Cisco IOS IP Addressing Services Configuration Guide

Release 12.4

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

About Cisco IOS and Cisco IOS XE Software Documentation

Last updated: August 6, 2008

This document describes the objectives, audience, conventions, and organization used in Cisco IOS and Cisco IOS XE software documentation, collectively referred to in this document as Cisco IOS documentation. Also included are resources for obtaining technical assistance, additional documentation, and other information from Cisco. This document is organized into the following sections:

Documentation Objectives

Cisco IOS documentation describes the tasks and commands available to configure and maintain Cisco networking devices.

Audience

The Cisco IOS documentation set is i ntended for users who configure and maintain Cisco networking devices (such as routers and switches) but who may not be familiar with the configuration and maintenance tasks, the relationship among tasks, or the Cisco IOS commands necessary to perform particular tasks. The Cisco IOS documentation set is also intended for those users experienced with Cisco IOS who need to know about new features, new configuration options, and new software characteristics in the current Cisco IOS release.

Documentation Conventions

In Cisco IOS documentation, the term *router* may be used to refer to various Cisco products; for example, routers, access servers, and switches. These and other networking devices that support Cisco IOS software are shown interchangeably in examples and are used only for illustrative purposes. An example that shows one product does not necessarily mean that other products are not supported.

This section includes the following topics:

- Typographic Conventions, page ii
- Command Syntax Conventions, page ii
- Software Conventions, page iii
- Reader Alert Conventions, page iii

Typographic Conventions

Cisco IOS documentation uses the following typographic conventions:

Convention	Description
^ or Ctrl	Both the ^ symbol and Ctrl represent the Control (Ctrl) key on a keyboard. For example, the key combination **^D** or **Ctrl-D** means that you hold down the Control key while you press the D key. (Keys are indicated in capital letters but are not case sensitive.)
string	A string is a nonquoted set of characters shown in italics. For example, when setting a Simple Network Management Protocol (SNMP) community string to *public*, do not use quotation marks around the string; otherwise, the string will include the quotation marks.

Command Syntax Conventions

Cisco IOS documentation uses the following command syntax conventions:

Convention	Description
bold	Bold text indicates commands and keywords that you enter as shown.
italic	Italic text indicates arguments for which you supply values.
[x]	Square brackets enclose an optional keyword or argument.
|	A vertical line, called a pipe, indicates a choice within a set of keywords or arguments.
[x | y]	Square brackets enclosing keywords or arguments separated by a pipe indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments separated by a pipe indicate a required choice.
[x {y | z}]	Braces and a pipe within square brackets indicate a required choice within an optional element.

Software Conventions

Cisco IOS uses the following program code conventions:

Convention	Description
`Courier font`	Courier font is used for information that is displayed on a PC or terminal screen.
`Bold Courier font`	Bold Courier font indicates text that the user must enter.
< >	Angle brackets enclose text that is not displayed, such as a password. Angle brackets also are used in contexts in which the italic font style is not supported; for example, ASCII text.
!	An exclamation point at the beginning of a line indicates that the text that follows is a comment, not a line of code. An exclamation point is also displayed by Cisco IOS software for certain processes.
[ ]	Square brackets enclose default responses to system prompts.

Reader Alert Conventions

The Cisco IOS documentation set uses the following conventions for reader alerts:

Caution Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

Note Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the manual.

Timesaver Means *the described action saves time*. You can save time by performing the action described in the paragraph.

Documentation Organization

This section describes the Cisco IOS documentation set, how it is organized, and how to access it on Cisco.com. Included are lists of configuration guides, command references, and supplementary references and resources that make up the documentation set. The following topics are included:

- Cisco IOS Documentation Set, page iv
- Cisco IOS Documentation on Cisco.com, page iv
- Configuration Guides, Command References, and Supplementary Resources, page v

Cisco IOS Documentation Set

Cisco IOS documentation consists of the following:

- Release notes and caveats provide information about platform, technology, and feature support for a release and describe severity 1 (catastrophic), severity 2 (severe), and severity 3 (moderate) defects in released Cisco IOS code. Review release notes before other documents to learn whether or not updates have been made to a feature.
- Sets of configuration guides and command references organized by technology and published for each standard Cisco IOS release.
  - Configuration guides—Compilations of documents that provide informational and task-oriented descriptions of Cisco IOS features.
  - Command references—Compilations of command pages that provide detailed information about the commands used in the Cisco IOS features and processes that make up the related configuration guides. For each technology, there is a single command reference that covers all Cisco IOS releases and that is updated at each standard release.
- Lists of all the commands in a specific release and all commands that are new, modified, removed, or replaced in the release.
- Command reference book for **debug** commands. Command pages are listed in alphabetical order.
- Reference book for system messages for all Cisco IOS releases.

Cisco IOS Documentation on Cisco.com

The following sections describe the documentation organization and how to access various document types.

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

New Features List

The New Features List for each release provides a list of all features in the release with hyperlinks to the feature guides in which they are documented.

Feature Guides

Cisco IOS features are documented in feature guides. Feature guides describe one feature or a group of related features that are supported on many different software releases and platforms. Your Cisco IOS software release or platform may not support all the features documented in a feature guide. See the Feature Information table at the end of the feature guide for information about which features in that guide are supported in your software release.

Configuration Guides

Configuration guides are provided by technology and release and comprise a set of individual feature guides relevant to the release and technology.

Command References

Command reference books describe Cisco IOS commands that are supported in many different software releases and on many different platforms. The books are provided by technology. For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or the *Cisco IOS Master Command List, All Releases*, at http://www.cisco.com/en/US/docs/ios/mcl/all_release/all_mcl.html.

Cisco IOS Supplementary Documents and Resources

Supplementary documents and resources are listed in Table 2 on page xi.

Configuration Guides, Command References, and Supplementary Resources

Table 1 lists, in alphabetical order, Cisco IOS and Cisco IOS XE software configuration guides and command references, including brief descriptions of the contents of the documents. The Cisco IOS command references are comprehensive, meaning that they include commands for both Cisco IOS software and Cisco IOS XE software, for all releases. The configuration guides and command references support many different software releases and platforms. Your Cisco IOS software release or platform may not support all these technologies.

For additional information about configuring and operating specific networking devices, go to the Product Support area of Cisco.com at http://www.cisco.com/web/psa/products/index.html.

Table 2 lists documents and resources that supplement the Cisco IOS software configuration guides and command references. These supplementary resources include release notes and caveats; master command lists; new, modified, removed, and replaced command lists; system messages; and the debug command reference.

Table 1 Cisco IOS and Cisco IOS XE Configuration Guides and Command References

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
Cisco IOS AppleTalk Configuration Guide *Cisco IOS XE AppleTalk Configuration Guide* *Cisco IOS AppleTalk Command Reference*	AppleTalk protocol.
Cisco IOS Asynchronous Transfer Mode Configuration Guide *Cisco IOS Asynchronous Transfer Mode Command Reference*	LAN ATM, multiprotocol over ATM (MPoA), and WAN ATM.

Table 1 Cisco IOS and Cisco IOS XE Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
Cisco IOS Bridging and IBM Networking Configuration Guide *Cisco IOS Bridging Command Reference* *Cisco IOS IBM Networking Command Reference*	• Transparent and source-route transparent (SRT) bridging, source-route bridging (SRB), Token Ring Inter-Switch Link (TRISL), and token ring route switch module (TRRSM). • Data-link switching plus (DLSw+), serial tunnel (STUN), block serial tunnel (BSTUN); logical link control, type 2 (LLC2), synchronous data link control (SDLC); IBM Network Media Translation, including Synchronous Data Logical Link Control (SDLLC) and qualified LLC (QLLC); downstream physical unit (DSPU), Systems Network Architecture (SNA) service point, SNA frame relay access, advanced peer-to-peer networking (APPN), native client interface architecture (NCIA) client/server topologies, and IBM Channel Attach.
Cisco IOS Broadband and DSL Configuration Guide *Cisco IOS XE Broadband and DSL Configuration Guide* *Cisco IOS Broadband and DSL Command Reference*	Point-to-Point Protocol (PPP) over ATM (PPPoA) and PPP over Ethernet (PPPoE).
Cisco IOS Carrier Ethernet Configuration Guide *Cisco IOS Carrier Ethernet Command Reference*	Connectivity fault management (CFM), Ethernet Local Management Interface (ELMI), IEEE 802.3ad link bundling, Link Layer Discovery Protocol (LLDP), media endpoint discovery (MED), and operations, administration, and maintenance (OAM).
Cisco IOS Configuration Fundamentals Configuration Guide *Cisco IOS XE Configuration Fundamentals Configuration Guide* *Cisco IOS Configuration Fundamentals Command Reference*	Autoinstall, Setup, Cisco IOS command-line interface (CLI), Cisco IOS file system (IFS), Cisco IOS web browser user interface (UI), basic file transfer services, and file management.
Cisco IOS DECnet Configuration Guide *Cisco IOS XE DECnet Configuration Guide* *Cisco IOS DECnet Command Reference*	DECnet protocol.
Cisco IOS Dial Technologies Configuration Guide *Cisco IOS XE Dial Technologies Configuration Guide* *Cisco IOS Dial Technologies Command Reference*	Asynchronous communications, dial backup, dialer technology, dial-in terminal services and AppleTalk remote access (ARA), large scale dialout, dial-on-demand routing, dialout, modem and resource pooling, ISDN, multilink PPP (MLP), PPP, virtual private dialup network (VPDN).
Cisco IOS Flexible NetFlow Configuration Guide *Cisco IOS Flexible NetFlow Command Reference*	Flexible NetFlow.

