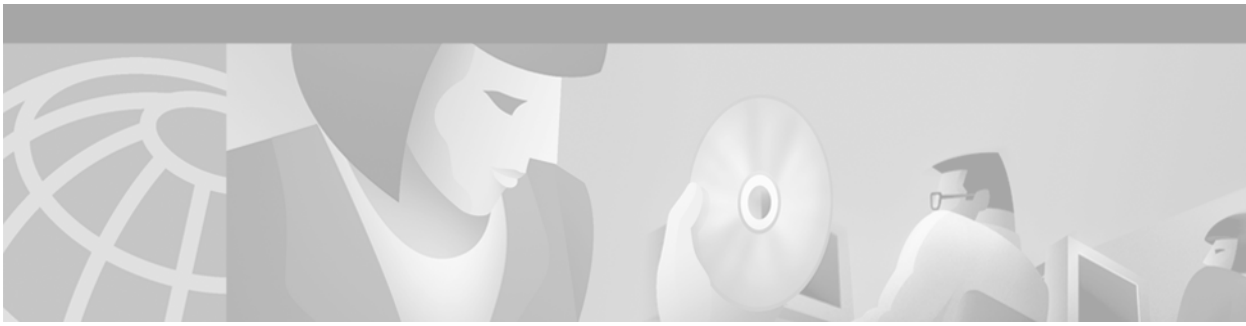


# Classification Overview

Classification entails using a traffic descriptor to categorize a packet within a specific group to define that packet and make it accessible for QoS handling on the network. Using packet classification, you can partition network traffic into multiple priority levels or classes of service. When traffic descriptors are used to classify traffic, the source agrees to adhere to the contracted terms and the network promises a quality of service. Traffic policers, such as the Traffic Policing feature and the rate-limiting feature of committed access rate (CAR), and traffic shapers, such as Generic Traffic Shaping (GTS), Distributed Traffic Shaping (DTS) and Frame Relay Traffic Shaping (FRTS), use traffic descriptor of a packet—that is, its classification—to ensure adherence to the contract.

Packet classification is pivotal to policy techniques that select packets traversing a network element or a particular interface for different types of QoS service. For example, you can use classification to mark certain packets for IP Precedence and you can identify others as belonging to a Resource Reservation Protocol (RSVP) flow.

Methods of classification were once limited to use of the contents of the packet header. Current methods of marking a packet with its classification allow you to set information in the Layer 2, 3, or 4 headers, or even by setting information within the payload of a packet. Criteria for classification of a group might be as broad as “traffic destined for subnetwork X” or as narrow as a single flow.

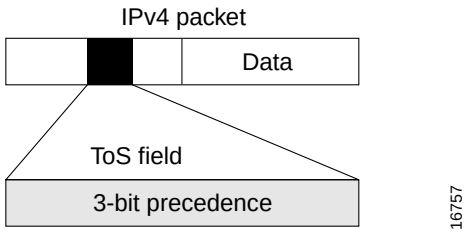
This chapter explains IP Precedence, then it gives a brief description of the kinds of traffic classification provided by the Cisco IOS QoS features. It discusses features described in the following sections:

- Policy-Based Routing
- QoS Policy Propagation via Border Gateway Protocol
- Committed Access Rate
- Class-Based Packet Marking
- QoS for Virtual Private Networks
- Network-Based Application Recognition

# About IP Precedence

Use of IP Precedence allows you to specify the class of service (CoS) for a packet. You use the three precedence bits in the type of service (ToS) field of the IP version 4 (IPv4) header for this purpose. Figure 2 shows the ToS field.

***Figure 2 IPv4 Packet Type of Service Field***




Using the ToS bits, you can define up to six classes of service. Other features configured throughout the network can then use these bits to determine how to treat the packet in regard to the ToS to grant it. These other QoS features can assign appropriate traffic-handling policies including congestion management strategy and bandwidth allocation. For example, although IP Precedence is not a queueing method, queueing methods such as weighted fair queueing (WFQ) and Weighted Random Early Detection (WRED) can use the IP Precedence setting of the packet to prioritize traffic.

By setting precedence levels on incoming traffic and using them in combination with the Cisco IOS QoS queueing features, you can create differentiated service. You can use features such as policy-based routing (PBR) and CAR to set precedence based on extended access list classification. These features afford considerable flexibility for precedence assignment. For example, you can assign precedence based on application or user, or by destination and source subnetwork.

So that each subsequent network element can provide service based on the determined policy, IP Precedence is usually deployed as close to the edge of the network or the administrative domain as possible. You can think of IP Precedence as an edge function that allows core, or backbone, QoS features such as WRED to forward traffic based on CoS. IP Precedence can also be set in the host or network client, but this setting can be overridden by policy within the network.

The following QoS features can use the IP Precedence field to determine how traffic is treated:

- Distributed WRED (DWRED)
- WFQ
- CAR

## How the IP Precedence Bits Are Used to Classify Packets

You use the three IP Precedence bits in the ToS field of the IP header to specify CoS assignment for each packet. You can partition traffic into up to six classes—the remaining two are reserved for internal network use—and then use policy maps and extended access lists to define network policies in terms of congestion handling and bandwidth allocation for each class.

For historical reasons, each precedence corresponds to a name. These names, which continue to evolve, are defined in the RFC 791 document. Table 3 lists the numbers and their corresponding names, from least to most important.

