



Configuring IP Services

This chapter describes how to configure optional IP services. For a complete description of the IP services commands in this chapter, refer to the “IP Services Commands” chapter of the *Cisco IOS IP Command Reference, Volume 1 of 3: Addressing and Services*. To locate documentation of other commands that appear in this chapter, use the command reference master index, or search online.

IP Services Task List

To configure optional IP services, perform any of the optional tasks described in the following sections:

- [Managing IP Connections](#) (Optional)
- [Filtering IP Packets Using Access Lists](#) (Optional)
- [Configuring the Hot Standby Router Protocol](#) (Optional)
- [Configuring IP Accounting](#) (Optional)
- [Configuring TCP Performance Parameters](#) (Optional)
- [Configuring IP over WANs](#) (Optional)
- [Configuring the MultiNode Load Balancing Forwarding Agent](#) (Optional)
- [Monitoring and Maintaining the IP Network](#) (Optional)

Remember that not all the tasks in these sections are required. The tasks you must perform will depend on your network and your needs.

At the end of this chapter, the examples in the “[IP Services Configuration Examples](#)” section illustrate how you might configure your network using IP.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the “[Identifying Supported Platforms](#)” section in the “[Using Cisco IOS Software](#)” chapter of this book.

Managing IP Connections

The IP suite offers a number of services that control and manage IP connections. Internet Control Message Protocol (ICMP) provides many of these services. ICMP messages are sent by routers or access servers to hosts or other routers when a problem is discovered with the Internet header. For detailed information on ICMP, see RFC 792.

To manage various aspects of IP connections, perform the optional tasks described in the following sections:

- [Enabling ICMP Protocol Unreachable Messages](#) (Optional)
- [Enabling ICMP Redirect Messages](#) (Optional)
- [Enabling ICMP Mask Reply Messages](#) (Optional)
- [Understanding Path MTU Discovery](#) (Optional)
- [Setting the MTU Packet Size](#) (Optional)
- [Enabling IP Source Routing](#) (Optional)
- [Configuring Simplex Ethernet Interfaces](#) (Optional)
- [Configuring a DRP Server Agent](#) (Optional)

See the “[ICMP Services Example](#)” section at the end of this chapter for examples of ICMP services.

Enabling ICMP Protocol Unreachable Messages

If the Cisco IOS software receives a nonbroadcast packet destined for itself that uses an unknown protocol, it sends an ICMP protocol unreachable message back to the source. Similarly, if the software receives a packet that it is unable to deliver to the ultimate destination because it knows of no route to the destination address, it sends an ICMP host unreachable message to the source. This feature is enabled by default.

To enable this service if it has been disabled, use the following command in interface configuration mode:

Command	Purpose
Router(config-if)# ip unreachable	Enables the sending of ICMP protocol unreachable and host unreachable messages.

To limit the rate that ICMP destination unreachable messages are generated, use the following command in global configuration mode:

Command	Purpose
Router(config)# ip icmp rate-limit unreachable [df] <i>milliseconds</i>	Limits the rate that ICMP destination unreachable messages are generated.

Enabling ICMP Redirect Messages

Routes are sometimes less than optimal. For example, it is possible for the router to be forced to resend a packet through the same interface on which it was received. If the router resends a packet through the same interface on which it was received, the Cisco IOS software sends an ICMP redirect message to the originator of the packet telling the originator that the router is on a subnet directly connected to the receiving device, and that it must forward the packet to another system on the same subnet. The software sends an ICMP redirect message to the originator of the packet because the originating host presumably could have sent that packet to the next hop without involving this device at all. The redirect message instructs the sender to remove the receiving device from the route and substitute a specified device representing a more direct path. This feature is enabled by default.

To enable the sending of ICMP redirect messages if this feature was disabled, use the following command in interface configuration mode:

Command	Purpose
Router(config-if)# ip redirects	Enables the sending of ICMP redirect messages to learn routes.

Enabling ICMP Mask Reply Messages

Occasionally, network devices must know the subnet mask for a particular subnetwork in the internetwork. To obtain this information, such devices can send ICMP mask request messages. ICMP mask reply messages are sent in reply from devices that have the requested information. The Cisco IOS software can respond to ICMP mask request messages if this function is enabled.

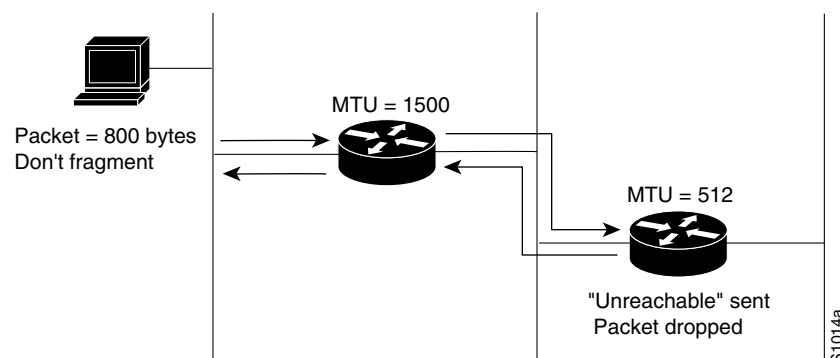
To enable the sending of ICMP mask reply messages, use the following command in interface configuration mode:

Command	Purpose
Router(config-if)# ip mask-reply	Enables the sending of ICMP mask reply messages.

Understanding Path MTU Discovery

The Cisco IOS software supports the IP Path MTU Discovery mechanism, as defined in RFC 1191. IP Path MTU Discovery allows a host to dynamically discover and cope with differences in the maximum allowable maximum transmission unit (MTU) size of the various links along the path. Sometimes a router is unable to forward a datagram because it requires fragmentation (the packet is larger than the MTU you set for the interface with the **ip mtu** interface configuration command), but the “don’t fragment” (DF) bit is set. The Cisco IOS software sends a message to the sending host, alerting it to the problem. The host will need to fragment packets for the destination so that they fit the smallest packet size of all the links along the path. This technique is shown in [Figure 17](#).

Figure 17 IP Path MTU Discovery



IP Path MTU Discovery is useful when a link in a network goes down, forcing the use of another, different MTU-sized link (and different routers). As shown in [Figure 17](#), suppose a router is sending IP packets over a network where the MTU in the first router is set to 1500 bytes, but the second router is set to 512 bytes. If the “Don’t fragment” bit of the datagram is set, the datagram would be dropped

because the 512-byte router is unable to forward it. All packets larger than 512 bytes are dropped in this case. The second router returns an ICMP destination unreachable message to the source of the datagram with its Code field indicating, “Fragmentation needed and DF set.” To support IP Path MTU Discovery, it would also include the MTU of the next hop network link in the low-order bits of an unused header field.

IP Path MTU Discovery is also useful when a connection is being established and the sender has no information at all about the intervening links. It is always advisable to use the largest MTU that the links will bear; the larger the MTU, the fewer packets the host must send.

**Note**

IP Path MTU Discovery is a process initiated by end hosts. If an end host does not support IP Path MTU Discovery, the receiving device will have no mechanism available to avoid fragmenting datagrams generated by the end host.

If a router that is configured with a small MTU on an outbound interface receives packets from a host that is configured with a large MTU (for example, receiving packets from a Token Ring interface and forwarding them to an outbound Ethernet interface), the router fragments received packets that are larger than the MTU of the outbound interface. Fragmenting packets slows the performance of the router. To keep routers in your network from fragmenting received packets, run IP Path MTU Discovery on all hosts and routers in your network, and always configure the largest possible MTU for each router interface type.

To enable IP Path MTU Discovery for connections initiated by the router (when the router is acting as a host), see the section “[Enabling TCP Path MTU Discovery](#)” later in this chapter.

Setting the MTU Packet Size

All interfaces have a default MTU packet size. You can adjust the IP MTU size so that the Cisco IOS software will fragment any IP packet that exceeds the MTU set for an interface.

Changing the MTU value (with the **mtu** interface configuration command) can affect the IP MTU value. If the current IP MTU value is the same as the MTU value and you change the MTU value, the IP MTU value will be modified automatically to match the new MTU. However, the reverse is not true; changing the IP MTU value has no effect on the value for the **mtu** interface configuration command.

Also, all devices on a physical medium must have the same protocol MTU in order to operate.

To set the MTU packet size for a specified interface, use the following command in interface configuration mode:

Command	Purpose
Router(config-if)# ip mtu <i>bytes</i>	Sets the IP MTU packet size for an interface.

Enabling IP Source Routing

The Cisco IOS software examines IP header options on every packet. It supports the IP header options Strict Source Route, Loose Source Route, Record Route, and Time Stamp, which are defined in RFC 791. If the software finds a packet with one of these options enabled, it performs the appropriate action. If it finds a packet with an invalid option, it sends an ICMP parameter problem message to the source of the packet and discards the packet.

IP provides a provision known as *source routing* that allows the source IP host to specify a route through the IP network. Source routing is specified as an option in the IP header. If source routing is specified, the software forwards the packet according to the specified source route. This feature is employed when you want to force a packet to take a certain route through the network. The default is to perform source routing.

To enable IP source-route header options if they have been disabled, use the following command in global configuration mode:

Command	Purpose
Router(config)# ip source-route	Enables IP source routing.

Configuring Simplex Ethernet Interfaces

You can configure simplex Ethernet interfaces. This feature is useful for setting up dynamic IP routing over a simplex circuit (a circuit that receives only or sends only). When a route is learned on a receive-only interface, the interface designated as the source of the route is converted to the interface you specify. When packets are routed out this specified interface, they are sent to the IP address of the source of the routing update. To reach this IP address on a transmit-only Ethernet link, a static Address Resolution Protocol (ARP) entry mapping this IP address to the hardware address of the other end of the link is required.

To assign a transmit interface to a receive-only interface, use the following command in interface configuration mode:

Command	Purpose
Router(config-if)# transmit-interface <i>type number</i>	Assigns a transmit interface to a receive-only interface.

See the “[Simplex Ethernet Interfaces Example](#)” section at the end of this chapter for an example of configuring a simplex Ethernet interface.

Configuring a DRP Server Agent

The Director Response Protocol (DRP) is a simple User Datagram Protocol (UDP)-based application developed by Cisco Systems. It enables the Cisco DistributedDirector product to query routers (DRP Server Agents) in the field for Border Gateway Protocol (BGP) and Interior Gateway Protocol (IGP) routing table metrics between distributed servers and clients. DistributedDirector, a separate standalone product, uses DRP to transparently redirect end-user service requests to the topologically closest responsive server. DRP enables DistributedDirector to provide dynamic, scalable, and “network intelligent” Internet traffic load distribution between multiple geographically dispersed servers.

DRP Server Agents are border routers (or peers to border routers) that support the geographically distributed servers for which DistributedDirector service distribution is desired. Note that, because DistributedDirector makes decisions based on BGP and IGP information, all DRP Server Agents must have access to full BGP and IGP routing tables.

Refer to the *Cisco DistributedDirector 2501 Installation and Configuration Guide* or the *Cisco DistributedDirector 4700-M Installation and Configuration Guide* for information on how to configure DistributedDirector.

To configure and maintain the DRP Server Agent, perform the tasks described in the following sections. The task in the first section is required; the tasks in the remaining sections are optional.

- [Enabling the DRP Server Agent](#) (Required)
- [Limiting the Source of DRP Queries](#) (Optional)
- [Configuring Authentication of DRP Queries and Responses](#) (Optional)

To monitor and maintain the DRP Server Agent, see the section “[Monitoring and Maintaining the DRP Server Agent](#)” later in this chapter.

For an example of configuring a DRP Server Agent, see the section “[DRP Server Agent Example](#)” at the end of this chapter.

Enabling the DRP Server Agent

The DRP Server Agent is disabled by default. To enable it, use the following command in global configuration mode:

Command	Purpose
Router(config)# ip drp server	Enables the DRP Server Agent.

Limiting the Source of DRP Queries

As a security measure, you can limit the source of valid DRP queries. If a standard IP access list is applied to the interface, the Server Agent will respond only to DRP queries originating from an IP address in the list. If no access list is configured, the Server Agent will answer all queries.

If both an access group and a key chain (described in the next section) have been configured, both security mechanisms must allow access before a request is processed.

To limit the source of valid DRP queries, use the following command in global configuration mode:

Command	Purpose
Router(config)# ip drp access-group <i>access-list-number</i>	Controls the sources of valid DRP queries by applying a standard IP access list.

Configuring Authentication of DRP Queries and Responses

Another available security measure is to configure the DRP Server Agent to authenticate DRP queries and responses. You define a key chain, identify the keys that belong to the key chain, and specify how long each key is valid. To do so, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# ip drp authentication key-chain <i>name-of-chain</i>	Identifies which key chain to use to authenticate all DRP requests and responses.
Step 2	Router(config)# key chain <i>name-of-chain</i>	Identifies a key chain (match the name configured in Step 1).
Step 3	Router(config-keychain)# key <i>number</i>	In key-chain configuration mode, identifies the key number.

	Command	Purpose
Step 4	Router(config-keychain-key)# key-string text	In key-chain key configuration mode, identifies the key string.
Step 5	Router(config-keychain-key)# accept-lifetime start-time {infinite end-time duration seconds}	(Optional) Specifies the time period during which the key can be received.
Step 6	Router(config-keychain-key)# send-lifetime start-time {infinite end-time duration seconds}	(Optional) Specifies the time period during which the key can be sent.

When configuring your key chains and keys, be aware of the following guidelines:

- The key chain configured for the DRP Server Agent in Step 1 must match the key chain in Step 2.
- The key configured in the primary agent in the remote router must match the key configured in the DRP Server Agent in order for responses to be processed.
- You can configure multiple keys with lifetimes, and the software will rotate through them.
- If authentication is enabled and multiple keys on the key chain happen to be active based on the **send-lifetime** values, the software uses only the first key it encounters for authentication.
- Use the **show key chain** command to display key chain information.