Table 1 Cisco IOS and Cisco IOS XE Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
Cisco IOS H.323 Configuration Guide	Gatekeeper enhancements for managed voice services, Gatekeeper Transaction Message Protocol, gateway codec order preservation and shutdown control, H.323 dual tone multifrequency relay, H.323 version 2 enhancements, Network Address Translation (NAT) support of H.323 v2 Registration, Admission, and Status (RAS) protocol, tokenless call authorization, and VoIP gateway trunk and carrier-based routing.
Cisco IOS High Availability Configuration Guide *Cisco IOS XE High Availability Configuration Guide* *Cisco IOS High Availability Command Reference*	A variety of High Availability (HA) features and technologies that are available for different network segments (from enterprise access to service provider core) to facilitate creation of end-to-end highly available networks. Cisco IOS HA features and technologies can be categorized in three key areas: system-level resiliency, network-level resiliency, and embedded management for resiliency.
Cisco IOS Integrated Session Border Controller Command Reference	A VoIP-enabled device that is deployed at the edge of networks. An SBC is a toolkit of functions, such as signaling interworking, network hiding, security, and quality of service (QoS).
Cisco IOS Intelligent Service Gateway Configuration Guide *Cisco IOS Intelligent Service Gateway Command Reference*	Subscriber identification, service and policy determination, session creation, session policy enforcement, session life-cycle management, accounting for access and service usage, session state monitoring.
Cisco IOS Interface and Hardware Component Configuration Guide *Cisco IOS XE Interface and Hardware Component Configuration Guide* *Cisco IOS Interface and Hardware Component Command Reference*	LAN interfaces, logical interfaces, serial interfaces, virtual interfaces, and interface configuration.
Cisco IOS IP Addressing Services Configuration Guide *Cisco IOS XE Addressing Services Configuration Guide* *Cisco IOS IP Addressing Services Command Reference*	Address Resolution Protocol (ARP), Network Address Translation (NAT), Domain Name System (DNS), Dynamic Host Configuration Protocol (DHCP), and Next Hop Address Resolution Protocol (NHRP).
Cisco IOS IP Application Services Configuration Guide *Cisco IOS XE IP Application Services Configuration Guide* *Cisco IOS IP Application Services Command Reference*	Enhanced Object Tracking (EOT), Gateway Load Balancing Protocol (GLBP), Hot Standby Router Protocol (HSRP), IP Services, Server Load Balancing (SLB), Stream Control Transmission Protocol (SCTP), TCP, Web Cache Communication Protocol (WCCP), User Datagram Protocol (UDP), and Virtual Router Redundancy Protocol (VRRP).
Cisco IOS IP Mobility Configuration Guide *Cisco IOS IP Mobility Command Reference*	Mobile ad hoc networks (MANet) and Cisco mobile networks.
Cisco IOS IP Multicast Configuration Guide *Cisco IOS XE IP Multicast Configuration Guide* *Cisco IOS IP Multicast Command Reference*	Protocol Independent Multicast (PIM) sparse mode (PIM-SM), bidirectional PIM (bidir-PIM), Source Specific Multicast (SSM), Multicast Source Discovery Protocol (MSDP), Internet Group Management Protocol (IGMP), and Multicast VPN (MVPN).

Table 1 Cisco IOS and Cisco IOS XE Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
Cisco IOS IP Routing Protocols Configuration Guide *Cisco IOS XE IP Routing Protocols Configuration Guide* *Cisco IOS IP Routing Protocols Command Reference*	Border Gateway Protocol (BGP), multiprotocol BGP, multiprotocol BGP extensions for IP multicast, bidirectional forwarding detection (BFD), Enhanced Interior Gateway Routing Protocol (EIGRP), Interior Gateway Routing Protocol (IGRP), Intermediate System-to-Intermediate System (IS-IS), on-demand routing (ODR), Open Shortest Path First (OSPF), and Routing Information Protocol (RIP).
Cisco IOS IP SLAs Configuration Guide *Cisco IOS XE IP SLAs Configuration Guide* *Cisco IOS IP SLAs Command Reference*	Cisco IOS IP Service Level Agreements (IP SLAs).
Cisco IOS IP Switching Configuration Guide *Cisco IOS XE IP Switching Configuration Guide* *Cisco IOS IP Switching Command Reference*	Cisco Express Forwarding, fast switching, and Multicast Distributed Switching (MDS).
Cisco IOS IPv6 Configuration Guide *Cisco IOS XE IPv6 Configuration Guide* *Cisco IOS IPv6 Command Reference*	For IPv6 features, protocols, and technologies, go to the IPv6 “Start Here” document at the following URL: http://www.cisco.com/en/US/docs/ios/ipv6/configuration/guide/ip6-roadmap.html
Cisco IOS ISO CLNS Configuration Guide *Cisco IOS XE ISO CLNS Configuration Guide* *Cisco IOS ISO CLNS Command Reference*	ISO connectionless network service (CLNS).
Cisco IOS LAN Switching Configuration Guide *Cisco IOS XE LAN Switching Configuration Guide* *Cisco IOS LAN Switching Command Reference*	VLANs, Inter-Switch Link (ISL) encapsulation, IEEE 802.10 encapsulation, IEEE 802.1Q encapsulation, and multilayer switching (MLS).
Cisco IOS Mobile Wireless Gateway GPRS Support Node Configuration Guide *Cisco IOS Mobile Wireless Gateway GPRS Support Node Command Reference*	Cisco IOS Gateway GPRS Support Node (GGSN) in a 2.5-generation general packet radio service (GPRS) and 3-generation universal mobile telecommunication system (UMTS) network.
Cisco IOS Mobile Wireless Home Agent Configuration Guide *Cisco IOS Mobile Wireless Home Agent Command Reference*	Cisco Mobile Wireless Home Agent, an anchor point for mobile terminals for which mobile IP or proxy mobile IP services are provided.
Cisco IOS Mobile Wireless Packet Data Serving Node Configuration Guide *Cisco IOS Mobile Wireless Packet Data Serving Node Command Reference*	Cisco Packet Data Serving Node (PDSN), a wireless gateway that is between the mobile infrastructure and standard IP networks and that enables packet data services in a code division multiple access (CDMA) environment.
Cisco IOS Mobile Wireless Radio Access Networking Configuration Guide *Cisco IOS Mobile Wireless Radio Access Networking Command Reference*	Cisco IOS radio access network products.

Table 1 Cisco IOS and Cisco IOS XE Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
Cisco IOS Multiprotocol Label Switching Configuration Guide *Cisco IOS XE Multiprotocol Label Switching Configuration Guide* *Cisco IOS Multiprotocol Label Switching Command Reference*	MPLS Label Distribution Protocol (LDP), MPLS Layer 2 VPNs, MPLS Layer 3 VPNs, MPLS Traffic Engineering (TE), and MPLS Embedded Management (EM) and MIBs.
Cisco IOS Multi-Topology Routing Configuration Guide *Cisco IOS Multi-Topology Routing Command Reference*	Unicast and multicast topology configurations, traffic classification, routing protocol support, and network management support.
Cisco IOS NetFlow Configuration Guide *Cisco IOS XE NetFlow Configuration Guide* *Cisco IOS NetFlow Command Reference*	Network traffic data analysis, aggregation caches, export features.
Cisco IOS Network Management Configuration Guide *Cisco IOS XE Network Management Configuration Guide* *Cisco IOS Network Management Command Reference*	Basic system management; system monitoring and logging; troubleshooting, logging, and fault management; Cisco Discovery Protocol; Cisco IOS Scripting with Tool Control Language (Tcl); Cisco networking services (CNS); DistributedDirector; Embedded Event Manager (EEM); Embedded Resource Manager (ERM); Embedded Syslog Manager (ESM); HTTP; Remote Monitoring (RMON); SNMP; and VPN Device Manager Client for Cisco IOS Software (XSM Configuration).
Cisco IOS Novell IPX Configuration Guide *Cisco IOS XE Novell IPX Configuration Guide* *Cisco IOS Novell IPX Command Reference*	Novell Internetwork Packet Exchange (IPX) protocol.
Cisco IOS Optimized Edge Routing Configuration Guide *Cisco IOS Optimized Edge Routing Command Reference*	Optimized edge routing (OER) monitoring, policy configuration, routing control, logging and reporting, and VPN IPsec/generic routing encapsulation (GRE) tunnel interface optimization.
Cisco IOS Quality of Service Solutions Configuration Guide *Cisco IOS XE Quality of Service Solutions Configuration Guide* *Cisco IOS Quality of Service Solutions Command Reference*	Class-based weighted fair queuing (CBWFQ), custom queuing, distributed traffic shaping (DTS), generic traffic shaping (GTS), IP- to-ATM class of service (CoS), low latency queuing (LLQ), modular QoS CLI (MQC), Network-Based Application Recognition (NBAR), priority queuing, Security Device Manager (SDM), Multilink PPP (MLPPP) for QoS, header compression, AutoQoS, QoS features for voice, Resource Reservation Protocol (RSVP), weighted fair queuing (WFQ), and weighted random early detection (WRED).
Cisco IOS Security Configuration Guide *Cisco IOS XE Security Configuration Guide* *Cisco IOS Security Command Reference*	Access control lists (ACLs), authentication, authorization, and accounting (AAA), firewalls, IP security and encryption, neighbor router authentication, network access security, network data encryption with router authentication, public key infrastructure (PKI), RADIUS, TACACS+, terminal access security, and traffic filters.