*Table 3 IP Precedence Values*

| Number | Name |
|---|---|
| 0 | routine |
| 1 | priority |
| 2 | immediate |
| 3 | flash |
| 4 | flash-override |
| 5 | critical |
| 6 | internet |
| 7 | network |

However, the IP Precedence feature allows you considerable flexibility for precedence assignment. That is, you can define your own classification mechanism. For example, you might want to assign precedence based on application or access router.

**Note** IP Precedence bit settings 6 and 7 are reserved for network control information such as routing updates.

# Setting or Changing the IP Precedence Value

By default, the Cisco IOS software leaves the IP Precedence value untouched, preserving the precedence value set in the header, allowing all internal network devices to provide service based on the IP Precedence setting. This policy follows the standard approach stipulating that network traffic should be sorted into various types of service at the basic perimeter of the network and that those types of service should be implemented in the core of the network. Routers in the core of the network can then use the precedence bits, for example, to determine the order of transmission, the likelihood of packet drop, and so on.

Because traffic coming into your network can have precedence set by outside devices, we recommend you reset the precedence for all traffic entering your network. By controlling IP Precedence settings, you prohibit users that have already set the IP precedence from acquiring better service for their traffic simply by setting a high precedence for all of their packets.

You can use any of the features described in the following sections to set the IP precedence in packets:

- Policy-Based Routing
- QoS Policy Propagation via Border Gateway Protocol
- Committed Access Rate

As mentioned previously, after a packet has been classified, you can use other QoS features such as CAR and WRED to specify and enforce business policies to fit your business model.

# Policy-Based Routing

PBR gives you a flexible means of routing packets by allowing you to configure a defined policy for traffic flows, lessening reliance on routes derived from routing protocols. To this end, PBR gives you more control over routing by extending and complementing the existing mechanisms provided by routing protocols. PBR allows you to set the IP precedence. It also allows you to specify a path for certain traffic, such as priority traffic over a high-cost link.

You can set up PBR as a way to route packets based on configured policies. For example, you can implement routing policies to allow or deny paths based on the identity of a particular end system, an application protocol, or the size of packets.

PBR allows you to perform the following tasks:

- Classify traffic based on extended access list criteria. Access lists, then, establish the match criteria.
- Set IP Precedence bits, giving the network the ability to enable differentiated classes of service.
- Route packets to specific traffic-engineered paths; you might need to route them to allow a specific QoS through the network.

Policies can be based on IP address, port numbers, protocols, or size of packets. For a simple policy, you can use any one of these descriptors; for a complicated policy, you can use all of them.

For example, classification of traffic through PBR allows you to identify traffic for different classes of service at the edge of the network and then implement QoS defined for each CoS in the core of the network using priority queueing (PQ), custom queueing (CQ), or WFQ techniques. This process obviates the need to classify traffic explicitly at each WAN interface in the core-backbone network.

For information on how to configure policy-based routing, see the chapter "Configuring Policy-Based Routing" in this book.

## How It Works

All packets received on an interface with PBR enabled are passed through enhanced packet filters known as route maps. The route maps used by PBR dictate the policy, determining to where the packets are forwarded.

Route maps are composed of statements. The route map statements can be marked as permit or deny, and they are interpreted in the following ways:

- If the packets do not match any route map statements, then all the set clauses are applied.
- If a statement is marked as deny, the packets meeting the match criteria are sent back through the normal forwarding channels and destination-based routing is performed.
- If the statement is marked as permit and the packets do not match any route map statements, the packets are sent back through the normal forwarding channels and destination-based routing is performed.

You specify PBR on the interface that receives the packet, not on the interface from which the packet is sent.

## When Should You Use Policy-Based Routing?

You might enable PBR if you want certain packets to be routed some way other than the obvious shortest path. For example, PBR can be used to provide the following functionality:

- equal access
- protocol-sensitive routing
- source-sensitive routing
- routing based on interactive versus batch traffic
- routing based on dedicated links

Some applications or traffic can benefit from QoS-specific routing; for example, you could transfer stock records to a corporate office on a higher-bandwidth, higher-cost link for a short time while sending routine application data such as e-mail over a lower-bandwidth, lower-cost link.

# QoS Policy Propagation via Border Gateway Protocol

The Border Gateway Protocol (BGP) is an interdomain routing protocol that exchanges reachability information with other BGP systems. It is defined by RFC 1163.

The Policy Propagation via BGP feature allows you to classify packets based on the following:

- Access lists.
- BGP community lists. A community is a group of destinations that share some common attribute. You use community lists to create groups of communities to use in a match clause of a route map. As with access lists, a series of community lists can be created.
- BGP autonomous system paths. An autonomous system path is a collection of networks under a common administration sharing a common routing strategy. BGP carries the autonomous system path in its routing updates. You can filter routing updates by specifying an access list on both incoming and outbound updates based on the BGP autonomous system path.
- IP Precedence. See the section "About IP Precedence" earlier in this chapter.
- Source and destination address lookup. You can specify whether the IP Precedence level is obtained from the source (input) address or destination (output) address entry in the route table.

After a packet has been classified using BGP, you can use other QoS features such as CAR and WRED to specify and enforce business policies to fit your business model.

BGP Policy Propagation leverages BGP to distribute QoS policy to remote routers in your network. It allows ingress routers to prioritize incoming traffic.

