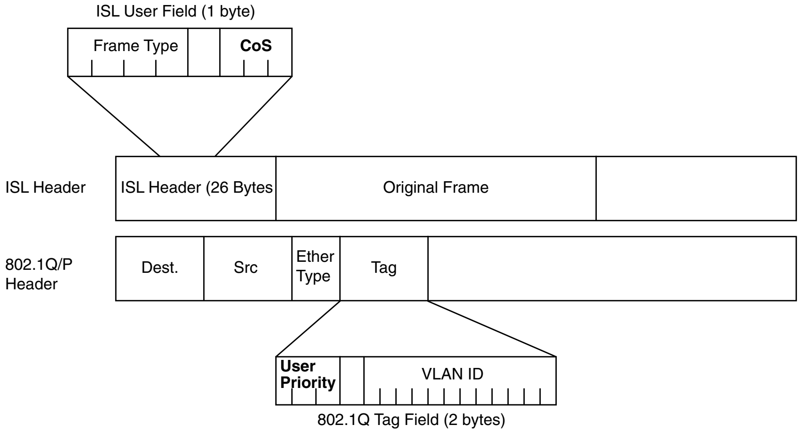
The two IP header QoS marking fields do not provide all the QoS marking fields needed today. One day, all other Layer 3 protocols besides IP may no longer be used. One day, all LAN switches will be capable of looking at IP headers, including IP DSCP and Precedence, and perform QoS based on those fields. Likewise, one day, all WAN services, including Frame Relay and ATM switches, will be able to perform QoS based on these same fields. However, today's reality is that even as more and more devices become capable of marking and reacting to IP precedence and DSCP, it will be a long time before all networking devices are both capable and configured to use these fields for QoS purposes. So, other QoS marking fields are needed.

LAN Class of Service (CoS)

Many LAN switches today can mark and react to a Layer 2 3-bit field called the Class of Service (CoS) located inside an Ethernet header. The CoS field only exists inside Ethernet frames when 802.1Q or Inter-Switch Link (ISL) trunking is used. You can use the field to set 8 different binary values, which can be used by the classification features of other QoS tools, just like IP precedence and DSCP.

Figure 3-3 shows the general location of the CoS field inside ISL and 802.1P headers.

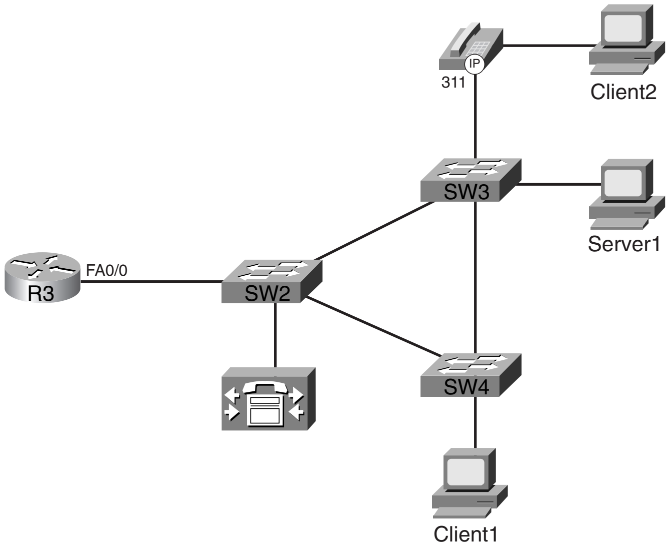
Figure 3-3 *LAN CoS Fields*



The term *CoS* really refers to two different fields—a field inside either the 802.1Q trunking header, or a field inside the ISL header. The IEEE 802.1Q standard uses the 3 most-significant bits of the 2-byte Tag Control field, called the user-priority bits. The Cisco proprietary ISL specification uses the 3 least-significant bits from the 1-byte User field, which is found inside the ISL header's user field byte. In general conversation, and in the QoS courses from Cisco, the term CoS applies to either of these two fields.

When can CoS be marked, and when can it be useful for classification to a QoS tool? First of all, trunking with 802.1Q or ISL must be enabled before the CoS field even exists! Second, as soon as the packet experiences Layer 3 forwarding, either with a router or some Layer 3 switch, the old LAN header gets discarded—which means you lose the CoS field. After a CoS field has been created and populated with the desired markings, routers and switches have several QoS tools that can react to these markings. Consider, for instance, a typical trunking environment, as shown in Figure 3-4, where all LAN switches are only performing Layer 2 switching.

Figure 3-4 *CoS—Trunking Locations in a Typical Network, Layer 2 Switches Only*



To mark the CoS bits, trunking must be used—and in Figure 3-4, trunking could be used on every Ethernet segment. Switches typically use trunking on the Ethernet segments to other switches, routers, and to IP Phones. Typically, switches do not need to use trunking on segments connected to PCs or servers. Because some networking cards have the capability to support 802.1Q or ISL trunking, however, servers and PCs can set the CoS bits.

NOTE Trunking requires a Fast Ethernet interface, or Gigabit, or 10 Gigabit—it is not supported over 10-Mbps Ethernet. This book does not distinguish among the different types of Ethernet upon each mention.

Both routers and switches use QoS tools that can react to the marked CoS bits. Cisco routers can indeed mark CoS bits for frames exiting an Ethernet interface that supports trunking. For instance, R3 could mark CoS 5 on a frame it forwards out its FA 0/0 interface. Other Cisco router QoS tools can react to the marked CoS bits on incoming frames as well. For instance, R3 could mark packets entering its FA0/0 interface with a particular DSCP value based on the incoming CoS value. Later in this chapter, you will see a sample configuration for class-based marking that performs both of these functions.

Cisco switches vary widely regarding their capabilities to set CoS bits and react to previously marked CoS bits. Switches can support marking of CoS, and more often today support marking of IP precedence and DSCP as well. LAN switches that do support QoS features generally perform output queuing, and sometimes input queuing, choosing queues based on CoS values. Congestion avoidance using Weighted Random Early Detection (WRED) is another typical switch QoS feature. In addition, some switches support policing tools, also based on CoS. Although campus QoS is not covered in depth on the QoS exams today, it is an important topic, particularly with converged voice, video, and data networks. Chapter 10, "LAN QoS," covers LAN QoS in additional depth.

Other Marking Fields

You can use single-bit fields in Frame Relay and ATM networks to mark a frame or cell for Layer 2 QoS. Unlike IP precedence, IP DSCP, and 802.1P/ISL CoS, however, these two fields are not intended for general, flexible use. Each of these single-bit fields, when set, imply that the frame or cell is a better candidate to be dropped, as compared with frames or cells that do not have the bit set. In other words, you can mark the bit, but the only expected action by another QoS tool is for the tool to drop the frame or cell.

Frame Relay defines the discard eligibility (DE) bit, and ATM defines the cell loss priority (CLP) bit. The general idea is that when a device, typically a WAN switch, experiences congestion, it needs to discard some frames or cells. If a frame or cell has the DE or CLP bit set, respectively, the switch may choose to discard those frames or cells, and not discard other frames or cells. If the DE or CLP bit is set, there is no requirement that the Frame Relay and ATM switches react to it—just like there is no guarantee that an IP packet with DSCP EF will get special treatment by another router. It's up to the owner of the Frame Relay or ATM switch to decide whether it will consider the DE and CLP bits, and how to react differently.

You can use two other QoS marking fields in specialized cases. The MPLS Experimental bits comprise a 3-bit field that you can use to map IP precedence into an MPLS label. This allows MPLS routers to perform QoS features indirectly based on the original IP Precedence field inside the IP packets encapsulated by MPLS, without the need to spend resources to open the IP packet header and examine the IP Precedence field.

Finally, the QoS Group field, an internal marking that exists only within the router, may be set as a packet passes through the fabric of a Cisco gigabit switch router (GSR) or edge services router (ESR). QoS processing can be performed more quickly inside the switching fabric by using the QoS group. Therefore, you may want to configure GSRs and ESRs to mark the QoS group on ingress so that QoS processing occurs more rapidly.

Summary of Marking Fields

Not all these marked fields receive the same amount of attention on the QoS exams. Refer to the Introduction of this book, and the website suggested there, for the latest information about where to focus your attention. Table 3-5 summarizes the marking fields.

Table 3-5 *Names of Marking Fields*

Field	Location	Length	Comments
IP Precedence	IP header	3 bits	Contained in the first 3 bits of the ToS byte.
IP DSCP	IP header	6 bits	Contained in the first 6 bits of the DS field, which replaces the ToS byte.
DS	IP header	1 byte	Replaces ToS byte per RFC 2475.
ToS	IP header	1 byte	Replaced by DS field per RFC 2475.
ToS	IP header	4 bits	A field inside the ToS byte; superseded by RFC 2475.
CoS	ISL and 802.1Q/P	3 bits	Cisco convention uses “CoS” to describe either trunking headers’ QoS field.
Priority bits	802.1Q/P	3 bits	The name used by IEEE 802.1P for the CoS bits.
Discard Eligible (DE)	Frame Relay header	1 bit	Frame Relay switches may discard DE-marked frames, avoiding discarding frames without DE marked, under congestion.
Cell Loss Priority (CLP)	ATM cell header	1 bit	ATM equivalent of the DE bit
MPLS Experimental values(s)	MPLS header	3 bits	Used to pass QoS marking information across an MPLS network.
QoS Group	Headers internal to IOS	N/A	Uses values between 1–99 inclusive. Used for marking only internal to a single router, specifically only on the GSR/ESR product lines.

The names of the various marking fields can be confusing. Quality of service (QoS) does not refer to any specific marking field, but it is a term that refers to a broad set of tools that effect bandwidth, delay, jitter, and loss. In other words, this whole book is about QoS. Class of service (CoS) refers to both of the two 3-bit fields in Ethernet trunking headers—one in the ISL header, and one in the 802.1Q trunking header. However, CoS also refers to a 2-bit field inside Systems Network Architecture (SNA) Layer 3 headers, which is also used for QoS functions. Type of service (ToS) is my personal favorite—ToS is the 1-byte field in the IP header, which includes a 3-bit Precedence field, and 4 ToS bits. And of course, DiffServ re-defines the ToS Byte as the DS-byte, with the DSCP field in the first 6 bits. Make sure you remember the true meanings of QoS, CoS, ToS, Precedence, and DSCP.

Classification and Marking Design Choices

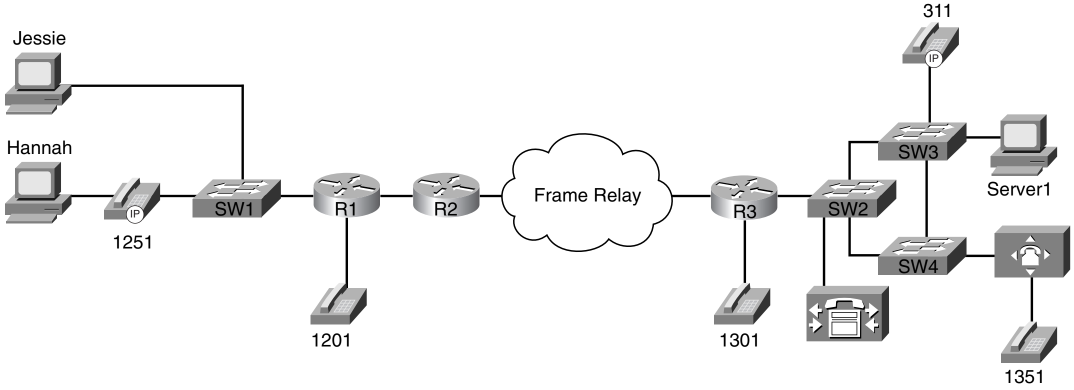
Classification and marking tools provide many options, but sometimes sorting out the best way to use the tools can be difficult. Classification and marking tools can classify based on a large number of frame and packet header fields. They can also mark a number of fields, the most notable being the IP Precedence and DSCP fields. You can use the classification and marking tools on all routers in the network, on many LAN switches, and even on IP Phones and host computers. This brief section discusses some of the classification and marking design choices.

The first step in making good classification and marking design choices is to choose where to mark. The general rule for choosing where to mark is as follows:

> Mark as close to the ingress edge of the network as is possible.

Figure 3-5 diagrams a typical enterprise IP network, which will be used to look more closely at the options for where to mark packets.

Figure 3-5 *Typical Enterprise Network*



Consider packets that flow left to right in Figure 3-5. Hannah and Jessie, both client PCs, can mark IP precedence, IP DSCP, and CoS if their Ethernet card supports ISL or 802.1Q. The IP Phone internally marks its own voice bearer traffic precedence 5, DSCP EF, and CoS 5 by default, its own voice signaling traffic precedence 3, DSCP 31, and CoS 3. The phone can also re-mark the CoS, precedence, and DSCP sent by Hannah's PC. (The phone default action is to re-mark 0 for all three values.) SW1, depending on the type of switch, might be able to re-mark CoS, re-mark precedence or DSCP, or make general (multifield) classification and marking decisions—in other words, it might be able to look at some of the fields listed earlier in Tables 3-2 and 3-3. Finally, R1 can use general multifield classification and marking before sending the packet over the WAN—but over the next link to R2, because the link is a PPP link, the only marking options would be in the IP header.

So marking can be done in many places near the ingress edge of the network—but whom do you trust? Classification and marking should not be performed before the frame/packet reaches a trusted device. This location in the network is called the *trust boundary*. For instance, Jessie formerly marked her packets with DSCP default, but because the user of the PC can change that value, Jessie changed to use DSCP EF to get better service. In most cases, the end-user PCs are beyond the trust boundary. IP Phones can reset CoS, precedence, and DSCP to 0 for Hannah's traffic, and mark the VoIP with CoS 5, precedence 5, and DSCP EF—with the added benefit that the phone user cannot reset those values. The IP Phone trust settings are controlled by the connected Cisco Catalyst switch, enabling the system administrator to trust markings received from the IP Phone while rewriting the values received from the attached PC.

The final consideration when deciding where to mark involves the function of the various devices, and personal preferences. For instance, IP Phones provide three classes—one for voice bearer traffic, one for voice signaling traffic, and one for all packets from the PC. However, a network may need multiple classes for data traffic, so further classification may be required by a switch or router. Some switches provide robust Layer 3 QoS classification and marking functions—in these cases, classification and marking may be performed on the switch; otherwise classification and marking must be performed on the router. Figure 3-6 outlines some of the strategies for classification and marking for three different LAN topologies.

Figure 3-6 shows three typical paths for frames between the end-user device and the first router. The first instance shows a typical installation near the end users—a switch that performs only Layer 2 QoS, and PCs connected to it. *Only Layer 2 QoS* just means that the switch can react to, or possibly set, CoS, but it cannot react to or mark IP precedence or DSCP. In this case, classification and marking is typically performed as packets enter R1's Ethernet interface. In addition, because SW1 can support CoS, but not precedence or DSCP, R1 may want to map incoming CoS values to the Precedence or DSCP fields.

The second part of Figure 3-6 shows a network with a Layer 3 QoS-capable switch. Depending on the type of switch, this switch may not be able to perform Layer 3 switching, but it does have the capability to react to or mark IP precedence or DSCP. In this case, you should classify and mark on the switch. Classification and marking on the Layer 3 switch allows classification and marking closer to the trust boundary of the network, and offers the added benefits of queuing, congestion avoidance, and policing based on the marked values. If only a few sites in the network have Layer 3 QoS-capable switches, you may prefer to perform classification and marking on the router, so all sites' configurations are similar. However, classifying and marking in the router places additional overhead on the router's CPU.

Finally, the third example shows a PC cabled through an IP Phone to a Layer 3 QoS-capable switch. The IP Phone can easily take care of classification and marking into two categories—voice and nonvoice. The switch and router can take advantage of those marked values. If more classes are needed for this network's QoS policy, SW3, or R3, can perform classification and marking. Of course, if the QoS policy for this network only requires the three classes—one for voice bearer traffic, one for voice signaling traffic, one for nonvoice—and all PCs are connected through the switch in the IP Phone, no classification and marking is needed on SW3 or R3!