Table 1 Cisco IOS and Cisco IOS XE Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
Cisco IOS Service Selection Gateway Configuration Guide *Cisco IOS Service Selection Gateway Command Reference*	Subscriber authentication, service access, and accounting.
Cisco IOS Software Activation Configuration Guide *Cisco IOS Software Activation Command Reference*	An orchestrated collection of processes and components to activate Cisco IOS software feature sets by obtaining and validating Cisco software licenses.
Cisco IOS Software Modularity Installation and Configuration Guide *Cisco IOS Software Modularity Command Reference*	Installation and basic configuration of software modularity images, including installations on single and dual route processors, installation rollbacks, software modularity binding, software modularity processes and patches.
Cisco IOS Terminal Services Configuration Guide *Cisco IOS Terminal Services Command Reference* *Cisco IOS XE Terminal Services Command Reference*	DEC, local-area transport (LAT), and X.25 packet assembler/disassembler (PAD).
Cisco IOS Virtual Switch Command Reference	Virtual switch redundancy, high availability, and packet handling; converting between standalone and virtual switch modes; virtual switch link (VSL); Virtual Switch Link Protocol (VSLP). **Note** For information about virtual switch configuration, refer to the product-specific software configuration information for the Cisco Catalyst 6500 series switch or for the Metro Ethernet 6500 series switch.
Cisco IOS Voice Configuration Library *Cisco IOS Voice Command Reference*	Cisco IOS support for voice call control protocols, interoperability, physical and virtual interface management, and troubleshooting. The library includes documentation for IP telephony applications.
Cisco IOS VPDN Configuration Guide *Cisco IOS XE VPDN Configuration Guide* *Cisco IOS VPDN Command Reference*	Layer 2 Tunneling Protocol (L2TP) dial-out load balancing and redundancy, L2TP extended failover, L2TP security VPDN, multihop by Dialed Number Identification Service (DNIS), timer and retry enhancements for L2TP and Layer 2 Forwarding (L2F), RADIUS Attribute 82: tunnel assignment ID, shell-based authentication of VPDN users, tunnel authentication via RADIUS on tunnel terminator.
Cisco IOS Wide-Area Networking Configuration Guide *Cisco IOS XE Wide-Area Networking Configuration Guide* *Cisco IOS Wide-Area Networking Command Reference*	Frame Relay, Layer 2 Tunneling Protocol Version 3 (L2TPv3), Link Access Procedure, Balanced (LAPB), Switched Multimegabit Data Service (SMDS), and X.25.
Cisco IOS Wireless LAN Configuration Guide *Cisco IOS Wireless LAN Command Reference*	Broadcast key rotation, IEEE 802.11x support, IEEE 802.1x authenticator, IEEE 802.1x local authentication service for Extensible Authentication Protocol-Flexible Authentication via Secure Tunneling (EAP-FAST), Multiple Basic Service Set ID (BSSID), Wi-Fi Multimedia (WMM) required elements, and Wi-Fi Protected Access (WPA).

Table 2 Cisco IOS Supplementary Documents and Resources

Document Title	Description
Cisco IOS Master Command List, All Releases	Alphabetical list of all the commands documented in all Cisco IOS releases.
Cisco IOS New, Modified, Removed, and Replaced Commands	List of all the new, modified, removed, and replaced commands for a Cisco IOS release.
Cisco IOS Software System Messages	List of Cisco IOS system messages and descriptions. System messages may indicate problems with your system; be informational only; or may help diagnose problems with communications lines, internal hardware, or the system software.
Cisco IOS Debug Command Reference	Alphabetical list of **debug** commands including brief descriptions of use, command syntax, and usage guidelines.
Release Notes and Caveats	Information about new and changed features, system requirements, and other useful information about specific software releases; information about defects in specific Cisco IOS software releases.
MIBs	Files used for network monitoring. To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator at the following URL: http://www.cisco.com/go/mibs
RFCs	Standards documents maintained by the Internet Engineering Task Force (IETF) that Cisco IOS documentation references where applicable. The full text of referenced RFCs may be obtained at the following URL: http://www.rfc-editor.org/

Additional Resources and Documentation Feedback

What's New in Cisco Product Documentation is published monthly and describes all new and revised Cisco technical documentation. The *What's New in Cisco Product Documentation* publication also provides information about obtaining the following resources:

- Technical documentation
- Cisco product security overview
- Product alerts and field notices
- Technical assistance

Cisco IOS technical documentation includes embedded feedback forms where you can rate documents and provide suggestions for improvement. Your feedback helps us improve our documentation.

Using the Command-Line Interface in Cisco IOS and Cisco IOS XE Software

Last updated: August 6, 2008

This document provides basic information about the command-line interface (CLI) in Cisco IOS and Cisco IOS XE software and how you can use some of the CLI features. This document contains the following sections:

- Initially Configuring a Device, page i
- Using the CLI, page ii
- Saving Changes to a Configuration, page xii
- Additional Information, page xii

For more information about using the CLI, see the "Using the Cisco IOS Command-Line Interface" section of the *Cisco IOS Configuration Fundamentals Configuration Guide*.

For information about the software documentation set, see the "About Cisco IOS and Cisco IOS XE Software Documentation" document.

Initially Configuring a Device

Initially configuring a device varies by platform. For information about performing an initial configuration, see the hardware installation documentation that is provided with the original packaging of the product or go to the Product Support area of Cisco.com at http://www.cisco.com/web/psa/products/index.html.

After you have performed the initial configuration and connected the device to your network, you can configure the device by using the console port or a remote access method, such as Telnet or Secure Shell (SSH), to access the CLI or by using the configuration method provided on the device, such as Security Device Manager.

Changing the Default Settings for a Console or AUX Port

There are only two changes that you can make to a console port and an AUX port:

- Change the port speed with the **config-register 0x** command. Changing the port speed is not recommended. The well-known default speed is 9600.
- Change the behavior of the port; for example, by adding a password or changing the timeout value.

Note The AUX port on the Route Processor (RP) installed in a Cisco ASR1000 series router does not serve any useful customer purpose and should be accessed only under the advisement of a customer support representative.

Using the CLI

This section describes the following topics:

Understanding Command Modes

The CLI command mode structure is hierarchical, and each mode supports a set of specific commands. This section describes the most common of the many modes that exist.

Table 1 lists common command modes with associated CLI prompts, access and exit methods, and a brief description of how each mode is used.

Table 1 CLI Command Modes

Command Mode	Access Method	Prompt	Exit Method	Mode Usage
User EXEC	Log in.	`Router>`	Issue the **logout** or **exit** command.	• Change terminal settings. • Perform basic tests. • Display device status.
Privileged EXEC	From user EXEC mode, issue the **enable** command.	`Router#`	Issue the **disable** command or the **exit** command to return to user EXEC mode.	• Issue **show** and **debug** commands. • Copy images to the device. • Reload the device. • Manage device configuration files. • Manage device file systems.
Global configuration	From privileged EXEC mode, issue the **configure terminal** command.	`Router(config)#`	Issue the **exit** command or the **end** command to return to privileged EXEC mode.	Configure the device.
Interface configuration	From global configuration mode, issue the **interface** command.	`Router(config-if)#`	Issue the **exit** command to return to global configuration mode or the **end** command to return to privileged EXEC mode.	Configure individual interfaces.
Line configuration	From global configuration mode, issue the **line vty** or **line console** command.	`Router(config-line)#`	Issue the **exit** command to return to global configuration mode or the **end** command to return to privileged EXEC mode.	Configure individual terminal lines.

Table 1 CLI Command Modes (continued)

Command Mode	Access Method	Prompt	Exit Method	Mode Usage
ROM monitor	From privileged EXEC mode, issue the **reload** command. Press the **Break** key during the first 60 seconds while the system is booting.	`rommon # >` The # symbol represents the line number and increments at each prompt.	Issue the **continue** command.	• Run as the default operating mode when a valid image cannot be loaded. • Access the fall-back procedure for loading an image when the device lacks a valid image and cannot be booted. • Perform password recovery when a CTRL-Break sequence is issued within 60 seconds of a power-on or reload event.
Diagnostic (available only on the Cisco ASR1000 series router)	The router boots or enters diagnostic mode in the following scenarios. When a Cisco IOS process or processes fail, in most scenarios the router will reload. • A user-configured access policy was configured using the **transport-map** command, which directed the user into diagnostic mode. • The router was accessed using an RP auxiliary port. • A break signal (**Ctrl-C**, **Ctrl-Shift-6**, or the **send break** command) was entered, and the router was configured to enter diagnostic mode when the break signal was received.	`Router(diag)#`	If a Cisco IOS process failure is the reason for entering diagnostic mode, the failure must be resolved and the router must be rebooted to exit diagnostic mode. If the router is in diagnostic mode because of a transport-map configuration, access the router through another port or using a method that is configured to connect to the Cisco IOS CLI. If the RP auxiliary port was used to access the router, use another port for access. Accessing the router through the auxiliary port is not useful for customer purposes.	• Inspect various states on the router, including the Cisco IOS state. • Replace or roll back the configuration. • Provide methods of restarting the Cisco IOS software or other processes. • Reboot hardware, such as the entire router, an RP, an ESP, a SIP, a SPA, or possibly other hardware components. • Transfer files into or off of the router using remote access methods such as FTP, TFTP, and SCP.

EXEC commands are not saved when the software reboots. Commands that you issue in a configuration mode can be saved to the startup configuration. If you save the running configuration to the startup configuration, these commands will execute when the software is rebooted. Global configuration mode is the highest level of configuration mode. From global configuration mode, you can enter a variety of other configuration modes, including protocol-specific modes.