## Restrictions

For the Policy Propagation via BGP feature to work, you must enable BGP and Cisco Express Forwarding (CEF)/distributed CEF (dCEF) on the router.

Subinterfaces on an ATM interface that has the **bgp-policy** command enabled must use CEF mode because dCEF is not supported. (Note that dCEF uses the Versatile Interface Processor (VIP) rather than the Route Switch Processor (RSP) to perform forwarding functions.)

For information on how to configure Policy Propagation via BGP, see the chapter "Configuring QoS Policy Propagation via Border Gateway Protocol" in this book.

# Committed Access Rate

CAR is a multifaceted feature that implements both classification services and policing through rate limiting. This section describes its classification capability. For information on its rate limiting features, see the chapter "Policing and Shaping Overview" in this book.

You can use the classification services of CAR to set the IP precedence for packets entering the network. This capability of CAR allows you to partition your network into multiple priority levels or classes of service. Networking devices within your network can then use the adjusted IP precedence to determine how to treat the traffic. For example, VIP-distributed WRED uses the IP precedence to determine the probability of whether a packet will be dropped.

As discussed in the section "About IP Precedence," you can use the three precedence bits in the ToS field of the IP header to define up to six classes of service.

You can classify packets using policies based on physical port, source or destination IP or MAC address, application port, IP protocol type, or other criteria specifiable by access lists or extended access lists. You can classify packets by categories external to the network, for example, by a customer. After a packet has been classified, a network can either accept or override and reclassify the packet according to a specified policy. CAR includes commands you can use to classify and reclassify packets.

CAR is supported on the majority of Cisco routers. Additionally, distributed CAR is supported on Cisco 7000 series routers with an RSP7000 interface processor or Cisco 7500 series routers with a VIP-based VIP2-40 or greater interface processor. A VIP2-50 interface processor is strongly recommended when the aggregate line rate of the port adapters on the VIP is greater than DS3. A VIP2-50 interface processor is required for OC-3 rates.

For information on how to configure CAR, see the chapter "Configuring Committed Access Rate" in this book.

# Class-Based Packet Marking

The Class-Based Packet Marking feature provides users with a means for efficient packet marking by which users can differentiate packets based on the designated markings.

The Class-Based Packet Marking feature allows users to perform the following tasks:

- Mark packets by setting the IP Precedence bits or the IP differentiated services code point (DSCP) in the IP ToS byte.
- Mark packets by setting the Layer 2 CoS value.
- Associate a local QoS group value with a packet.
- Set the Cell Loss Priority (CLP) bit setting in the ATM header of a packet from 0 to 1.

The Class-Based Packet Marking feature supports the following RFCs:

- RFC 2474, *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*
- RFC 2475, *An Architecture for Differentiated Services Framework*
- RFC 2597, *Assured Forwarding PHB*
- RFC 2598, *An Expedited Forwarding PHB*

For information on how to configure Class-Based Packet Marking, see the chapter "Configuring Class-Based Packet Marking" in this book.

## IP Precedence and IP DSCP Marking

Associating a packet with an IP Precedence or IP DSCP marking allows users to classify traffic based on IP Precedence and IP DSCP value, depending on which value is marked. These markings can be used to identify traffic within the network, and other interfaces can match traffic based on the IP Precedence or DSCP markings.

IP Precedence and DSCP markings are used to decide how packets should be treated in Weighted Random Early Detection (WRED).

The IP DSCP value is the first 6 bits in the ToS byte, while the IP Precedence value is the first 3 bits in the ToS value. The IP Precedence value is actually part of the IP DSCP value. Therefore, both values cannot be set simultaneously. If both values are set simultaneously, the packet is marked with the IP DSCP value.

If you need to mark packets in your network and all of your devices support IP DSCP marking, use the IP DSCP marking to mark your packets, since the IP DSCP markings provide more packet marking options. If marking by IP DSCP is undesirable, however, or if you are unsure if the devices in your network support IP DSCP values, use the IP precedence value to mark your packets. The IP precedence value is likely supported by all devices in the network.

A user can set up to 8 different IP precedence markings and 64 different IP DSCP markings.

## CoS Value Marking

Associating a packet with a local CoS value allows users to associate a Layer 2 CoS value with a packet. The value can then be used to classify packets based on user-defined requirements. Layer 2 to Layer 3 mapping can also be configured by matching on the CoS value, because switches already have the capability to match and set CoS values. If a packet that needs to be marked to differentiate user-defined QoS services is leaving a router and entering a switch, the router should set the CoS value of the packet, because the switch can process the Layer 2 CoS header marking.

A user can set up to 8 different CoS markings.

## QoS Group Value Marking

Associating a packet with a local QoS group allows users to associate a group ID with a packet. The group ID can be used to classify packets into QoS groups based on prefix, autonomous system, or community string. This QoS group marking can only be used to classify traffic within a router, and cannot be used to mark packets leaving the router.

A user can set up to 100 different QoS group markings.

## ATM CLP Bit Setting

Changing the CLP bit setting in the ATM header of a cell provides a method of controlling the discarding of cells in congested ATM environments. A CLP bit contains two settings: 0 or 1. Cells with a CLP bit setting of 1 are discarded before cells with a CLP bit setting of 0 when congestion occurs.