Note

To configure lifetimes for DRP authentication, you must configure time services for your router. For information on setting time services, see the Network Time Protocol (NTP) and calendar commands in the “Performing Basic System Management” chapter of the *Cisco IOS Configuration Fundamentals Configuration Guide*.

Filtering IP Packets Using Access Lists

Packet filtering helps control packet movement through the network. Such control can help limit network traffic and restrict network use by certain users or devices. To permit or deny packets from crossing specified interfaces, we provide *access lists*.

You can use access lists in the following ways:

- To control the transmission of packets on an interface
- To control vty access
- To restrict contents of routing updates

This section summarizes how to create IP access lists and how to apply them.

See the “[IP Services Configuration Examples](#)” section at the end of this chapter for examples of configuring IP access lists.

An access list is a sequential collection of permit and deny conditions that apply to IP addresses. The Cisco IOS software tests addresses against the conditions in an access list one by one. The first match determines whether the software accepts or rejects the address. Because the software stops testing conditions after the first match, the order of the conditions is critical. If no conditions match, the software rejects the address.

The two main tasks involved in using access lists are as follows:

1. Create an access list by specifying an access list number or name and access conditions.

2. Apply the access list to interfaces or terminal lines.

These and other tasks are described in this section and are labeled as required or optional. Either the first or second task is required, depending on whether you identify your access list with a number or a name.

- [Creating Standard and Extended Access Lists Using Numbers](#) (Required)
- [Creating Standard and Extended Access Lists Using Names](#) (Required)
- [Specifying IP Extended Access Lists with Fragment Control](#) (Optional)
- [Enabling Turbo Access Control Lists](#) (Optional)
- [Applying Time Ranges to Access Lists](#) (Optional)
- [Including Comments About Entries in Access Lists](#) (Optional)
- [Applying Access Lists](#) (Required)

Creating Standard and Extended Access Lists Using Numbers

Cisco IOS software supports the following types of access lists for IP:

- Standard IP access lists that use source addresses for matching operations.
- Extended IP access lists that use source and destination addresses for matching operations, and optional protocol type information for finer granularity of control.
- Dynamic extended IP access lists that grant access per user to a specific source or destination host basis through a user authentication process. In essence, you can allow user access through a firewall dynamically, without compromising security restrictions. Dynamic access lists and lock-and-key access are described in the “Configuring Traffic Filters” chapter of the *Cisco IOS Security Configuration Guide*.
- Reflexive access lists that allow IP packets to be filtered based on session information. Reflexive access lists contain temporary entries, and are nested within an extended, named IP access list. For information on reflexive access lists, refer to the “Configuring IP Session Filtering (Reflexive Access Lists)” chapter in the *Cisco IOS Security Configuration Guide* and the “Reflexive Access List Commands” chapter in the *Cisco IOS Security Command Reference*.



Note

Release 11.1 introduced substantial changes to IP access lists. These extensions are backward compatible; migrating from a release earlier than Release 11.1 to the current release will convert your access lists automatically. However, the current implementation of access lists is incompatible with Cisco IOS Release 11.1 or earlier. If you create an access list using the current Cisco IOS release and then load older Cisco IOS software, the resulting access list will not be interpreted correctly. This condition could cause you severe security problems. Save your old configuration file before booting Release 11.1 or earlier images.

To create a standard access list, use the following commands in global configuration mode:

	Command	Purpose
Step 1	Router(config)# access-list <i>access-list-number</i> remark <i>remark</i>	Indicates the purpose of the deny or permit statement. ¹
Step 2	Router(config)# access-list <i>access-list-number</i> { deny permit } <i>source</i> [<i>source-wildcard</i>] [log]	Defines a standard IP access list using a source address and wildcard.
	or Router(config)# access-list <i>access-list-number</i> { deny permit } any [log]	Defines a standard IP access list using an abbreviation for the source and source mask of 0.0.0.0 255.255.255.255.

1. This example configures the remark before the **deny** or **permit** statement. The remark can be configured after the **deny** or **permit** statement.

The Cisco IOS software can provide logging messages about packets permitted or denied by a standard IP access list. That is, any packet that matches the access list will cause an informational logging message about the packet to be sent to the console. The level of messages logged to the console is controlled by the **logging console** global configuration command.

The first packet that triggers the access list causes an immediate logging message, and subsequent packets are collected over 5-minute intervals before they are displayed or logged. The logging message includes the access list number, whether the packet was permitted or denied, the source IP address of the packet, and the number of packets from that source permitted or denied in the prior 5-minute interval.

However, you can use the **ip access-list log-update** command to set the number of packets that, when match an access list (and are permitted or denied), cause the system to generate a log message. You might want to do this to receive log messages more frequently than at 5-minute intervals.



Caution

If you set the *number-of-matches* argument to 1, a log message is sent right away, rather than caching it; every packet that matches an access list causes a log message. A setting of 1 is not recommended because the volume of log messages could overwhelm the system.

Even if you use the **ip access-list log-update** command, the 5-minute timer remains in effect, so each cache is emptied at the end of 5 minutes, regardless of the count of messages in each cache. Regardless of when the log message is sent, the cache is flushed and the count reset to 0 for that message the same way it is when a threshold is not specified.



Note

The logging facility might drop some logging message packets if there are too many to be handled or if there is more than one logging message to be handled in 1 second. This behavior prevents the router from crashing due to too many logging packets. Therefore, the logging facility should not be used as a billing tool or an accurate source of the number of matches to an access list.



Note

If you enable CEF and then create an access list that uses the **log** keyword, the packets that match the access list are not CEF switched. They are fast switched. Logging disables CEF.

For an example of a standard IP access list using logs, see the section “[Numbered Access List Examples](#)” at the end of this chapter.

To create an extended access list, use the following commands in global configuration mode:

	Command	Purpose
Step 1	Router(config)# access-list <i>access-list-number</i> remark <i>remark</i>	Indicates the purpose of the deny or permit statement. ¹
Step 2	Router(config)# access-list <i>access-list-number</i> { deny permit } <i>protocol</i> <i>source</i> <i>source-wildcard</i> <i>destination</i> <i>destination-wildcard</i> [precedence <i>precedence</i>] [tos <i>tos</i>] [established] [log log-input] [time-range <i>time-range-name</i>] [fragments]	Defines an extended IP access list number and the access conditions. Specifies a time range to restrict when the permit or deny statement is in effect. Use the log keyword to get access list logging messages, including violations. Use the log-input keyword to include input interface, source MAC address, or VC in the logging output.
	OR	OR
	Router(config)# access-list <i>access-list-number</i> { deny permit } <i>protocol</i> any any [log log-input] [time-range <i>time-range-name</i>] [fragments]	Defines an extended IP access list using an abbreviation for a source and source wildcard of 0.0.0.0 255.255.255.255, and an abbreviation for a destination and destination wildcard of 0.0.0.0 255.255.255.255.
	OR	OR
	Router(config)# access-list <i>access-list-number</i> { deny permit } <i>protocol</i> host <i>source</i> host <i>destination</i> [log log-input] [time-range <i>time-range-name</i>] [fragments]	Defines an extended IP access list using an abbreviation for a source and source wildcard of <i>source</i> 0.0.0.0, and an abbreviation for a destination and destination wildcard of <i>destination</i> 0.0.0.0.
	OR	OR
	Router(config)# access-list <i>access-list-number</i> [dynamic <i>dynamic-name</i> [timeout <i>minutes</i>]] { deny permit } <i>protocol</i> <i>source</i> <i>source-wildcard</i> <i>destination</i> <i>destination-wildcard</i> [precedence <i>precedence</i>] [tos <i>tos</i>] [established] [log log-input] [time-range <i>time-range-name</i>] [fragments]	Defines a dynamic access list. For information about lock-and-key access, refer to the “Configuring Traffic Filters” chapter in the <i>Cisco IOS Security Configuration Guide</i> .

1. This example configures the remark before the **deny** or **permit** statement. The remark can be configured after the **deny** or **permit** statement.



Note

The **fragments** keyword is described in the [Specifying IP Extended Access Lists with Fragment Control](#) section.

After you create an access list, you place any subsequent additions (possibly entered from the terminal) at the end of the list. In other words, you cannot selectively add or remove access list command lines from a specific access list.



Note

When creating an access list, remember that, by default, the end of the access list contains an implicit deny statement for everything if it did not find a match before reaching the end.

**Note**

In a standard access list, if you omit the mask from an associated IP host address access list specification, 0.0.0.0 is assumed to be the mask.

**Note**

Autonomous switching is not used when you have extended access lists.

After creating an access list, you must apply it to a line or interface, as shown in the section “[Applying Access Lists](#)” later in this chapter. See the “[Implicit Masks in Access Lists Examples](#)” section at the end of this chapter for examples of implicit masks.

Creating Standard and Extended Access Lists Using Names

You can identify IP access lists with an alphanumeric string (a name) rather than a number. Named access lists allow you to configure more IP access lists in a router than if you were to use numbered access lists. If you identify your access list with a name rather than a number, the mode and command syntax are slightly different. Currently, only packet and route filters can use a named list.

Consider the following guidelines before configuring named access lists:

- Access lists specified by name are not compatible with Cisco IOS Releases prior to 11.2.
- Not all access lists that accept a number will accept a name. Access lists for packet filters and route filters on interfaces can use a name.
- A standard access list and an extended access list cannot have the same name.
- Numbered access lists are also available, as described in the previous section, “[Creating Standard and Extended Access Lists Using Numbers](#).”

**Note**

Named access lists will not be recognized by any software release prior to Cisco IOS Release 11.2.

To create a standard access list, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# ip access-list standard <i>name</i>	Defines a standard IP access list using a name and enters standard named access list configuration mode.
Step 2	Router(config-std-nacl)# remark <i>remark</i>	Allows you to comment about the following deny or permit statement in a named access list. ¹
Step 3	Router(config-std-nacl)# deny { <i>source</i> [<i>source-wildcard</i>] any } [log] and/or Router(config-std-nacl)# permit { <i>source</i> [<i>source-wildcard</i>] any } [log]	Specifies one or more conditions allowed or denied, which determines whether the packet is passed or dropped.
Step 4	Router(config-std-nacl)# exit	Exits access-list configuration mode.

1. This example configures the remark before the **deny** or **permit** statement. The remark can be configured after the **deny** or **permit** statement.

To create an extended access list, use the following commands beginning in global configuration mode:

Step 1	Router(config)# ip access-list extended <i>name</i>	Defines an extended IP access list using a name and enters extended named access list configuration mode.
Step 2	Router(config-ext-nacl)# remark <i>remark</i>	Allows you to comment about the following deny or permit statement in a named access list. ¹
Step 3	<pre>Router(config-ext-nacl)# deny permit <i>protocol</i> <i>source source-wildcard destination</i> <i>destination-wildcard</i> [precedence <i>precedence</i>] [tos <i>tos</i>] [established] [log log-input] [time-range <i>time-range-name</i>] [fragments]</pre> <p>or</p> <pre>Router(config-ext-nacl)# deny permit <i>protocol any</i> [log log-input] [time-range <i>time-range-name</i>] [fragments]</pre> <p>or</p> <pre>Router(config-ext-nacl) deny permit <i>protocol host</i> <i>source host destination</i> [log log-input] [time-range <i>time-range-name</i>] [fragments]</pre> <p>or</p> <pre>Router(config-ext-nacl)# dynamic <i>dynamic-name</i> [timeout <i>minutes</i>] {deny permit} <i>protocol source</i> <i>source-wildcard destination destination-wildcard</i> [precedence <i>precedence</i>] [tos <i>tos</i>] [established] [log log-input] [time-range <i>time-range-name</i>] [fragments]</pre>	<p>In access-list configuration mode, specifies the conditions allowed or denied. Specifies a time range to restrict when the permit or deny statement is in effect. Use the log keyword to get access list logging messages, including violations. Use the log-input keyword to include input interface, source MAC address, or VC in the logging output.</p> <p>or</p> <p>Defines an extended IP access list using an abbreviation for a source and source wildcard of 0.0.0.0 255.255.255.255, and an abbreviation for a destination and destination wildcard of 0.0.0.0 255.255.255.255.</p> <p>or</p> <p>Defines an extended IP access list using an abbreviation for a source and source wildcard of <i>source</i> 0.0.0.0, and an abbreviation for a destination and destination wildcard of <i>destination</i> 0.0.0.0.</p> <p>or</p> <p>Defines a dynamic access list.</p>

1. This example configures the remark before the deny or permit statement. The remark can be configured after the deny or permit statement.

**Note**

Autonomous switching is not used when you have extended access lists.

**Note**

The **fragments** keyword is described in the [Specifying IP Extended Access Lists with Fragment Control](#) section.

After you initially create an access list, you place any subsequent additions (possibly entered from the terminal) at the end of the list. In other words, you cannot selectively add access list command lines to a specific access list. However, you can use **no permit** and **no deny** commands to remove entries from a named access list.

**Note**

When making the standard and extended access list, remember that, by default, the end of the access list contains an implicit deny statement for everything if it did not find a match before reaching the end. Further, with standard access lists, if you omit the mask from an associated IP host address access list specification, 0.0.0.0 is assumed to be the mask.

After creating an access list, you must apply it to a line or interface, as shown in section “[Applying Access Lists](#)” later in this chapter.

See the “[Named Access List Example](#)” section at the end of this chapter for an example of a named access list.

Specifying IP Extended Access Lists with Fragment Control



This section describes the functionality added to IP extended named and numbered access lists. You can now specify whether the system examines noninitial IP fragments of packets when applying an IP extended access list.