ROM monitor mode is a separate mode that is used when the software cannot load properly. If a valid software image is not found when the software boots or if the configuration file is corrupted at startup, the software might enter ROM monitor mode. Use the question symbol (?) to view the commands that you can use while the device is in ROM monitor mode.

```
rommon 1 > ?
alias                   set and display aliases command
boot                    boot up an external process
confreg                 configuration register utility
cont                    continue executing a downloaded image
context                 display the context of a loaded image
cookie                  display contents of cookie PROM in hex
.
.
.
rommon 2 >
```

The following example shows how the command prompt changes to indicate a different command mode:

```
Router> enable
Router# configure terminal
Router(config)# interface ethernet 1/1
Router(config-if)# ethernet
Router(config-line)# exit
Router(config)# end
Router#
```

Note A keyboard alternative to the **end** command is Ctrl-Z.

Using the Interactive Help Feature

The CLI includes an interactive Help feature. Table 2 describes how to use the Help feature.

Table 2 CLI Interactive Help Commands

Command	Purpose
help	Provides a brief description of the help feature in any command mode.
?	Lists all commands available for a particular command mode.
*partial command***?**	Provides a list of commands that begin with the character string (no space between the command and the question mark).
partial command<**Tab**>	Completes a partial command name (no space between the command and <Tab>).
command **?**	Lists the keywords, arguments, or both associated with the command (space between the command and the question mark).
command keyword **?**	Lists the arguments that are associated with the keyword (space between the keyword and the question mark).

The following examples show how to use the help commands:

help

```
Router> help
Help may be requested at any point in a command by entering a question mark '?'.  If
nothing matches, the help list will be empty and you must backup until entering a '?'
shows the available options.
Two styles of help are provided:
1. Full help is available when you are ready to enter a command argument (e.g. 'show ?')
and describes each possible argument.
2. Partial help is provided when an abbreviated argument is entered and you want to know
what arguments match the input (e.g. 'show pr?'.)
```

?

```
Router# ?
Exec commands:
  access-enable        Create a temporary access-List entry
  access-profile       Apply user-profile to interface
  access-template      Create a temporary access-List entry
  alps                 ALPS exec commands
  archive              manage archive files
<snip>
```

partial command?

```
Router(config)# zo?
zone  zone-pair
```

partial command<Tab>

```
Router(config)# we<Tab> webvpn
```

command ?

```
Router(config-if)# pppoe ?
  enable         Enable pppoe
  max-sessions   Maximum PPPOE sessions
```

command keyword ?

```
Router(config-if)# pppoe enable ?
  group  attach a BBA group
  <cr>
```

Understanding Command Syntax

Command syntax is the format in which a command should be entered in the CLI. Commands include the name of the command, keywords, and arguments. Keywords are alphanumeric strings that are used literally. Arguments are placeholders for values that a user must supply. Keywords and arguments may be required or optional.

Specific conventions convey information about syntax and command elements. Table 3 describes these conventions.

Table 3 CLI Syntax Conventions

Symbol/Text	Function	Notes
< > (angle brackets)	Indicate that the option is an argument.	Sometimes arguments are displayed without angle brackets.
A.B.C.D.	Indicates that you must enter a dotted decimal IP address.	Angle brackets (< >) are not always used to indicate that an IP address is an argument.
WORD (all capital letters)	Indicates that you must enter one word.	Angle brackets (< >) are not always used to indicate that a WORD is an argument.
LINE (all capital letters)	Indicates that you must enter more than one word.	Angle brackets (< >) are not always used to indicate that a LINE is an argument.
<cr> (carriage return)	Indicates the end of the list of available keywords and arguments, and also indicates when keywords and arguments are optional. When <cr> is the only option, you have reached the end of the branch or the end of the command if the command has only one branch.	—

The following examples show syntax conventions:

```
Router(config)# ethernet cfm domain ?
  WORD  domain name
Router(config)# ethernet cfm domain dname ?
  level
Router(config)# ethernet cfm domain dname level ?
  <0-7>  maintenance level number
Router(config)# ethernet cfm domain dname level 7 ?
  <cr>
Router(config)# snmp-server file-transfer access-group 10 ?
  protocol  protocol options
  <cr>
Router(config)# logging host ?
  Hostname or A.B.C.D  IP address of the syslog server
  ipv6                 Configure IPv6 syslog server
Router(config)# snmp-server file-transfer access-group 10 ?
  protocol  protocol options
  <cr>
```

Understanding Enable and Enable Secret Passwords

Some privileged EXEC commands are used for actions that impact the system, and it is recommended that you set a password for these commands to prevent unauthorized use. Two types of passwords, enable (not encrypted) and enable secret (encrypted), can be set. The following commands set these passwords and are issued in global configuration mode:

- **enable** *password*
- **enable secret** *password*

Using an enable secret password is recommended because it is encrypted and more secure than the enable password. When you use an enable secret password, text is encrypted (unreadable) before it is written to the config.text file. When you use an enable password, the text is written as entered (readable) to the config.text file.

Each type of password is case sensitive, can contain from 1 to 25 uppercase and lowercase alphanumeric characters, and can start with a number. Spaces are also valid password characters; for example, “two words” is a valid password. Leading spaces are ignored, but trailing spaces are recognized.

Note Both password commands have numeric keywords that are single integer values. If you choose a number for the first character of your password followed by a space, the system will read the number as if it were the numeric keyword and not as part of your password.

When both passwords are set, the enable secret password takes precedence over the enable password.

To remove a password, use the **no** form of the commands: **no enable** *password* or **no enable secret** *password*.

For more information about password recovery procedures for Cisco products, see http://www.cisco.com/en/US/products/sw/iosswrel/ps1831/products_tech_note09186a00801746e6.shtml.

Using the Command History Feature

The CLI command history feature saves the commands you enter during a session in a command history buffer. The default number of commands saved is 10, but the number is configurable within the range of 0 to 256. This command history feature is particularly useful for recalling long or complex commands.

To change the number of commands saved in the history buffer for a terminal session, issue the **terminal history size** command:

```
Router# terminal history size num
```

A command history buffer is also available in line configuration mode with the same default and configuration options. To set the command history buffer size for a terminal session in line configuration mode, issue the **history** command:

```
Router(config-line)# history [size num]
```

To recall commands from the history buffer, use the following methods:

- Press Ctrl-P or the up arrow key—Recalls commands beginning with the most recent command. Repeat the key sequence to recall successively older commands.

- Press Ctrl-N or the down arrow key—Recalls the most recent commands in the history buffer after they have been recalled using Ctrl-P or the up arrow key. Repeat the key sequence to recall successively more recent commands.

Note The arrow keys function only on ANSI-compatible terminals such as the VT100.

- Issue the **show history** command in user EXEC or privileged EXEC mode—Lists the most recent commands that you entered. The number of commands that are displayed is determined by the setting of the **terminal history size** and **history** commands.

  The CLI command history feature is enabled by default. To disable this feature for a terminal session, issue the **terminal no history** command in user EXEC or privileged EXEC mode or the **no history** command in line configuration mode.

Abbreviating Commands

Typing a complete command name is not always required for the command to execute. The CLI recognizes an abbreviated command when the abbreviation contains enough characters to uniquely identify the command. For example, the **show version** command can be abbreviated as **sh ver**. It cannot be abbreviated as **s ver** because **s** could mean **show**, **set**, or **systat**. The **sh v** abbreviation also is not valid because the **show** command has **vrrp** as a keyword in addition to **version**. (Command and keyword examples from Cisco IOS Release 12.4(13)T.)

Using Aliases for CLI Commands

To save time and the repetition of entering the same command multiple times, you can use a command alias. An alias can be configured to do anything that can be done at the command line, but an alias cannot move between modes, type in passwords, or perform any interactive functions.

Table 4 shows the default command aliases.

Table 4 Default Command Aliases

Command Alias	Original Command
h	help
lo	logout
p	ping
s	show
u or **un**	undebug
w	where

To create a command alias, issue the **alias** command in global configuration mode. The syntax of the command is **alias** *mode command-alias original-command*. Following are some examples:

- Router(config)# **alias exec prt partition**—privileged EXEC mode
- Router(config)# **alias configure sb source-bridge**—global configuration mode
- Router(config)# **alias interface rl rate-limit**—interface configuration mode

To view both default and user-created aliases, issue the **show alias** command.

For more information about the **alias** command, see http://www.cisco.com/en/US/docs/ios/fundamentals/command/reference/cf_book.html.

Using the no and default Forms of Commands

Most configuration commands have a **no** form that is used to reset a command to its default value or disable a feature or function. For example, the **ip routing** command is enabled by default. To disable this command, you would issue the **no ip routing** command. To re-enable IP routing, you would issue the **ip routing** command.

Configuration commands may also have a **default** form, which returns the command settings to their default values. For commands that are disabled by default, using the **default** form has the same effect as using the **no** form of the command. For commands that are enabled by default and have default settings, the **default** form enables the command and returns the settings to their default values.

The **no** and **default** forms of commands are described in the command pages of command references.

Using the debug Command

A **debug** command produces extensive output that helps you troubleshoot problems in your network. These commands are available for many features and functions within Cisco IOS and Cisco IOS XE software. Some **debug** commands are **debug all**, **debug aaa accounting**, and **debug mpls packets**. To use **debug** commands during a Telnet session with a device, you must first enter the **terminal monitor** command. To turn off debugging completely, you must enter the **undebug all** command.

For more information about **debug** commands, see the *Cisco IOS Debug Command Reference* at http://www.cisco.com/en/US/docs/ios/debug/command/reference/db_book.html.

Caution Debugging is a high priority and high CPU utilization process that can render your device unusable. Use **debug** commands only to troubleshoot specific problems. The best times to run debugging are during periods of low network traffic and when few users are interacting with the network. Debugging during these periods decreases the likelihood that the **debug** command processing overhead will affect network performance or user access or response times.

Filtering Output Using Output Modifiers

Many commands produce lengthy output that may use several screens to display. Using output modifiers, you can filter this output to show only the information that you want to see.