# Support for ATM Virtual Circuits

With the Class-Based Packet Marking feature, packet marking is supported on ATM virtual circuits (VCs). Users can configure the marking action in the same policy map where they configure the queueing actions, on a per-VC basis. Previously, packet marking was supported on the main interface or subinterface configuration level.

# Additional Statistics

With the Class-Based Packet Marking feature, output from the **show policy-map interface** command is enhanced to provide additional statistics such as the incoming traffic rate, the dropped packet rate, the number of matched packets, and the number of matched bytes for traffic classes that are attached to the specified interface.

The Class-Based Packet Marking feature is configured with the Modular QoS CLI. For additional information on the Modular QoS CLI, see the chapter "Modular Quality of Service Command-Line Interface Overview" in this book.

# Benefits

## Packet Marking Through IP Precedence, QoS Group, CoS Value, and IP DSCP Value Setting

Packet marking allows you to partition your network into multiple priority levels or classes of service, as follows:

- Use QoS packet marking to set the IP Precedence or IP DSCP values for packets entering the network. Networking devices within your network can then use the newly marked IP Precedence values to determine how the traffic should be treated. For example, class-based WRED uses IP Precedence values to determine the probability that a packet will be dropped. In addition, voice packets can be marked with a particular color (precedence/DSCP). Low latency queueing (LLQ) can then be configured to put all packets of that mark into the priority queue.
- Use QoS packet marking to assign packets to a QoS group. The router uses the QoS group to determine how to prioritize packets for transmission.
- Use CoS packet marking to assign packets to set the priority value of 802.1p/Inter-Switch Link (ISL) packets. The router uses the CoS value to determine how to prioritize packets for transmission and can use this marking to perform Layer 2 to Layer 3 mapping.

## Improved Bandwidth Management on ATM Networks

The ability to set the ATM CLP bit allows users to extend their IP QoS policies into an ATM network. As congestion occurs in the ATM network, cells with the CLP bit set are more likely to be dropped, resulting in improved network performance for higher priority traffic and applications.

## Restrictions

The following restrictions apply to the Class-Based Packet Marking feature:

- It can mark only packets traveling on CEF switching paths. In order to use the Class-Based Packet Marking feature, you must configure CEF on both the interface receiving the packet and the interface sending the packet.
- It can be configured on an interface, a subinterface, or an ATM permanent virtual circuit (PVC), but is not supported on the following interface types:
  - Fast EtherChannel
  - Tunnel
  - PRI
  - ATM switched virtual circuit (SVC)
  - Frame Relay data-link connection identifier (DLCI)
  - Any interface that does not support CEF
- Before modifying the encapsulation type from IEEE 802.1 Q to ISL, or vice versa, on a subinterface, detach the policy map from the subinterface. After changing the encapsulation type, reattach the policy map.
- To use the **set atm-clp** command available with the Class-Based Packet Marking feature, you must have one of the following adapters: the Enhanced ATM Port Adapter (PA-A3), the ATM Inverse Multiplexer over ATM Port Adapter with 8 T1 Ports (PA-A3-8T1IMA), or the ATM Inverse Multiplexer over ATM Port Adapter with 8 E1 Ports (PA-A3-8E1IMA). Therefore, the **set atm-clp** command is not supported on any platform that does not support these adapters. For more information, refer to the documentation for your specific router.
- A policy map containing the **set atm-clp** command can be attached as an output policy only. The **set atm-clp** command does not support packets that originate from the router.

## Prerequisites

CEF must be configured on the interface before the Class-Based Packet Marking feature can be used.

For additional information on CEF, refer to the *Cisco IOS Switching Services Configuration Guide*.

# QoS for Virtual Private Networks

The QoS for Virtual Private Networks (VPNs) feature is designed for tunnel interfaces. When the feature is enabled, the QoS features on the output interface classify packets before encryption, allowing traffic flows to be adjusted in congested environments. The result is more effective packet tunneling.

The QoS for VPNs feature provides a solution for making Cisco IOS QoS services operate in conjunction with tunneling and encryption on an interface. Cisco IOS software can classify packets and apply the appropriate QoS service before the data is encrypted and tunneled. The QoS for VPN feature allows users to look inside the packet so that packet classification can be done based on original port numbers and based on source and destination IP addresses. This allows the service provider to treat mission-critical or multiservice traffic with higher priority across its network.

To use this feature, the system must be able to configure QoS features.

## Restrictions

Interfaces running cascading QoS features, such as Generic Traffic Shaping (GTS) or CQ, are required to have QoS for VPNs enabled or disabled on all cascading features. If the QoS for VPNs feature is enabled on one cascading feature, the QoS for VPNs feature must be enabled on all cascading features. Similarly, if the QoS for VPNs feature is disabled on one cascading feature, the QoS for VPNs feature must be disabled on all cascading features.

For information on how to configure the QoS for VPNs feature, see the chapter "Configuring QoS for Virtual Private Networks" in this book.