Prior to this feature, nonfragmented packets and the initial fragment of a packet were processed by IP extended access lists (if such an access list was applied), but noninitial fragments were permitted by default. The IP Extended Access Lists with Fragment Control feature now allows more granularity of control over noninitial packets.

Because noninitial fragments contain only Layer 3 information, access-list entries containing only Layer 3 information can and now are applied to noninitial fragments. The fragment has all the information the system needs to filter, so the entry is applied to the fragments.

This feature adds the optional **fragments** keyword to four IP access list commands [**access-list (IP extended)**, **deny (IP)**, **dynamic**, and **permit (IP)**]. By specifying the **fragments** keyword in an access list entry, that particular access list entry applies only to noninitial fragments of packets; the fragment is either permitted or denied accordingly.

The behavior of access-list entries regarding the presence or absence of the **fragments** keyword can be summarized as follows:

If the Access-List Entry has...	Then..
<p>...no fragments keyword, and assuming all of the access-list entry information matches,</p>	<p>For an access-list entry containing only Layer 3 information:</p> <ul style="list-style-type: none"> • The entry is applied to nonfragmented packets, initial fragments and noninitial fragments. <p>For an access list entry containing Layer 3 and Layer 4 information:</p> <ul style="list-style-type: none"> • The entry is applied to nonfragmented packets and initial fragments. <ul style="list-style-type: none"> – If the entry matches and is a permit statement, the packet or fragment is permitted. – If the entry matches and is a deny statement, the packet or fragment is denied. • The entry is also applied to noninitial fragments in the following manner. Because noninitial fragments contain only Layer 3 information, only the Layer 3 portion of an access-list entry can be applied. If the Layer 3 portion of the access-list entry matches, and <ul style="list-style-type: none"> – If the entry is a permit statement, the noninitial fragment is permitted. – If the entry is a deny statement, the next access-list entry is processed. <p> Note Note that the deny statements are handled differently for noninitial fragments versus nonfragmented or initial fragments.</p>
<p>...the fragments keyword, and assuming all of the access-list entry information matches,</p>	<p>The access-list entry is applied only to noninitial fragments.</p> <p> Note The fragments keyword cannot be configured for an access-list entry that contains any Layer 4 information.</p>

Be aware that you should not simply add the **fragments** keyword to every access list entry because the first fragment of the IP packet is considered a nonfragment and is treated independently of the subsequent fragments. An initial fragment will not match an access list **permit** or **deny** entry that contains the **fragments** keyword, the packet is compared to the next access list entry, and so on, until it is either permitted or denied by an access list entry that does not contain the **fragments** keyword. Therefore, you may need two access list entries for every **deny** entry. The first **deny** entry of the pair will not include the **fragments** keyword, and applies to the initial fragment. The second **deny** entry of the pair will include the **fragments** keyword and applies to the subsequent fragments. In the cases where there are multiple **deny** access list entries for the same host but with different Layer 4 ports, a single **deny** access-list entry with the **fragments** keyword for that host is all that needs to be added. Thus all the fragments of a packet are handled in the same manner by the access list.

The **fragments** keyword can be applied to dynamic access lists also.

Packet fragments of IP datagrams are considered individual packets and each counts individually as a packet in access list accounting and access list violation counts.

**Note**

The **fragments** keyword cannot solve all cases involving access lists and IP fragments.

Turbo Access Lists

A turbo access list treats fragments and uses the **fragments** keyword in the same manner as a nonturbo access list.

Policy Routing

Fragmentation and the fragment control feature affect policy routing if the policy routing is based on the **match ip address** command and the access list had entries that match on Layer 4 through 7 information. It is possible that noninitial fragments pass the access list and are policy routed, even if the first fragment was not policy routed or the reverse.

By using the **fragments** keyword in access list entries as described earlier, a better match between the action taken for initial and noninitial fragments can be made and it is more likely policy routing will occur as intended.

Benefits of Fragment Control in an IP Extended Access List

If the **fragments** keyword is used in additional IP access list entries that deny fragments, the fragment control feature provides the following benefits:

Additional Security

You are able to block more of the traffic you intended to block, not just the initial fragment of such packets. The unwanted fragments no longer linger at the receiver until the reassembly timeout is reached because they are blocked before being sent to the receiver. Blocking a greater portion of unwanted traffic improves security and reduces the risk from potential hackers.

Reduced Cost

By blocking unwanted noninitial fragments of packets, you are not paying for traffic you intended to block.

Reduced Storage

By blocking unwanted noninitial fragments of packets from ever reaching the receiver, that destination does not have to store the fragments until the reassembly timeout period is reached.

Expected Behavior is Achieved

The noninitial fragments will be handled in the same way as the initial fragment, which is what you would expect. There are fewer unexpected policy routing results and fewer fragment of packets being routed when they should not be.

For an example of fragment control in an IP extended access list, see the [IP Extended Access List with Fragment Control Example](#).

Enabling Turbo Access Control Lists

The Turbo Access Control Lists (Turbo ACL) feature processes access lists more expediently than conventional access lists. This feature enables Cisco 7200 and 7500 series routers, and Cisco 12000 series Gigabit Switch Routers, to evaluate ACLs for more expedient packet classification and access checks.

ACLs are normally searched sequentially to find a matching rule, and ACLs are ordered specifically to take this factor into account. Because of the increasing needs and requirements for security filtering and packet classification, ACLs can expand to the point that searching the ACL adds a substantial amount of time and memory when packets are being forwarded. Moreover, the time taken by the router to search the list is not always consistent, adding a variable latency to the packet forwarding. A high CPU load is necessary for searching an access list with several entries.

The Turbo ACL feature compiles the ACLs into a set of lookup tables, while maintaining the first match requirements. Packet headers are used to access these tables in a small, fixed number of lookups, independently of the existing number of ACL entries. The benefits of this feature include the following:

- For ACLs larger than three entries, the CPU load required to match the packet to the predetermined packet-matching rule is lessened. The CPU load is fixed, regardless of the size of the access list, allowing for larger ACLs without incurring any CPU overhead penalties. The larger the access list, the greater the benefit.
- The time taken to match the packet is fixed, so that latency of the packets is smaller (substantially in the case of large access lists) and, more importantly, consistent, allowing better network stability and more accurate transit times.



Note

Access lists containing specialized processing characteristics such as evaluate and time-range entries are excluded from Turbo ACL acceleration.

The Turbo ACL builds a set of lookup tables from the ACLs in the configuration; these tables increase the internal memory usage, and in the case of large and complex ACLs, tables containing 2 MB to 4 MB of memory are usually required. Routers enabled with the Turbo ACL feature should allow for this amount of memory usage. The **show access-list compiled** EXEC command displays the memory overhead of the Turbo ACL tables for each access list.

To configure the Turbo ACL feature, perform the tasks described in the following sections. The task in the first section is required; the task in the remaining section is optional:

- [Configuring Turbo ACLs](#) (Required)
- [Verifying Turbo ACLs](#) (Optional)

Configuring Turbo ACLs

To enable the Turbo ACL feature, use the following command in global configuration mode:

Command	Purpose
Router(config)# access-list compiled	Enables the Turbo ACL feature.

Verifying Turbo ACLs

Use the **show access-list compiled EXEC** command to verify that the Turbo ACL feature has been successfully configured on your router. This command also displays the memory overhead of the Turbo ACL tables for each access list. The command output contains the following states:

- **Operational**—The access list has been compiled by Turbo ACL, and matching to this access list is performed through the Turbo ACL tables at high speed.
- **Unsuitable**—The access list is not suitable for compiling, perhaps because it has time-range enabled entries, evaluate references, or dynamic entries.
- **Deleted**—No entries are in this access list.
- **Building**—The access list is being compiled. Depending on the size and complexity of the list, and the load on the router, the building process may take a few seconds.
- **Out of memory**—An access list cannot be compiled because the router has exhausted its memory.

The following is sample output from the **show access-lists compiled EXEC** command:

```
Router# show access-lists compiled

Compiled ACL statistics:
12 ACLs loaded, 12 compiled tables
ACL          State      Tables  Entries  Config  Fragment  Redundant  Memory
1            Operational  1       2        1       0         0         1Kb
2            Operational  1       3        2       0         0         1Kb
3            Operational  1       4        3       0         0         1Kb
4            Operational  1       3        2       0         0         1Kb
5            Operational  1       5        4       0         0         1Kb
9            Operational  1       3        2       0         0         1Kb
20           Operational  1       9        8       0         0         1Kb
21           Operational  1       5        4       0         0         1Kb
101          Operational  1       15       9       7         2         1Kb
102          Operational  1       13       6       6         0         1Kb
120          Operational  1       2        1       0         0         1Kb
199          Operational  1       4        3       0         0         1Kb

First level lookup tables:
Block      Use              Rows      Columns  Memory used
0          TOS/Protocol     6/16     12/16    66048
1          IP Source (MS)   10/16    12/16    66048
2          IP Source (LS)   27/32    12/16    132096
3          IP Dest (MS)     3/16     12/16    66048
4          IP Dest (LS)     9/16     12/16    66048
5          TCP/UDP Src Port 1/16     12/16    66048
6          TCP/UDP Dest Port 3/16     12/16    66048
7          TCP Flags/Fragment 3/16     12/16    66048
```

Applying Time Ranges to Access Lists

You can implement access lists based on the time of day and week using the **time-range** global configuration command. To do so, first define the name and times of the day and week of the time range, then reference the time range by name in an access list to apply restrictions to the access list.

Currently, IP and Internetwork Packet Exchange (IPX) named or numbered extended access lists are the only functions that can use time ranges. The time range allows the network administrator to define when the **permit** or **deny** statements in the access list are in effect. Prior to this feature, access list statements were always in effect once they were applied. The **time-range** keyword is referenced in the named and numbered extended access list task tables in the previous sections “[Creating Standard and Extended Access Lists Using Numbers](#)” and “[Creating Standard and Extended Access Lists Using Names](#).” The

time-range command is described in the “Performing Basic System Management” chapter of the *Cisco IOS Configuration Fundamentals Configuration Guide*. See the “[Time Range Applied to an IP Access List Example](#)” section at the end of this chapter for a configuration example of IP time ranges.

Possible benefits of using time ranges include the following:

- The network administrator has more control over permitting or denying a user access to resources. These resources could be an application (identified by an IP address/mask pair and a port number), policy routing, or an on-demand link (identified as interesting traffic to the dialer).
- Network administrators can set time-based security policy, including the following:
 - Perimeter security using the Cisco IOS Firewall feature set or access lists
 - Data confidentiality with Cisco Encryption Technology or IP Security Protocol (IPSec)
- Policy-based routing (PBR) and queueing functions are enhanced.
- When provider access rates vary by time of day, it is possible to automatically reroute traffic cost effectively.
- Service providers can dynamically change a committed access rate (CAR) configuration to support the quality of service (QoS) service level agreements (SLAs) that are negotiated for certain times of day.
- Network administrators can control logging messages. Access list entries can log traffic at certain times of the day, but not constantly. Therefore, administrators can simply deny access without needing to analyze many logs generated during peak hours.

Including Comments About Entries in Access Lists

You can include comments (remarks) about entries in any named IP access list using the **remark** access-list configuration command. The remarks make the access list easier for the network administrator to understand and scan. Each remark line is limited to 100 characters.

The remark can go before or after a **permit** or **deny** statement. You should be consistent about where you put the remark so it is clear which remark describes which **permit** or **deny** statement. For example, it would be confusing to have some remarks *before* the associated **permit** or **deny** statements and some remarks *after* the associated statements. The standard and extended access list task tables in the previous sections “[Creating Standard and Extended Access Lists Using Numbers](#)” and “[Creating Standard and Extended Access Lists Using Names](#)” include the **remark** command. See the “[Commented IP Access List Entry Examples](#)” section at the end of this chapter for examples of commented IP access list entries.

Remember to apply the access list to an interface or terminal line after the access list is created. See the following section “[Applying Access Lists](#)” for more information.

Applying Access Lists

After creating an access list, you must reference the access list to make it work. To use an access list, perform the tasks described in the following sections. The tasks in the first section are required; the tasks in the remaining sections are optional:

- [Controlling Access to a Line or Interface](#) (Required)
- [Controlling Policy Routing and the Filtering of Routing Information](#) (Optional)
- [Controlling Dialer Functions](#) (Optional)

Controlling Access to a Line or Interface

After you create an access list, you can apply it to one or more interfaces. Access lists can be applied on *either* outbound or inbound interfaces. This section describes guidelines on how to accomplish this task for both terminal lines and network interfaces. Remember the following:

- When controlling access to a line, you must use a number.
- When controlling access to an interface, you can use a name or number.

To restrict access to a vty and the addresses in an access list, use the following command in line configuration mode. Only numbered access lists can be applied to lines. Set identical restrictions on all the virtual terminal lines, because a user can attempt to connect to any of them.

Command	Purpose
Router(config-line)# access-class <i>access-list-number</i> { in out }	Restricts incoming and outgoing connections between a particular vty (into a device) and the addresses in an access list.

To restrict access to an interface, use the following command in interface configuration mode:

Command	Purpose
Router(config-if)# ip access-group { <i>access-list-number</i> <i>access-list-name</i> } { in out }	Controls access to an interface.

For inbound access lists, after receiving a packet, the Cisco IOS software checks the source address of the packet against the access list. If the access list permits the address, the software continues to process the packet. If the access list rejects the address, the software discards the packet and returns an ICMP host unreachable message.

For outbound access lists, after receiving and routing a packet to a controlled interface, the software checks the source address of the packet against the access list. If the access list permits the address, the software sends the packet. If the access list rejects the address, the software discards the packet and returns an ICMP host unreachable message.