Three output modifiers are available and are described as follows:

- **begin** *regular expression*—Displays the first line in which a match of the regular expression is found and all lines that follow.
- **include** *regular expression*—Displays all lines in which a match of the regular expression is found.
- **exclude** *regular expression*—Displays all lines except those in which a match of the regular expression is found.

To use one of these output modifiers, type the command followed by the pipe symbol (|), the modifier, and the regular expression that you want to search for or filter. A regular expression is a case-sensitive alphanumeric pattern. It can be a single character or number, a phrase, or a more complex string.

The following example illustrates how to filter output of the **show interface** command to display only lines that include the expression "protocol."

```
Router# show interface | include protocol

FastEthernet0/0 is up, line protocol is up
Serial4/0 is up, line protocol is up
Serial4/1 is up, line protocol is up
Serial4/2 is administratively down, line protocol is down
Serial4/3 is administratively down, line protocol is down
```

Understanding CLI Error Messages

You may encounter some error messages while using the CLI. Table 5 shows the common CLI error messages.

Table 5 Common CLI Error Messages

Error Message	Meaning	How to Get Help
% Ambiguous command: "show con"	You did not enter enough characters for the command to be recognized.	Reenter the command followed by a space and a question mark (?). The keywords that you are allowed to enter for the command appear.
% Incomplete command.	You did not enter all the keywords or values required by the command.	Reenter the command followed by a space and a question mark (?). The keywords that you are allowed to enter for the command appear.
% Invalid input detected at "^" marker.	You entered the command incorrectly. The caret (^) marks the point of the error.	Enter a question mark (?) to display all the commands that are available in this command mode. The keywords that you are allowed to enter for the command appear.

For more system error messages, see the following documents:

- *Cisco IOS Release 12.2SR System Message Guide*
- *Cisco IOS System Messages, Volume 1 of 2* (Cisco IOS Release 12.4)
- *Cisco IOS System Messages, Volume 2 of 2* (Cisco IOS Release 12.4)

Saving Changes to a Configuration

To save changes that you made to the configuration of a device, you must issue the **copy running-config startup-config** command or the **copy system:running-config nvram:startup-config** command. When you issue these commands, the configuration changes that you made are saved to the startup configuration and saved when the software reloads or power to the device is turned off or interrupted. The following example shows the syntax of the **copy running-config startup-config** command:

```
Router# copy running-config startup-config
Destination filename [startup-config]?
```

You press Enter to accept the startup-config filename (the default), or type a new filename and then press Enter to accept that name. The following output is displayed indicating that the configuration was saved:

```
Building configuration...
[OK]
Router#
```

On most platforms, the configuration is saved to NVRAM. On platforms with a Class A flash file system, the configuration is saved to the location specified by the CONFIG_FILE environment variable. The CONFIG_FILE variable defaults to NVRAM.

Additional Information

- "Using the Cisco IOS Command-Line Interface" section of the *Cisco IOS Configuration Fundamentals Configuration Guide*:

  http://www.cisco.com/en/US/docs/ios/fundamentals/configuration/guide/cf_cli-basics.html

  or

  "Using Cisco IOS XE Software" chapter of the *Cisco ASR1000 Series Aggregation Services Routers Software Configuration Guide*:

  http://www.cisco.com/en/US/docs/routers/asr1000/configuration/guide/chassis/using_cli.html

- Cisco Product Support Resources

  http://www.cisco.com/web/psa/products/index.html

- Support area on Cisco.com (also search for documentation by task or product)

  http://www.cisco.com/en/US/support/index.html

- *White Paper: Cisco IOS Reference Guide*

  http://www.cisco.com/en/US/products/sw/iosswrel/ps1828/products_white_paper09186a008018305e.shtml

- Software Download Center (downloads; tools; licensing, registration, advisory, and general information) (requires Cisco.com User ID and password)

  http://www.cisco.com/kobayashi/sw-center/

- Error Message Decoder, a tool to help you research and resolve error messages for Cisco IOS software

  http://www.cisco.com/pcgi-bin/Support/Errordecoder/index.cgi

- Command Lookup Tool, a tool to help you find detailed descriptions of Cisco IOS commands (requires Cisco.com user ID and password)

  http://tools.cisco.com/Support/CLILookup

- Output Interpreter, a troubleshooting tool that analyzes command output of supported **show** commands

  https://www.cisco.com/pcgi-bin/Support/OutputInterpreter/home.pl\

IP Addressing

Configuring IPv4 Addresses

First Published: December 14th, 2007

This chapter contains information about, and instructions for configuring IPv4 addresses on interfaces that are part of a networking device.

Note All further references to IPv4 addresses in this document use only IP in the text, not IPv4.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for IP Addresses" section on page 28.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Information About IP Addresses

To configure IP addresses, you should understand the following concepts:

- Binary Numbering, page 2
- IP Address Structure, page 4
- IP Address Classes, page 5
- IP Network Subnetting, page 7
- IP Network Address Assignments, page 8
- Classless Inter-Domain Routing, page 11
- Prefixes, page 11

Binary Numbering

IP addresses are 32 bits long. The 32 bits are divided into four octets (8-bits). A basic understanding of binary numbering is very helpful if you are going to manage IP addresses in a network because changes in the values of the 32 bits indicate either a different IP network address or IP host address.

A value in binary is represented by the number (0 or 1) in each position multiplied by the number 2 to the power of the position of the number in sequence, starting with 0 and increasing to 7, working right to left. Figure 1 is an example of an 8-digit binary number.

Figure 1 Example of an 8-digit Binary Number

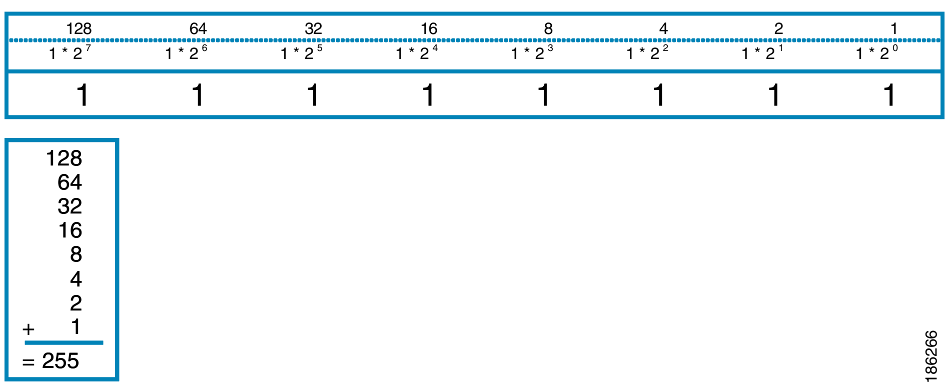


Figure 2 provides binary to decimal number conversion for 0 through 255.

Figure 2 Binary to Decimal Number Conversion for 0 to 134

00000000 = 000	00011011 = 027	00110110 = 054	01010001 = 081	01101100 = 108
00000001 = 001	00011100 = 028	00110111 = 055	01010010 = 082	01101101 = 109
00000010 = 002	00011101 = 029	00111000 = 056	01010011 = 083	01101110 = 110
00000011 = 003	00011110 = 030	00111001 = 057	01010100 = 084	01101111 = 111
00000100 = 004	00011111 = 031	00111010 = 058	01010101 = 085	01110000 = 112
00000101 = 005	00100000 = 032	00111011 = 059	01010110 = 086	01110001 = 113
00000110 = 006	00100001 = 033	00111100 = 060	01010111 = 087	01110010 = 114
00000111 = 007	00100010 = 034	00111101 = 061	01011000 = 088	01110011 = 115
00001000 = 008	00100011 = 035	00111110 = 062	01011001 = 089	01110100 = 116
00001001 = 009	00100100 = 036	00111111 = 063	01011010 = 090	01110101 = 117
00001010 = 010	00100101 = 037	01000000 = 064	01011011 = 091	01110110 = 118
00001011 = 011	00100110 = 038	01000001 = 065	01011100 = 092	01110111 = 119
00001100 = 012	00100111 = 039	01000010 = 066	01011101 = 093	01111000 = 120
00001101 = 013	00101000 = 040	01000011 = 067	01011110 = 094	01111001 = 121
00001110 = 014	00101001 = 041	01000100 = 068	01011111 = 095	01111010 = 122
00001111 = 015	00101010 = 042	01000101 = 069	01100000 = 096	01111011 = 123
00010000 = 016	00101011 = 043	01000110 = 070	01100001 = 097	01111100 = 124
00010001 = 017	00101100 = 044	01000111 = 071	01100010 = 098	01111101 = 125
00010010 = 018	00101101 = 045	01001000 = 072	01100011 = 099	01111110 = 126
00010011 = 019	00101110 = 046	01001001 = 073	01100100 = 100	01111111 = 127
00010100 = 020	00101111 = 047	01001010 = 074	01100101 = 101	10000000 = 128
00010101 = 021	00110000 = 048	01001011 = 075	01100110 = 102	10000001 = 129
00010110 = 022	00110001 = 049	01001100 = 076	01100111 = 103	10000010 = 130
00010111 = 023	00110010 = 050	01001101 = 077	01101000 = 104	10000011 = 131
00011000 = 024	00110011 = 051	01001110 = 078	01101001 = 105	10000100 = 132
00011001 = 025	00110100 = 052	01001111 = 079	01101010 = 106	10000101 = 133
00011010 = 026	00110101 = 053	01010000 = 080	01101011 = 107	10000110 = 134

186267

Figure 3 provides binary to decimal number conversion for 135 through 255.