Figure 3-6 *Three Classification and Marking Placement Strategies*

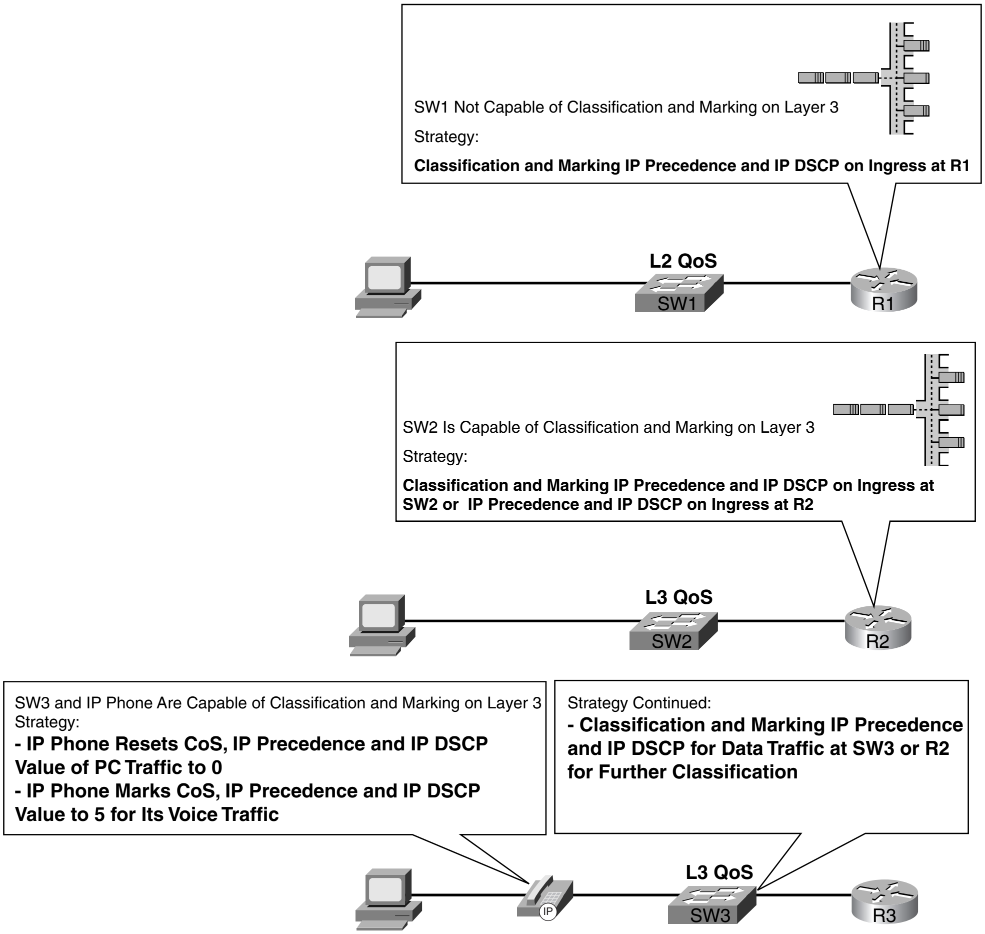
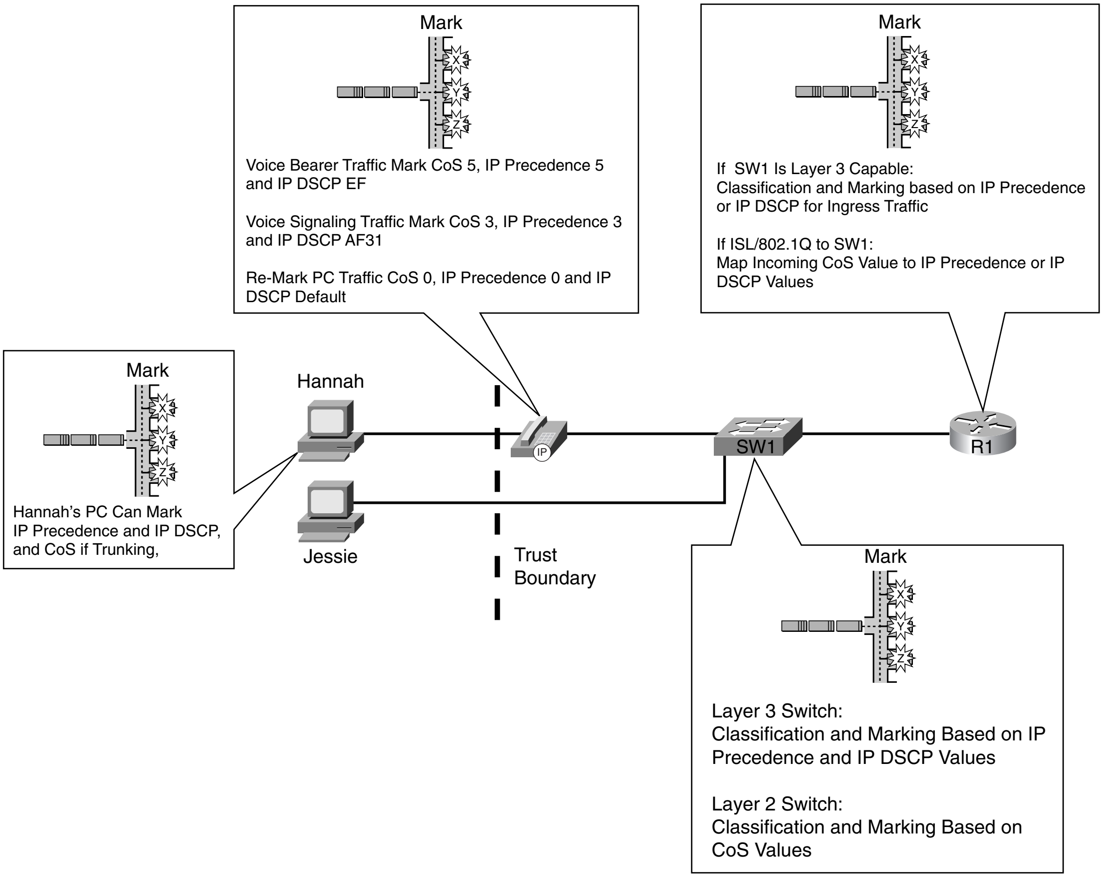

SW1 Not Capable of Classification and Marking on Layer 3
Strategy:
Classification and Marking IP Precedence and IP DSCP on Ingress at R1
L2 QoS
SW1
R1
SW2 Is Capable of Classification and Marking on Layer 3
Strategy:
Classification and Marking IP Precedence and IP DSCP on Ingress at SW2 or IP Precedence and IP DSCP on Ingress at R2
L3 QoS
SW2
R2
SW3 and IP Phone Are Capable of Classification and Marking on Layer 3
Strategy:
- IP Phone Resets CoS, IP Precedence and IP DSCP Value of PC Traffic to 0
- IP Phone Marks CoS, IP Precedence and IP DSCP Value to 5 for Its Voice Traffic
Strategy Continued:
- Classification and Marking IP Precedence and IP DSCP for Data Traffic at SW3 or R2 for Further Classification
IP
L3 QoS
SW3
R3

Figure 3-7 summarizes some of the design options for where to classify and mark, showing the remote site from Figure 3-5.

Figure 3-7 *Classification and Marking Options Applied to a Typical Enterprise Network*

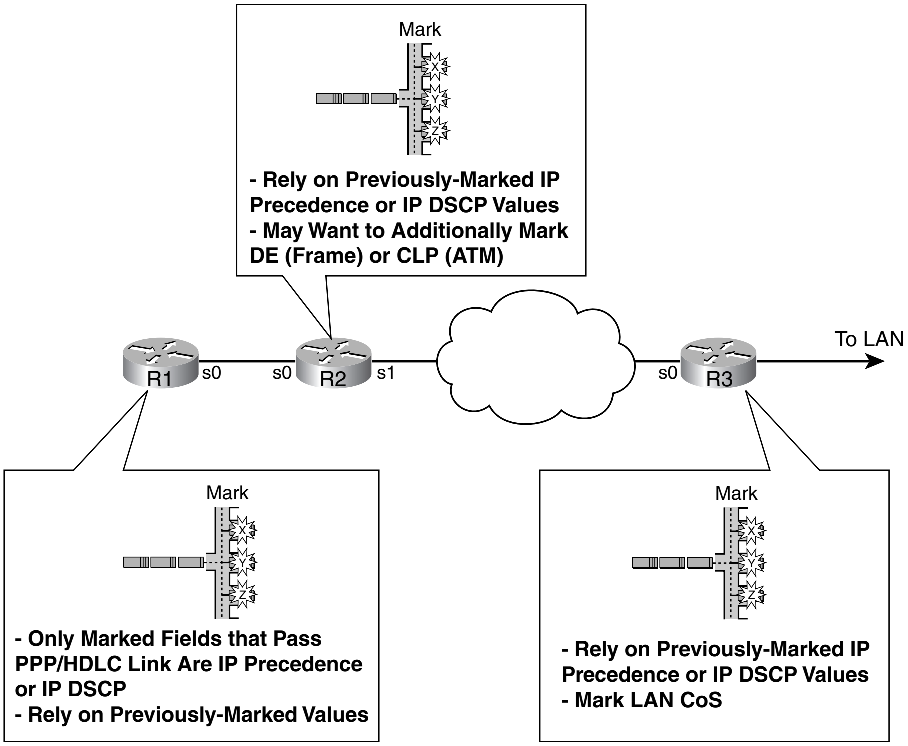

Mark
X
Y
Z
Voice Bearer Traffic Mark CoS 5, IP Precedence 5 and IP DSCP EF
Voice Signaling Traffic Mark CoS 3, IP Precedence 3 and IP DSCP AF31
Re-Mark PC Traffic CoS 0, IP Precedence 0 and IP DSCP Default
Mark
X
Y
Z
If SW1 Is Layer 3 Capable:
Classification and Marking based on IP Precedence or IP DSCP for Ingress Traffic
If ISL/802.1Q to SW1:
Map Incoming CoS Value to IP Precedence or IP DSCP Values
Mark
X
Y
Z
Hannah's PC Can Mark IP Precedence and IP DSCP, and CoS if Trunking,
Hannah
Jessie
IP
Trust Boundary
SW1
R1
Mark
X
Y
Z
Layer 3 Switch:
Classification and Marking Based on IP Precedence and IP DSCP Values
Layer 2 Switch:
Classification and Marking Based on CoS Values

The choices of where to perform classification and marking can be summarized as follows:

- Classify and mark as close to the ingress edge as possible.
- Consider the trust boundary in the network, making sure to mark or re-mark traffic after it reaches a trusted device in the network
- Because the two IP QoS marking fields—Precedence and DSCP—are carried end to end, mark one of these fields to maximize the benefits of reducing classification overhead by the other QoS tools enabled in the network.

Typically, when the packet makes it to the first WAN router, the initial marking has occurred. However, there may be other instances where marking should take place. Consider Figure 3-8, which shows several additional options for where marking can occur.

Figure 3-8 *Classification and Marking Options—Typical Enterprise WAN*


Mark
X
Y
Z
- Rely on Previously-Marked IP Precedence or IP DSCP Values
- May Want to Additionally Mark DE (Frame) or CLP (ATM)
To LAN
R1
s0
s0
R2
s1
s0
R3
Mark
X
Y
Z
- Only Marked Fields that Pass PPP/HDLC Link Are IP Precedence or IP DSCP
- Rely on Previously-Marked Values
Mark
X
Y
Z
- Rely on Previously-Marked IP Precedence or IP DSCP Values
- Mark LAN CoS

Most QoS tools can classify based on IP precedence and DSCP. However, the Frame Relay or ATM switches can also react to the DE and CLP bits, respectively. Therefore, you might want to set DE or CLP for the least-important traffic. If the LAN switches connected to R3 react to CoS settings, but not precedence or DSCP, which is typical of switches that only support Layer 2 QoS, you might want to mark the CoS bits on R3 before sending frames onto the Ethernet.

Finally, when you do mark CoS, IP precedence, and IP DSCP, what values should you use? Well, the "bigger is better" attitude is suggested for CoS and precedence, whereas the DiffServ PHB RFCs should be followed for DSCP settings. Cisco also suggests some specific values in cases where your policies allow for voice payload, video payload, voice/video signaling, and two classes of data. Table 3-6 lists these recommended values.

Table 3-6 *Cisco's Recommended Values for Marking*

Type of Traffic	CoS	Precedence	DSCP
Voice payload	5	5	EF
Video payload	4	4	AF41
Voice/Video signaling	3	3	AF31
High-priority or gold data classes*	2	2	AF21 AF22 AF23
Medium-priority or silver data*	1	1	AF11 AF12 AF13
All else	0	0	Default

* Note: The table lists the current recommendations as of early 2003. The DQOS course, and presumably the exam, was created much earlier, when the recommendation for high-priority data was to mark with AF21, with no recommendation for medium-priority data. Keep that in mind when answering exam questions. Also check www.cisco.com and www.ciscopress.com/1587200589 for more information when the exams do change!

In summary, classification and marking tools classify packets based on a large number of different fields inside data link and network layer headers. Based on the classification, the tools then mark a field in a frame or packet header, with the goal that other QoS tools can more easily classify and perform specific QoS actions based on these marked fields. Among all the fields that can be marked, IP Precedence and DSCP, because they are part of the IP header, are the only fields that can be marked and carried from end to end in the network.

Classification and Marking Tools

Three classification and marking tools provide the multifield classifier function of DiffServ—namely, class-Based marking (CB marking), committed access rate (CAR), and policy-based routing (PBR). All three tools enable you to configure matching parameters for a wide variety of fields in a packet header.

Class-Based Marking (CB Marking)

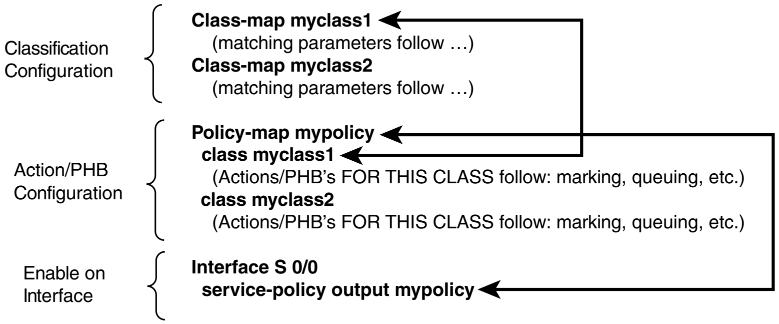
Cisco added CB marking to IOS after all the other classification and marking tools discussed in this book. As of IOS 12.1(5)T and 12.2 mainline, CB marking represents the only classification and marking tool specifically intended for the classification and marking function. NBAR, CAR, PBR, and dial peers have other purposes, whereas CB marking focuses entirely on classification and marking.

You must use a new IOS syntax called the *Modular QoS command-line interface* (MQC) to configure CB marking. After hearing of the term MQC for the first time, many people think that Cisco has created a totally new CLI, different from IOS configuration mode, to configure CB marking. In reality, MQC defines a new set of configuration commands—commands that are typed in using the same IOS CLI, in configuration mode. So, why bother to name these new commands with the term "MQC?" Well, newer IOS QoS tools, as well as future IOS QoS tools, and to some degree older QoS tools, will all use the same MQC commands for QoS configuration in the future. Instead of more than 30 IOS QoS tools, each with different configuration commands, the commands will slowly converge to use the MQC commands. Therefore, MQC is really more of a standard for new and revised configuration commands for QoS features.

All IOS QoS tools that begin with the phrase "class based" use the MQC commands as of IOS 12.2 mainline. These tools include CB marking, CB Weighted Fair Queuing (CBWFQ), CB policing, and CB shaping. Most QoS tools need to perform classification functions; all MQC supporting tools use the same commands for classification. The person configuring the router only needs to learn one set of commands for classification for all four of these tools, which reduces effort and reduces mistakes.

MQC separates the classification, per-hop behavior (PHB), and enabling functions into three separate commands. The **class-map** command defines the matching parameters for classifying packets. Because different tools create different PHBs, the PHB actions (marking, queuing, and so on) are configured under a **policy-map** command. Finally, because these tools operate on packets that either enter or exit an interface, the policy is then enabled on an interface using a **service-policy** command. Figure 3-9 shows the general flow of commands.

Figure 3-9 *MQC Commands and Their Correlation*



```
Class-map myclass1
  (matching parameters follow …)
Class-map myclass2
  (matching parameters follow …)

Policy-map mypolicy
 class myclass1
  (Actions/PHB's FOR THIS CLASS follow: marking, queuing, etc.)
 class myclass2
  (Actions/PHB's FOR THIS CLASS follow: marking, queuing, etc.)

Interface S 0/0
 service-policy output mypolicy
```

In this example, the network's QoS policy calls for two classes of packets. (The actual types of packets that are placed into each class are not shown, just to keep the focus on the general flow of how the main commands work together.) To classify packets into two classes, two **class-map** commands are used. Each **class-map** would be followed by a **match** subcommand, which defines the actual parameters compared to packet header contents to match packets for classification. For each class, some QoS action needs to be applied—but configuration for these actions is made under the **policy-map** command. Under a single policy map, multiple classes will be referenced—two classes in this example, myclass1 and myclass2. Inside the single policy called mypolicy, under each of the two classes myclass1 and myclass2, you can configure separate QoS actions. For instance, you could apply different marking to packets in class myclass1 and myclass2 at this point. Finally, when the **service-policy** command is applied to an interface, the QoS features are enabled.

MQC provides some good advantages when compared to building each QoS tool with different sets of configuration commands. In many cases, you will use multiple policies in one router, but you need the same classifications. For instance, you might apply slightly different queuing parameters to five different serial links, but because packets have already been marked near the ingress edge of the network, all the classification logic is the same in each case. Therefore, all five policy maps could refer to the same class maps for classification purposes. With multiple tools sharing the same commands, QoS configuration becomes less confusing. Learning how to configure a new MQC QoS tool will be easy as well—for instance, when you know how to configure CB marking, you only need to learn one more command to learn how to configure CBWFQ!

Several specific examples appear over the next several pages. First, Table 3-7 lists the MQC commands used for CB marking. The table shows all the classification options available using the **match** command, and all the marking options available using the **set** command. Table 3-8 lists the **show** commands related to CB marking.

Table 3-7 *Command Reference for CB Marking*

Command	Mode and Function
class-map *class-map-name*	Global config; names a class map, where classification options are configured
Match …	Class-map subcommand; defines specific classification parameters
match access-group {*access-group* | **name** *access-group-name*}	Class-map subcommand; matches an ACL
match source-address mac *address-destination*	Class-map subcommand; matches a source MAC address
match ip precedence *ip-precedence-value* [*ip-precedence-value ip-precedence-value ip-precedence-value*]	Class-map subcommand; Matches an IP precedence value

continues

Table 3-7 *Command Reference for CB Marking (Continued)*

Command	Mode and Function
match mpls experimental *number*	Class-map subcommand; matches an MPLS Experimental value
match cos *cos-value* [*cos-value cos-value cos-value*]	Class-map subcommand; matches a CoS value
match destination-address mac *address*	Class-map subcommand; matches a destination MAC address
match input-interface *interface-name*	Class-map subcommand; matches an input interface
match ip dscp *ip-dscp-value* [*ip-dscp-value ip-dscp-value ip-dscp-value ip-dscp-value ip-dscp-value ip-dscp-value ip-dscp-value*]	Class-map subcommand; matches an IP DSCP value
match ip rtp *starting-port-number port-range*	Class-map subcommand; matches the RTP's UDP port-number range
match qos-group *qos-group-value*	Class-map subcommand; matches a QoS group
match protocol *protocol-name*	Class-map subcommand; matches NBAR protocol types
match protocol citrix app *application-name-string*	Class-map subcommand; matches NBAR Citrix applications
match protocol http [**url** *url-string* | **host** *hostname-string* | **mime** *MIME-type*]	Class-map subcommand; matches a host name and URL string
match any	Class-map subcommand; matches all packets
policy-map *policy-map-name*	Global config; names a policy, which is a set of actions to perform
class *class-name*	Policy-map subcommand; identifies which packets on which to perform some action by referring to the classification logic in a class map
set	Class subcommand; for the class, marks (sets) particular QoS fields
set ip precedence *ip-precedence-value*	Class subcommand; set the value for IP precedence
set ip dscp *ip-dscp-value*	Class subcommand; set the value for IP DSCP

Table 3-7 *Command Reference for CB Marking (Continued)*

Command	Mode and Function
set cos *cos-value*	Class subcommand; set the value for CoS
set ip qos-group *group-id*	Class subcommand; set the value for the QoS group
set atm-clp	Class subcommand; set the value for the ATM CLP bit
set fr-de	Class subcommand; set the value for the Frame Relay DE bit

Table 3-8 *Exec Command Reference for CB Marking*

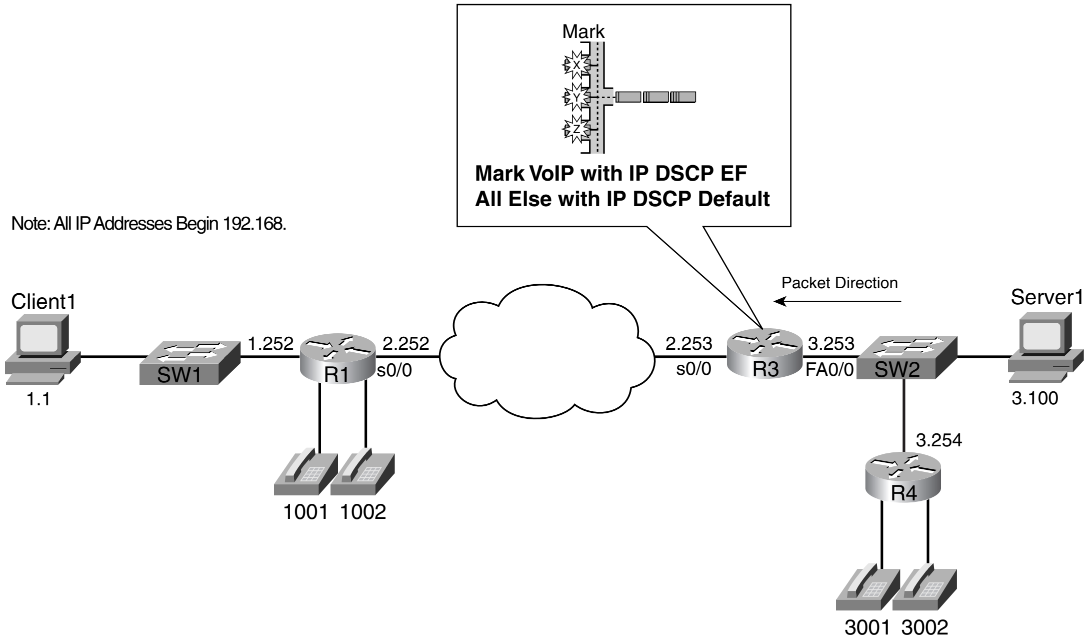
Command	Function
show policy-map *policy-map-name*	Lists configuration information about all MQC-based QoS tools
show policy-map *interface-spec* [***input*** | ***output***] [**class** *class-name*]	Lists statistical information about the behavior of all MQC-based QoS tools

QoS configuration should follow the process of planning the QoS policies for the network. After those policies have been defined, and the location of where to perform the marking functions has been determined, however, the CB marking configuration that follows becomes an exercise in deciding how to match or classify the packets, and how to configure the commands correctly. In the first MQC configuration example, for example, the policy has been defined as follows:

- All VoIP traffic should be marked with DSCP EF.
- All other traffic should be marked with DSCP Default.