# Network-Based Application Recognition

The Network-Based Application Recognition (NBAR) feature adds intelligent network classification to network infrastructures. NBAR is a classification engine that recognizes a wide variety of applications, including web-based and other difficult-to-classify protocols that utilize dynamic TCP/User Datagram Ports (UDP) port assignments. When an application is recognized and classified by NBAR, a network can invoke services for that specific application. NBAR ensures that network bandwidth is used efficiently by working with QoS features to provide the following features:

- Guaranteed bandwidth
- Bandwidth limits
- Traffic shaping
- Packet coloring

NBAR introduces the following new classification features:

- Classification of applications that dynamically assign TCP/UDP port numbers
- Classification of HTTP traffic by URL, HOST, or Multipurpose Internet Mail Extension (MIME) type
- Classification of Citrix Independent Computing Architecture (ICA) traffic by application name
- Classification of application traffic using subport information

NBAR can also classify static port protocols. Although access control lists (ACLs) can also be used for this purpose, NBAR is easier to configure and can provide classification statistics that are not available when ACLs are used.

NBAR provides a special Protocol Discovery feature that determines which application protocols are traversing a network at any given time. The Protocol Discovery feature captures key statistics associated with each protocol in a network. These statistics can be used to define traffic classes and QoS policies for each traffic class.

NBAR addresses IP QoS classification requirements by classifying application-level protocols so that QoS policies can be applied to the classified traffic. NBAR addresses the ongoing need to extend the classification engine for the many existing and emerging application protocols by providing an extensible Packet Description Language (PDL). NBAR can determine which protocols and applications are currently running on a network so that an appropriate QoS policy can be created based upon the current traffic mix and application requirements.

NBAR can now perform subport classification of HTTP traffic by host name in addition to classification by MIME-type or URL. This ability enables users to classify HTTP traffic by web server names. With URL matching, only the portion of the URL following the host name can be specified for a match. To perform a match on the host name portion of the URL, use the new HOST matching criterion. For example, a HOST match on http://www.cisco.com/latest/whatsnew.html will classify all traffic from the web server www.cisco.com, whereas a URL match can be performed on the /latest/whatsnew.html portion of the URL.

NBAR supports the following RFCs:

- RFC 742, *NAME/FINGER Protocol*
- RFC 759, *Internet Message Protocol*
- RFC 792, *Internet Control Message Protocol*
- RFC 793, *Transmission Control Protocol*
- RFC 821, *Simple Mail Transfer Protocol*
- RFC 827, *Exterior Gateway Protocol*
- RFC 854, *Telnet Protocol Specification*
- RFC 888, *STUB Exterior Gateway Protocol*
- RFC 904, *Exterior Gateway Protocol formal specification.*
- RFC 951, *Bootstrap Protocol*
- RFC 959, *File Transfer Protocol*
- RFC 977, *Network News Transfer Protocol*
- RFC 1001, *Protocol Standard for a NetBIOS Service on a TCP/UDP Transport: Concepts and Methods*
- RFC 1002, *Protocol Standard for a NetBIOS Service on a TCP/UDP Transport: Detailed Specifications*
- RFC 1057, *RPC: Remote Procedure Call*
- RFC 1094, *NFS: Network File System Protocol Specification*
- RFC 1112, *Host Extensions for IP multicasting*
- RFC 1157, *Simple Network Management Protocol*
- RFC 1282, *BSD Rlogin*
- RFC 1288, *The Finger User Information Protocol*
- RFC 1305, *Network Time Protocol*
- RFC 1350, *The TFTP Protocol (Revision 2)*
- RFC 1436, *The Internet Gopher Protocol*
- RFC 1459, *Internet Relay Chat Protocol*
- RFC 1510, *The Kerberos Network Authentication Service*

- RFC 1542, *Clarifications and Extensions for the Bootstrap Protocol*
- RFC 1579, *Firewall-Friendly FTP*
- RFC 1583, *OSPF Version 2*
- RFC 1657, *Definitions of Managed Objects for the Fourth Version of the Border Gateway Protocol*
- RFC 1701, *Generic Routing Encapsulation*
- RFC 1730, *Internet Message Access Protocol - Version 4*
- RFC 1771, *A Border Gateway Protocol 4 (BGP-4)*
- RFC 1777, *Lightweight Directory Access Protocol*
- RFC 1831, *RPC: Remote Procedure Call Protocol Specification Version 2*
- RFC 1928, *SOCKS Protocol Version 5*
- RFC 1939, *Post Office Protocol - Version 3*
- RFC 1945, *Hypertext Transfer Protocol -- HTTP/1.0*
- RFC 1964, *The Kerberos Version 5 GSS-API Mechanism*
- RFC 2060, *Internet Message Access Protocol - Version 4rev1*
- RFC 2068, *Hypertext Transfer Protocol -- HTTP/1.1*
- RFC 2131, *Dynamic Host Configuration Protocol*
- RFC 2205, *Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification*
- RFC 2236, *Internet Group Management Protocol, Version 2*
- RFC 2251, *Lightweight Directory Access Protocol (v3)*
- RFC 2252, *Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions*
- RFC 2253, *Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names*
- RFC 2326, *Real Time Streaming Protocol (RTSP)*
- RFC 2401, *Security Architecture for the Internet Protocol*
- RFC 2406, *IP Encapsulating Security Payload*
- RFC 2453, *RIP Version 2*
- RFC 2616, *Hypertext Transfer Protocol -- HTTP/1.1*

NBAR supports the following RFCs:

- 0009, *File Transfer Protocol (FTP)*
- 0013, *Domain Names - Concepts and Facilities*
- 0033, *The TFTP Protocol (Revision 2)*
- 0034, *Routing Information Protocol*
- 0053, *Post Office Protocol - Version 3*
- 0056, *RIP Version 2*

For information on how to configure NBAR, see the chapter "Configuring Network-Based Application Recognition" in this book.