Figure 3 Binary to Decimal Number Conversion for 135 to 255

10000111 = 135	10100010 = 162	10111101 = 189	11011000 = 216	11110011 = 243
10001000 = 136	10100011 = 163	10111110 = 190	11011001 = 217	11110100 = 244
10001001 = 137	10100100 = 164	10111111 = 191	11011010 = 218	11110101 = 245
10001010 = 138	10100101 = 165	11000000 = 192	11011011 = 219	11110110 = 246
10001011 = 139	10100110 = 166	11000001 = 193	11011100 = 220	11110111 = 247
10001100 = 140	10100111 = 167	11000010 = 194	11011101 = 221	11111000 = 248
10001101 = 141	10101000 = 168	11000011 = 195	11011110 = 222	11111001 = 249
10001110 = 142	10101001 = 169	11000100 = 196	11011111 = 223	11111010 = 250
10001111 = 143	10101010 = 170	11000101 = 197	11100000 = 224	11111011 = 251
10010000 = 144	10101011 = 171	11000110 = 198	11100001 = 225	11111100 = 252
10010001 = 145	10101100 = 172	11000111 = 199	11100010 = 226	11111101 = 253
10010010 = 146	10101101 = 173	11001000 = 200	11100011 = 227	11111110 = 254
10010011 = 147	10101110 = 174	11001001 = 201	11100100 = 228	11111111 = 255
10010100 = 148	10101111 = 175	11001010 = 202	11100101 = 229	
10010101 = 149	10110000 = 176	11001011 = 203	11100110 = 230	
10010110 = 150	10110001 = 177	11001100 = 204	11100111 = 231	
10010111 = 151	10110010 = 178	11001101 = 205	11101000 = 232	
10011000 = 152	10110011 = 179	11001110 = 206	11101001 = 233	
10011001 = 153	10110100 = 180	11001111 = 207	11101010 = 234	
10011010 = 154	10110101 = 181	11010000 = 208	11101011 = 235	
10011011 = 155	10110110 = 182	11010001 = 209	11101100 = 236	
10011100 = 156	10110111 = 183	11010010 = 210	11101101 = 237	
10011101 = 157	10111000 = 184	11010011 = 211	11101110 = 238	
10011110 = 158	10111001 = 185	11010100 = 212	11101111 = 239	
10011111 = 159	10111010 = 186	11010101 = 213	11110000 = 240	
10100000 = 160	10111011 = 187	11010110 = 214	11110001 = 241	
10100001 = 161	10111100 = 188	11010111 = 215	11110010 = 242	

186271

IP Address Structure

An IP host address identifies a device to which IP packets can be sent. An IP network address identifies a specific network segment to which one or more hosts can be connected. The following are characteristics of IP addresses:

- IP addresses are 32 bits long
- IP addresses are divided into four sections of one byte (octet) each
- IP addresses are typically written in a format known as dotted decimal

Table 1 shows some examples of IP addresses.

Table 1 Examples of IP Addresses

IP Addresses in Dotted Decimal	IP Addresses in Binary
10.34.216.75	00001010.00100010.11011000.01001011
172.16.89.34	10101100.00010000.01011001.00100010
192.168.100.4	11000000.10101000.01100100.00000100

Note The IP addresses in Table 1 are from RFC 1918, *Address Allocation for Private Internets*. These IP addresses are not routable on the Internet. They are intended for use in private networks. For more information on RFC1918, see http://www.ietf.org/rfc/rfc1918.txt.

IP addresses are further subdivided into two sections known as network and host. The division is accomplished by arbitrarily ranges of IP addresses to classes. For more information see RFC 791 Internet Protocol at http://www.ietf.org/rfc/rfc0791.txt.

IP Address Classes

In order to provide some structure to the way IP addresses are assigned, IP addresses are grouped into classes. Each class has a range of IP addresses. The range of IP addresses in each class is determined by the number of bits allocated to the network section of the 32-bit IP address. The number of bits allocated to the network section is represented by a mask written in dotted decimal or with the abbreviation /*n* where *n* = the numbers of bits in the mask.

Table 2 lists ranges of IP addresses by class and the masks associated with each class. The digits in bold indicate the network section of the IP address for each class. The remaining digits are available for host IP addresses. For example, IP address 10.90.45.1 with a mask of 255.0.0.0 is broken down into a network IP address of 10.0.0.0 and a host IP address of 0.90.45.1.

Table 2 IP Address Ranges by Class with Masks

Class	Range
A (range/mask in dotted decimal)	**0**.0.0.0 to 127.0.0.0/8 (255.0.0.0)
A (range in binary)	**00000000**.00000000.00000000.00000000 to **01111111**.00000000.00000000.00000000
A (mask in binary)	11111111.00000000.00000000.00000000/8
B (range/mask in dotted decimal)	**128**.0.0.0 to **191**.**255**.0.0/16 (255.255.0.0)
B (range in binary)	**10000000**.**00000000**.00000000.00000000 to **10111111**.**11111111**.00000000.00000000
B (mask in binary)	**11111111**.**11111111**.00000000.00000000/16
C (range/mask in dotted decimal)	**192**.**0**.**0**.0 to **223**.**255**.**255**.0/24 (255.255.255.0)
C (range in binary)	**11000000**.**00000000**.**00000000**.00000000 to **11011111**.**11111111**.**11111111**.00000000
C (mask in binary)	11111111.11111111.11111111.0000000/24
D[1] (range/mask in dotted decimal)	**224**.**0**.**0**.**0** to **239**.**255**.**255**.**255**/32 (255.255.255.255)
D (range in binary)	**11100000**.**00000000**.**00000000**.**00000000** to **11101111**.**11111111**.**11111111**.**11111111**
D (mask in binary)	11111111.11111111.11111111.11111111/32
E[2] (range/mask in dotted decimal)	**240**.**0**.**0**.**0** to **255**.**255**.**255**.**255**/32 (255.255.255.255)
E (range in binary)	**11110000**.**00000000**.**00000000**.**00000000** to **11111111**.**11111111**.**11111111**.**11111111**
E (mask in binary)	11111111.11111111.11111111.11111111/32

1. Class D IP addresses are reserved for multicast applications.

2. Class E IP addresses are reserved for broadcast traffic.

Note Some IP addresses in these ranges are reserved for special uses. For more information refer to RFC 3330, *Special-Use IP Addresses*, at http://www.ietf.org/rfc/rfc3330.txt.

When a digit that falls within the network mask changes from 1 to 0 or 0 to 1 the network address is changed. For example, if you change 10101100.00010000.01011001.00100010/16 to 10101100.00110000.01011001.00100010/16 you have changed the network address from 172.16.89.34/16 to 172.48.89.34/16.

When a digit that falls outside the network mask changes from 1 to 0 or 0 to 1 the host address is changed. For example, if you change 10101100.00010000.01011001.00100010/16 to 10101100.00010000.01011001.00100011/16 you have changed the host address from 172.16.89.34/16 to 172.16.89.35/16.

Each class of IP address supports a specific range of IP network addresses and IP host addresses. The range of IP network addresses available for each class is determined with the formula 2 to the power of the number of available bits. In the case of class A addresses, the value of the first bit in the 1st octet (as shown in Table 2) is fixed at 0. This leaves 7 bits for creating additional network addresses. Therefore there are 128 IP network addresses available for class A ($2^7 = 128$).

The number of IP host addresses available for an IP address class is determined by the formula 2 to the power of the number of available bits minus 2. There are 24 bits available in a class A addresses for IP host addresses. Therefore there are 16,777,214 IP hosts addresses available for class A ($((2^{24}) - 2 = 16{,}777{,}214)$).

Note The 2 is subtracted because there are 2 IP addresses that cannot be used for a host. The all 0's host address cannot be used because it is the same as the network address. For example, 10.0.0.0 cannot be both a IP network address and an IP host address. The all 1's address is a broadcast address that is used to reach all hosts on the network. For example, an IP datagram addressed to 10.255.255.255 will be accepted by every host on network 10.0.0.0.

Table 3 shows the network and host addresses available for each class of IP address.

Table 3 Network and Host Addresses Available for Each Class of IP Address

Class	Network Addresses	Host Addresses
A	128	16,777,214
B	16,384[1]	65534
C	2,097,152[2]	254

1. There are only 14 bits available for class B IP network addresses because the first 2 bits are fixed at 10 as shown in Table 2.
2. There are only 21 bits available for class C IP network addresses because the first 3bits are fixed at 110 as shown in Table 2.

IP Network Subnetting

The arbitrary subdivision of network and host bits in IP address classes resulted in an inefficient allocation of IP space. For example, if your network has 16 separate physical segments you will need 16 IP network addresses. If you use 16 class B IP network addresses, you would be able to support 65,534 hosts on each of the physical segments. Your total number of supported host IP addresses is 1,048,544 (16 * 65,534 = 1,048,544). Very few network technologies can scale to having 65,534 hosts on a single network segment. Very few companies need 1,048,544 IP host addresses. This problem required the development of a new strategy that permitted the subdivision of IP network addresses into smaller groupings of IP subnetwork addresses. This strategy is known as subnetting.

If your network has 16 separate physical segments you will need 16 IP subnetwork addresses. This can be accomplished with one class B IP address. For example, start with the class B IP address of 172.16.0.0 you can reserve 4 bits from the third octet as subnet bits. This gives you 16 subnet IP addresses $2^4 = 16$. Table 4 shows the IP subnets for 172.16.0.0/20.