Figure 3-10 is used for many example configurations in this book. In the first example, marking is performed for packets entering R3's FA0/0 interface. In reality, it also makes sense to mark packets near R1 for packet flows from left to right in the figure. To keep the configurations less cluttered, however, only one direction, right to left, is shown. Example 3-1 lists the configuration for this first example.

Figure 3-10 *CB Marking Sample Configuration 1*



Example 3-1 *CB Marking: Sample 1 Configuration*

```
R3#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)#ip cef
R3(config)#!
R3(config)#class-map voip-rtp
R3(config-cmap)#match ip rtp 16384 16383
R3(config-cmap)#policy-map voip-and-be
R3(config-pmap)# class voip-rtp
R3(config-pmap-c)#  set ip DSCP EF
R3(config-pmap-c)# class class-default
R3(config-pmap-c)#   set ip dscp default
R3(config-pmap-c)#interface e 0/0
R3(config-if)# service-policy input voip-and-be
R3(config-if)#end
R3#
R3#show running-config
```

Example 3-1 *CB Marking: Sample 1 Configuration (Continued)*

```
Building configuration...
!Portions removed to save space…
ip cef
!
class-map match-all voip-rtp
  match ip rtp 16384 16383
!
!
policy-map voip-and-be
  class voip-rtp
   set ip dscp 46
  class class-default
   set ip dscp 0
!
interface Fastethernet0/0
 description connected to SW2, where Server1 is connected
 ip address 192.168.3.253 255.255.255.0
 service-policy input voip-and-be
```

First, focus on the command prompts in Example 3-1. Note that the **class-map** command moves the CLI into class-map configuration mode, with the prompt R3(config-cmap). The **policy-map** command moves the CLI into policy-map configuration mode, and the **class** command that follows (not **class-map**, but just **class**) moves the CLI into an additional subconfiguration mode that has no specific name.

NOTE I tend to call configuration mode you are in after using the **policy-map** command, and then the **class** command, the *policy-map class* mode when teaching QoS classes.

Next, examine the **match** commands. The solution could have referred to IP ACL 101 with the **match ip access-group 101** command, with ACL 101 matching UDP ports between 16,384 and 32,767, inclusive, to match all VoIP traffic. However, the **match ip rtp** command matches only the even-numbered ports in this same UDP port range. (VoIP payload only uses the even port numbers.) Therefore, the **match ip rtp** command is more efficient for matching VoIP, easier to configure, and only matches the VoIP payload. The other **match** command, **match any**, does exactly that: It matches anything.

Class maps allow multiple **match** commands in a single **class-map** command. You may have noticed the **match-all** parameter on the **class-map** output from **show run**; IOS added the **match-all** parameter, even though it was not typed in. If a **class-map** command has multiple **match** commands, with the default setting of **match-all**, all the **match** commands must match a packet before the packet is considered to be part of the class. The other alternative is to configure the keyword **match-any**, which means that if one or more of the **match** commands in a single class map matches a packet, the packet is part of that class.

Continuing down the configuration, examine the **policy-map set** commands. The first command sets a DSCP of EF for all traffic that matches **class-map voip-rtp**. The other **set** command, which follows the **class class-default** command, sets DSCP of Default for traffic that matches the **class-default class-map**. In other words, the policy map sets DSCP EF for packets that match one class, and DSCP Default, using the keyword **default**, for the other class. IOS includes a class that matches all remaining traffic, called **class-default**, in every **policy-map**. Although the command **class class-default** was not specified in the configuration, note that the **class class-default** command is automatically added to the end of a policy map to match all unspecified traffic. **class-default** was used in the **policy-map** to match all remaining traffic and then mark that traffic as BE.

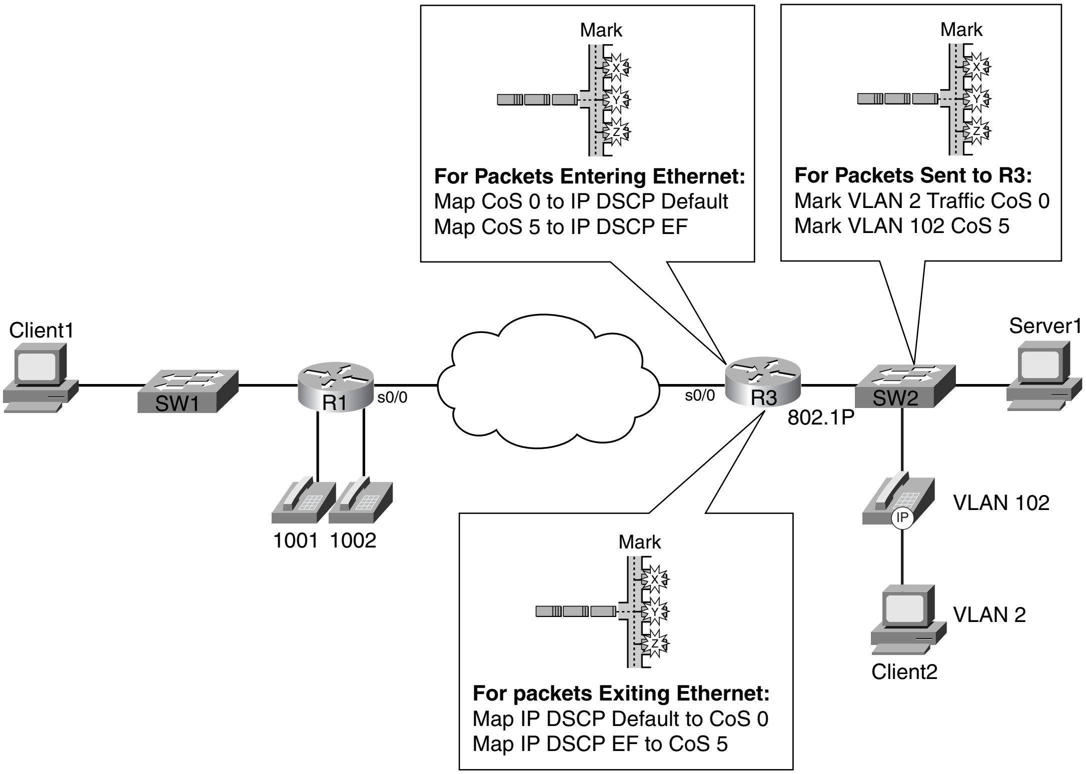
Finally, the **service-policy** command enables CB marking for ingress packets with the **service-policy input voip-and-be** interface subcommand. When enabled, IOS applies the policy map classes in the order they appear in the **policy-map** command. In this example, for instance, the VoIP-RTP class is used to examine the packet first; if a match appears, the packet is marked with DSCP EF. After the packet has been matched and marked, it exits the policy map. If no match occurs, only then is the next class, class-default, used to examine the packet.

Consider two caveats before moving on to more examples. First, examine the output of the **show run** command in Example 3-1, and look for the **set** commands. The MQC enables you to type in the text names of the DSCP values, but IOS records the configuration using the decimal version of the DSCP value! Therefore, you may need a handy reference for the actual DSCP values, as is shown in Table 3-4.

The other caveat only occurs if you already know how to configure Frame Relay traffic shaping (FRTS). FRTS uses a command called **map-class**. Many people who know how to configure **map-class** first look at any MQC-based tool, see the **class-map** command, and don't realize that the commands are indeed two totally different commands. So, ignore all you have learned about FRTS until you learn about MQC configurations using the **class-map** command.

The next example is a CB marking configuration that uses the same network as the one used in Example 3-1. R3 is performing the CB marking function again, but this time R3 expects that SW2 has already set CoS bits to either 0 or 5. The engineers in the meeting to discuss QoS policies for this network decided that SW2 would mark CoS with either 0 or 5, and then R3 would map CoS 0 to DSCP Default, and CoS 5 to DSCP EF. For packets moving left to right, R3 should map DSCP Default back to CoS 0, and DSCP EF back to CoS 5. Figure 3-11 depicts the network and QoS policies, and Example 3-2 lists the configuration.

Figure 3-11 *CB Marking Sample Configuration 2*



For Packets Entering Ethernet:
Map CoS 0 to IP DSCP Default
Map CoS 5 to IP DSCP EF

For Packets Sent to R3:
Mark VLAN 2 Traffic CoS 0
Mark VLAN 102 CoS 5

For packets Exiting Ethernet:
Map IP DSCP Default to CoS 0
Map IP DSCP EF to CoS 5

Example 3-2 *CB Marking: Sample 2 Configuration*

```
class-map cos0
 match cos 0
!
class-map cos5
 match cos 5
!
class-map BE
 match ip dscp default
!
class-map EF
 match ip dscp EF
!
policy-map map-cos-to-dscp
 class cos0
  set ip DSCP default
```

continues

Example 3-2 *CB Marking: Sample 2 Configuration (Continued)*

```
 class cos5
  set ip DSCP EF
 class class-default
   set ip dscp default
!
policy-map map-dscp-to-cos
 class BE
  set cos 0
 class EF
  set cos 5
 class class-default
   set cos 0
!
interface FastEthernet0/0
!
interface FastEthernet0/0.1
 encapsulation dot1Q 102
 service-policy input map-cos-to-dscp
 service-policy output map-dscp-to-cos
!
interface FastEthernet0/0.2
 encapsulation dot1Q 2 native
```

As you learned earlier in this chapter, to mark and classify CoS values, a VLAN trunking header must exist on the packet. On R3 in this example, subinterface Fast Ethernet 0/0.1 and subinterface Fast Ethernet 0/0.2 have been created and assigned to the voice VLAN 102 and the data VLAN 2, respectively, using 802.1Q trunking. This configuration creates an 802.1Q header for traffic in the voice VLAN 102, without creating a VLAN header for the data VLAN 2 traffic.

The QoS policy required two policy maps in this example. Policy map map-cos-to-dscp matched CoS values for frames entering R3's FA 0/0.1 interface, and marked DSCP values, for packets flowing right to left in the figure. Therefore, the policy map was enabled on input of R3's FA 0/0.1 interface. Policy map map-dscp-to-cos matched DSCP values on packets exiting R3's FA 0/0.1 interface, and marked CoS, for packets flowing left to right in the figure. Therefore, the policy map was enabled on output of R3's FA 0/0.1 interface. Neither policy map could be applied on the WAN interface, because only interfaces configured for 802.1Q accept **service-policy** commands that reference policy maps that either classify or mark based on CoS.

Note that you cannot enable a **policy-map** that refers to CoS on interface FA0/0.2 in this example. That subinterface is in the native VLAN, meaning that no 802.1Q header is used. In a real network, you would probably want to enable a **policy-map** on the subinterface in order to mark traffic, but it must classify based on something beside CoS.

Network-Based Application Recognition (NBAR)

CB marking, and other MQC-based tools, can use NBAR to help classify traffic. By using the **match protocol** class-map subcommand, MQC can match protocols recognized by NBAR. This section describes NBAR, and includes examples of CB marking with NBAR.

NBAR classifies packets that are normally difficult to classify. For instance, some applications use dynamic port numbers, so a statically configured **match** command, looking for a particular UDP or TCP port number, just could not classify the traffic. NBAR can look past the UDP and TCP header, looking at the host name, URL, or MIME type in HTTP requests. NBAR can also look past the TCP and UDP headers to recognize application-specific information. For instance, NBAR allow recognition of different Citrix application types, and allows for searching for a portion of a URL string.

NBAR uses the classification information for two purposes. NBAR, without the help of other IOS features, can classify these difficult-to-classify protocols for the purpose of gathering statistics about the protocols. In fact, NBAR by itself provides classification and statistics, but no marking. NBAR also provides classification help for other QoS tools. Specifically, all MQC tools can refer to NBAR classifications for matching traffic.

The connection between NBAR and CB marking, or any other MQC tool, is through the **match protocol** class-map subcommand. An MQC tool can include the **match protocol** command under a **class-map** command. To do so, NBAR must be enabled on the same interface on which the class map is indirectly enabled through the **service-policy** interface subcommand.

A sample configuration and statistical display may help you make sense of NBAR. Tables 3-9 and 3-10 list the NBAR configuration and exec commands, respectively. Following the tables, Figure 3-12 diagrams the familiar network, where R3 performs CB marking based on NBAR classification of the URL string. Finally, Example 3-3 lists a sample NBAR and CB marking configuration, where CB marking matches a portion of an HTTP URL. The example includes a listing of NBAR statistics gathered on the interface.

Table 3-9 *Configuration Command Reference for NBAR*

Command	Mode and Function
ip nbar protocol-discovery	Interface mode; enables NBAR for traffic entering the interface.
ip nbar port-map *protocol-name* [**tcp** | **udp**] *port-number*	Global; tells NBAR to search for a protocol using a different port number than the well-known port. Also defines ports to be used by custom packet description language modules (PDLMs).
ip nbar pdlm *pdlm-name*	Global; extends the list of protocols recognized by NBAR by adding additional PDLMs.

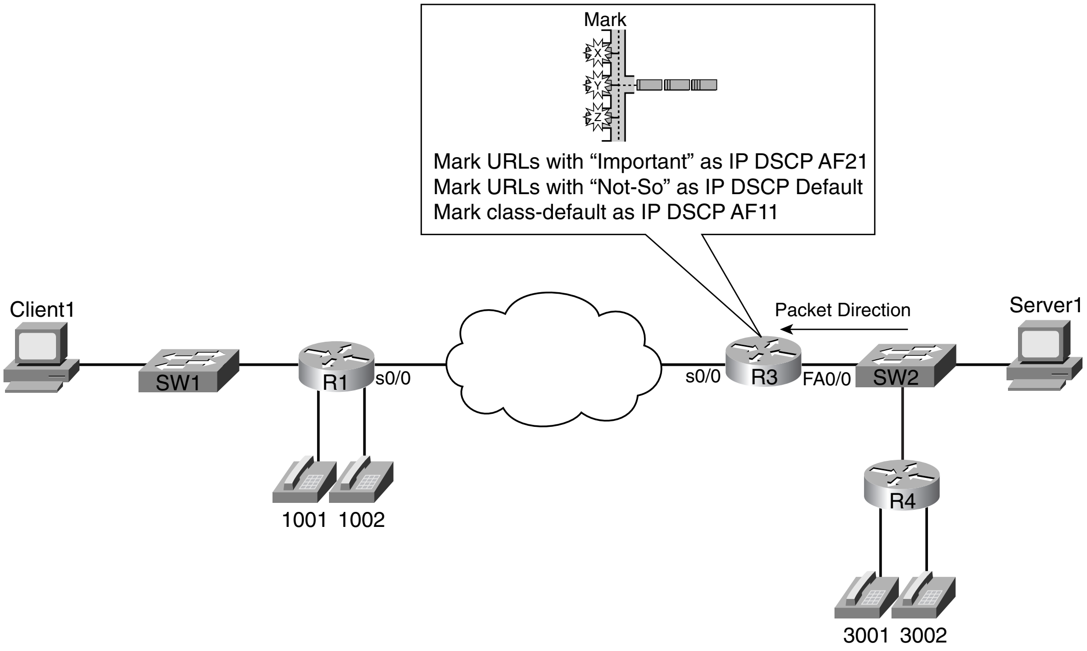
NOTE You can download additional PDLMs from Cisco.com:

www.cisco.com/cgi-bin/tablebuild.pl/pdlm

Table 3-10 *Exec Command Reference for NBAR*

Command	Function
show ip nbar protocol-discovery [**interface** *interface-spec*] [**stats** {**byte-count** | **bit-rate** | **packet-count**}][{**protocol** *protocol-name* | **top-n** *number*}]	Lists information about statistics for the discovered protocols. Statistics can be listed by interface, by protocol, or for just the top *n* protocols by volume.
show ip nbar port-map [*protocol-name*]	Lists the current ports in use by the discovered protocols.

Figure 3-12 *CB Marking Sample Configuration 3*


Mark
X
Y
Z
Mark URLs with "Important" as IP DSCP AF21
Mark URLs with "Not-So" as IP DSCP Default
Mark class-default as IP DSCP AF11
Packet Direction
Client1
SW1
R1
s0/0
1001 1002
s0/0
R3
FA0/0
SW2
Server1
R4
3001 3002

Example 3-3 uses the following criteria for marking packets:

- Any HTTP traffic whose URL contains the string "important" anywhere in the URL is marked with AF21.
- Any HTTP traffic whose URL contains the string "not-so" anywhere in the URL is marked with DSCP default.
- All other traffic is marked with AF11.

Example 3-3 shows the configuration.