When you apply an access list that has not yet been defined to an interface, the software will act as if the access list has not been applied to the interface and will accept all packets. Remember this behavior if you use undefined access lists as a means of security in your network.

Controlling Policy Routing and the Filtering of Routing Information

To use access lists to control policy routing and the filtering of routing information, see the “Configuring IP Routing Protocol-Independent Features” chapter of this document.

Controlling Dialer Functions

To use access lists to control dialer functions, refer to the “Preparing to Configure DDR” chapter in the *Cisco IOS Dial Technologies Configuration Guide*.

Configuring the Hot Standby Router Protocol

The Hot Standby Router Protocol (HSRP) provides high network availability because it routes IP traffic from hosts on Ethernet, FDDI, or Token Ring networks without relying on the availability of any single router. HSRP is used in a group of routers for selecting an active router and a standby router. (An active router is the router of choice for routing packets; a standby router is a router that takes over the routing duties when an active router fails, or when preset conditions are met.)

HSRP is useful for hosts that do not support a router discovery protocol (such as ICMP Router Discovery Protocol [IRDP]) and cannot switch to a new router when their selected router reloads or loses power. Because existing TCP sessions can survive the *failover*, this protocol also provides a more transparent recovery for hosts that dynamically choose a next hop for routing IP traffic.

When the HSRP is configured on a network segment, it provides a virtual MAC address and an IP address that is shared among a group of routers running HSRP. The address of this HSRP group is referred to as the *virtual IP address*. One of these devices is selected by the protocol to be the active router. The active router receives and routes packets destined for the MAC address of the group. For n routers running HSRP, $n + 1$ IP and MAC addresses are assigned.

HSRP detects when the designated active router fails, at which point a selected standby router assumes control of the MAC and IP addresses of the Hot Standby group. A new standby router is also selected at that time.

Devices that are running HSRP send and receive multicast UDP-based hello packets to detect router failure and to designate active and standby routers.

Previously, when HSRP was configured on an interface, ICMP redirect messages were disabled by default. With Cisco IOS Release 12.1(3)T, ICMP redirection on interfaces configured with HSRP are enabled by default. See the [“Enabling HSRP Support for ICMP Redirect Messages”](#) section later in this document for more information.

You can configure multiple Hot Standby groups on an interface, thereby making fuller use of redundant routers and load sharing. To do so, specify a group number for each Hot Standby command you configure for the interface.

**Note**

Token Ring interfaces allow up to three Hot Standby groups each, the group numbers being 0, 1, and 2.

**Note**

The Cisco 1000 series, Cisco 2500 series, Cisco 3000 series, Cisco 4000 series, and Cisco 4500 routers that use Lance Ethernet hardware do not support multiple Hot Standby groups on a single Ethernet interface. The Cisco 800 series, Cisco 1000 series, and Cisco 1600 series that use PQIICC Ethernet hardware do not support multiple Hot Standby groups on a single Ethernet interface. You can configure a workaround solution by using the **standby use-bia** interface configuration command, which uses the burned-in address of the interface as its virtual MAC address, instead of the preassigned MAC address.

HSRP is supported over Inter-Switch Link (ISL) encapsulation. Refer to the “Configuring Routing Between VLANs with ISL Encapsulation” chapter in the *Cisco IOS Switching Services Configuration Guide*.

With Cisco IOS Release 12.1(3)T, HSRP can provide support for a Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) interface. See the section “[Enabling HSRP Support for MPLS VPNs](#)” later in this chapter for more information.

To configure HSRP, perform the tasks described in the following sections. The tasks in the first section are required; the tasks in the remaining sections are optional.

- [Enabling HSRP](#) (Required)
- [Configuring HSRP Group Attributes](#) (Optional)
- [Changing the HSRP MAC Refresh Interval](#) (Optional)
- [Enabling HSRP MIB Traps](#) (Optional)
- [Enabling HSRP Support for MPLS VPNs](#) (Optional)
- [Enabling HSRP Support for ICMP Redirect Messages](#) (Optional)

For more information about HSRP and how to configure it on a Cisco router, see the chapter “Using HSRP for Fault-Tolerant IP Routing” in the *Cisco CCIE Fundamentals: Case Studies* publication.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the “Identifying Supported Platforms” section in the “Using Cisco IOS Software” chapter in this book.

Enabling HSRP

To enable the HSRP on an interface, use the following command in interface configuration mode:

Command	Purpose
Router(config-if)# standby [group-number] ip [ip-address [secondary]]	Enables the HSRP.

Configuring HSRP Group Attributes

To configure other Hot Standby group attributes that affect how the local router participates in HSRP, use the following commands in interface configuration mode as needed:

Command	Purpose
Router(config-if)# standby [<i>group-number</i>] timers [<i>msec</i>] <i>hellotime</i> [<i>msec</i>] <i>holdtime</i>	Configures the time between hello packets and the hold time before other routers declare the active router to be down.
Router(config-if)# standby [<i>group-number</i>] priority <i>priority</i>	Set the Hot Standby priority used in choosing the active router. The priority value range is from 1 to 255, where 1 denotes the lowest priority and 255 denotes the highest priority. Specify that, if the local router has priority over the current active router, the local router should attempt to take its place as the active router.
Router(config-if)# standby [<i>group-number</i>] preempt [delay { <i>minimum delay</i> <i>reload delay</i> <i>sync delay</i> }]	Configure a preemption delay, after which the Hot Standby router preempts and becomes the active router.
Router(config-if)# standby [<i>group-number</i>] track <i>type number</i> [<i>interface-priority</i>]	Configures the interface to track other interfaces, so that if one of the other interfaces goes down, the Hot Standby priority of the device is lowered.
Router(config-if)# standby [<i>group-number</i>] authentication text <i>string</i>	Selects an authentication string to be carried in all HSRP messages.
Router(config-if)# standby delay minimum <i>min-delay</i> reload <i>reload-delay</i>	Configures the delay period before the initialization of Hot Standby Router Protocol (HSRP) groups.
Router(config-if)# standby [<i>group-number</i>] mac-address <i>macaddress</i>	Specifies a virtual MAC address for the virtual router.
Router(config-if)# standby use-bia [<i>scope interface</i>]	Configures HSRP to use the burned-in address of an interface as its virtual MAC address instead of the preassigned MAC address (on Ethernet and FDDI) or the functional address (on Token Ring).

Changing the HSRP MAC Refresh Interval

When HSRP runs over FDDI, you can change the interval at which a packet is sent to refresh the MAC cache on learning bridges or switches. HSRP hello packets use the burned-in address (BIA) instead of the MAC virtual address. Refresh packets keep the MAC cache on switches and learning bridges current.

You can change the refresh interval on FDDI rings to a longer or shorter interval, thereby using bandwidth more efficiently. You can prevent the sending of any MAC refresh packets if you do not need them (if you have FDDI but do not have a learning bridge or switch). When changing the HSRP MAC refresh interval, be aware of the following guidelines:

- This feature applies to HSRP running over FDDI only.
- You need not configure the MAC refresh interval if you have the **standby use-bia** interface configuration command configured.

By default, a packet is sent every 10 seconds to refresh the MAC cache on learning bridges or switches. To change the interval, use the following command in interface configuration mode:

Command	Purpose
Router(config-if)# standby mac-refresh <i>seconds</i>	Changes the interval at which refresh packets are sent.

For examples of this feature, see the section “[HSRP MAC Refresh Interval Examples](#)” at the end of this chapter.

Enabling HSRP MIB Traps

With Cisco IOS Release 12.0(3)T, the software supports the HSRP Management MIB feature. HSRP MIB supports Simple Network Management Protocol (SNMP) Get operations, to allow network devices to get reports about HSRP groups in a network from the network management station.

Enabling HSRP MIB trap support is done from the command-line interface (CLI), and the MIB is used for getting the reports. A trap notifies the network management station when a router leaves or enters the active or standby state. When an entry is configured from the CLI, the RowStatus for that group in the MIB immediately goes to the active state.

The Cisco IOS software supports a read-only version of the MIB, and set operations are not supported.

This feature supports four MIB tables, as follows:

- cHsrpGrpEntry table defined in CISCO-HSRP-MIB.my
- cHsrpExtIfTrackedEntry, cHsrpExtSecAddrEntry, and cHsrpExtIfEntry defined in CISCO-HSRP-EXT-MIB.my

The cHsrpGrpEntry table consists of all the group information defined in RFC 2281, *Cisco Hot Standby Router Protocol*; the other tables consist of the Cisco extensions to RFC 2281, which are defined in CISCO-HSRP-EXT-MIB.my.

To enable HSRP MIB trap support, use the following commands in global configuration mode:

	Command	Purpose
Step 1	Router(config)# snmp-server enable traps hsrp	Enables the router to send SNMP traps and informs, and HSRP notifications.
Step 2	Router(config)# snmp-server host <i>host community-string hsrp</i>	Specifies the recipient of an SNMP notification operation, and that HSRP notifications be sent to the host.

See the section “[HSRP MIB Trap Example](#)” later in this chapter for an example of how to configure HSRP MIB trap support in your network. See the “Configuring SNMP” chapter in the *Cisco IOS Configuration Fundamentals Configuration Guide* for more information on configuring SNMP.

Enabling HSRP Support for MPLS VPNs

HSRP support on an MPLS VPN interface is useful when an Ethernet is connected between two provider edges (PEs) with either of the following conditions:

- A customer edge (CE) with a default route to the HSRP virtual IP address
- One or more hosts with the HSRP virtual IP address configured as the default gateway

Each VPN is associated with one or more VPN routing/forwarding (VRF) instances. A VRF consists of the following elements:

- IP routing table
- Cisco Express Forwarding (CEF) table
- Set of interfaces that use the CEF forwarding table
- Set of rules and routing protocol parameters to control the information in the routing tables

VPN routing information is stored in the IP routing table and the CEF table for each VRF. A separate set of routing and CEF tables is maintained for each VRF. These tables prevent information from being forwarded outside a VPN and also prevent packets that are outside a VPN from being forwarded to a router within the VPN.

HSRP currently adds ARP entries and IP hash table entries (aliases) using the default routing table instance. However, a different routing table instance is used when VRF forwarding is configured on an interface, causing ARP and ICMP echo requests for the HSRP virtual IP address to fail.

HSRP support for MPLS VPNs ensures that the HSRP virtual IP address is added to the correct IP routing table and not to the default routing table.

To configure this feature, perform the required tasks described in the following sections:

- [Defining VPNs](#) (Required)
- [Enabling HSRP](#) (Required)

Defining VPNs

To define VPNs, use the following commands on the PE routers beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# ip vrf <i>vrf-name</i>	Enters VRF configuration mode and assigns a VRF name.
Step 2	Router(config-vrf)# rd <i>route-distinguisher</i>	Creates routing and forwarding tables.
Step 3	Router(config-vrf)# route-target { import export both } <i>route-target-ext-community</i>	Creates a list of import or export route target communities for the specified VRF.
Step 4	Router(config-vrf)# exit	Exits the current configuration mode and enters global configuration mode.
Step 5	Router(config)# interface <i>type number</i>	Specifies an interface and enters interface configuration mode.
Step 6	Router(config-if)# ip vrf forwarding <i>vrf-name</i>	Associates a VRF with an interface or subinterface.

Enabling HSRP

To enable the HSRP on an interface, use the following command in interface configuration mode:

Command	Purpose
Router (config-if)# standby [<i>group-number</i>] ip <i>ip-address</i>	Enables the HSRP.

Verifying HSRP Support for MPLS VPNs

The following example shows how to use **show EXEC** commands to verify that the HSRP virtual IP address is in the correct ARP and CEF tables:

```
Router# show ip arp vrf vrf1

Protocol Address          Age (min) Hardware Addr  Type   Interface
→ Internet 10.2.0.1             -          00d0.bbd3.bc22 ARPA   Ethernet0/2
Internet 10.2.0.20           -          0000.0c07.ac01 ARPA   Ethernet0/2

Router# show ip cef vrf vrf1

Prefix          Next Hop          Interface
0.0.0.0/0       10.3.0.4          Ethernet0/3
0.0.0.0/32      receive
10.1.0.0/16     10.2.0.1          Ethernet0/2
10.2.0.0/16     attached          Ethernet0/2
10.2.0.1/32     receive
→ 10.2.0.20/32   receive
224.0.0.0/24    receive
255.255.255.255/32 receive
```

Enabling HSRP Support for ICMP Redirect Messages

Previously, ICMP redirect messages were automatically disabled on interfaces configured with HSRP. ICMP is a network layer Internet protocol that provides message packets to report errors and other information relevant to IP processing. ICMP provides many diagnostic functions and can send and redirect error packets to the host. See the section [“Enabling ICMP Redirect Messages”](#) earlier in this chapter for more information on ICMP redirect messages.

When running HSRP, it is important to prevent hosts from discovering the interface (or real) MAC addresses of routers in the HSRP group. If a host is redirected by ICMP to the real MAC address of a router, and that router later fails, then packets from the host will be lost.

With Cisco IOS Release 12.1(3)T and later, ICMP redirect messages are automatically enabled on interfaces configured with HSRP. This functionality works by filtering outgoing ICMP redirect messages through HSRP, where the next hop IP address may be changed to an HSRP virtual IP address.