Table 4 Examples of IP Subnet Addresses using 172.16.0.0/20

Number	IP Subnet Addresses in Dotted Decimal	IP Subnet Addresses in Binary
0[1]	172.16.0.0	10101100.00010000.00000000.00000000
1	172.16.16.0	10101100.00010000.00010000.00000000
2	172.16.16.32.0	10101100.00010000.00100000.00000000
3	172.16.48.0	10101100.00010000.00110000.00000000
4	172.16.64.0	10101100.00010000.01000000.00000000
5	172.16.80.0	10101100.00010000.01010000.00000000
6	172.16.96.0	10101100.00010000.01100000.00000000
7	172.16.112.0	10101100.00010000.01110000.00000000
8	172.16.128.0	10101100.00010000.10000000.00000000
9	172.16.144.0	10101100.00010000.10010000.00000000
10	172.16.160.0	10101100.00010000.10100000.00000000
11	172.16.176.0	10101100.00010000.10110000.00000000
12	172.16.192.0	10101100.00010000.11000000.00000000
13	172.16.208.0	10101100.00010000.11010000.00000000
14	172.16.224.0	10101100.00010000.11100000.00000000
15	172.16.240.0	10101100.00010000.11110000.00000000

1. The first subnet that has all of the subnet bits set to 0 is referred to as *subnet 0*. It is indistinguishable from the network address and must be used carefully.

When a digit that falls within the subnetwork (subnet) mask changes from 1 to 0 or 0 to 1 the subnetwork address is changed. For example, if you change 10101100.00010000.01011001.00100010/20 to 10101100.00010000.01111001.00100010/20 you have changed the network address from 172.16.89.34/20 to 172.48.121.34/20.

When a digit that falls outside the subnet mask changes from 1 to 0 or 0 to 1 the host address is changed. For example, if you change 10101100.00010000.01011001.00100010/20 to 10101100.00010000.01011001.00100011/20 you have changed the host address from 172.16.89.34/20 to 172.16.89.35/20.

Timesaver To avoid having to do manual IP network, subnetwork, and host calculations, use one of the free IP subnet calculators available on the Internet.

Some people get confused about the terms *network address* and *subnet* or *subnetwork addresses* and when to use them. In the most general sense the term *network address* means "the IP address that routers use to route traffic to a specific network segment so that the intended destination IP host on that segment can receive it". Therefore the term *network address* can apply to both non-subnetted and subnetted IP network addresses. When you are troubleshooting problems with forwarding traffic from a router to a specific IP network address that is actually a subnetted network address, it can help to be more specific by referring to the destination network address as a subnet network address because some routing protocols handle advertising subnet network routes differently from network routes. For example, the default behavior for RIP v2 is to automatically summarize the subnet network addresses that it is connected to their non-subnetted network addresses (172.16.32.0/24 is advertised by RIP v2 as 172.16.0.0/16) when sending routing updates to other routers. Therefore the other routers might have knowledge of the IP network addresses in the network, but not the subnetted network addresses of the IP network addresses.

Tip The term *IP address space* is sometimes used to refer to a range of IP addresses. For example, "We have to allocate a new IP network address to our network because we have used all of the available IP addresses in the current *IP address space*".

IP Network Address Assignments

Routers keep track of IP network addresses to understand the network IP topology (layer 3 of the OSI reference model) of the network to ensure that IP traffic can be routed properly. In order for the routers to understand the network layer (IP) topology, every individual physical network segment that is separated from any other physical network segment by a router must have a unique IP network address.

Figure 4 shows an example of a simple network with correctly configured IP network addresses. The routing table in R1 looks like Table 5.

Table 5 Routing Table for a Correctly Configured Network

Interface Ethernet 0	Interface Ethernet 1
172.31.32.0/24 (Connected)	172.31.16.0/24 (Connected)

Figure 4 Correctly Configured Network

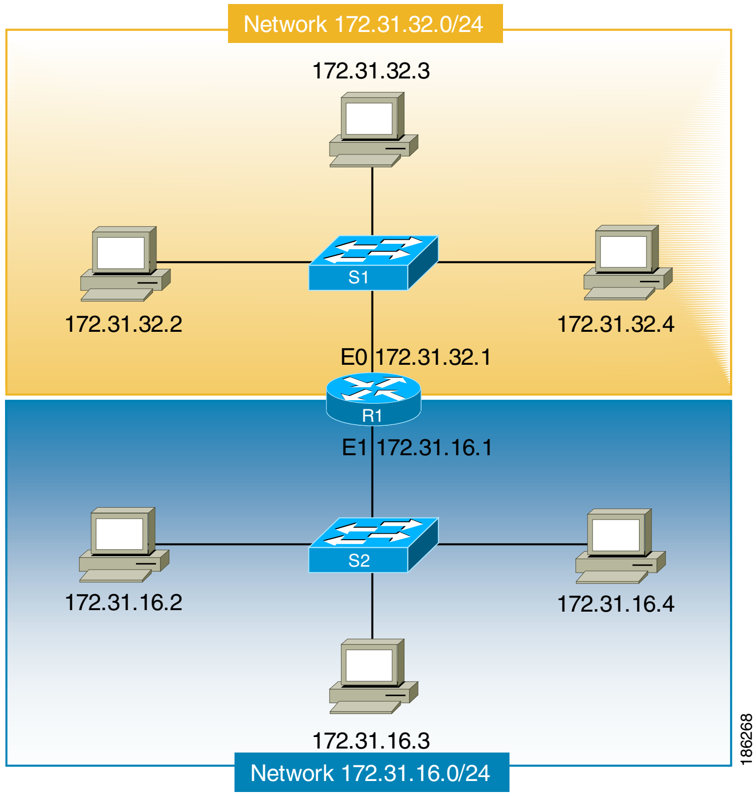
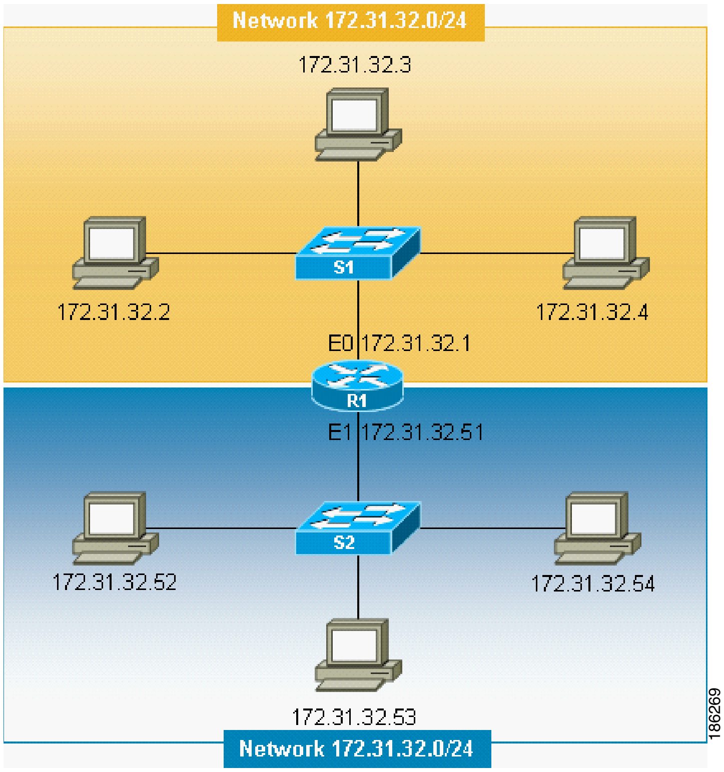


Figure 5 shows an example of a simple network with incorrectly configured IP network addresses. The routing table in R1 looks like Table 6. If the PC with IP address 172.31.32.3 attempts to send IP traffic to the PC with IP address 172.31.32.54, router R1 cannot determine which interface that the PC with IP address 172.31.32.54 is connected to.

Table 6 Routing Table in Router R1 for an Incorrectly Configured Network (Example 1)

Ethernet 0	Ethernet 1
172.31.32.0/24 (Connected)	172.31.32.0/24 (Connected)

Figure 5 Incorrectly Configured Network (Example 1)



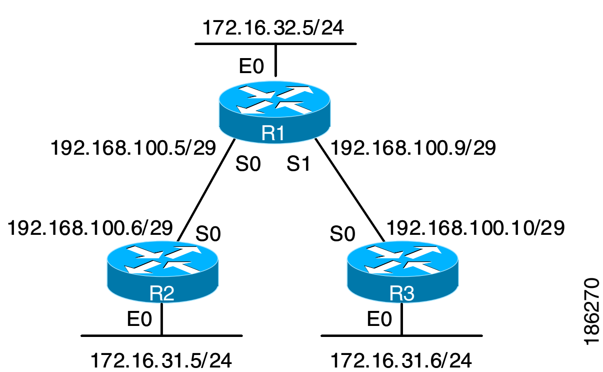
To help prevent mistakes as shown in Figure 5, Cisco IOS-based networking devices will not allow you to configure the same IP network address on two or more interfaces in the router when IP routing is enabled.

The only way to prevent the mistake shown in Figure 6, where 172.16.31.0/24 is used in R2 and R3, is to have very accurate network documentation that shows where you have assigned IP network addresses.

Table 7 Routing Table in Router R1 for an Incorrectly Configured Network (Example 2)

Ethernet 0	Serial 0	Serial 1
172.16.32.0/24 (Connected)	192.168.100.4/29 (Connected) 172.16.31.0/24 RIP	192.168.100.8/29 (Connected) 172.16.31.0/24 RIP

Figure 6 Incorrectly Configured Network (example 2)



Fore a more thorough explanation of IP routing, see the "Related Documents" section on page 26 for a list of documents related to IP routing.

Classless Inter-Domain Routing

Due to the continuing increase in internet use and the limitations on how IP addresses can be assigned using the class structure shown in Table 2, a more flexible method for allocating IP addresses was required. The new method is documented in RFC 1519 *Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy*. CIDR allows network administrators to apply arbitrary masks to IP addresses to create an IP addressing plan that meets the requirements of the networks that they administrate.