You must enable CEF before you configure NBAR. For more information on CEF, refer to the *Cisco IOS Switching Services Configuration Guide*.

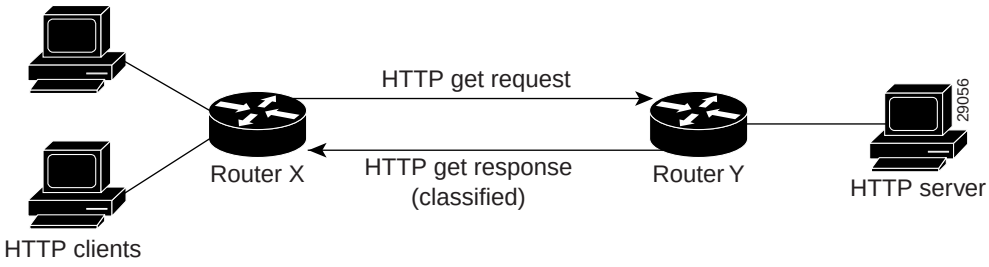
# Classification of HTTP by URL, HOST, or MIME

NBAR can classify application traffic by looking beyond the TCP/UDP port numbers of a packet. This ability is called subport classification. NBAR looks into the TCP/UDP payload itself and classifies packets on content within the payload such as transaction identifier, message type, or other similar data.

Classification of HTTP by URL, HOST, or MIME type is an example of subport classification. NBAR classifies HTTP traffic by text within the URL or HOST fields of a GET request using regular expression matching. NBAR uses the UNIX filename specification as the basis for the URL or HOST specification format. The NBAR engine then converts the specified match string into a regular expression.

NBAR recognizes HTTP GET packets containing the URL and classifies all packets that are sent to the source of the HTTP GET request. Figure 3 illustrates a network topology with NBAR in which Router Y is the NBAR-enabled router.

***Figure 3 Network Topology with NBAR***




When specifying a URL for classification, include only the portion of the URL following the www.hostname.domain in the match statement. For example, for the URL www.cisco.com/latest/whatsnew.html, include only /latest/whatsnew.html.

HOST specification is identical to URL specification. NBAR performs a regular expression match on the HOST field contents inside an HTTP GET packet and classifies all packets from that host. For example, for the URL www.cisco.com/latest/whatsnew.html, include only www.cisco.com.

For MIME type matching, the MIME type can contain any user-specified text string. A list of the Internet Assigned Numbers Authority (IANA)-supported MIME types can be found at the IANA web site.

In MIME type matching, NBAR classifies the packet containing the MIME type and all subsequent packets, which are sent to the source of the HTTP GET request.

NBAR supports URL and HOST classification in the presence of persistent HTTP. NBAR does not classify packets that are part of a pipelined request. With pipelined requests, multiple requests are pipelined to the server before previous requests are serviced.

# Classification of Citrix ICA Traffic by Application Name

NBAR can classify ICA traffic and perform subport classification of Citrix traffic based on Citrix published applications. NBAR can monitor Citrix ICA client requests for a published application destined to a Citrix ICA Master browser. After the client makes a request to the published application, the Citrix ICA Master browser directs the client to the server with the most available memory. The Citrix ICA client then connects to this Citrix ICA server for the application.

NBAR statefully tracks Citrix ICA client/server messages and classifies requests for given Citrix application names and traffic. A Citrix application is named when published on a Citrix ICA server. NBAR performs a regular expression match using a user-specified application name string on the

contents of the Citrix ICA control packets carrying the published application name. Therefore, users need to specify a regular expression that will result in a match for the published application name if they wish to match a specified application. Refer to the **match protocol citrix** command in the *Cisco IOS Quality of Service Solution Command Reference* for additional information.

Citrix ICA clients can be configured in various modes. NBAR cannot distinguish among Citrix applications in all modes of operation. Therefore, network administrators might need to collaborate with Citrix administrators to ensure that NBAR properly classifies Citrix traffic.

A Citrix administrator can configure Citrix to publish Citrix applications individually or as the entire desktop. In the Published Desktop mode of operation, all applications within the published desktop of a client use the same TCP session. Therefore, differentiation among applications is impossible, and NBAR can only be used to classify Citrix applications as aggregates (by looking at port 1494).

The Published Application mode for Citrix ICA clients is recommended when you use NBAR. In Published Application mode, a Citrix administrator can configure a Citrix client in either seamless or nonseamless (windows) modes of operation. In nonseamless mode, each Citrix application uses a separate TCP connection, and NBAR can be used to provide interapplication differentiation based on the name of the published application.

Seamless mode clients can operate in one of two submodes: session sharing or nonsession sharing:

- In seamless session sharing mode, all clients share the same TCP connection, and NBAR cannot differentiate among applications. Seamless sharing mode is enabled by default on some software releases.
- In seamless nonsession sharing mode, each application for each particular client uses a separate TCP connection. NBAR can provide interapplication differentiation in seamless nonsession sharing mode.