Redirects to Active HSRP Routers

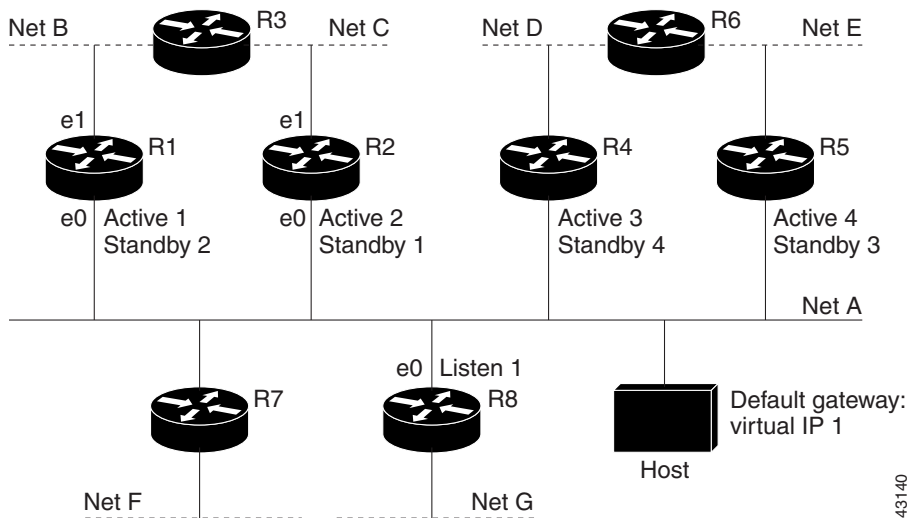
The next-hop IP address is compared to the list of active HSRP routers on that network; if a match is found, then the real next-hop IP address is replaced with a corresponding virtual IP address and the redirect message is allowed to continue.

If no match is found, then the ICMP redirect message is sent only if the router corresponding to the new next hop IP address is not running HSRP. Redirects to passive HSRP routers are not allowed (a passive HSRP router is a router running HSRP, but which contains no active HSRP groups on the interface).

For optimal operation, every router in a network that is running HSRP should contain at least one active HSRP group on an interface to that network. Every HSRP router need not be a member of the same group. Each HSRP router will snoop on all HSRP packets on the network to maintain a list of active routers (virtual IP addresses versus real IP addresses).

Consider the network shown in [Figure 18](#), which supports the HSRP ICMP redirection filter.

Figure 18 Network Supporting the HSRP ICMP Redirection Filter



43140

If the host wants to send a packet to another host on Net D, then it first sends it to its default gateway, the virtual IP address of HSRP group 1.

The following is the packet received from the host:

```
dest MAC      = HSRP group 1 virtual MAC
source MAC    = Host MAC
dest IP       = host-on-netD IP
source IP     = Host IP
```

Router R1 receives this packet and determines that router R4 can provide a better path to Net D, so it prepares to send a redirect message that will redirect the host to the real IP address of router R4 (because only real IP addresses are in its routing table).

The following is the initial ICMP redirect message sent by router R1:

```
dest MAC      = Host MAC
source MAC    = router R1 MAC
dest IP       = Host IP
source IP     = router R1 IP
gateway to use = router R4 IP
```

Before this redirect occurs, the HSRP process of router R1 determines that router R4 is the active HSRP router for group 3, so it changes the next hop in the redirect message from the real IP address of router R4 to the virtual IP address of group 3. Furthermore, it determines from the destination MAC address of the packet that triggered the redirect message that the host used the virtual IP address of group 1 as its gateway, so it changes the source IP address of the redirect message to the virtual IP address of group 1.

The modified ICMP redirect message showing the two modified fields (*) is as follows:

```
dest MAC      = Host MAC
source MAC    = router R1 MAC
dest IP       = Host IP
source IP*    = HSRP group 1 virtual IP
gateway to use* = HSRP group 3 virtual IP
```

This second modification is necessary because hosts compare the source IP address of the ICMP redirect message with their default gateway. If these addresses do not match, the ICMP Redirect message is ignored. The routing table of the host now consists of the default gateway, virtual IP address of group 1, and a route to Net D through the virtual IP address of group 3.

Redirects to Passive HSRP Routers

Redirects to passive HSRP routers are not permitted. Redundancy may be lost if hosts learn the real IP addresses of HSRP routers.

In the previous example, redirects to router R8 are not allowed because R8 is a passive HSRP router. In this case, packets from the host to Net D will first go to router R1 and then be forwarded to router R4, that is, they will traverse the network twice.

A network configuration with passive HSRP routers is considered a misconfiguration. For HSRP ICMP redirection to operate optimally, every router on the network that is running HSRP should contain at least one active HSRP group.

Redirects to Non-HSRP Routers

Redirects to routers not running HSRP on their local interface are permitted. No redundancy is lost if hosts learn the real IP address of non-HSRP routers.

In the example, redirection to router R7 is allowed because R7 is not running HSRP. In this case, the next hop IP address is unchanged. The source IP address is changed dependent upon the destination MAC address of the original packet. You can specify the **no standby redirects unknown** command to stop these redirects from being sent.

Passive HSRP Router Advertisements

Passive HSRP routers send out HSRP advertisement messages both periodically and when entering or leaving the passive state. Thus, all HSRP routers can determine the HSRP group state of any HSRP router on the network. These advertisements inform other HSRP routers on the network of the HSRP interface state, as follows:

- **Dormant**—Interface has no HSRP groups, single advertisements sent once when last group is removed
- **Passive**—Interface has at least one non-active group and no active groups, advertisements sent out periodically
- **Active**—Interface has at least one active group, single advertisement sent out when first group becomes active

You can adjust the advertisement interval and holddown time using the **standby redirects timers** command.

Redirects Not Sent

If the HSRP router cannot uniquely determine the IP address used by the host when it sends the packet that caused the redirect, the redirect message will not be sent. The router uses the destination MAC address in the original packet to make this determination. In certain configurations, such as the use of the **standby use-bia** interface configuration command specified on an interface, redirects cannot be sent. In this case, the HSRP groups use the interface MAC address as their virtual MAC address. The router now cannot determine if the default gateway of the host is the real IP address or one of the HSRP virtual IP addresses that are active on the interface.

Using HSRP with ICMP redirects is not possible in the Cisco 800 series, Cisco 1000 series, Cisco 1600 series, Cisco 2500 series, Cisco 3000 series, and Cisco 4500 series routers because the Ethernet controller can only support one MAC address.

The IP source address of an ICMP packet must match the gateway address used by the host in the packet that triggered the ICMP packet, otherwise the host will reject the ICMP redirect packet. An HSRP router uses the destination MAC address to determine the gateway IP address of the host. If the HSRP router is using the same MAC address for multiple IP addresses then it is not possible to uniquely determine the gateway IP address of the host and the redirect message is not sent.

The following is sample output from the **debug standby events icmp EXEC** command if HSRP could not uniquely determine the gateway used by the host:

```
10:43:08: SB: ICMP redirect not sent to 20.0.0.4 for dest 30.0.0.2
10:43:08: SB: could not uniquely determine IP address for mac 00d0.bbd3.bc22
```

Configuring HSRP Support for ICMP Redirect Messages

By default, HSRP filtering of ICMP redirect messages is enabled on routers running HSRP. To reenabling this feature on your router if it is disabled, use the following command in interface configuration mode:

Command	Purpose
Router (config-if)# standby redirects [enable disable] [timers advertisement holdown] [unknown]	Enables HSRP filtering of ICMP redirect messages

Configuring IP Accounting

Cisco IP accounting support provides basic IP accounting functions. By enabling IP accounting, users can see the number of bytes and packets switched through the Cisco IOS software on a source and destination IP address basis. Only transit IP traffic is measured and only on an outbound basis; traffic generated by the software or terminating in the software is not included in the accounting statistics. To maintain accurate accounting totals, the software maintains two accounting databases: an active and a checkpointed database.

Cisco IP accounting support also provides information identifying IP traffic that fails IP access lists. Identifying IP source addresses that violate IP access lists alerts you to possible attempts to breach security. The data also indicates that you should verify IP access list configurations. To make this feature available to users, you must enable IP accounting of access list violations using the **ip accounting access-violations** interface configuration command. Users can then display the number of bytes and packets from a single source that attempted to breach security against the access list for the source destination pair. By default, IP accounting displays the number of packets that have passed access lists and were routed.

To enable IP accounting, use one of the following commands for each interface in interface configuration mode:

Command	Purpose
Router(config-if)# ip accounting	Enables basic IP accounting.
Router(config-if)# ip accounting access-violations	Enables IP accounting with the ability to identify IP traffic that fails IP access lists.

To configure other IP accounting functions, use the following commands in global configuration mode, as needed:

Command	Purpose
Router(config)# ip accounting-threshold <i>threshold</i>	Sets the maximum number of accounting entries to be created.
Router(config)# ip accounting-list <i>ip-address wildcard</i>	Filters accounting information for hosts.
Router(config)# ip accounting-transits <i>count</i>	Controls the number of transit records that will be stored in the IP accounting database.

To display IP access violations for a specific IP accounting database, use the following command in EXEC mode:

Command	Purpose
Router# show ip accounting [checkpoint] access-violations	Displays IP access violation information.

To display IP access violations, include the **access-violations** keyword in the **show ip accounting** EXEC command. If you do not specify the keyword, the command defaults to displaying the number of packets that have passed access lists and were routed. The access violations output displays the number of the access list failed by the last packet for the source and destination pair. The number of packets reveals how aggressive the attack is upon a specific destination.

Use the **show ip accounting** EXEC command to display the active accounting database, and traffic coming from a remote site and transiting through a router. To display the checkpointed database, use the **show ip accounting checkpoint** EXEC command. The **clear ip accounting** EXEC command clears the active database and creates the checkpointed database.

Configuring IP MAC Accounting

The MAC address accounting functionality provides accounting information for IP traffic based on the source and destination MAC addresses on LAN interfaces. MAC accounting calculates the total packet and byte counts for a LAN interface that receives or sends IP packets to or from a unique MAC address. It also records a timestamp for the last packet received or sent. For example, with IP MAC accounting, you can determine how much traffic is being sent to and/or received from various peers at NAPS/peering points. IP MAC accounting is supported on Ethernet, FastEthernet, and FDDI interfaces and supports Cisco Express Forwarding (CEF), distributed CEF (dCEF), flow, and optimum switching.

To configure the interface for IP accounting based on the MAC address, perform the following steps beginning in global configuration:

	Command	Purpose
Step 1	Router(config)# interface <i>type number</i>	Specifies the interface and enters interface configuration mode.
Step 2	Router(config-if)# ip accounting mac-address { input output }	Configures IP accounting based on the MAC address of received (input) or transmitted (output) packets

To remove IP accounting based on the MAC address from the interface, use the **no ip accounting mac-address** command.

Use the EXEC command **show interface mac** to display MAC accounting information for interfaces configured for MAC accounting.

Configuring IP Precedence Accounting

The precedence accounting feature provides accounting information for IP traffic based on the precedence on any interface. This feature calculates the total packet and byte counts for an interface that receives or sends IP packets and sorts the results based on IP precedence. This feature is supported on all interfaces and subinterfaces and supports CEF, dCEF, flow, and optimum switching.

To configure the interface for IP accounting based on IP precedence, perform the following steps beginning in global configuration model:

	Command	Purpose
Step 1	Router(config)# interface <i>type number</i>	Specifies the interface (or subinterface) and enters interface configuration mode.
Step 2	Router(config-if)# ip accounting precedence { input output }	Configures IP accounting based on the precedence of received (input) or transmitted (output) packets

To remove IP accounting based on IP precedence from the interface, use the **no ip accounting precedence** command.

Use the EXEC command **show interface precedence** to display precedence accounting information for interfaces configured for precedence accounting.

Configuring TCP Performance Parameters

To tune IP performance, perform any of the optional tasks described in the following sections. To configure various switching options, refer to the “Cisco IOS Switching Paths” chapter in the *Cisco IOS Switching Services Configuration Guide*.

- [Compressing TCP Packet Headers](#) (Optional)
- [Setting the TCP Connection Attempt Time](#) (Optional)
- [Enabling TCP Path MTU Discovery](#) (Optional)
- [Enabling TCP Selective Acknowledgment](#) (Optional)
- [Enabling TCP Time Stamp](#) (Optional)
- [Setting the TCP Maximum Read Size](#) (Optional)
- [Setting the TCP Window Size](#) (Optional)
- [Setting the TCP Outgoing Queue Size](#) (Optional)

Compressing TCP Packet Headers

You can compress the headers of your TCP/IP packets in order to reduce their size, thereby increasing performance. Header compression is particularly useful on networks with a large percentage of small packets (such as those supporting many Telnet connections). To enable TCP header compression, use the following command in interface configuration mode:

Command	Purpose
Router(config-if)# ip tcp header-compression [passive]	Enables TCP header compression.

The **ip tcp header-compression** interface configuration command only compresses the TCP header; it has no effect on UDP packets or other protocol headers. The TCP header compression technique is supported on serial lines using High-Level Data Link Control (HDLC) or PPP encapsulation. You must enable compression on both ends of a serial connection.

By using the **passive** keyword, you can optionally specify outgoing packets to be compressed only if TCP incoming packets on the same interface are compressed. If you specify the command without the **passive** keyword, the software will compress all traffic. Without the command, the default is no compression.



Note

Fast processors can handle several fast interfaces, such as T1 lines, that are running header compression. However, you should think carefully about the traffic characteristics of your network before compressing TCP headers. You might want to use the monitoring commands to compare network utilization before and after enabling TCP header compression.

Expressing TCP Header Compression

Before Cisco IOS Release 12.0(7)T, if compression of TCP headers was enabled, compression was performed in the process switching path. Compression performed in the process switching path meant that packets traversing interfaces that had TCP header compression enabled were queued and passed up to the process to be switched. This procedure slowed down transmission of the packet, and therefore some users preferred to fast switch uncompressed TCP packets.

In Cisco IOS Release 12.1, if TCP header compression is enabled, it occurs by default in the fast-switched path or the CEF-switched path, depending on which switching method is enabled on the interface.

If neither fast switching nor CEF switching is enabled, then if TCP header compression is enabled, it will occur in the process-switched path as before.