Example 3-3 *Sample 3: CB Marking Based on URLs, Using NBAR for Classification*

```
ip cef
!
class-map http-impo
 match protocol http url "*important*"
!
class-map http-not
 match protocol http url "*not-so*"
!
class-map all-else
 match any
!
policy-map http
 class http-impo
  set ip dscp AF21
!
 class http-not
  set ip dscp default
!
 class class-default
  set ip DSCP AF11
!
interface fastethernet 0/0
 ip nbar protocol-discovery
 service-policy input http
!
!

R3# show ip nbar protocol-discovery top-n 5

 FastEthernet0/0
                            Input                    Output
   Protocol                 Packet Count             Packet Count
                            Byte Count               Byte Count
                            5 minute bit rate (bps)  5 minute bit rate (bps)
   ------------------------ ------------------------ ------------------------
   eigrp                    76                       0
                            5624                     0
                            0                        0
   bgp                      0                        0
                            0                        0
                            0                        0
   citrix                   0                        0
                            0                        0
                            0                        0
   cuseeme                  0                        0
                            0                        0
                            0                        0
```

continues

Example 3-3 *Sample 3: CB Marking Based on URLs, Using NBAR for Classification (Continued)*

```
   custom-01                0                        0
                            0                        0
                            0                        0
   unknown                  5610                     0
                            5665471                  0
                            135000                   0
   Total                    5851                     0
                            5845277                  0
                            135000                   0

R3#show ip nbar protocol-discovery interface fastethernet 0/0 stats packet-count top-n 5

 FastEthernet0/0
                            Input                    Output
   Protocol                 Packet Count             Packet Count
   ------------------------ ------------------------ ------------------------
   http                     721                      428
   eigrp                    635                      0
   netbios                  199                      0
   icmp                     1                        1
   bgp                      0                        0
   unknown                  46058                    63
   Total                    47614                    492
```

Notice that the class map configuration does not specifically use the term NBAR. Two class maps, http-impo and http-not, use the **match** command, with the **protocol** keyword, which implies that the actual classification uses NBAR. NBAR has been enabled on FA0/0 with the **ip nbar protocol discovery** command—had NBAR not been enabled, the **service-policy** command would have been rejected. Also note that CEF forwarding must be enabled, using the **ip cef** global command, before NBAR will work.

NBAR can match URLs exactly, or with some wildcards. You can use the asterisk (*) to match any characters of any length. In this case, as long as the phrases "important" or "not-so" appear in the URL, the packets are matched by one of the two class maps, respectively. Interestingly, when downloading an object with HTTP, the URL does not flow in every packet. *When classifying based on URL, NBAR matches all packets beginning with the matched URL, and then until another HTTP request for another URL flows inside the same TCP connection.*

The **show ip nbar protocol-discovery** command lists statistics for NBAR-classified packets. However, just using that command in live networks does not help much, because it lists three lines of output per type of protocol that can be discovered by NBAR—not just the protocols NBAR actually discovered. Therefore, the optional parameters on the command are more useful. For instance, both commands shown in the preceding example use the **top-n** parameter to limit the output based on the highest-volume protocols. The **show** command can also limit the statistics for a single interface, or it can limit the statistics to just packet count, or byte count, or bit rate.

Unlike most other IOS features, you can upgrade NBAR without changing to a later IOS version. Cisco uses a feature called packet descriptor language modules (PDLMs) to define new protocols that NBAR should match. When Cisco decides to add one or more new protocols to the list of protocols that NBAR should recognize, it creates and compiles a PDLM. You can then download the PDLM from Cisco, copy it into Flash memory, and add the **ip nbar pdlm** *pdlm-name* command to the configuration, where *pdlm-name* is the name of the PDLM file in Flash memory. NBAR can then classify based on the protocol information from the new PDLM.

CB Marking **show** Commands

CB marking provides only one **show** command that provides statistical information: **show policy-map interface**. The statistics do provide some good insight to the packet volumes being marked by CB marking. The next sample configuration includes a new configuration and several variations of the **show policy-map** command.

The same network is used for the next example as was used in the other CB marking examples, but with different marking criteria. In this case, traffic is generated so that the **show** command output is more meaningful. The following traffic is generated:

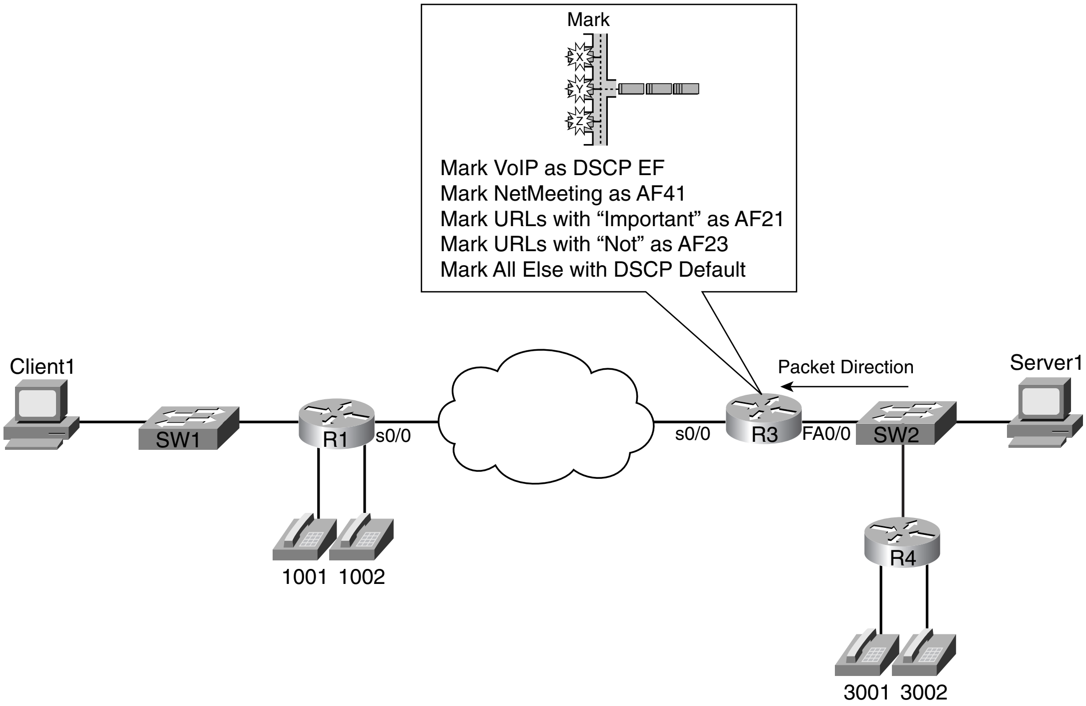
- Two G.711 VoIP calls between R4 and R1 using Foreign Exchange Station (FXS) cards on these two routers. Voice Activation Detection (VAD) is disabled.
- One FTP connection from the client PC to the server, with an FTP get of a 40-MB file called big.zip.
- One Microsoft NetMeeting video/audio conference between the client and server.
- One web page download from the server to the client. The web page has a few small objects. The web page includes two panes, each with a different JPG file: one called important.jpg; the other called not-so.jpg. The JPGs are exact copies of each other, and each JPG is 687 KB. In later examples, the differing performance of the download of these examples is used to demonstrate the behavior of other QoS tools.

Figure 3-13 depicts the same familiar network, and lists the criteria in with the figure for easy reference.

The new criteria for Example 3-4 is as follows:

- VoIP payload is marked with DSCP EF.
- NetMeeting voice and video from Server 1 to Client 1 is marked with DSCP AF41.
- Any HTTP traffic whose URL contains the string “important” anywhere in the URL is marked with AF21.
- Any HTTP traffic whose URL contains the string “not-so” anywhere in the URL is marked with AF23.
- All other traffic is marked with DSCP Default.

Figure 3-13 *Three Classification and Marking Placement Strategies*



Mark VoIP as DSCP EF
Mark NetMeeting as AF41
Mark URLs with "Important" as AF21
Mark URLs with "Not" as AF23
Mark All Else with DSCP Default

Example 3-4 shows the configuration, including the appropriate **show** commands.

Example 3-4 *CB Marking Sample 4, with* **show** *Command output*

```
ip cef
!
interface fastethernet 0/0
 ip nbar protocol-discovery
!
access-list 101 permit udp host 192.168.3.101 gt 16383 192.168.1.0 0.0.0.255 gt 16383
!
class-map voip-rtp
 match ip rtp 16384 16383
!
class-map http-impo
 match protocol http url "*important*"
!
class-map http-not
 match protocol http url "*not-so*"
!
```

Example 3-4 *CB Marking Sample 4, with* **show** *Command output (Continued)*

```
class-map NetMeet
 match access-group 101
!
policy-map laundry-list
!
 class voip-rtp
  set ip dscp EF
!
 class NetMeet
  set ip dscp AF41
!
class http-impo
  set ip dscp AF21
!
 class http-not
  set ip dscp AF23
!
 class class-default
  set ip DSCP default
!
 interface Fastethernet 0/0
 service-policy input laundry-list
end

R3#show policy-map
  Policy Map laundry-list
    Class voip-rtp
      set ip dscp 46
    Class NetMeet
      set ip dscp 34
    Class http-impo
      set ip dscp 18
    Class http-not
      set ip dscp 22
    Class class-default
      set ip dscp 0

R3#show policy-map laundry-list
  Policy Map laundry-list
    Class voip-rtp
      set ip dscp 46
    Class NetMeet
      set ip dscp 34
    Class http-impo
      set ip dscp 18
    Class http-not
      set ip dscp 22
```

continues

Example 3-4 *CB Marking Sample 4, with* **show** *Command output (Continued)*

```
    Class class-default
      set ip dscp 0
R3#show policy-map interface fastethernet 0/0 input
 Fastethernet0/0

  Service-policy input: laundry-list

    Class-map: voip-rtp (match-all)
      35268 packets, 2609832 bytes
      5 minute offered rate 59000 bps, drop rate 0 bps
      Match: ip rtp 16384 16383
      QoS Set
        ip dscp 46
          Packets marked 35268

    Class-map: NetMeet (match-all)
      817 packets, 328768 bytes
      5 minute offered rate 19000 bps, drop rate 0 bps
      Match: access-group 101
      QoS Set
        ip dscp 34
          Packets marked 817

    Class-map: http-impo (match-all)
      2843 packets, 3462611 bytes
      5 minute offered rate 56000 bps, drop rate 0 bps
      Match: protocol http url "*important*"
      QoS Set
        ip dscp 18
          Packets marked 2855

    Class-map: http-not (match-all)
      2828 packets, 3445409 bytes
      5 minute offered rate 56000 bps, drop rate 0 bps
      Match: protocol http url "*not-so*"
      QoS Set
        ip dscp 22
          Packets marked 2842

    Class-map: class-default (match-all)
      33216 packets, 43649458 bytes
      5 minute offered rate 747000 bps, drop rate 0 bps
      Match: any
      QoS Set
        ip dscp 0
          Packets marked 33301
```

Review the configuration before taking a closer look at the **show** commands. The only part of the configuration that was not covered in the first three examples on CB marking is the matching of the Microsoft NetMeeting traffic. NetMeeting uses RTP for the audio and video flows. ACL

101 matches all UDP port numbers over 16,384, for traffic from Server 1 going to the client. This may catch other traffic besides NetMeeting, but it definitely catches all the NetMeeting traffic. Also note that the NetMeet class map uses a combination of capital letters and lowercase letters, as does the **class** command that refers to it. Class map names are case sensitive—you may want to choose to use only uppercase letters for names to avoid confusion.

The **show policy-map laundry-list** command just lists a summary of the configuration. You can gather the same information with a **show running-config** command, but it is summarized nicely with **show policy-map**. The **show policy-map** command lists the same configuration information, but it lists the information for all the configured policy maps in this router.

The **show policy-map** command using the **interface** option provides statistical information about the number of packets and bytes that have matched each class inside the policy maps. Because CB marking is configured, it also notes the number of packets that have been marked. You can select all interfaces, just one interface, either input or output, and even select a single class inside a single policy map for display.

Finally, the **load-interval** interface subcommand can also be useful when looking at any QoS tool's statistics. The **load-interval** command defines the time interval over which IOS measures packet and bit rates on an interface. With a lower load interval, the statistics change more quickly; with a larger load interval, the statistics change more slowly. In a lab when you are just learning to use QoS tools, set the load interval to the minimum of 30 seconds, so you can see the results of new traffic, or changes to the configuration, quickly. (The default setting is 5 minutes.)

CB Marking Summary

Class-based marking provides the most functional general classification and marking tool in IOS, as of IOS 12.2 mainline. Class-based marking provides the largest number of fields for classifying packets, and the largest number of fields that can be marked. It uses MQC for configuration, separating the classification details from the QoS action.

Refer to Table 3-17, in the "Foundation Summary" section of this chapter, for a complete list of classification and marking fields used by CB marking.

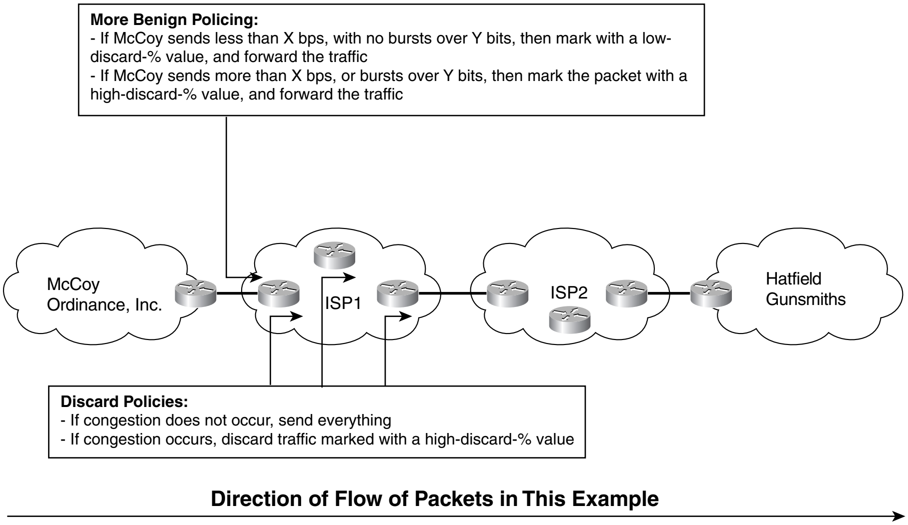
Committed Access Rate (CAR)

CAR provides policing functions and marking. Chapter 5, "Traffic Policing and Shaping," covers the policing details of CAR and CB policing. However, a quick review of policing before getting into CAR's marking features will help you appreciate why CAR includes marking.

Policing, in its most basic form, discards traffic that exceeds a particular traffic contract. The contract has two components: a rate, stated either in bits per second or bytes per second; and a burst size, stated in either bits or bytes. The traffic conforms to the contract if it sends at the rate, or below, and it does not send a burst of traffic greater than the burst size. If the traffic exceeds the traffic rate over time, or exceeds the single burst size limit, the policing function drops the traffic in excess of the rate and the burst size. Therefore, the simplest form of policing has two rigid actions: either to forward packets or to drop them.

CAR's marking function allows for additional policing action besides just forwarding or dropping a packet. Consider a typical case where policing is used, as in Figure 3-14. ISP1 needs to police traffic to protect customers who conform to their contracts from congestion created by customers who do not conform. If the network is not congested, however, it might be nice to go ahead and forward the nonconforming customer traffic. Doing so doesn't really cost the ISP anything, so long as the network is not congested. If the network is congested, however, ISP1 wants to discard the traffic that exceeds the contract before discarding traffic that is within its respective contract.

Figure 3-14 *Policing: Excess Traffic Marked with Higher Discard Value*


More Benign Policing:
- If McCoy sends less than X bps, with no bursts over Y bits, then mark with a low-discard-% value, and forward the traffic
- If McCoy sends more than X bps, or bursts over Y bits, then mark the packet with a high-discard-% value, and forward the traffic

Discard Policies:
- If congestion does not occur, send everything
- If congestion occurs, discard traffic marked with a high-discard-% value

For instance, the conforming traffic can be marked with DSCP AF41, and the nonconforming traffic with DSCP Default. The congestion-avoidance QoS tools in ISP1 can be configured to aggressively discard all DSCP Default traffic at the first signs of congestion. So, when ISP1 experiences congestion, policing indirectly causes the excess traffic to be discarded; in periods of no congestion, ISP1 provides service beyond what the customer has paid for.

You can also use CAR to just mark the traffic. CAR classifies traffic based on a large number of fields in the packet header, including anything that can be matched with an IP ACL. Once matched, CAR can be configured to do one action for conforming traffic, and another for excess traffic. If the two actions (conform and exceed actions) are the same action, in effect, CAR has not policed, but rather has just marked packets in the same way.

CAR configuration includes the classification, marking, and enabling features all in a single configuration command: the **rate-limit** interface subcommand. (CB marking, you may recall, separates classification, marking, and enabling on an interface into three separate commands.) Tables 3-11, 3-12, and 3-13 list the pertinent CAR configuration and exec commands, respectively.