To turn off session sharing, perform the following steps:

**Step 1** At the command prompt of the Citrix server, open the registry editor by entering the **regedit** command.

**Step 2** Create the following registry entry (which overrides session sharing):

```
[HKLM]\SYSTEM\CurrentControlSet\Control\Citrix\WFSHELL\TWI
Value name: "SeamlessFlags", type DWORD, possible values :0 or 1
```

Setting this registry value to 1 overrides session sharing. Note that this flag is SERVER GLOBAL.

**Note** NBAR operates properly in ICA secure mode. Pipelined Citrix ICA client requests are not supported.

# Protocol Discovery

So that QoS policies can be developed and applied, NBAR includes a Protocol Discovery feature that provides an easy way to discover application protocols that are transiting an interface. The Protocol Discovery feature discovers any protocol traffic supported by NBAR. Protocol Discovery can be applied to interfaces and can be used to monitor both input and output traffic. Protocol Discovery maintains the following per-protocol statistics for enabled interfaces: total number of input and output packets and bytes, and input and output bit rates.

# Packet Description Language Module

An external Packet Description Language Module (PDLM) can be loaded at run time to extend the NBAR list of recognized protocols. PDLMs can also be used to enhance an existing protocol recognition capability. PDLMs allow NBAR to recognize new protocols without requiring a new Cisco IOS image or a router reload.

New PDLMs will be released only by Cisco and can be loaded from Flash memory. Please contact your local Cisco representative to request additions or changes to the set of protocols classified by NBAR.

# Memory Management

NBAR uses approximately 150 bytes of DRAM for each flow that requires stateful inspection. Table 4 lists the stateful protocols supported by NBAR that require stateful inspection. When NBAR is configured, it allocates 1 MB of DRAM to support up to 5000 concurrent flows. NBAR checks to determine if it needs more memory to handle additional concurrent stateful flows. If such a need is detected, NBAR expands its memory usage in increments of 200 Kb to 400 Kb.

***Table 4 TCP and UDP Stateful Protocols***

| Cisco IOS Release[1] | Protocol | Type | Description | Syntax |
|---|---|---|---|---|
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | FTP | TCP | File Transfer Protocol | ftp |
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | Exchange | TCP | MS-RPC for Exchange | exchange |
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T<br>(HTTP Host classification is not available on the 12.0 XE train) | HTTP | TCP | HTTP with URL, MIME, or HOST classification | http |
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | Netshow | TCP/<br>UDP | Microsoft Netshow | netshow |
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | RealAudio | TCP/<br>UDP | RealAudio Streaming Protocol | realaudio |
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | r-commands | TCP | rsh, rlogin, rexec | rcmd |
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | StreamWorks | UDP | Xing Technology Stream Works audio and video | streamwork |

*Table 4 TCP and UDP Stateful Protocols (continued)*

| Cisco IOS Release[1] | Protocol | Type | Description | Syntax |
|---|---|---|---|---|
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | SQL*NET | TCP/<br>UDP | SQL*NET for Oracle | sqlnet |
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | SunRPC | TCP/<br>UDP | Sun Remote Procedure Call | sunrpc |
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | TFTP | UDP | Trivial File Transfer Protocol | tftp |
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | VDOLive | TCP/<br>UDP | VDOLive Streaming Video | vdolive |

1. Indicates the Cisco IOS maintenance release that first supported the protocol.

# Supported Protocols

NBAR can classify the following three types of protocols:

- TCP and UDP protocols that dynamically assign port numbers and therefore require stateful inspection
- Non-UDP and non-TCP IP protocols
- TCP and UDP protocols that use statically assigned port numbers

Table 5 lists all of the non-UDP and non-TCP protocols than NBAR can classify. Table 6 lists the TCP and UDP static port protocols.

*Table 5 Non-UDP and Non-TCP Protocols*

| Cisco IOS Release[1] | Protocol | Type | Well-Known Port Number | Description | Syntax |
|---|---|---|---|---|---|
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | EGP | IP | 8 | Exterior Gateway Protocol | egp |
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | GRE | IP | 47 | Generic Routing Encapsulation | gre |
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | ICMP | IP | 1 | Internet Control Message Protocol | icmp |
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | IPINIP | IP | 4 | IP in IP | ipinip |

*Table 5 Non-UDP and Non-TCP Protocols (continued)*

| Cisco IOS Release[1] | Protocol | Type | Well-Known Port Number | Description | Syntax |
|---|---|---|---|---|---|
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | IPSec | IP | 50, 51 | IP Encapsulating Security Payload/Authentication Header | ipsec |
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | EIGRP | IP | 88 | Enhanced Interior Gateway Routing Protocol | eigrp |

1. Indicates the Cisco IOS maintenance release that first supported the protocol.

*Table 6 TCP and UDP Static Port Protocols*

| Cisco IOS Release[1] | Protocol | Type | Well-Known Port Number | Description | Syntax |
|---|---|---|---|---|---|
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | BGP | TCP/UDP | 179 | Border Gateway Protocol | bgp |
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | CU-SeeMe | TCP/UDP | 7648, 7649 | Desktop video conferencing | cuseeme |
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | CU-SeeMe | UDP | 24032 | Desktop video conferencing | cuseeme |
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | DHCP/<br>BOOTP | UDP | 67, 68 | Dynamic Host Configuration Protocol/ Bootstrap Protocol | dhcp |
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | DNS | TCP/UDP | 53 | Domain Name System | dns |
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | Finger | TCP | 79 | Finger user information protocol | finger |
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | Gopher | TCP/UDP | 70 | Internet Gopher Protocol | gopher |
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | HTTP | TCP | 80 | Hypertext Transfer Protocol | http |
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | HTTPS | TCP | 443 | Secured HTTP | secure-http |
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | IMAP | TCP/UDP | 143, 220 | Internet Message Access Protocol | imap |