The Express TCP Header Compression feature reduces network overhead and speeds up transmission of TCP packets. The faster speed provides a greater benefit on slower links than faster links.

In order for Express TCP Header Compression to work, the following conditions must be in place:

- CEF switching or fast switching must be enabled on the interface.
- HDLC, PPP, or Frame Relay encapsulation must be configured.
- TCP header compression must be enabled.

The CEF and fast-switching aspects of the Express TCP Header Compression feature are related to these documents:

- *Cisco IOS Switching Services Configuration Guide*
- *Cisco IOS Switching Services Command Reference*

For information about compressing RTP headers, see the chapter “Configuring IP Multicast Routing” in this document.

Changing the Number of TCP Header Compression Connections

You also can specify the total number of header compression connections that can exist on an interface. You should configure one connection for each TCP connection through the specified interface.

When specifying the total number of header compression connections that can exist on an interface, be aware of the following conditions:

- By default, for Frame Relay encapsulation, there can be 256 TCP header compression connections (128 calls). The maximum value is fixed, not configurable.
- By default, for PPP or HDLC encapsulation, the software allows 32 TCP header compression connections (16 calls). This default can be increased to a maximum of 256 TCP header compression connections.

To specify the number of connections, use the following command in interface configuration mode:

Command	Purpose
Router(config-if)# ip tcp compression-connections <i>number</i>	Specifies the total number of TCP header compression connections that can exist on an interface.

Setting the TCP Connection Attempt Time

You can set the amount of time the Cisco IOS software will wait to attempt to establish a TCP connection. Because the connection attempt time is a host parameter, it does not pertain to traffic going through the device, just to traffic originated at the device.

To set the TCP connection attempt time, use the following command in global configuration mode:

Command	Purpose
Router(config)# ip tcp synwait-time <i>seconds</i>	Sets the amount of time the Cisco IOS software will wait to attempt to establish a TCP connection. The default is 30 seconds.

Enabling TCP Path MTU Discovery

Path MTU Discovery is a method for maximizing the use of available bandwidth in the network between the endpoints of a TCP connection, and is described in RFC 1191. By default, this feature is disabled. Existing connections are not affected when this feature is turned on or off.

To enable Path MTU Discovery, use the following command in global configuration mode:

Command	Purpose
Router(config)# ip tcp path-mtu-discovery [age-timer { <i>minutes</i> infinite }]	Enables Path MTU Discovery.

Customers using TCP connections to move bulk data between systems on distinct subnets would benefit most by enabling this feature. Customers using remote source-route bridging (RSRB) with TCP encapsulation, serial tunnel (STUN), X.25 Remote Switching (also known as XOT or X.25 over TCP), and some protocol translation configurations might also benefit from enabling this feature.

The **ip tcp path-mtu-discovery** global configuration command is to enable Path MTU Discovery for connections initiated by the router when it is acting as a host. For a discussion of how the Cisco IOS software supports Path MTU Discovery when the device is acting as a router, see the section “[Understanding Path MTU Discovery](#)” earlier in this chapter.

The age-timer is a time interval for how often TCP should reestimate the path MTU with a larger maximum segment size (MSS). The default Path MTU Discovery age-timer is 10 minutes; its maximum is 30 minutes. You can turn off the age timer by setting it to infinite.

Enabling TCP Selective Acknowledgment

The TCP selective acknowledgment feature improves performance in the event that multiple packets are lost from one TCP window of data.

Prior to this feature, with the limited information available from cumulative acknowledgments, a TCP sender could learn about only one lost packet per round-trip time. An aggressive sender could choose to resend packets early, but such re-sent segments might have already been successfully received.

The TCP selective acknowledgment mechanism helps improve performance. The receiving TCP host returns selective acknowledgment packets to the sender, informing the sender of data that have been received. In other words, the receiver can acknowledge packets received out of order. The sender can then resend only the missing data segments (instead of everything since the first missing packet).

Prior to selective acknowledgment, if TCP lost packets 4 and 7 out of an 8-packet window, TCP would receive acknowledgment of only packets 1, 2, and 3. Packets 4 through 8 would need to be re-sent. With selective acknowledgment, TCP receives acknowledgment of packets 1, 2, 3, 5, 6, and 8. Only packets 4 and 7 must be re-sent.

Refer to RFC 2018 for more detailed information on TCP selective acknowledgment.

The feature is used only when multiple packets are dropped within one TCP window. There is no performance impact when the feature is enabled but not used. To enable TCP selective acknowledgment, use the following command in global configuration mode:

Command	Purpose
Router(config)# ip tcp selective-ack	Enables TCP selective acknowledgment.

Enabling TCP Time Stamp

The TCP time-stamp option provides better TCP round-trip time measurements. Because the time stamps are always sent and echoed in both directions and the time-stamp value in the header is always changing, TCP header compression will not compress the outgoing packet. To allow TCP header compression over a serial link, the TCP time-stamp option is disabled.

Refer to RFC 1323 for more detailed information on TCP time stamp.

To enable TCP time stamp, use the following command in global configuration mode:

Command	Purpose
Router(config)# ip tcp timestamp	Enables TCP time stamp.

If you want to use TCP header compression over a serial line, TCP time stamp and TCP selective acknowledgment must be disabled. Both features are disabled by default. To disable TCP selective acknowledgment once it is enabled, see the previous [“Enabling TCP Selective Acknowledgment”](#) section.

Setting the TCP Maximum Read Size

By default, for Telnet and rlogin, the maximum number of characters that TCP reads from the input queue at once is a very large number (the largest possible 32-bit positive number). We do not recommend that you change this value. However, to change that value, use the following command in global configuration mode:

Command	Purpose
Router(config)# ip tcp chunk-size <i>characters</i>	Sets the TCP maximum read size for Telnet or rlogin.

Setting the TCP Window Size

The default TCP window size is 2144 bytes. We recommend you keep the default value unless you know your router is sending large packets (greater than 536 bytes). To change the default window size, use the following command in global configuration mode:

Command	Purpose
Router(config)# ip tcp window-size <i>bytes</i>	Sets the TCP window size.

Setting the TCP Outgoing Queue Size

The default TCP outgoing queue size per connection is 5 segments if the connection has a TTY associated with it (like a Telnet connection). If no TTY connection is associated with it, the default queue size is 20 segments. To change the 5-segment default value, use the following command in global configuration mode:

Command	Purpose
Router(config)# <code>ip tcp queue-max packets</code>	Sets the TCP outgoing queue size.

Configuring IP over WANs

You can configure IP over X.25, Switched Multimegabit Data Service (SMDS), Frame Relay, and dial-on-demand routing (DDR) networks. When configuring IP over X.25, SMDS, or Frame Relay, configure the address mappings as described in the appropriate chapters of the *Cisco IOS Wide-Area Networking Configuration Guide*. For DDR, refer to the “Preparing to Configure DDR” chapter of the *Cisco IOS Dial Technologies Configuration Guide* publication.

Configuring the MultiNode Load Balancing Forwarding Agent

The MultiNode Load Balancing (MNLB) Forwarding Agent is the Cisco IOS-based packet redirector component of the MNLD Feature Set for LocalDirector, a product in the Cisco family of load balancing solutions.

The Forwarding Agent discovers the destination of specific connection requests and forwards packets between the client and the chosen destination. When a Forwarding Agent receives a connection request, the request is forwarded to the MNLB services manager, the LocalDirector-based component of the MNLD Feature Set for LocalDirector. The services manager makes the load-balancing decision and sends the Forwarding Agent the optimal destination. After the destination is specified, session data is forwarded directly to the destination by the Forwarding Agent, without further services manager participation. There is no limit to the number of Forwarding Agents that can be configured in the MNLD Feature Set for LocalDirector.

The MNLD Feature Set for LocalDirector comprises hardware and software that runs on multiple network components. The services manager runs on the Cisco LocalDirector chassis and makes the load-balancing decisions. The Forwarding Agents run on Cisco IOS router and switch platforms and forward packets to and from the selected destination. Separating the decision-making and packet-forwarding tasks enables much faster packet throughput. The underlying Cisco architecture, ContentFlow architecture, enables the following features:

- High availability
- Unbounded scalability
- Application-aware balancing
- No single point of failure
- Unmatched performance

Configure the Forwarding Agent only if you are installing the MNLD Feature Set for LocalDirector. If you are installing the MNLD Feature Set for LocalDirector, refer to the *MultiNode Load Balancing Feature Set for LocalDirector User Guide* for information about which other hardware and software components are required.

The MNLB Forwarding Agent is an implementation of the Cisco ContentFlow architecture flow delivery agent (FDA).

Refer to the *MultiNode Load Balancing Feature Set for LocalDirector User Guide* for more information about how the Forwarding Agent is configured and for more information about the product.

MNLB Forwarding Agent Configuration Task List

To configure the MNLB Forwarding Agent, perform the tasks described in the following sections. The tasks are all required except for the task in the second section, which is optional but strongly recommended.

- [Enabling CEF](#) (Required)
- [Enabling NetFlow Switching](#) (Optional but strongly recommended)
- [Enabling IP Multicast Routing](#) (Required)
- [Configuring the Router as a Forwarding Agent](#) (Required)

Enabling CEF

CEF is advanced Layer 3 IP switching technology. CEF optimizes network performance and scalability for networks with large and dynamic traffic patterns, such as the Internet, on networks characterized by intensive Web-based applications, or interactive sessions.

To enable CEF, use the following command in global configuration mode:

Command	Purpose
Router(config)# ip cef distributed	Enables CEF.



Note

When you enable CEF globally, all interfaces that support CEF are enabled by default. If you want to turn off CEF on a particular interface, you can do so.

Refer to the “Cisco Express Forwarding” part of the *Cisco IOS Switching Services Configuration Guide* for more information on how to configure CEF.

Enabling NetFlow Switching

You must enable NetFlow switching on all interfaces that will carry ContentFlow traffic. To enable NetFlow switching, use the following commands beginning in interface configuration mode:

	Command	Purpose
Step 1	Router(config-if)# interface <i>type</i> <i>slot/port-adapter/port</i> (Cisco 7500 series routers) or Router(config-if)# interface <i>type slot/port</i> (Cisco 7200 series routers)	Specifies the interface, and enters interface configuration mode.
Step 2	Router(config-if)# ip route-cache flow	Enables flow switching on the interface.

Normally the size of the NetFlow cache will meet your needs. To increase or decrease the number of entries maintained in the cache, use the following command in global configuration mode:

Command	Purpose
Router(config)# ip flow-cache entries <i>number</i>	Changes the number of entries maintained in the NetFlow cache. The number of entries can be from 1024 to 524288. The default is 64536.

Refer to the “Netflow Switching” part of the *Cisco IOS Switching Services Configuration Guide* for more information on how to configure NetFlow switching.

Enabling IP Multicast Routing

You must enable IP multicast routing on all interfaces to the services manager.

To enable multicast routing on all interfaces, use the following command in global configuration mode:

Command	Purpose
Router(config)# ip multicast routing	Enables multicast routing.

To have the router join a multicast group and enable IGMP, use the following command in interface configuration mode:

Command	Purpose
Router(config-if)# ip igmp join-group <i>group-address</i>	Joins a multicast group. This command must be configured on all interfaces that will listen for the services manager multicasts. The group address must match that configured within the services manager configuration.

See the “Configuring IP Multicast Routing” chapter of this document for more information on how to configure IP multicast routing.

Configuring the Router as a Forwarding Agent

To configure the router as a Forwarding Agent, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# ip casa <i>control-address igmp-address</i>	Specifies the IP address and IGMP address of the Forwarding Agent. The recommended IGMP address is 224.0.1.2.
Step 2	Router(config-casa)# forwarding-agent pools <i>initial-affinity-pool max-affinity-pool</i>	Adjusts the memory allocated for the affinity pools of the Forwarding Agent. The default pool size is 5000, and there is no maximum pool size.
Step 3	Router(config-casa)# forwarding-agent <i>port-number</i> [<i>password [timeout]</i>]	Specifies the port number. The default is port 1637.



Note

The Forwarding Agent IGMP address and port must match the IGMP address and port configured on the services manager using the **ip igmp join-group** interface configuration command.

Monitoring and Maintaining the IP Network

To monitor and maintain your network, perform any of the optional tasks described in the following sections:

- [Clearing Caches, Tables, and Databases](#) (Optional)
- [Monitoring and Maintaining the DRP Server Agent](#) (Optional)
- [Clearing the Access List Counters](#) (Optional)
- [Displaying System and Network Statistics](#) (Optional)
- [Monitoring the MNLB Forwarding Agent](#) (Optional)
- [Monitoring and Maintaining HSRP Support for ICMP Redirect Messages](#) (Optional)

Clearing Caches, Tables, and Databases

You can remove all contents of a particular cache, table, or database. Clearing a cache, table, or database can become necessary when the contents of the particular structure have become or are suspected to be invalid.

To clear caches, tables, and databases, use the following commands in EXEC mode, as needed:

Command	Purpose
Router# clear ip accounting [checkpoint]	Clears the active IP accounting or checkpointed database when IP accounting is enabled.
Router# clear tcp statistics	Clears TCP statistics.

Monitoring and Maintaining the DRP Server Agent

To monitor and maintain the DRP Server Agent, use the following commands in EXEC mode:

Command	Purpose
Router# clear ip drp	Clears statistics being collected on DRP requests and responses.
Router# show ip drp	Displays information about the DRP Server Agent.

Clearing the Access List Counters

The system counts how many packets pass each line of an access list; the counters are displayed by the **show access-lists** EXEC command. To clear the counters of an access list, use the following command in EXEC mode:

Command	Purpose
Router# clear access-list counters {access-list-number access-list-name}	Clears the access list counters.