Table 3-11 *Configuration Command Reference for CAR*

Command	Mode and Function
rate-limit {**input** | **output**} [**access-group** [**rate-limit**] *acl-index*] *bps burst-normal burst-max* **conform-action** *conform-action* **exceed-action** *exceed-action*	Interface mode; configures classification, marking, policing, and enabling CAR on the interface
access-list rate-limit *acl-index* {*precedence* | *mac-address* | *exp* **mask** *mask*}	Global mode; creates a CAR ACL, which can match IP precedence, MAC addresses, and MPLS Experimental bits

Table 3-12 *Possible Actions with CAR* **rate-limit** *Command*

rate-limit Conform and Exceed Options	Function
Continue	Evaluates the next **rate-limit** command
Drop	Drops the packet
set-dscp-continue	Sets the differentiated services code point (DSCP) (0–63) and evaluates the next **rate-limit** command
set-dscp-transmit	Sets the DSCP and transmits the packet
set-mpls-exp-continue	Sets the MPLS Experimental bits (0–7) and evaluates the next **rate-limit** command
set-mpls-exp-transmit	Sets the MPLS Experimental bits (0–7) and sends the packet
set-prec-continue	Sets the IP precedence (0–7) and evaluates the next **rate-limit** command
set-prec-transmit	Sets the IP precedence (0–7) and sends the packet
set-qos-continue	Sets the QoS group ID (1–99) and evaluates the next **rate-limit** command
set-qos-transmit	Sets the QoS group ID (1–99) and sends the packet
Transmit	Sends the packet

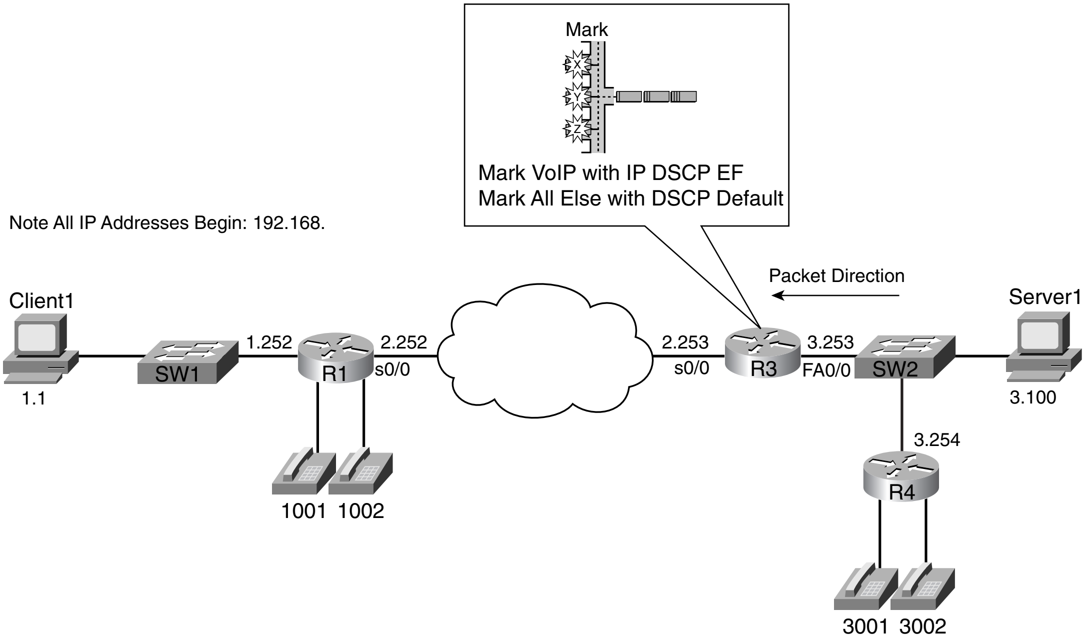
Table 3-13 *Exec Command Reference for CAR*

Command	Function
show interfaces [*interface-type interface-number*] **rate-limit**	Displays CAR statistics on the interface specified, or on all interfaces if the interface is not specified
show access-lists rate-limit [*acl-index*]	Lists information about the configuration of rate-limit ACLs

The first CAR marking example, shown in Example 3-5, uses the following criteria for marking packets. In this example, R3 is marking packets that flow right to left in Figure 3-15. (This example's criteria matches that in Example 3-1 for CB marking.)

- All VoIP payload traffic is marked with DSCP EF.
- All other traffic is marked with DSCP Default.

Figure 3-15 *CAR Marking Sample 1: VoIP Marked with DSCP EF, Everything Else Marked BE*



Example 3-5 *CAR Marking, VoIP as DSCP EF, Everything Else as BE*

```
no ip cef
!
access-list 102 permit udp any range 16384 32768 any range 16384 32768
!
interface fastethernet 0/0
 rate-limit input access-group 102 10000 20000 30000 conform-action
   set-dscp-transmit 46 exceed-action set-dscp-transmit 46
 rate-limit input 10000 20000 30000 conform-action set-dscp-transmit 0
   exceed-action set-dscp-transmit 0
end
```

The configuration does not take nearly as many different commands as the CB marking example, because most of the interesting parameters are contained in the **rate-limit** commands. Cisco Express Forwarding (CEF) is disabled, just to make the point that although you can use CEF with CAR, it is not required. ACL 102 defines some classification parameters that CAR will use to match VoIP packets, looking at UDP ports between 16,384 and 32,767. The ACL logic matches all VoIP payload, but it will also match VoIP Real Time Control Protocol (RTCP) traffic, which uses the odd-numbered UDP ports in the same port range. Finally, two **rate-limit** commands under FA0/0 enable CAR, define policing limits, classification details, and marking details.

The first of the two **rate-limit** commands matches a subset of all traffic using classification, whereas the second **rate-limit** command just matches all traffic. CAR uses the information configured in these two commands sequentially; in other words, if a packet matches the first CAR statement's classification details, the statement is matched, and its actions are followed. If not, CAR compares the next statement, and so on. In this example, the first CAR **rate-limit** command matches VoIP packets by referring to ACL 102, and the second statement, because it does not refer to an ACL, matches all packets.

NOTE CAR can actually match multiple statements on the same interface. Some CAR actions include the keyword **continue**, which means that even after the statement is matched, CAR should keep searching the statements for further matches. This allows CAR to nest statements, to perform features such as "police all traffic at 500 kbps, but police subsets at 250 kbps, 200 kbps, and 150 kbps."

Now examine the first **rate-limit** command, **rate-limit input access-group 102 10000 20000 30000 conform-action set-dscp-transmit 46 exceed-action set-dscp-transmit 46**, in detail. The **input** keyword means that CAR examines traffic entering the interface. The **access-group**

102 command means that packets permitted by ACL 102 are considered to match this **rate-limit** command. The next three values represent the committed rate, the burst size, and the excess size, which make up the traffic contract. The **conform-action** keyword identifies that the next parameter defines the action applied to conforming traffic, and the **exceed-action** keyword identifies that the next parameter defines the action applied to traffic that exceeds the traffic contract. In this example, both the conform and exceed actions are identical: **set-dscp-transmit 46**, which marks the DSCP value to decimal 46, or DSCP EF. (The **rate-limit** command does not allow the use of DSCP names.)

In this example, the actual traffic contract does not matter, because the actions for conforming traffic and excess traffic are the same. The true goal of this example is just to use CAR to mark packets VoIP—not to actually police the traffic. Chapter 5 includes CAR examples with different conform and exceed actions. The three values represent the committed rate (bps), the committed burst size (bytes), and the committed burst plus the excess burst (bytes). The excess burst parameter essentially provides a larger burst during the first measurement interval after a period of inactivity. (Chapter 5 covers the details of these settings.)

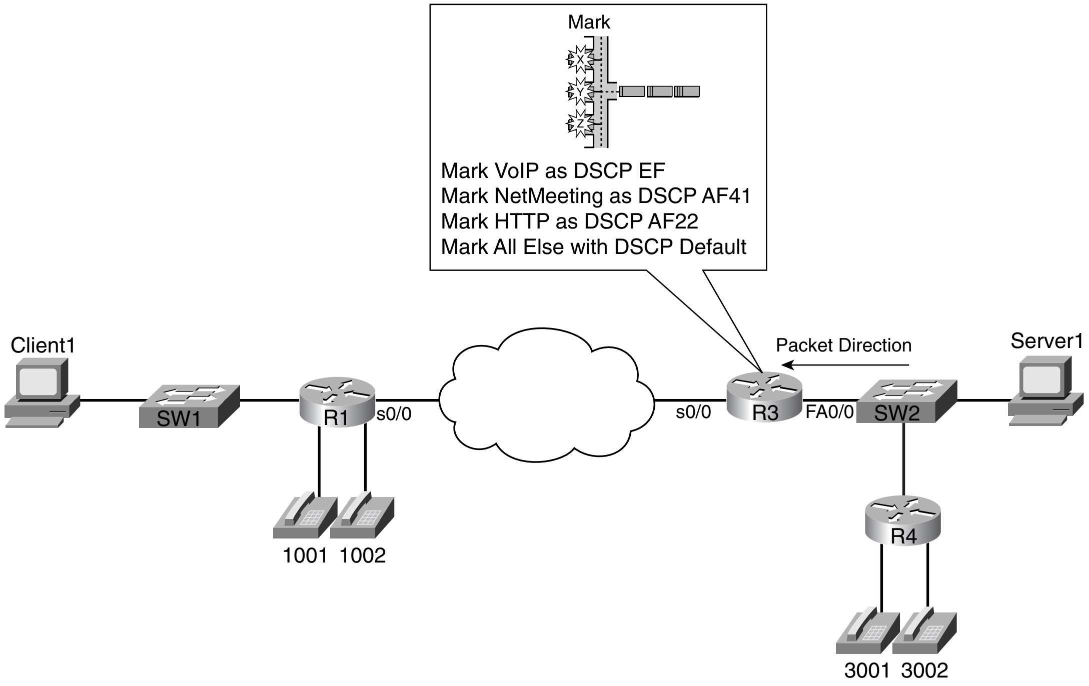
The second **rate-limit** command, **rate-limit input 10000 20000 30000 conform-action set-dscp-transmit 0 exceed-action set-dscp-transmit 0**, matches all remaining traffic. The only way that CAR can classify packets is to refer to an IP ACL, or a CAR rate-limit ACL, from the **rate-limit** command. The second **rate-limit** command does not refer to an ACL with the **access-group** keyword, so by implication, the statement matches all packets. Both actions set the DSCP value to zero. Essentially, this example uses CAR to mark traffic with either DSCP 46 or 0 (decimal), without discarding any packets due to policing.

The second sample CAR configuration, Example 3-6, includes classification options similar to CB marking Example 3-4. Because CAR cannot take advantage of NBAR, CAR cannot look at the URL for HTTP requests, as the CB marking example did. The slightly modified criteria for CAR marking in Example 3-6 is as follows:

- VoIP payload is marked with DSCP EF.
- NetMeeting voice and video from Server1 to Client1 is marked with DSCP AF41.
- Any HTTP traffic is marked with AF22.
- All other traffic is marked with DSCP Default.

Figure 3-16 shows the network in which the configuration is applied, and Example 3-6 shows the configuration.

Figure 3-16 *CAR Marking Sample 2 Network*



Example 3-6 ***CAR Marking Sample 2:*** *VoIP, NetMeeting Audio/Video, HTTP URLs, and Everything Else*

```
no ip cef
!
access-list 110 permit udp any range 16384 32768 any range 16384 32768
!
access-list 111 permit udp host 192.168.1.100 gt 16383 192.168.3.0 0.0.0.255 gt 16383
!
access-list 112 permit tcp any eq www any
access-list 112 permit tcp any any eq www
!
!
interface fastethernet 0/0
rate-limit input access-group 111 8000 20000 30000 conform-action
  set-dscp-transmit 34 exceed-action set-dscp-transmit 34
rate-limit input access-group 110 8000 20000 30000 conform-action
  set-dscp-transmit 46 exceed-action set-dscp-transmit 46
rate-limit input access-group 112 8000 20000 30000 conform-action
  set-dscp-transmit 20 exceed-action set-dscp-transmit 20
```

continues

Example 3-6 ***CAR Marking Sample 2:*** *VoIP, NetMeeting Audio/Video, HTTP URLs, and Everything Else (Continued)*

```
rate-limit input 8000 20000 30000 conform-action set-dscp-transmit 0
  exceed-action set-dscp-transmit 0

end
R3#show interface fastethernet 0/0 rate-limit
Fastethernet0/0 connected to SW2, where Server1 is connected
  Input
    matches: access-group 111
      params:  8000 bps, 20000 limit, 30000 extended limit
      conformed 1346 packets, 341169 bytes; action: set-dscp-transmit 34
      exceeded 2683 packets, 582251 bytes; action: set-dscp-transmit 34
      last packet: 56ms ago, current burst: 29952 bytes
      last cleared 00:07:11 ago, conformed 6000 bps, exceeded 10000 bps
    matches: access-group 110
      params:  8000 bps, 20000 limit, 30000 extended limit
      conformed 6118 packets, 452856 bytes; action: set-dscp-transmit 46
      exceeded 34223 packets, 2552218 bytes; action: set-dscp-transmit 46
      last packet: 12ms ago, current burst: 29989 bytes
      last cleared 00:07:11 ago, conformed 8000 bps, exceeded 47000 bps
    matches: access-group 112
      params:  8000 bps, 20000 limit, 30000 extended limit
      conformed 677 packets, 169168 bytes; action: set-dscp-transmit 20
      exceeded 3631 packets, 5084258 bytes; action: set-dscp-transmit 20
      last packet: 8ms ago, current burst: 29638 bytes
      last cleared 00:07:12 ago, conformed 3000 bps, exceeded 94000 bps
    matches: all traffic
      params:  8000 bps, 20000 limit, 30000 extended limit
      conformed 671 packets, 279572 bytes; action: set-dscp-transmit 0
```

The **show interface Fastethernet 0/0 rate-limit** command lists the pertinent statistical information about CAR's performance. The output has one stanza correlating to each **rate-limit** command on the interface, as highlighted in the example. Under each stanza, the number of packets and bytes that conformed, and the number of packets and bytes that exceeded the traffic contract, are listed. Because this CAR configuration was intended only for marking traffic, the number of packets and bytes in each category does not matter; Chapter 5 takes a closer look at the two values. For comparison purposes, however, consider the bps rates of the combined conformed and exceeded values. For instance, the second **rate-limit** command referenced ACL 110, which matched the two VoIP calls between R1 and R4. These two values total 55 kbps, which is the amount of traffic expected from a pair of G.729a calls over an Ethernet network.

CAR Marking Summary

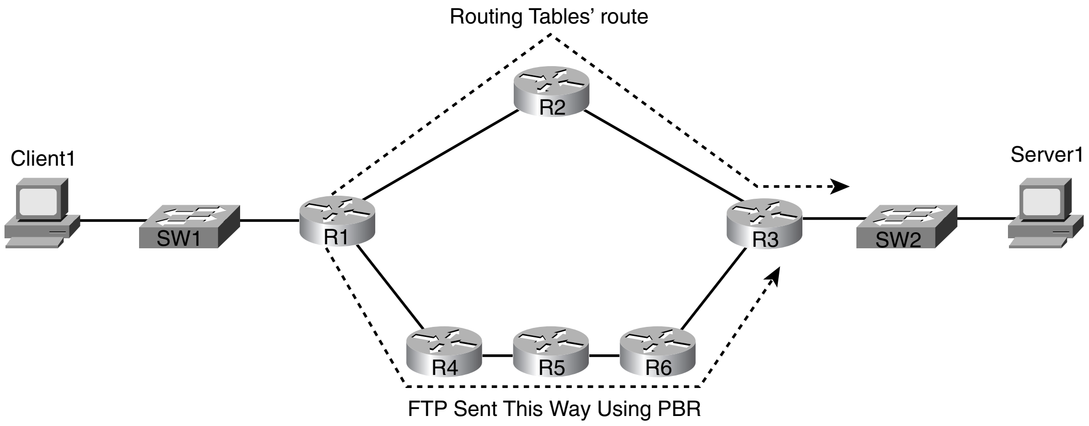
CAR is another tool that examines packet header information to classify and mark packets. CAR provides fewer options for classification and marking than does CB marking, but CAR is considered to be DiffServ compliant because it can classify DSCP using an ACL and mark the DSCP field directly. CAR, along with CB marking and PBR, makes classification decisions based on the contents of packet headers and marks QoS fields based on those classifications. Dial peers provide very different classification options, so fewer direct comparisons can be drawn.

Refer to Table 3-17 for a complete list of classification and marking fields used by CAR.

Policy-Based Routing (PBR)

PBR enables you to route a packet based on other information, in addition to the destination IP address. In most cases, engineers are happy with the choices of routes made by the routing protocol, with routing occurring based on the destination IP address in each packet. For some specialized cases, however, an engineer may want some packets to take a different path. One path through the network may be more secure, for instance, so some packets could be directed through a longer, but more secure, path. Some packets that can tolerate high latency may be routed through a path that uses satellite links, saving bandwidth on the lower-latency terrestrial circuits for delay-sensitive traffic. Regardless of the reasons, PBR can classify packets and choose a different route. Figure 3-17 shows a simple example, where FTP traffic is directed over the longer path in the network.

Figure 3-17 *PBR: FTP Traffic Routed over Longer Path*



PBR supports packet marking and policy routing. As you learned in previous sections, CAR supports marking because CAR's main feature, policing, benefits from having the marking feature available as well. Similarly, PBR includes a marking feature, because in some cases, PBR is used to pick a different route for QoS reasons—for instance, to affect the latency of a packet. So, PBR's core function can benefit from marking a packet, so that the appropriate QoS action can be taken as the packet traverses the network. Just as with CAR, you can use PBR's marking feature without actually using its core feature. In other words, you can use PBR just for classification and marking, without choosing a different route. The examples in this chapter focus only on PBR as a marking tool.

Unlike CB marking and CAR, PBR only processes packets entering an interface; you cannot enable it for packets exiting an interface. The reason PBR only processes incoming packets relates to its core function: policy routing. PBR needs to process packets before a routing decision has been made. Therefore, PBR processes packets entering an interface, preempting the normal routing logic based on destination IP address.

Finally, one other difference between PBR and the other classification and marking tools covered so far (CB marking and CAR) is that PBR can classify based on routing information, instead of totally relying on information in the frame or packet header. PBR can look up the entry in the routing table that matches a packet's destination address, for instance, and then classify based on information about that route. For example, the metric associated with that route, the source of the routing information, or the next-hop interface associated with the route can be checked. In most cases, this routing information does not help you with differentiating between different types of traffic. An FTP server, an IP Phone, a video server, and some web servers may all be in the same subnet, for instance, but the routing information about that subnet could not help PBR distinguish between those different types of traffic. Therefore, typically the most useful classification feature of PBR, when used for marking, is just to refer to an IP ACL.

PBR configuration uses yet another totally different set of configuration commands as compared to CB marking and CAR. PBR does separate the classification, marking, and enabling features into different commands. Tables 3-14 and 3-15 list the pertinent PBR configuration and exec commands, respectively. Following the tables, two example PBR configurations are shown. The two examples use the same criteria as the two CAR samples.

Table 3-14 *Configuration Command Reference for PBR*

Command	Mode and Function
ip local policy route-map *map-tag*	Global; specifies that packets generated by this router should be candidates for policy routing
ip policy route-map *map-tag*	Interface subcommand; refers to a route map, which in turn classifies packets and specifies actions; actions include specifying a different route, and setting IP precedence
route-map *map-tag* **[permit | deny] [***sequence-number***]**	Global command; creates a route map entry
match ip address {*access-list-number* **|** *access-list-name***} [***... access-list-number* **|** *... access-list-name***]**	Route-map subcommand; used to match IP packets based on parameters that can be matched with an IP ACL
match length *minimum-length maximum-length*	Route-map subcommand; used to match IP packets based on their length

Table 3-14 *Configuration Command Reference for PBR (Continued)*

Command	Mode and Function
set ip precedence *number* | *name*	Route-map subcommand; sets IP precedence value using the decimal number of name
set ip next-hop *ip-address* **[**...*ip-address***]**	Route-map subcommand; defines the IP address(es) of the next-hop router(s) to be used for forwarding packets that match this route map entry
ip route-cache policy	Global command; enables fast switching of PBR-routed packets

Note: Not all PBR-related commands are shown in this table, but commands specifically related to marking are shown.

Table 3-15 *Exec Command Reference for PBR Marking*

Command	Function
show ip policy	Lists configured PBR details, and statistics for numbers of packets matched by each clause.
show route-map	Lists statistical information about packets matched with a route map. PBR uses route maps to classify and mark traffic.