*Table 6 TCP and UDP Static Port Protocols (continued)*

| Cisco IOS Release[1] | Protocol | Type | Well-Known Port Number | Description | Syntax |
|---|---|---|---|---|---|
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | IRC | TCP/UDP | 194 | Internet Relay Chat | irc |
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | Kerberos | TCP/UDP | 88, 749 | Kerberos Network Authentication Service | kerberos |
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | L2TP | UDP | 1701 | L2F/L2TP tunnel | l2tp |
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | LDAP | TCP/UDP | 389 | Lightweight Directory Access Protocol | ldap |
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | MS-PPTP | TCP | 1723 | Microsoft Point-to-Point Tunneling Protocol for Virtual Private Networks (VPNs) | pptp |
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | MS-SQLServer | TCP | 1433 | Microsoft SQL Server Desktop Videoconferencing | sqlserver |
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | NetBIOS | TCP | 137, 139 | NetBIOS over IP (MS Windows) | netbios |
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | NetBIOS | UDP | 137, 138 | NetBIOS over IP (MS Windows) | netbios |
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | NFS | TCP/UDP | 2049 | Network File System | nfs |
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | NNTP | TCP/UDP | 119 | Network News Transfer Protocol | nntp |
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | Notes | TCP/UDP | 1352 | Lotus Notes | notes |
| 12.1(2)E<br>12.1(5)T | Novadigm | TCP/UDP | 3460 to 3465 | Novadigm Enterprise Desktop Manager (EDM) | novadigm |
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | NTP | TCP/UDP | 123 | Network Time Protocol | ntp |
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | PCAnywhere | TCP | 5631, 65301 | Symantec PCAnywhere | pcanywhere |

*Table 6 TCP and UDP Static Port Protocols (continued)*

| Cisco IOS Release[1] | Protocol | Type | Well-Known Port Number | Description | Syntax |
|---|---|---|---|---|---|
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | PCAnywhere | UDP | 22, 5632 | Symantec PCAnywhere | pcanywhere |
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | POP3 | TCP/UDP | 110 | Post Office Protocol | pop3 |
| 12.1(2)E<br>12.1(5)T | Printer | TCP/UDP | 515 | Printer | printer |
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | RIP | UDP | 520 | Routing Information Protocol | rip |
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | RSVP | UDP | 1698,1699 | Resource Reservation Protocol | rsvp |
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | SFTP | TCP | 990 | Secure FTP | secure-ftp |
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | SHTTP | TCP | 443 | Secure HTTP | secure-http |
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | SIMAP | TCP/UDP | 585, 993 | Secure IMAP | secure-imap |
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | SIRC | TCP/UDP | 994 | Secure IRC | secure-irc |
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | SLDAP | TCP/UDP | 636 | Secure LDAP | secure-ldap |
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | SNNTP | TCP/UDP | 563 | Secure NNTP | secure-nntp |
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | SMTP | TCP | 25 | Simple Mail Transfer Protocol | smtp |
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | SNMP | TCP/UDP | 161, 162 | Simple Network Management Protocol | snmp |
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | SOCKS | TCP | 1080 | Firewall security protocol | socks |

*Table 6 TCP and UDP Static Port Protocols (continued)*

| Cisco IOS Release[1] | Protocol | Type | Well-Known Port Number | Description | Syntax |
|---|---|---|---|---|---|
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | SPOP3 | TCP/UDP | 995 | Secure POP3 | secure-pop3 |
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | SSH | TCP | 22 | Secured Shell | ssh |
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | STELNET | TCP | 992 | Secure Telnet | secure-telnet |
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | Syslog | UDP | 514 | System Logging Utility | syslog |
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | Telnet | TCP | 23 | Telnet Protocol | telnet |
| 12.0(5)XE2<br>12.1(1)E<br>12.1(5)T | X Window System | TCP | 6000-6003 | X11, X Window System | xwindows |

1. Indicates the Cisco IOS maintenance release that first supported the protocol.

# Restrictions

The NBAR feature does not support the following:

- More than 24 concurrent URLs, HOSTs, or MIME type matches
- Matching beyond the first 400 bytes in a URL
- Non-IP traffic
- Multicast and other non-CEF switching modes
- Fragmented packets
- Pipelined persistent HTTP requests
- URL/HOST/MIME/ classification with secure HTTP
- Asymmetric flows with stateful protocols
- Packets originating from or destined to the router running NBAR

NBAR is not configurable on the following logical interfaces:

- Fast EtherChannel
- Interfaces where tunneling or encryption is used
- VLANs

**Note** NBAR is configurable on VLANs as of Cisco IOS Release 12.1(13)E, but supported in the software switching path only.

- Dialer interfaces
- Multilink PPP

**Note** NBAR cannot be used to classify output traffic on a WAN link where tunneling or encryption is used. Therefore, NBAR should be configured on other interfaces on the router (such as a LAN link) to perform input classification before the traffic is switched to the WAN link for output.