Displaying System and Network Statistics

You can display specific statistics such as the contents of IP routing tables, caches, and databases. The resulting information can be used to determine resource utilization and to solve network problems.

To display specific statistics such as the contents of IP routing tables, caches, and databases, use the following commands in privileged EXEC mode, as needed. Refer to the “IP Services Commands” chapter in the *Cisco IOS IP Command Reference, Volume 1 of 3: Addressing and Services* for details about the commands listed in these tasks.

Command	Purpose
Router# show access-lists [access-list-number access-list-name]	Displays the contents of one or all current access lists.
Router# show access-list compiled	Displays information regarding compiled access lists, including the state of each compiled access list.
Router# show ip access-list [access-list-number name]	Displays the contents of current IP access lists.
Router# show ip accounting [checkpoint]	Displays the active IP accounting or checkpointed database.

Command	Purpose
Router# show ip redirects	Displays the address of the default router and the address of hosts for which an ICMP redirect message has been received.
Router# show ip sockets	Displays IP socket information.
Router# show ip tcp header-compression	Displays statistics on TCP header compression.
Router# show ip traffic	Displays IP protocol statistics.
Router# show standby [<i>interface</i> [<i>group</i>]] [active init listen standby][brief]	Displays the status of the standby router.
Router# show standby delay [<i>type number</i>]	Displays HSRP information about delay periods
Router# show tcp statistics	Displays TCP statistics.

Monitoring the MNLB Forwarding Agent

To monitor the status of the Forwarding Agent, use the following commands in EXEC mode, as needed:

Command	Purpose
Router# show ip casa affinities	Displays the status of affinities.
Router# show ip casa oper	Displays the operational status of the Forwarding Agent.
Router# show ip casa stats	Displays statistical information about the Forwarding Agent.
Router# show ip casa wildcard	Displays information about wildcard blocks.

Monitoring and Maintaining HSRP Support for ICMP Redirect Messages

To display the status of ICMP redirect messages, use the following commands in EXEC mode, as needed:

Command	Purpose
Router# debug standby events icmp	Displays debug messages for HSRP-filtered ICMP redirect messages.
Router# debug ip icmp	Displays information on ICMP transactions.

IP Services Configuration Examples

This section provides the following IP configuration examples:

- [ICMP Services Example](#)
- [Simplex Ethernet Interfaces Example](#)
- [DRP Server Agent Example](#)
- [Numbered Access List Examples](#)
- [Named Access List Example](#)
- [IP Extended Access List with Fragment Control Example](#)

- [Time Range Applied to an IP Access List Example](#)
- [Commented IP Access List Entry Examples](#)
- [IP Accounting Example](#)
- [HSRP Load Sharing Example](#)
- [HSRP MAC Refresh Interval Examples](#)
- [HSRP MIB Trap Example](#)
- [HSRP Support for MPLS VPNs Example](#)
- [HSRP Support for ICMP Redirect Messages Example](#)
- [MNLB Forwarding Agent Examples](#)

ICMP Services Example

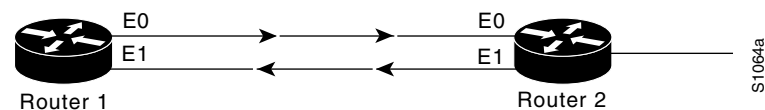
The following example changes some of the ICMP defaults for the first Ethernet interface 0. Disabling the sending of redirects could mean that you do not expect your devices on this segment to ever need to send a redirect message. Disabling the unreachable messages will have a secondary effect—it also will disable IP Path MTU Discovery, because path discovery works by having the Cisco IOS software send Unreachables messages. If you have a network segment with a small number of devices and an absolutely reliable traffic pattern—which could easily happen on a segment with a small number of little-used user devices—you would be disabling options that your device would be unlikely to use anyway.

```
interface ethernet 0
  no ip unreachable
  no ip redirects
```

Simplex Ethernet Interfaces Example

The following is an example of configuring a simplex Ethernet interface. [Figure 19](#) illustrates how to configure IP on two routers sharing transmit-only and receive-only Ethernet connections.

Figure 19 Simplex Ethernet Connections



Router 1 Configuration

```
interface ethernet 0
  ip address 128.9.1.1
  !
interface ethernet 1
  ip address 128.9.1.1
  transmit-interface ethernet 0
  !
!use show interfaces command to find router2-MAC-address-E0
arp 128.9.1.4 router2-MAC-address-E0 arpa
```

Router 2 Configuration

```
interface ethernet 0
```

```

ip address 128.9.1.2
transmit-interface ethernet 1
!
interface ethernet 1
ip address 128.9.1.2
!
!use show interfaces command to find router1-MAC-address-E1
arp 128.9.1.1 router1-MAC-address-E1 arpa

```

DRP Server Agent Example

The following example enables the DRP Server Agent. Sources of DRP queries are limited by access list 1, which permits only queries from the host at address 33.45.12.4. Authentication is also configured for the DRP queries and responses.

```

ip drp server
access-list 1 permit 33.45.12.4
ip drp access-group 1
ip drp authentication key-chain mktg
key chain mktg
key 1
key-string internal

```

Numbered Access List Examples

In the following example, network 36.0.0.0 is a Class A network whose second octet specifies a subnet; that is, its subnet mask is 255.255.0.0. The third and fourth octets of a network 36.0.0.0 address specify a particular host. Using access list 2, the Cisco IOS software would accept one address on subnet 48 and reject all others on that subnet. The last line of the list shows that the software would accept addresses on all other network 36.0.0.0 subnets.

```

access-list 2 permit 36.48.0.3
access-list 2 deny 36.48.0.0 0.0.255.255
access-list 2 permit 36.0.0.0 0.255.255.255
interface ethernet 0
ip access-group 2 in

```

The following example defines access lists 1 and 2, both of which have logging enabled:

```

interface ethernet 0
ip address 1.1.1.1 255.0.0.0
ip access-group 1 in
ip access-group 2 out
!
access-list 1 permit 5.6.0.0 0.0.255.255 log
access-list 1 deny 7.9.0.0 0.0.255.255 log
!
access-list 2 permit 1.2.3.4 log
access-list 2 deny 1.2.0.0 0.0.255.255 log

```

If the interface receives 10 packets from 5.6.7.7 and 14 packets from 1.2.23.21, the first log will look like the following:

```

list 1 permit 5.6.7.7 1 packet
list 2 deny 1.2.23.21 1 packet

```

Five minutes later, the console will receive the following log:

```

list 1 permit 5.6.7.7 9 packets
list 2 deny 1.2.23.21 13 packets

```

Turbo Access Control List Example

The following is a Turbo ACL configuration example. The **access-list compiled** global configuration command output indicates that Turbo ACL is enabled.

```
interface Ethernet2/7
  no ip address
  ip access-group 20 out
  no ip directed-broadcast
  shutdown
!
no ip classless
ip route 192.168.0.0 255.255.255.0 10.1.1.1
!
access-list compiled
access-list 1 deny any
access-list 2 deny 192.168.0.0 0.0.0.255
access-list 2 permit any
```

Implicit Masks in Access Lists Examples

IP access lists contain *implicit* masks. For instance, if you omit the mask from an associated IP host address access list specification, 0.0.0.0 is assumed to be the mask. Consider the following example configuration:

```
access-list 1 permit 0.0.0.0
access-list 1 permit 131.108.0.0
access-list 1 deny 0.0.0.0 255.255.255.255
```

For this example, the following masks are implied in the first two lines:

```
access-list 1 permit 0.0.0.0 0.0.0.0
access-list 1 permit 131.108.0.0 0.0.0.0
```

The last line in the configuration (using the **deny** keyword) can be left off, because IP access lists implicitly *deny* all other access. Leaving off the last line in the configuration is equivalent to finishing the access list with the following command statement:

```
access-list 1 deny 0.0.0.0 255.255.255.255
```

The following access list only allows access for those hosts on the three specified networks. It assumes that subnetting is not used; the masks apply to the host portions of the network addresses. Any hosts with a source address that does not match the access list statements will be rejected.

```
access-list 1 permit 192.5.34.0 0.0.0.255
access-list 1 permit 128.88.0.0 0.0.255.255
access-list 1 permit 36.0.0.0 0.255.255.255
! (Note: all other access implicitly denied)
```

To specify a large number of individual addresses more easily, you can omit the address mask that is all 0s from the **access-list** global configuration command. Thus, the following two configuration commands are identical in effect:

```
access-list 2 permit 36.48.0.3
access-list 2 permit 36.48.0.3 0.0.0.0
```

Extended Access List Examples

In the following example, the first line permits any incoming TCP connections with destination ports greater than 1023. The second line permits incoming TCP connections to the Simple Mail Transfer Protocol (SMTP) port of host 128.88.1.2. The last line permits incoming ICMP messages for error feedback.

```
access-list 102 permit tcp 0.0.0.0 255.255.255.255 128.88.0.0 0.0.255.255 gt 1023
access-list 102 permit tcp 0.0.0.0 255.255.255.255 128.88.1.2 0.0.0.0 eq 25
access-list 102 permit icmp 0.0.0.0 255.255.255.255 128.88.0.0 255.255.255.255
interface ethernet 0
 ip access-group 102 in
```

For another example of using an extended access list, suppose you have a network connected to the Internet, and you want any host on an Ethernet to be able to form TCP connections to any host on the Internet. However, you do not want IP hosts to be able to form TCP connections to hosts on the Ethernet except to the mail (SMTP) port of a dedicated mail host.

SMTP uses TCP port 25 on one end of the connection and a random port number on the other end. The same two port numbers are used throughout the life of the connection. Mail packets coming in from the Internet will have a destination port of 25. Outbound packets will have the port numbers reversed. The fact that the secure system behind the router always will be accepting mail connections on port 25 is what makes possible separate control of incoming and outgoing services. The access list can be configured on either the outbound or inbound interface.

In the following example, the Ethernet network is a Class B network with the address 128.88.0.0, and the address of the mail host is 128.88.1.2. The **established** keyword is used only for the TCP protocol to indicate an established connection. A match occurs if the TCP datagram has the ACK or RST bits set, which indicate that the packet belongs to an existing connection.

```
access-list 102 permit tcp 0.0.0.0 255.255.255.255 128.88.0.0 0.0.255.255 established
access-list 102 permit tcp 0.0.0.0 255.255.255.255 128.88.1.2 0.0.0.0 eq 25
interface ethernet 0
 ip access-group 102 in
```

Named Access List Example

The following configuration creates a standard access list named `Internet_filter` and an extended access list named `marketing_group`:

```
interface Ethernet0/5
 ip address 2.0.5.1 255.255.255.0
 ip access-group Internet_filter out
 ip access-group marketing_group in
...
ip access-list standard Internet_filter
 permit 1.2.3.4
 deny any
ip access-list extended marketing_group
 permit tcp any 171.69.0.0 0.0.255.255 eq telnet
 deny tcp any any
 permit icmp any any
 deny udp any 171.69.0.0 0.0.255.255 lt 1024
 deny ip any any log
```

IP Extended Access List with Fragment Control Example

The first statement will match and deny only noninitial fragments destined for host 1.1.1.1. The second statement will match and permit only the remaining nonfragmented and initial fragments that are destined for host 1.1.1.1 TCP port 80. The third statement will deny all other traffic. In order to block noninitial fragments for any TCP port, we must block noninitial fragments for all TCP ports, including port 80 for host 1.1.1.1.

```
access-list 101 deny ip any host 1.1.1.1 fragments
access-list 101 permit tcp any host 1.1.1.1 eq 80
access-list 101 deny ip any any
```

Time Range Applied to an IP Access List Example

The following example denies HTTP traffic on Monday through Friday from 8:00 a.m. to 6:00 p.m. on IP. The example allows UDP traffic on Saturday and Sunday from noon to 8:00 p.m. only.

```
time-range no-http
 periodic weekdays 8:00 to 18:00
 !
time-range udp-yes
 periodic weekend 12:00 to 20:00
 !
ip access-list extended strict
 deny tcp any any eq http time-range no-http
 permit udp any any time-range udp-yes
 !

interface ethernet 0
 ip access-group strict in
```

Commented IP Access List Entry Examples

In the following example of a numbered access list, the workstation belonging to Jones is allowed access and the workstation belonging to Smith is not allowed access:

```
access-list 1 remark Permit only Jones workstation through
access-list 1 permit 171.69.2.88
access-list 1 remark Do not allow Smith workstation through
access-list 1 deny 171.69.3.13
```

In the following example of a numbered access list, the Winter and Smith workstations are not allowed to browse the web:

```
access-list 100 remark Do not allow Winter to browse the web
access-list 100 deny host 171.69.3.85 any eq http
access-list 100 remark Do not allow Smith to browse the web
access-list 100 deny host 171.69.3.13 any eq http
```

In the following example of a named access list, the Jones subnet is not allowed access:

```
ip access-list standard prevention
 remark Do not allow Jones subnet through
 deny 171.69.0.0 0.0.255.255
```

In the following example of a named access list, the Jones subnet is not allowed to use outbound Telnet:

```
ip access-list extended telnetting
 remark Do not allow Jones subnet to telnet out
```

```
deny tcp 171.69.0.0 0.0.255.255 any eq telnet
```