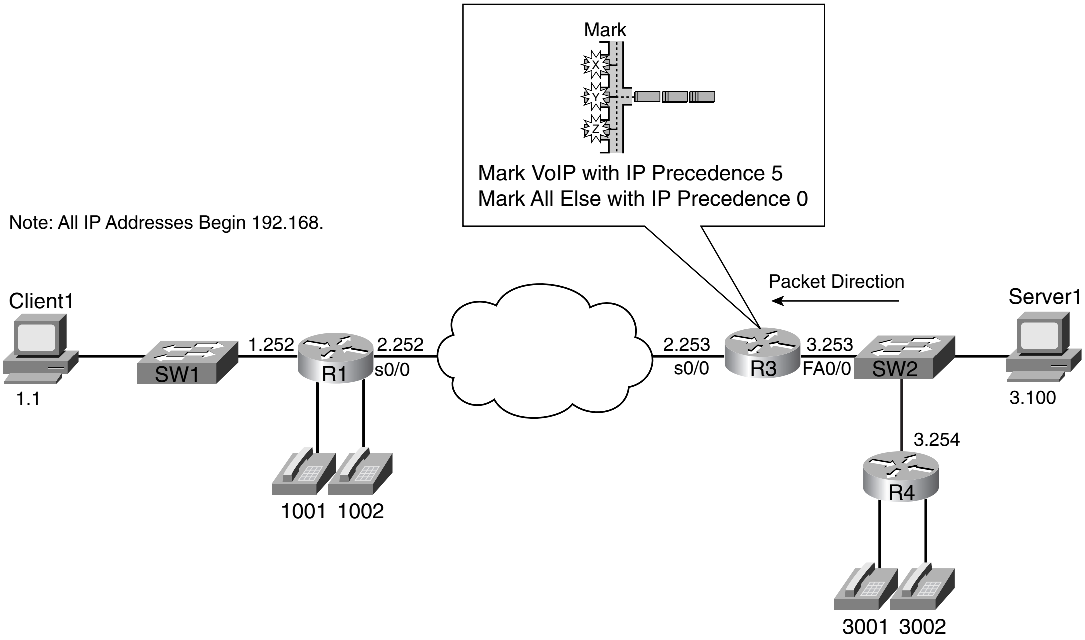
Example 3-7 shows the first PBR marking example, which uses the same criteria as Example 3-1 for CB marking and Example 3-5 for CAR. In this example, R3 is marking packets that flow right to left in Figure 3-18.

- All VoIP payload traffic is marked with IP precedence 5.
- All other traffic is marked with IP precedence 0.

Example 3-7 *PBR Marking, VoIP as DSCP EF, Everything Else as BE*

```
ip route-cache policy
!
ip access-list extended VoIP-ACL
 permit udp any range 16384 32767 any range 16384 32767
!
int fastethernet 0/0
 ip policy route-map voip-routemap
!
route-map voip-routemap permit 10
 match ip address VoIP-ACL
 set ip precedence 5
!
route-map voip-routemap permit 20
set ip precedence 0
```

Figure 3-18 *PBR Marking Sample 1: VoIP Marked with IP Precedence 5, Everything Else Marked IP Precedence 0*



PBR uses **route-map** commands, along with **match** and **set** route-map subcommands, to classify and mark the packets. This configuration uses a route map named voip-routemap, which includes two clauses. The first clause, clause 10, uses a **match** command that refers to VoIP-ACL, which is a named IP ACL. VoIP-ACL matches UDP port numbers between 16,384 and 32,767, which matches all VoIP traffic. If the ACL permits a packet, the route map's first clause acts on the **set** command, which specifies that IP precedence should be set to 5.

The second route map clause, clause 20, matches the rest of the traffic. The route map could have referred to another IP ACL to match all packets; however, by not specifying a match statement in clause 20, all packets will match this clause by default. By not having to refer to another IP ACL to match all packets, less processing overhead is required. The **set** command then specifies to set precedence to zero.

The **ip policy route-map voip-routemap** command enables PBR on interface FA0/0 for incoming packets. Notice that the direction, input or output, is not specified, because PBR can only process incoming packets.

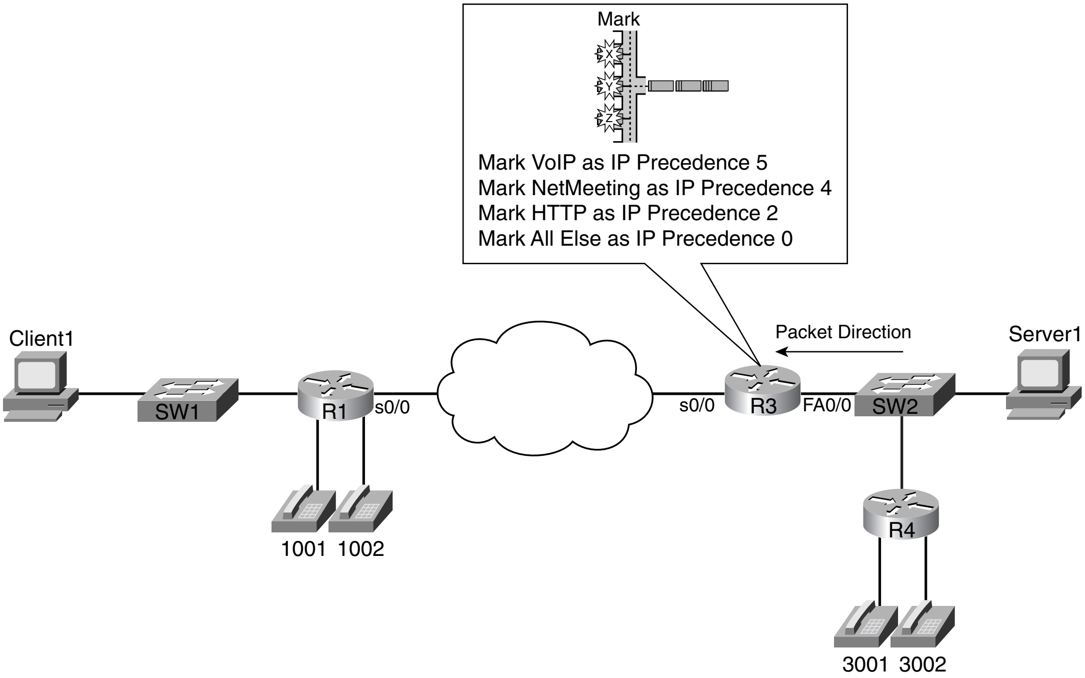
The last PBR-specific command is **ip route-cache policy**. IOS process-switches PBR traffic by default; to use fast switching on PBR traffic, use the **ip route-cache policy** command.

The second PBR configuration (Example 3-8) includes classification options identical to CAR example 2 (see Example 3-6). A major difference between PBR and CAR is that PBR cannot set the DSCP field, so it sets the IP Precedence field instead. The slightly modified criteria, as compared with CAR example 2, for PBR example 2 is as follows:

- VoIP payload is marked with precedence 5.
- NetMeeting voice and video from Server1 to Client1 is marked with precedence 4.
- Any HTTP traffic is marked with precedence 2.
- All other traffic is marked with precedence 0.

Figure 3-19 shows the network in which the configuration is applied, and Example 3-8 shows the configuration.

Figure 3-19 *PBR Marking Sample 2 Network*


Mark
X
Y
Z
Mark VoIP as IP Precedence 5
Mark NetMeeting as IP Precedence 4
Mark HTTP as IP Precedence 2
Mark All Else as IP Precedence 0
Client1
SW1
R1
s0/0
s0/0
R3
FA0/0
SW2
Packet Direction
Server1
1001 1002
R4
3001 3002

Example 3-8 *PBR Marking Sample 2: VoIP, NetMeeting Audio/Video, HTTP URLs, and Everything Else*

```
ip route-cache policy
!
ip access-list extended VoIP-ACL
 permit udp any range 16384 32768 any range 16384 32768
!
ip access-list extended NetMeet-ACL
 permit udp host 192.168.1.100 range 16384 32768 192.168.3.0 0.0.0.255 range 16384 32768
!
!
ip access-list extended http-acl
 permit tcp any eq www any
 permit tcp any any eq www
!
interface fastethernet 0/0
 ip policy route-map voip-routemap
!
route-map voip-routemap permit 10
 match ip-address NetMeet-ACL
 set ip precedence 4
!
route-map voip-routemap permit 20
 match ip-address VoIP-ACL
 set ip precedence 5
!
route-map voip-routemap permit 30
 match ip-address http-acl
 set ip precedence 2
!
route-map voip-routemap permit 40
set ip precedence 0
!
end
R3#sh ip policy
Interface      Route map
Fastethernet0/0    voip-routemap

R3#show route-map
route-map voip-routemap, permit, sequence 10
  Match clauses:
    ip address (access-lists): NetMeet-ACL
  Set clauses:
    ip precedence flash-override
  Policy routing matches: 3 packets, 222 bytes
route-map voip-routemap, permit, sequence 20
  Match clauses:
    ip address (access-lists): VoIP-ACL
  Set clauses:
    ip precedence critical
  Policy routing matches: 14501 packets, 1080266 bytes
```

Example 3-8 *PBR Marking Sample 2: VoIP, NetMeeting Audio/Video, HTTP URLs, and Everything Else (Continued)*

```
route-map voip-routemap, permit, sequence 30
  Match clauses:
    ip address (access-lists): http-acl
  Set clauses:
    ip precedence immediate
  Policy routing matches: 834 packets, 1007171 bytes
route-map voip-routemap, permit, sequence 40
  Match clauses:
  Set clauses:
    ip precedence routine
  Policy routing matches: 8132 packets, 11263313 bytes
```

The output of the **show ip policy** command lists only sparse information. The **show route-map** command enables you to view statistical information about what PBR has performed. This command lists statistics for any activities performed by a route map, including when one is used for PBR. Notice that the four sets of classification criteria seen in the configuration are listed in the highlighted portions of the **show route-map** output, as are packet and byte counters.

PBR Marking Summary

PBR provides another classification and marking tool that examines packet header information to classify and mark packets. PBR is unique compared to the other tools in that it can classify based on information about the route that would be used for forwarding a packet. However, PBR has fewer options for matching header fields for classification as compared with the other tools.

PBR can mark IP precedence, QoS group, as well as the ToS bits. Refer to Table 3-17, in the summary for this chapter, for a complete list of classification and marking fields used by PBR.

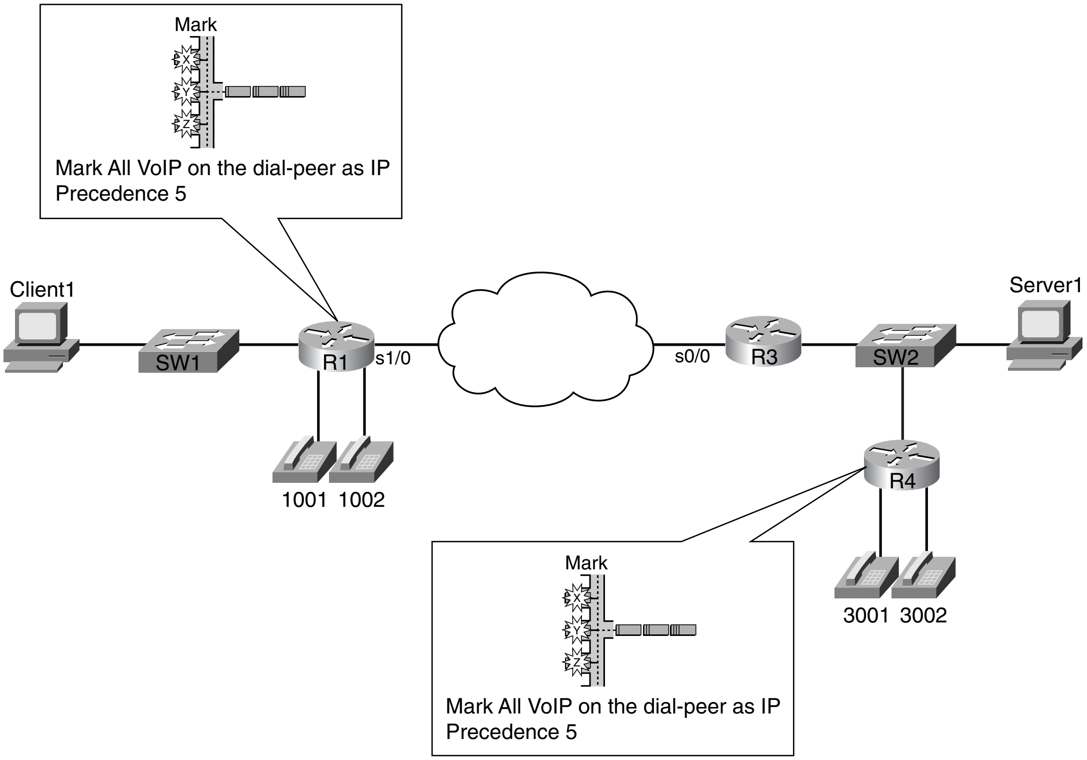
PBR provides a strong option for classification and marking in two cases. For applications when marking based on routing information is useful, PBR can look at details about the route used for each packet, and make marking choices. The other application for PBR marking is when policy routing is already needed, and marking needs to be done at the same time. For more general cases of classification and marking, CB marking or CAR is recommended.

VoIP Dial Peer

IOS voice gateways provide many services to connect the packetized, VoIP network to non-packetized, traditional voice services, including analog and digital trunks. IOS gateways perform many tasks, but one of the most important tasks is to convert from packetized voice to nonpacketized voice, and vice versa. In other words, voice traffic entering a router on an analog or digital trunk is not carried inside an IP packet, but the IOS gateway converts the incoming voice to a digital signal (analog trunks only) and adds the appropriate IP, UDP, and RTP headers around the digital voice (both analog and digital trunks). Conversely, when a VoIP packet arrives, and the voice needs to be sent out a trunk, the IOS gateway removes the packet headers, converts the voice to analog (analog trunks only), and sends the traffic out the trunk.

Although this book does not attempt to explain voice configuration and concepts to much depth, some appreciation for IOS gateway configuration is required for some of the functions covered in this book. In particular, Chapter 8, "Call Admission Control and QoS Signaling," which covers Voice call admission control (CAC), requires a little deeper examination of voice. To understand classification and marking using dial peers, however, only a cursory knowledge of voice configuration is required. Consider Figure 3-20, for instance, which shows two analog IOS voice gateways, R1 and R4, along with Examples 3-9 and 3-10, which show the pertinent configuration on R1 and R4.

Figure 3-20 *Network with Two Analog Voice Gateways*


Mark
X
Y
Z
Mark All VoIP on the dial-peer as IP Precedence 5
Client1
SW1
R1
s1/0
s0/0
R3
SW2
Server1
1001 1002
R4
3001 3002
Mark
X
Y
Z
Mark All VoIP on the dial-peer as IP Precedence 5

Example 3-9 *R1 Voice Gateway Configuration*

```
hostname R1
!
int fastethernet 0/0
ip address 192.168.1.251 255.255.255.0
!
dial-peer voice 3001 voip
 destination-pattern 3001
 session target ipv4:192.168.3.254
!
dial-peer voice 3002 voip
 destination-pattern 3002
 session target ipv4:192.168.3.254
!
dial-peer voice 1001 pots
 destination-pattern 1001
 port 3/0
!
dial-peer voice 1002 pots
 destination-pattern 1002
 port 3/1
```

Example 3-10 *R4 Voice Gateway Configuration*

```
hostname R4
!
int fastethernet 0/0
ip address 192.168.3.254 255.255.255.0
!
dial-peer voice 1001 voip
 destination-pattern 1001
 session target ipv4:192.168.1.251
!
dial-peer voice 1002 voip
 destination-pattern 1002
 session target ipv4:192.168.1.251
!
dial-peer voice 3001 pots
 destination-pattern 3001
 port 3/0
!
dial-peer voice 3002 pots
 destination-pattern 3002
 port 3/1
```

The highlighted portions of the examples focus on the configuration for the physical voice ports on R1, and the VoIP configuration on R4. Both R1 and R4 use **dial-peer** commands to define their local analog voice trunks and to define peers to which VoIP calls can be made. In Example 3-9, for instance, the highlighted portion of the configuration shows R1's configuration of the two local analog lines. The two highlighted **dial-peer** statements use the keyword **pots**, which stands for plain-old telephone service. The **pots** keyword implies that the ports associated with this dial peer are traditional analog or digital telephony ports. The physical analog ports are correlated to each dial peer with the **port** command; in each of these configurations, a two-port FXS card sits inside slot 3 of a 1760-V router. Finally, on R1, the phone number, or dial pattern, associated with each of the analog ports is configured. With just the highlighted configuration in R1, voice calls could be placed between the two extensions (x1001 and x1002).

To place calls to extensions 1001 and 1002 from R4, the **dial-peer** commands highlighted in Example 3-10 are required. These two **dial-peer** commands use a **voip** keyword, which means this dial peer configures information about an entity to which VoIP calls can be placed. The phone number, or dial pattern, is defined with the **destination-pattern** command again—notice that extensions 1001 and 1002 are again configured. Finally, because these two dial peers configure details about a VoIP call, a local physical port is not referenced. Instead, the **session-target ipv4:192.168.1.251** command implies that when these phone numbers are called, to establish a VoIP call, using the IP version 4 IP address shown.

Similarly, R4 defines the local phone numbers and ports for the locally connected phones, and R1 defines VoIP dial peers referring to R4's phones, so that calls can be initiated from R1.

Dial-peer classification and marking, when you know how to configure the basic dial-peer parameters, is easy. POTS dial peers refer to analog or digital trunks, over which no IP packet is in use—so there is nothing to mark. On VoIP dial peers, the dial peer refers to the IP address of another gateway to which a call is placed. So, by placing the **ip precedence 5** dial-peer subcommand under each voip **dial-peer**, the packets generated for calls matching each dial peer will be marked with IP precedence 5. Example 3-11 lists the R4 configuration, with these changes made; the equivalent changes would be made to R1 as well.

Example 3-11 *R4 Voice Gateway Configuration*

```
hostname R4
!
interface fastethernet 0/0
ip address 192.168.3.254 255.255.255.0
!
dial-peer voice 1001 voip
 destination-pattern 1001
 session target ipv4:192.168.1.251
 ip precedence 5
 no vad
!
dial-peer voice 1002 voip
 destination-pattern 1002
 session target ipv4:192.168.1.251
```

Example 3-11 *R4 Voice Gateway Configuration (Continued)*

```
 ip precedence 5
 no vad
!
dial-peer voice 3001 pots
 destination-pattern 3001
 port 3/0
!
dial-peer voice 3002 pots
 destination-pattern 3002
 port 3/1
```

In the example, the highlighted text shows the **ip precedence 5** commands under each **voip dial-peer**. Packets created for VoIP calls for the configured dial patterns of 1001 and 1002 will be marked with IP precedence 5. The identical commands would be added to R1's configuration on the VoIP dial peers to achieve the same effect.