IP Accounting Example

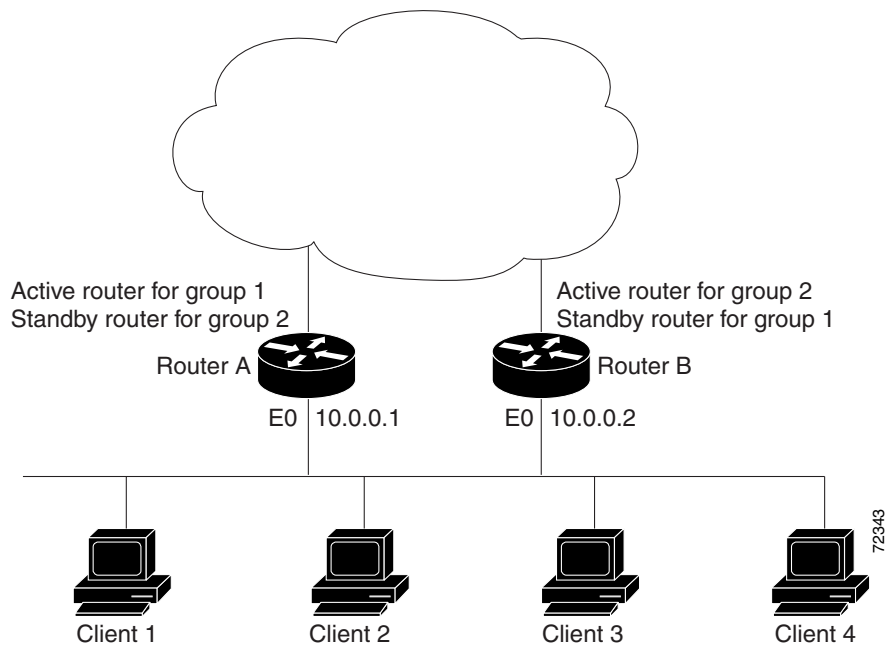
The following example enables IP accounting based on the source and destination MAC address and based on IP precedence for received and transmitted packets:

```
interface Ethernet0/5
 ip accounting mac-address input
 ip accounting mac-address output
 ip accounting precedence input
 ip accounting precedence output
```

HSRP Load Sharing Example

You can use HSRP or Multiple HSRP when you configure load sharing. In [Figure 20](#), half of the clients are configured for Router A, and half of the clients are configured for Router B. Together, the configuration for Routers A and B establish two Hot Standby groups. For group 1, Router A is the default active router because it has the assigned highest priority, and Router B is the standby router. For group 2, Router B is the default active router because it has the assigned highest priority, and Router A is the standby router. During normal operation, the two routers share the IP traffic load. When either router becomes unavailable, the other router becomes active and assumes the packet-transfer functions of the router that is unavailable. The **standby preempt** interface configuration command is necessary so that if a router goes down and then comes back up, preemption occurs and restores load sharing.

Figure 20 HSRP Load Sharing Example



The following example shows Router A configured as the active router for group 1 with a priority of 110 and Router B configured as the active router for group 2 with a priority of 110. The default priority level is 100. Group 1 uses a virtual IP address of 10.0.0.3 and Group 2 uses a virtual IP address of 10.0.0.4.

Router A Configuration

```
hostname RouterA
!
interface ethernet 0
  ip address 10.0.0.1 255.255.255.0
  standby 1 ip 10.0.0.3
  standby 1 priority 110
  standby 1 preempt
  standby 2 ip 10.0.0.4
  standby 2 preempt
```

Router B Configuration

```
hostname RouterB
!
interface ethernet 0
  ip address 10.0.0.2 255.255.255.0
  standby 1 ip 10.0.0.3
  standby 1 preempt
  standby 2 ip 10.0.0.4
  standby 2 priority 110
  standby 2 preempt
```

HSRP MAC Refresh Interval Examples

This section provides the following HSRP MAC refresh interval examples:

- [No Switch or Learning Bridge Present Example](#)
- [Switch or Learning Bridge Present Example](#)

No Switch or Learning Bridge Present Example

The following HSRP example of changing the MAC refresh interval is applicable if no switch or learning bridge is in your network. It prevents the sending of refresh packets.

```
interface fddi 1/0/0
  ip address 10.1.1.1 255.255.255.0
  standby ip 10.1.1.250
  standby mac-refresh 0
```

Switch or Learning Bridge Present Example

The following HSRP example of changing the MAC refresh interval is applicable if a switch or learning bridge is in your network. It will reduce the number of extra packets you send to refresh the MAC cache on the switch or learning bridge to two per minute.

```
interface fddi 1/0/0
  ip address 10.1.1.1 255.255.255.0
  standby ip 10.1.1.250
  standby mac-refresh 30
```

HSRP MIB Trap Example

The following example shows how to configure HSRP on two routers and enable the HSRP MIB trap feature. As in many environments, one router is preferred as the active one by configuring it at a higher priority level and enabling preemption. In this example, the active router is referred to as the primary router. The second router is referred to as the backup router.

Primary Router Configuration

```
interface Ethernet1
 ip address 15.1.1.1 255.255.0.0
 no ip redirects
 standby priority 200
 standby preempt
 standby ip 15.1.1.3
 snmp-server enable traps hsrp
 snmp-server host yourhost.cisco.com public hsrp
```

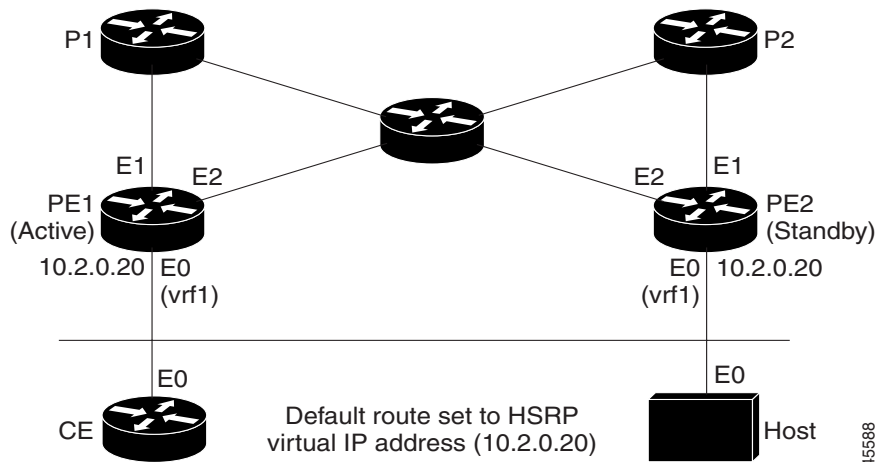
Backup Router Configuration

```
interface Ethernet1
 ip address 15.1.1.2 255.255.0.0
 no ip redirects
 standby priority 101
 standby ip 15.1.1.3
 snmp-server enable traps hsrp
 snmp-server host myhost.cisco.com public hsrp
```

HSRP Support for MPLS VPNs Example

Figure 21 shows two PEs with HSRP running between their VRF interfaces. The CE is configured with the HSRP virtual IP address as its default route. HSRP is configured to track the interfaces connecting the PEs to the rest of the provider network. For example, if interface E1 of PE1 fails, the HSRP priority will be reduced such that PE2 takes over forwarding packets to the HSRP virtual IP address.

Figure 21 Topology Showing HSRP Support Between Two VRF Interfaces



45588

Router PE1 Configuration

```
configure terminal
!
ip cef
!
ip vrf vrf1
  rd 100:1
  route-target export 100:1
  route-target import 100:1
!
interface ethernet0
  ip vrf forwarding vrf1
  ip address 10.2.0.1 255.255.0.0
  standby 1 ip 10.2.0.20
  standby 1 priority 105
  standby preempt delay minimum 10
  standby 1 timers 3 1
  standby 1 track ethernet1 10
  standby 1 track ethernet2 10
```

Router PE2 Configuration

```
configure terminal
!
ip cef
!
ip vrf vrf1
  rd 100:1
  route-target export 100:1
  route-target import 100:1
!
interface ethernet0
  ip vrf forwarding vrf1
  ip address 10.2.0.2 255.255.0.0
  standby 1 ip 10.2.0.20
  standby 1 priority 100
  standby preempt delay minimum 10
  standby 1 timers 3 1
  standby 1 track ethernet1 10
  standby 1 track ethernet2 10
```

HSRP Support for ICMP Redirect Messages Example

The following is a configuration example for two HSRP groups that allow the filtering of ICMP redirect messages:

Router A Configuration—Active for Group 1 and Standby for Group 2

```
interface Ethernet1
  ip address 1.0.0.10 255.0.0.0
  standby redirects
  standby 1 priority 120
  standby 1 preempt delay minimum 20
  standby 1 ip 1.0.0.1
  standby 2 priority 100
  standby 2 preempt delay minimum 20
  standby 2 ip 1.0.0.2
```

Router B Configuration—Standby for Group 1 and Active for Group 2

```
interface Ethernet1
```

```

ip address 1.0.0.11 255.0.0.0
standby redirects
standby 1 priority 100
standby 1 preempt delay minimum 20
standby 1 ip 1.0.0.1
standby 2 priority 120
standby 2 preempt delay minimum 20
standby 2 ip 1.0.0.2

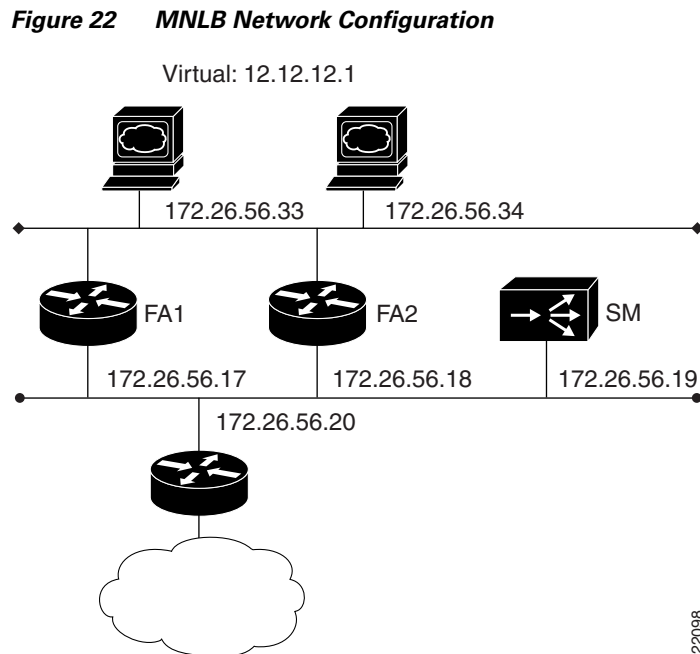
```

MNLB Forwarding Agent Examples

This section provides the following configuration examples:

- [Forwarding Agent Configuration for FA2 Example](#)
- [Services Manager Configuration for SM Example](#)

The network configured is shown in [Figure 22](#).



22098

Forwarding Agent Configuration for FA2 Example

The following is a sample of a router configured as a Forwarding Agent. In this example all disabled interfaces have been omitted to simplify the display.

```

FA2# wr t
Building configuration...

Current configuration:
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service udp-small-servers

```

```

service tcp-small-servers
!
hostname FA2
!
!
microcode CIP flash slot0:cip26-5
microcode reload
ip subnet-zero
no ip domain-lookup
!
ip cef distributed
ip casa 206.10.20.34 224.0.1.2
  forwarding-agent 1637
!
interface Ethernet0/0
  ip address 172.26.56.18 255.255.255.0
  no ip directed-broadcast
  ip route-cache flow
  ip igmp join-group 224.0.1.2
  no ip mroute-cache

!
interface Ethernet0/1
  ip address 172.26.56.37 255.255.255.0
  no ip directed-broadcast
!
!
!
router eigrp 777
  network 172.26.0.0
!

no ip classless
!
line con 0
  exec-timeout 0 0
  transport input none
line aux 0
line vty 0 4
  exec-timeout 0 0
login
!
end

```

Services Manager Configuration for SM Example

```

SM# wr t
Building configuration...
: Saved
: LocalDirector 420 Version 3.0.0.127
syslog output 20.3
no syslog console
enable password 00000000000000000000000000000000 encrypted
hostname SM
no shutdown ethernet 0
no shutdown ethernet 1
no shutdown ethernet 2
no shutdown ethernet 3
interface ethernet 0 auto
interface ethernet 1 auto
interface ethernet 2 auto
interface ethernet 3 auto

```

```
mtu 0 1500
mtu 1 1500
mtu 2 1500
mtu 3 1500
multiring all
no secure 0
no secure 1
no secure 2
no secure 3
ping-allow 0
ping-allow 1
ping-allow 2
ping-allow 3
ip address 172.26.56.19 255.255.255.248
route 172.26.10.249 255.255.255.255 172.26.56.20 1
route 206.10.20.33 255.255.255.255 172.26.56.17 1
route 206.10.20.34 255.255.255.255 172.26.56.18 1
no rip passive
failover ip address 0.0.0.0
failover
password cisco
telnet 161.0.0.0 255.0.0.0
no snmp-server contact
no snmp-server location
casa service-manager port 1638
casa service-manager multicast-ttl 60
tftp-server 172.26.10.249 /tftpboot/LD
virtual 172.26.56.13:0:0:tcp is
virtual 172.26.56.2:0:0:tcp is
redirection 172.26.56.13:0:0:tcp dispatched casa wildcard-ttl 60 fixed-ttl 60 igmp
224.0.1.2 port 1637
redirection 172.26.56.2:0:0:tcp dispatched casa wildcard-ttl 60 fixed-ttl 60 igmp
224.0.1.2 port 1637
real 172.26.56.34:0:0:tcp is
real 172.26.56.33:0:0:tcp is
real 172.26.56.6:0:0:tcp is
real 172.26.56.10:0:0:tcp is
bind 172.26.56.13:0:0:tcp 172.26.56.33:0:0:tcp
bind 172.26.56.13:0:0:tcp 172.26.56.34:0:0:tcp
bind 172.26.56.2:0:0:tcp 172.26.56.10:0:0:tcp
bind 172.26.56.2:0:0:tcp 172.26.56.6:0:0:tcp
: end
```