Beginning in IOS Releases 12.2(2)XB and 12.2(2)T the **ip precedence** command has been replaced with the **ip qos dscp** command. This allows the dial peer to set the IP precedence or the DSCP value for VoIP payload and signaling traffic. Also keep in mind that the current DQOS exam, at the time this book was published, was based on IOS 12.1(5)T—so this command would not be on the current exam. Check the URLs listed in the Introduction for any possible changes.

The command uses the following syntax:

```
ip qos dscp [number | set-af | set-cs | default | ef][media | signaling]
```

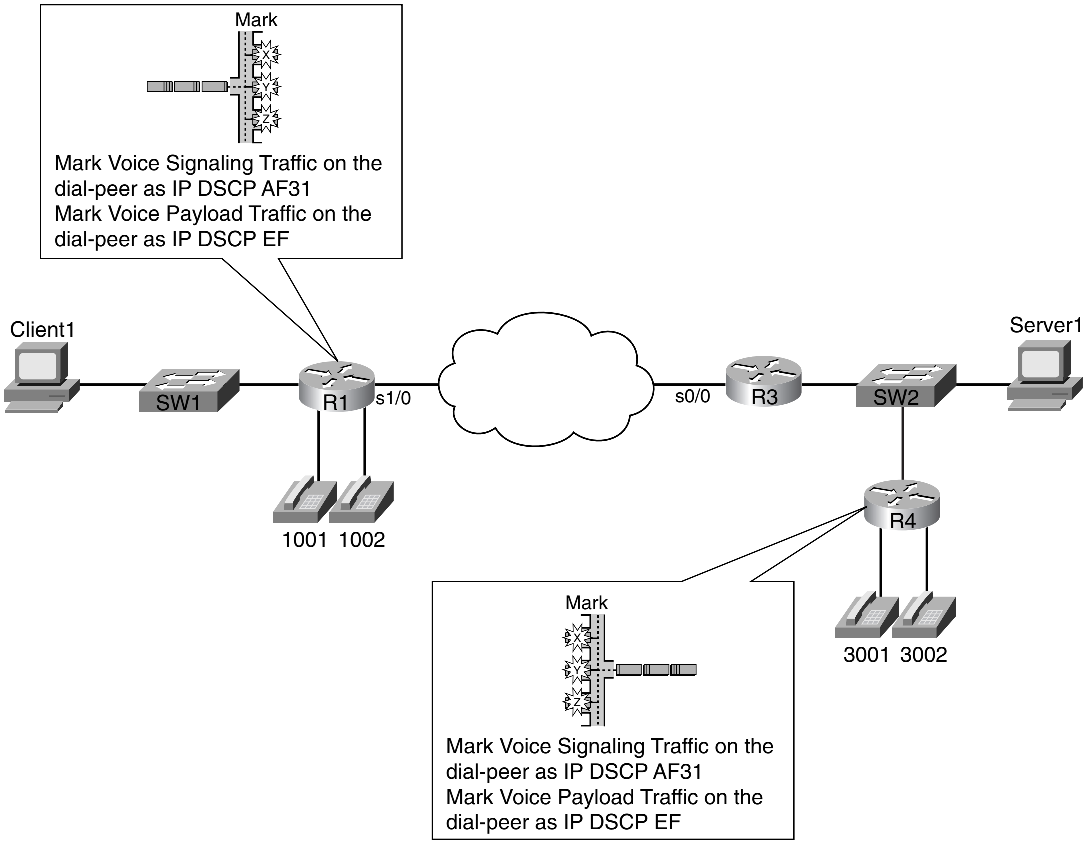
Table 3-16 outlines the meaning of the parameters of the command.

Table 3-16 *IP QoS DSCP Command Options*

IP QoS DSCP Options	Function
number	DSCP value. Valid entries are from 0 to 63.
set-af	Sets DSCP to assured forwarding bit pattern. Acceptable values are as follows: AF11, AF12, AF13, AF21, AF22, AF23, AF31, AF32, AF33, AF41, AF42, AF43
set-cs	Sets DSCP to class selector code point. Acceptable values are as follows: CS1, CS2, CS3, CS4, CS5, CS6, CS7
default	Sets DSCP to default bit pattern 000000.
ef	Sets DSCP to expedited forwarding bit pattern 101110.
media	Applies the specified DSCP value to the media payload packets.
signaling	Applies the specified DSCP value to the signaling packets.

The **ip qos dscp** command enables you to have much more granular control of how a VoIP packet is marked than the **ip precedence** command, while providing a method to preserve backward compatibility. Examples 3-12 and 3-13 show how R1 and R4 can be configured to use the **ip qos dscp** command to mark voice payload traffic with a DSCP value of EF and voice signaling traffic with a DSCP value of AF31. Figure 3-21 shows the now-familiar network, with the new criteria listed.

Figure 3-21 *Mark Voice Payload Traffic*


Mark
X
Y
Z
Mark Voice Signaling Traffic on the dial-peer as IP DSCP AF31
Mark Voice Payload Traffic on the dial-peer as IP DSCP EF
Client1
SW1
R1
s1/0
s0/0
R3
SW2
Server1
1001 1002
R4
3001 3002
Mark
X
Y
Z
Mark Voice Signaling Traffic on the dial-peer as IP DSCP AF31
Mark Voice Payload Traffic on the dial-peer as IP DSCP EF

Example 3-12 *R1 IP QoS DSCP Dial-Peer Configuration*

```
hostname R1
!
int fastethernet 0/0
ip address 192.168.1.251 255.255.255.0
```

Example 3-12 *R1 IP QoS DSCP Dial-Peer Configuration (Continued)*

```
!
dial-peer voice 3001 voip
 destination-pattern 3001
 ip qos dscp ef media
 ip qos dscp af31 signaling
 session target ipv4:192.168.3.254
!
dial-peer voice 3002 voip
 destination-pattern 3002
 ip qos dscp ef media
 ip qos dscp af31 signaling
 session target ipv4:192.168.3.254
!
dial-peer voice 1001 pots
 destination-pattern 1001
 port 3/0
!
dial-peer voice 1002 pots
 destination-pattern 1002
 port 3/1
```

Example 3-13 *R4 IP QoS DSCP Dial-Peer Configuration*

```
hostname R4
!
int fastethernet 0/0
ip address 192.168.3.254 255.255.255.0
!
dial-peer voice 1001 voip
 destination-pattern 1001
 ip qos dscp ef media
 ip qos dscp af31 signaling
 session target ipv4:192.168.1.251
!
dial-peer voice 1002 voip
 destination-pattern 1002
 ip qos dscp ef media
 ip qos dscp af31 signaling
 session target ipv4:192.168.1.251
!
dial-peer voice 3001 pots
 destination-pattern 3001
 port 3/0
!
dial-peer voice 3002 pots
 destination-pattern 3002
 port 3/1
```

In this example, the highlighted text shows the **ip qos dscp** commands used to mark voice signaling with DSCP AF31 and voice payload with DSCP EF. For networks that cannot yet support DSCP markings, you can use the **set-cs** option to mark the voice traffic with IP precedence, providing backward-compatible support.

VoIP Dial-Peer Summary

For voice traffic passing through an IOS gateway, marking the traffic using dial peers provides an easy-to-configure, low-overhead way to mark the packets. Prior to IOS Releases 12.2(2)XB and 12.2(2)T the **ip precedence** command was used to mark all VoIP traffic with an IP precedence value. After these IOS releases, you can use the **ip qos dscp** command to separate and individually mark the voice signaling and voice payload traffic. These markings can be DCSP values, or IP precedence values if backward compatibility is needed. Refer to Tables 3-14 and 3-15 for **ip qos dscp** command options.

Summary of Classification and Marking QoS Features

Classification and marking tools can be easily compared based on three general categories. First, some classification and marking tools are specialized, and some are more general. CB marking, CAR, and PBR all perform the same basic function of matching packets based on fields inside the packet header, and marking based on those fields, so these three tools are the more generalized classification and marking tools. Dial peers perform specialized classification and marking functions. This tool is different from the other three because it does not classify packets based on a variety of fields in a packet header, instead classifying all traffic to the particular dial peer.

The other two points for comparison of classification and marking tools concern what each tool can match for classification, and what each can mark for the marking feature. Tables 3-17 and 3-18 summarize the fields that you can use for classification and marking in the tools, respectively.

Table 3-17 *Classification Fields Used by Classification and Marking Tools*

Classification Field*	CB Marking	CAR	PBR	Dial Peers
Extended IP ACLs	X	X	X	
DSCP	X			
Precedence	X	X		
QoS Group	X			
CoS	X			
NBAR	X			
VoIP Payload (even-numbered RTP UDP ports)	X			

Table 3-17 *Classification Fields Used by Classification and Marking Tools (Continued)*

Classification Field*	CB Marking	CAR	PBR	Dial Peers
MPLS Experimental bits	X	X		
Input Interface	X			
Source MAC address	X	X		
Destination MAC address	X			
BGP ASN			X	
BGP Community			X	
Outgoing Interface			X	
Next-hop IP Address			X	
Routing Protocol Metric			X	
Source of Routing Information			X	
Packet Length			X	
VoIP Dial Peers				X

* Some fields can be matched via ACL as well as matched directly. For instance, DSCP can be matched with an ACL. Because CB marking is the only tool that can directly configure a match for DSCP, only CB marking's box is checked in the table. Refer to Table 3-2, earlier in this chapter, for a list of fields matchable with IP extended ACLs.

Table 3-18 *Marking Fields Used by Classification and Marking Tools*

Marking Field	CB Marking	CAR	PBR	Dial Peers
DSCP	X	X		X
Precedence	X	X	X	X
QoS Group	X	X	X	
CoS	X			
MPLS Experimental bits	X	X		
Frame Relay DE	X			
ATM CLP	X			
IP ToS bits			X	

The choice of when to use each tool can be confusing. Dial peers have specific applications, so the number of instances where they are useful directs you when to consider each tool. Dial peers are used only in IOS voice gateways, which is a convenient place to set IP precedence or DSCP.

The three general classification and marking tools create a more difficult choice, because each can be enabled on almost any interface. CB marking may be ruled out based on the IOS revision being used, because CB marking did not appear in a T-train IOS until 12.1(5)T, and in mainline IOS until 12.2.

NOTE Refer to www.cisco.com/go/fn to use the Cisco feature navigator to find IOS revision levels required for different features.

When at a level of code that support CB marking, the general recommendation is to use CB marking, unless one of the next two statements is true. If you need to use PBR to perform policy routing, and you also need to mark, go ahead and use PBR for marking. Similarly, if you need to perform policing and marking at the same point in the network, use CAR for both.

Foundation Summary

The "Foundation Summary" is a collection of tables and figures that provide a convenient review of many key concepts in this chapter. For those of you already comfortable with the topics in this chapter, this summary could help you recall a few details. For those of you who just read this chapter, this review should help solidify some key facts. For any of you doing your final prep before the exam, these tables and figures are a convenient way to review the day before the exam.

Table 3-19 shows the list of items that can be matched with an extended IP ACL. Table 3-20 lists the fields that can be matched by classification and marking tools without use of an ACL. Note that some header fields can be matched by an ACL or directly through some other style of configuration—in those cases, it is typically better to match the field directly, rather than with an ACL.

Table 3-19 *IP Extended ACL Matchable Fields—IOS 12.2*

Field	Comments
Source IP address	A range of source IP addresses can be matched by using a wildcard mask.
Destination IP address	A range of source IP addresses can be matched by using a wildcard mask.
IP Precedence	Format of command uses names for precedence. The following table lists the decimal value for each name. Name IP precedence value **routine** 0 **priority** 1 **immediate** 2 **flash** 3 **flash-override** 4

continues

Table 3-19 *IP Extended ACL Matchable Fields—IOS 12.2 (Continued)*

Field	Comments
IP Precedence *(Continued)*	**Critic** 5 **internet** 6 **network** 7
IP DSCP	Format of the command allows use of differentiated services code point (DSCP) names, as well as decimal values.
IP ToS	Can check to see whether a single Type of Service (ToS) field bit is toggled on; keywords are **normal** (binary **0000**), **max-reliability** (binary **1000**), **max-throughput** (binary **0100**), **min-delay** (binary **0010**), and **min-monetary-cost** (binary **0001**).
TCP ports	Can check source and destination ports; can also check a range of port numbers, whether a port number is larger or smaller than a single value.
TCP Established	Although not typically useful for QoS classification, ACLs can match all TCP segments after the initial segment used for connection establishment.
UDP	Checks the source and destination ports; can also check a range of port numbers, whether a port number is larger or smaller than a single value.
ICMP	Checks a larger variety of ICMP messages and code types (for example, echo request and echo reply).
IGMP	Checks for Internet Group Management Protocol (IGMP) message types.

Table 3-20 *Fields* Directly *Matchable by Classification and Marking tools*

Field	Tool	Comments
Source MAC address	CAR, CB marking	Committed access rate (CAR) uses special "access-rate" ACLs; class-based (CB) marking uses the **match** command.
IP Precedence	CAR, CB marking	CAR uses special "access-rate" ACLs specific to CAR; CB marking uses the **match** command; both can match a subset of values.
MPLS Experimental	CAR, CB marking	CAR uses special "access-rate" ACLs specific to CAR; CB marking uses the **match** command; both can match a subset of values.

Table 3-20 *Fields* Directly *Matchable by Classification and Marking tools (Continued)*

Field	Tool	Comments
CoS	CB marking	Checks incoming ISL/802.1P CoS bits. Can match multiple values.
Destination MAC address	CB marking	Checks for destination MAC address. Can match multiple values.
Input Interface	CB marking	Checks for input interface. Can match multiple values.
IP DSCP	CB marking	Can check for multiple values using multiple **match** commands.
RTP's UDP port-number range	CB marking	RTP uses even-numbered UDP ports from 16,384 to 32,767. This option allows matching a subset of these values, even-numbered ports only, because RTP only uses even-numbered ports.
QoS Group	CB marking	The QoS Group field is used to tag packets internal to a single router.
NBAR protocol types	CB marking	Refer to the "Network Based Application Recognition (NBAR)" section in this chapter for more details.
NBAR Citrix applications	CB marking	NBAR can recognize different types of Citrix applications; CB marking can use NBAR to classify based on these application types.
Host name and URL string	CB marking	NBAR can also match URL strings, including the host name, using regular expressions. CB marking can use NBAR to match these strings for classification.
Outgoing Interface	Policy-based routing (PBR)	Checks the routing table and finds all valid routes for the packet; matches based on the outgoing interface.
Next-Hop	PBR	Similar to the outgoing interface, but it checks the next-hop routers' IP addresses.
Metric	PBR	Checks the routing table entry for this packet, and compares the metric value to match the packet.
Route type	PBR	Checks the routing table, looking at the source of the routing table entry that matches the packet.
Dial Peer	Dial peers	Based on the dial peer and used to connect a VoIP call.

Figure 3-22 outlines the two IP marking fields and their positions inside an IP header: The suggested values for these fields, and their names, are listed in Table 3-21.

Figure 3-22 *IP Precedence and IP DSCP Fields*

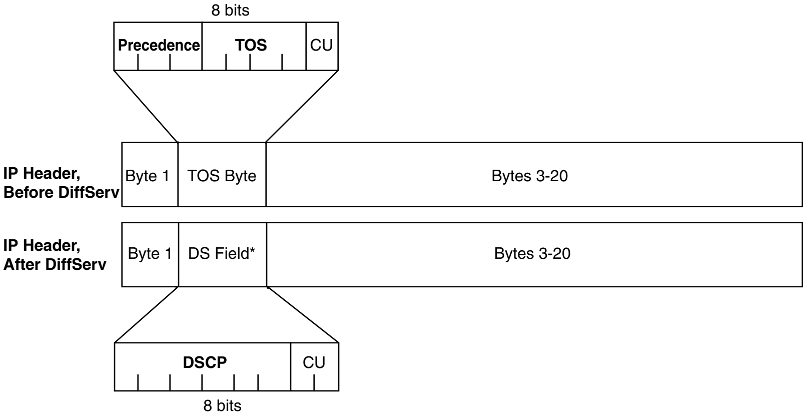


Table 3-21 *IP Precedence and DSCP—Popular Values and Names*

Field and Value (Decimal)	Binary Value	Name	Defined by This RFC
Precedence 0	**000**	**routine**	791
Precedence 1	**001**	**priority**	791
Precedence 2	**010**	**immediate**	791
Precedence 3	**011**	**flash**	791
Precedence 4	**100**	**flash override**	791
Precedence 5	**101**	**critic**	791
Precedence 6	**110**	**internetwork control**	791
Precedence 7	**111**	**network control**	791
DSCP 0	**000**000	**best effort** or **default**	2475
DSCP 8	**001**000	CS1	2475
DSCP 16	**010**000	CS2	2475
DSCP 24	**011**000	CS3	2475
DSCP 32	**100**000	CS4	2475
DSCP 40	**101**000	CS5	2475
DSCP 48	**110**000	CS6	2475
DSCP 56	**111**000	CS7	2475

Table 3-21 *IP Precedence and DSCP—Popular Values and Names (Continued)*

Field and Value (Decimal)	Binary Value	Name	Defined by This RFC
DSCP 10	**001**010	AF11	2597
DSCP 12	**001**100	AF12	2597
DSCP 14	**001**110	AF13	2597
DSCP 18	**010**010	AF21	2597
DSCP 20	**010**100	AF22	2597
DSCP 22	**010**110	AF23	2597
DSCP 26	**011**010	AF31	2597
DSCP 28	**011**100	AF32	2597
DSCP 30	**011**110	AF33	2597
DSCP 34	**100**010	AF41	2597
DSCP 36	**100**100	AF42	2597
DSCP 38	**100**110	AF43	2597
DSCP 46	**101**110	EF	2598

CS = Class Selector
AF = Assured Forwarding
EF = Expedited Forwarding

Figure 3-23 shows the general location of the CoS field inside ISL and 802.1P headers.

Figure 3-23 *LAN Class Of Service Fields*

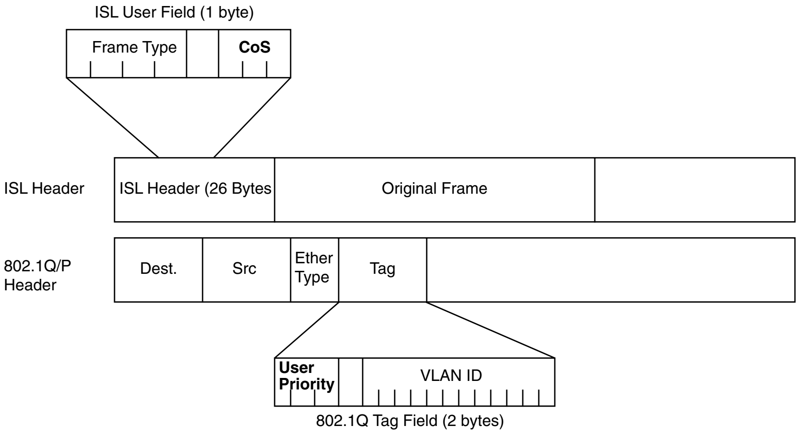


Table 3-22 summarizes the marking fields.

Table 3-22 *Names of Marking Fields*

Field	Location	Length	Comments
IP Precedence	IP header	3 bits	Contained in the first 3 bits of the ToS byte.
IP DSCP	IP header	6 bits	Contained in the first 6 bits of the DS field, which replaces the ToS byte.
DS	IP header	1 byte	Replaces ToS byte per RFC 2475.
ToS	IP header	1 byte	Replaced by DS field per RFC 2475.
ToS	IP header	4 bits	A field inside the ToS byte; superseded by RFC 2475.
CoS	ISL and 802.1Q/P	3 bits	Cisco convention uses "CoS" to describe either trunking headers' QoS field.
Priority bits	802.1Q/P	3 bits	The name used by IEEE 802.1P for the CoS bits.
Discard Eligible (DE)	Frame Relay header	1 bit	Frame Relay switches may discard DE-marked frames, avoiding discarding frames without DE marked, under congestion.
Cell Loss Priority (CLP)	ATM cell header	1 bit	ATM equivalent of the DE bit
MPLS Experimental values(s)	MPLS header	3 bits	Used to pass QoS marking information across an MPLS network.
QoS Group	Headers internal to IOS	N/A	Uses values between 1–99 inclusive. Used for marking only internal to a single router, specifically only on the GSR/ESR product lines.

Table 3-23 lists the MQC commands used for CB marking. The table shows all the classification options available using the **match** command, and all the marking options available using the **set** command. Table 3-24 lists the **show** commands related to CB marking.

Table 3-23 *Command Reference for Class-Based Marking*

Command	Mode and Function
class-map *class-map-name*	Global config; names a class map, where classification options are configured
Match …	Class-map subcommand; defines specific classification parameters

Table 3-23 *Command Reference for Class-Based Marking (Continued)*

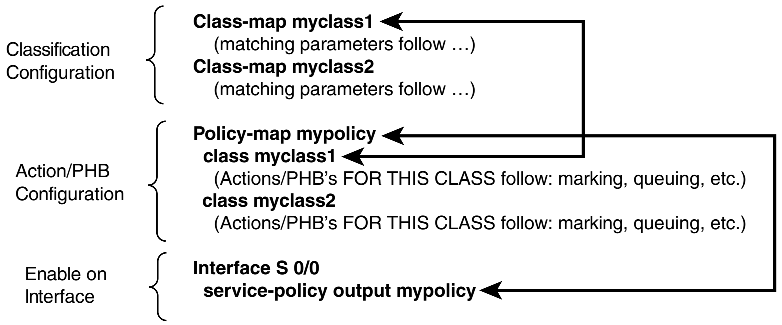
Command	Mode and Function
match access-group {*access-group* | **name** *access-group-name*}	ACL
match source-address mac *address-destination*	Source MAC address
match ip precedence *ip-precedence-value* [*ip-precedence-value ip-precedence-value ip-precedence-value*]	IP precedence
match mpls experimental *number*	MPLS Experimental
match cos *cos-value* [*cos-value cos-value cos-value*]	CoS
match destination-address mac *address*	Destination MAC address
match input-interface *interface-name*	Input interface
match ip dscp *ip-dscp-value* [*ip-dscp-value ip-dscp-value ip-dscp-value ip-dscp-value ip-dscp-value ip-dscp-value ip-dscp-value*]	IP DSCP
match ip rtp *starting-port-number port-range*	RTP's UDP port-number range
match qos-group *qos-group-value*	QoS group
match protocol *protocol-name*	NBAR protocol types
match protocol citrix [**app** *application-name-string*]	NBAR Citrix applications
match protocol http [**url** *url-string* | **host** *hostname-string* | **mime** *MIME-type*]	Host name and URL string
match any	All packets
policy-map *policy-map-name*	Global config; names a policy, which is a set of actions to perform.
class *class-name*	Policy-map subcommand; identifies which packets on which to perform some action by referring to the classification logic in a class map
set	For the class, marks (sets) particular QoS fields
set ip precedence *ip-precedence-value*	IP precedence
set ip dscp *ip-dscp-value*	IP DSCP
set cos *cos-value*	CoS
set ip qos-group *group-id*	QoS group
set atm-clp	ATM CLP bit
Set fr-de	Frame Relay DE bit

Table 3-24 *Exec Command Reference for Class-Based Marking*

Command	Function
show policy-map *policy-map-name*	Lists configuration information about all MQC-based QoS tools
show policy-map *interface-spec* [***input*** | ***output***] [**class** *class-name*]	Lists statistical information about the behavior of all MQC-based QoS tools

Figure 3-24 shows the general flow of MQC commands.

Figure 3-24 *MQC Commands and Their Correlation*



Classification Configuration
Class-map myclass1
(matching parameters follow …)
Class-map myclass2
(matching parameters follow …)
Action/PHB Configuration
Policy-map mypolicy
class myclass1
(Actions/PHB's FOR THIS CLASS follow: marking, queuing, etc.)
class myclass2
(Actions/PHB's FOR THIS CLASS follow: marking, queuing, etc.)
Enable on Interface
Interface S 0/0
service-policy output mypolicy

Tables 3-25 and 3-26 list the NBAR configuration and exec commands, respectively.

Table 3-25 *Configuration Command Reference for NBAR*

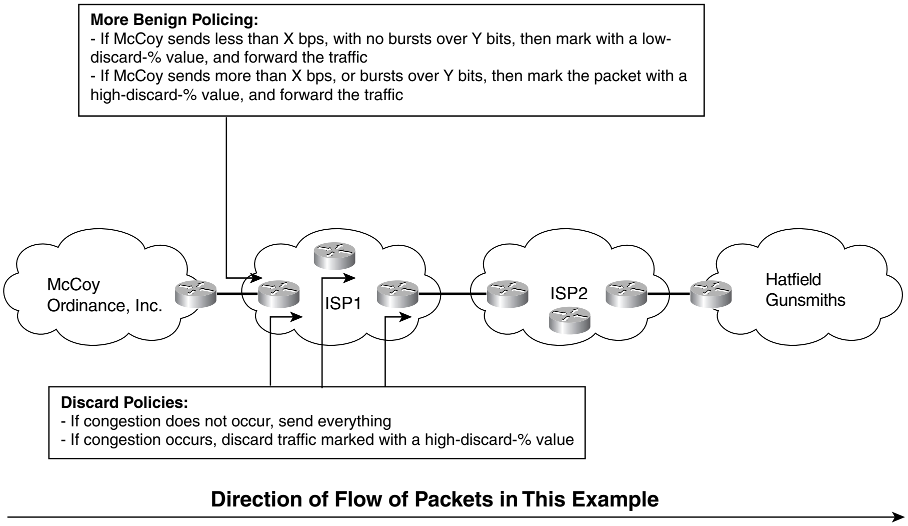
Command	Mode and Function
ip nbar protocol-discovery	Interface mode; enables NBAR for traffic entering the interface.
ip nbar port-map *protocol-name* [**tcp** | **udp**] *port-number*	Global; tells NBAR to search for a protocol using a different port number than the well-known port. Also defines ports to be used by custom packet description language modules (PDLM).
ip nbar pdlm *pdlm-name*	Global; extends the list of protocols recognized by NBAR by adding additional PDLMs.

You can use CAR for policing, but instead of discarding packets, CAR can instead mark nonconforming packets with a value that increases the packets' chances of being discarded when congestion occurs, as seen in Figure 3-25.

Table 3-26 *Exec Command Reference for NBAR*

Command	Function
show ip nbar protocol-discovery [**interface** *interface-spec*] [**stats** {**byte-count** | **bit-rate** | **packet-count**}][{**protocol** *protocol-name* | **top-n** *number*}]	Lists information about statistics for the discovered protocols. Statistics can be listed by interface, by protocol, or for just the top *n* protocols by volume.
show ip nbar port-map [*protocol-name*]	Lists the current ports in use by the discovered protocols.

Figure 3-25 *Policing: Excess Traffic Marked with Lower Value*



More Benign Policing:
- If McCoy sends less than X bps, with no bursts over Y bits, then mark with a low-discard-% value, and forward the traffic
- If McCoy sends more than X bps, or bursts over Y bits, then mark the packet with a high-discard-% value, and forward the traffic

Discard Policies:
- If congestion does not occur, send everything
- If congestion occurs, discard traffic marked with a high-discard-% value

Tables 3-27, 3-28, and 3-29 list the pertinent CAR configuration and exec commands, respectively.

Table 3-27 *Configuration Command Reference for CAR*

Command	Mode and Function
rate-limit {**input** | **output**} [**access-group** [**rate-limit**] *acl-index*] *bps burst-normal burst-max* **conform-action** *conform-action* **exceed-action** *exceed-action*	Interface mode; configures classification, marking, policing, and enabling CAR on the interface
access-list rate-limit *acl-index* {*precedence* | *mac-address* | *exp* **mask** *mask*}	Global mode; creates a CAR ACL, which can match IP precedence, MAC addresses, and MPLS Experimental bits

Table 3-28 *Possible Actions with CAR* **rate-limit** *Command*

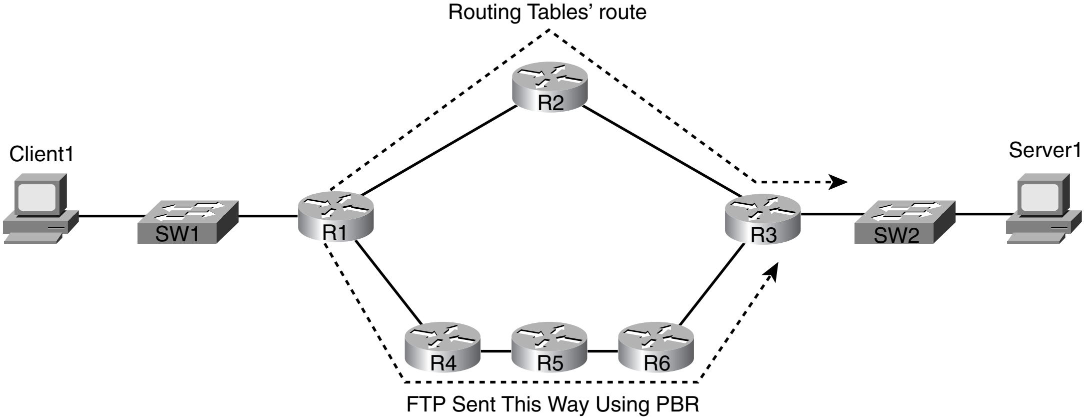
rate-limit Conform and Exceed Options	Function
Continue	Evaluates the next **rate-limit** command
Drop	Drops the packet
set-dscp-continue	Sets the differentiated services code point (DSCP) (0–63) and evaluates the next **rate-limit** command
set-dscp-transmit	Sets the DSCP and transmits the packet
set-mpls-exp-continue	Sets the MPLS Experimental bits (0–7) and evaluates the next **rate-limit** command
set-mpls-exp-transmit	Sets the MPLS Experimental bits (0–7) and sends the packet
set-prec-continue	Sets the IP precedence (0–7) and evaluates the next **rate-limit** command
set-prec-transmit	Sets the IP precedence (0–7) and sends the packet
set-qos-continue	Sets the QoS group ID (1–99) and evaluates the next **rate-limit** command
set-qos-transmit	Sets the QoS group ID (1–99) and sends the packet
Transmit	Sends the packet

Table 3-29 *Exec Command Reference for CAR*

Command	Function
show interfaces [*interface-type interface-number*] **rate-limit**	Displays CAR statistics on the interface specified, or on all interfaces if the interface is not specified
show access-lists rate-limit [*acl-index*]	Lists information about the configuration of rate-limit ACLs

Policy-based routing (PBR) enables you to route a packet based on some other information besides the destination IP address. Figure 3-26 shows a simple example, where FTP traffic is directed over the longer path in the network.

Figure 3-26 *PBR: FTP Traffic Routed over Longer Path*



Tables 3-30 and 3-31 list the pertinent PBR configuration and exec commands, respectively.

Table 3-30 *Configuration Command Reference for PBR*

Command	Mode and Function
ip local policy route-map *map-tag*	Global; specifies that packets generated by this router should be candidates for policy routing
ip policy route-map *map-tag*	Interface subcommand; refers to a route map, which in turn classifies packets and specifies actions; actions include specifying a different route, and setting IP precedence
route-map *map-tag* **[permit | deny]** [*sequence-number*]	Global command; creates a route map entry
match ip address {*access-list-number* | *access-list-name*} [... *access-list-number* | ... *access-list-name*]	Route-map subcommand; used to match IP packets based on parameters that can be matched with an IP ACL
match length *minimum-length maximum-length*	Route-map subcommand; used to match IP packets based on their length
set ip precedence *number* | *name*	Route-map subcommand; sets IP precedence value using the decimal number of name

continues

Table 3-30 *Configuration Command Reference for PBR (Continued)*

Command	Mode and Function
set ip next-hop *ip-address* **[**...*ip-address***]**	Route-map subcommand; defines the IP address(es) of the next-hop router(s) to be used for forwarding packets that match this route map entry
ip route-cache policy	Global command; enables fast switching of PBR-routed packets

Note: Not all PBR-related commands are shown in this table, but commands specifically related to marking are shown.

Table 3-31 *Exec Command Reference for PBR Marking*

Command	Function
show ip policy	Lists configured PBR details, and statistics for numbers of packets matched by each clause.
show route-map	Lists statistical information about packets matched with a route map. PBR uses route maps to classify and mark traffic.

Q&A

As mentioned in the Introduction, you have two choices for review questions. The questions that follow next give you a more difficult challenge than the exam itself by using an open-ended question format. By reviewing now with this more difficult question format, you can exercise your memory better, and prove your conceptual and factual knowledge of this chapter. You can find the answers to these questions in Appendix A.

The second option for practice questions is to use the CD-ROM included with this book. It includes a testing engine and more than 200 multiple-choice questions. You should use this CD-ROM nearer to the end of your preparation, for practice with the actual exam format. You can even customize the CD-ROM exam to include, or not include, the topics that are only on the CCIP QoS.

1. Describe the difference between classification and marking.
2. Describe, in general, how a queuing feature could take advantage of the work performed by a classification and marking feature.
3. Characterize what must be true before the CoS field may be useful for marking packets.
4. Most other QoS tools, besides classification and marking tools, also have a classification feature. Describe the advantage of classification, in terms of overall QoS design and policies, and explain why classification and marking is useful, in spite of the fact that other tools also classify the traffic.
5. Which of the following classification and marking tools can classify based on the contents of an HTTP URL: class-based marking (CB marking), policy-based routing (PBR), committed access rate (CAR), network-based application recognition (NBAR), or dial peers?
6. Describe the differences between IP extended ACLs as compared with NBAR for matching TCP and UDP port numbers.
7. Which of the following classification and marking tools can classify based on the outgoing interface of the route used for a packet: class-based marking (CB marking), policy-based routing (PBR), committed access rate (CAR), network-based application recognition (NBAR), or dial peers?
8. Which of the following classification and marking tools can classify based on the destination TCP port number of a packet, without using an IP ACL: class-based marking (CB marking), policy-based routing (PBR), committed access rate (CAR), network-based application recognition (NBAR), or dial peers?

9 Which of the following classification and marking tools can classify based on the DSCP, without using an IP ACL: class-based marking (CB marking), policy-based routing (PBR), committed access rate (CAR), network-based application recognition (NBAR), or dial peers?

10 Which of the following classification and marking tools can classify based on either the source or destination MAC address: class-based marking (CB marking), policy-based routing (PBR), committed access rate (CAR), network-based application recognition (NBAR), or dial peers?

11 Which of the following classification and marking tools can classify based on the even numbered UDP ports used for RTP traffic, with or without using an IP ACL: class-based marking (CB marking), policy-based routing (PBR), committed access rate (CAR), network-based application recognition (NBAR), or dial-peers?

12 Which of the following QoS marking fields are carried inside an 802.1Q header: QoS, CoS, DE, ToS byte, User Priority, ToS bits, CLP, Precedence, QoS Group, DSCP, MPLS Experimental, or DS?

13 Which of the following QoS marking fields are carried inside an IP header: QoS, CoS, DE, ToS byte, User Priority, ToS bits, CLP, Precedence, QoS Group, DSCP, MPLS Experimental, or DS?

14 Which of the following QoS marking fields are never marked inside a frame that exits a router: QoS, CoS, DE, ToS byte, User Priority, ToS bits, CLP, Precedence, QoS Group, DSCP, MPLS Experimental, or DS?

15 Describe the goal of marking near the edge of a network in light of the meaning of the term "trust boundary."

16 Define the meaning of MQC, and spell out what the acronym abbreviates.

17 What configuration command lists the classification details when configuring CB marking? What configuration mode must you use to configure the command? What commands must you issue to place the configuration mode user into that mode?

18 What configuration command lists the marking details when configuring CB marking? What configuration mode must you use to configure the command? What commands must you issue to place the configuration mode user into that mode?

19 What configuration command enables CB marking? What configuration mode must you use to configure the command? What commands must you issue to place the configuration mode user into that mode?

20 What configuration command lists the classification details when configuring CAR? What configuration mode must you use to configure the command? What commands must you issue to place the configuration mode user into that mode?

21 What configuration command lists the marking details when configuring CAR? What configuration mode must you use to configure the command? What commands must you issue to place the configuration mode user into that mode?

22 What configuration command enables CAR? What configuration mode must you use to configure the command? What commands must you issue to place the configuration mode user into that mode?

23 What configuration command lists the classification details when configuring PBR? What configuration mode must you use to configure the command? What commands must you issue to place the configuration mode user into that mode?

24 What configuration command lists the marking details when configuring PBR? What configuration mode must you use to configure the command? What commands must you issue to place the configuration mode user into that mode?

25 What configuration command enables PBR? What configuration mode must you use to configure the command? What commands must you issue to place the configuration mode user into that mode?

26 Describe the process dial peers use to classify and mark traffic.

27 What configuration command(s) lists the marking details when configuring dial peers? What configuration mode must you use to configure the command? What commands must you issue to place the configuration mode user into that mode?

28 What QoS values can a dial peer mark?