For more information on CIDR, refer to RFC 1519 at http://www.ietf.org/rfc/rfc1519.txt).

Prefixes

The term *prefix* is often used to refer to the number of bits of an IP network address that are of importance for building routing tables. If you are using only classful (strict adherence to A, B, and C network address boundaries) IP addresses, the prefixes are the same as the masks for the classes of addresses. For example, using classful IP addressing, a class C IP network address such as 192.168.10.0 uses a 24-bit mask (/24 or 255.255.255.0) and can also be said to have a 24-bit prefix.

If you are using CIDR, the prefixes are arbitrarily assigned to IP network addresses based on how you want to populate the routing tables in your network. For example, a group of class C IP addresses such as 192.168.10.0, 192.168.11.0, 192.168.12.0, 192.168.13.0 can be advertised as a single route to 192.168.0.0 with a 16-bit prefix (192.168.0.0/16). This results in a 4:1 reduction in the number of routes that the routers in your network need to manage.

How to Configure IP Addresses

This section contains the following tasks:

Establishing IP Connectivity to a Network by Assigning an IP Address to an Interface

Perform this task to configure an IP address on an interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no shutdown**
5. **ip address** *ip-address mask*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface fastethernet 0/0`	Specifies an interface and enters interface configuration mode.
Step 4	**no shutdown** **Example:** `Router(config-if)# no shutdown`	Enables the interface.
Step 5	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 172.16.16.1 255.255.240.0`	Configures the IP address on the interface.
Step 6	**end** **Example:** `Router(config-if)# end`	Exits the current configuration mode and returns to privileged EXEC mode.

Troubleshooting Tips

The following commands can help troubleshoot IP addressing:

- **show ip interface**—Displays the IP parameters for the interface.
- **show ip route connected**—Displays the IP networks the networking device is connected to.

Increasing the Number of IP Hosts that Are Supported on a Network by Using Secondary IP Addresses

If you have a situation in which you need to connect more IP hosts to a network segment and you have used all of the available IP host addresses for the subnet to which you have assigned the segment, you can avoid having to readdress all of the hosts with a different subnet by adding a second IP network address to the network segment.

Perform this task to configure a secondary IP address on an interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no shutdown**
5. **ip address** *ip-address mask*
6. **ip address** *ip-address mask* **secondary**
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`enable`** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**`configure terminal`** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**`interface`** *`type number`* **Example:** `Router(config)# interface fastethernet 0/0`	Specifies an interface and enters interface configuration mode.
Step 4	**`no shutdown`** **Example:** `Router(config-if)# no shutdown`	Enables the interface.
Step 5	**`ip address`** *`ip-address mask`* **Example:** `Router(config-if)# ip address 172.16.16.1 255.255.240.0`	Configures the IP address on the interface.

	Command or Action	Purpose
Step 6	**ip address** *ip-address mask* **secondary** **Example:** `Router(config-if)# ip address 172.16.32.1 255.255.240.0 secondary`	Configures the secondary IP address on the interface.
Step 7	**end** **Example:** `Router(config-if)# end`	Exits the current configuration mode and returns to privileged EXEC mode.

Troubleshooting Tips

The following commands can help troubleshoot IP addressing:

- **show ip interface**—Displays the IP parameters for the interface.
- **show ip route connected**—Displays the IP networks the networking device is connected to.

What to Do Next

If your network has two or more routers and you have already configured a routing protocol, make certain that the other routers can reach the new IP network that you assigned. You might need to modify the configuration for the routing protocol on the router so that it advertises the new network. Consult the *Cisco IOS IP Routing Protocols Configuration Guide*, Release 12.4, at this URL for information on configuring routing protocols:
http://www.cisco.com/en/US/products/ps6350/products_configuration_guide_book09186a0080437e22.html.

Reducing the Number of IP Addresses Required to Establish IP Connectivity by Using IP Unnumbered on Point-to-Point WAN Interfaces

If you have a limited number of IP network or subnet addresses and you have point-to-point WANs in your network, you can use the IP Unnumbered Interfaces feature to enable IP connectivity on the point-to-point WAN interfaces without actually assigning an IP address to them.

Perform this task to configure the IP Unnumbered Interfaces feature on a point-to-point WAN interface.

- IP Unnumbered Feature, page 14
- Restrictions, page 15
- SUMMARY STEPS, page 15
- DETAILED STEPS, page 15
- Troubleshooting Tips, page 17

IP Unnumbered Feature

The IP Unnumbered Interfaces feature enables IP processing on a point-to-point WAN interface without assigning it an explicit IP address. The IP unnumbered point-to-point WAN interface uses the IP address of another interface to enable IP connectivity, which conserves network addresses.

Restrictions

The following restrictions apply to the IP Unnumbered Interfaces feature:

- The IP Unnumbered Interfaces feature is only supported on point-to-point (non-multiaccess) WAN interfaces
- You cannot netboot a Cisco IOS image over an interface that is using the IP Unnumbered Interfaces feature

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no shutdown**
5. **ip address** *ip-address mask*
6. **interface** *type number*
7. **no shutdown**
8. **ip unnumbered** *type number*
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface fastethernet 0/0`	Specifies an interface and enters interface configuration mode.
Step 4	**no shutdown** **Example:** `Router(config-if)# no shutdown`	Enables the interface.
Step 5	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 172.16.16.1 255.255.240.0`	Configures the IP address on the interface.

	Command or Action	Purpose
Step 6	**interface** *type number* **Example:** `Router(config-if)# interface serial 0/0`	Specifies a point-to-point WAN interface and enters interface configuration mode.
Step 7	**no shutdown** **Example:** `Router(config-if)# no shutdown`	Enables the point-to-point WAN interface.
Step 8	**ip unnumbered** *type number* **Example:** `Router(config-if)# ip unnumbered fastethernet 0/0`	Enables the IP unnumbered feature on the point-to-point WAN interface. In this example the point-to-point WAN interface uses IP address 172.16.16.1 from Fast Ethernet 0/0.
Step 9	**end** **Example:** `Router(config-if)# end`	Exits the current configuration mode and returns to privileged EXEC mode.

Troubleshooting Tips

The following commands can help troubleshoot IP addressing:

- **show ip interface**—Displays the IP parameters for the interface.
- **show ip route connected**—Displays the IP networks the networking device is connected to.

Reducing the Number of IP Addresses Required to Establish IP Connectivity by Using IP addresses with 31-Bit Prefixes on Point-to-Point WAN Interfaces

You can reduce the number of IP subnets used by networking devices to establish IP connectivity to point-to-point WANs that they are connected to by using IP Addresses with 31-bit Prefixes as defined in RFC 3021.

Perform this task to configure an IP address with a 31-bit prefix on a point-to-point WAN interface.

- RFC 3021, page 17
- Prerequisites, page 18
- Restrictions, page 18
- SUMMARY STEPS, page 18
- DETAILED STEPS, page 19
- Troubleshooting Tips, page 19

RFC 3021

Prior to RFC 3021, *Using 31-bit Prefixes on IPv4 Point-to-Point Links*, many network administrators assigned IP address with a 30-bit subnet mask (255.255.255.252) to point-to-point interfaces to conserve IP address space. Although this practice does conserve IP address space compared to assigning IP addresses with shorter subnet masks such as 255.255.255.240, IP addresses with a 30-bit subnet mask still require four addresses per link: two host addresses (one for each host interface on the link), one all-zeros network address, and one all-ones broadcast network address.

Table 8 shows an example of the four IP addresses that are created when a 30-bit (otherwise known as 255.255.255.252 or /30) subnet mask is applied to the IP address 192.168.100.4. The bits that are used to specify the host IP addresses in bold.

Table 8 Four IP Addresses Created When a 30-Bit Subnet Mask (/30) is Used

Address	Description	Binary
192.168.100.4/30	All-zeros IP address	11000000.10101000.01100100.000001**00**
192.168.100.5/30	First host addresses	11000000.10101000.01100100.000001**01**
192.168.100.6/30	Second host address	11000000.10101000.01100100.000001**10**
192.168.100.7/30	All-ones broadcast address	11000000.10101000.01100100.000001**11**

Point-to-point links only have two endpoints (hosts) and do not require broadcast support because any packet that is transmitted by one host is always received by the other host. Therefore the all-ones broadcast IP address is not required for a point-to-point interface.

The simplest way to explain RFC 3021 is to say that the use of a 31-bit prefix (created by applying a 31-bit subnet mask to an IP address) allows the all-zeros and all-ones IP addresses to be assigned as host addresses on point-to-point networks. Prior to RFC 3021 the longest prefix in common use on point-to-point links was 30-bits, which meant that the all-zeros and all-ones IP addresses were wasted.

Table 9 shows an example of the two IP addresses that are created when a 31-bit (otherwise known as 255.255.255.254 or /31) subnet mask is applied to the IP address 192.168.100.4. The bit that is used to specify the host IP addresses in bold

Table 9 Four IP Addresses Created When a 31-Bit Subnet Mask (/31) is Used

Address	Description	Binary
192.168.100.4/31	First host address	11000000.10101000.01100100.0000010**0**
192.168.100.5/31	Second host address	11000000.10101000.01100100.0000010**1**

The complete text for RFC 3021 is available at http://www.ietf.org/rfc/rfc3021.txt.

Prerequisites

You must have classless IP addressing configured on your networking device before you configure an IP address with a 31-bit prefix on a point-to-point interface. Classless IP addressing is enabled by default in many versions of Cisco IOS software. If you are not certain that your networking device has IP classless addressing configured, enter the **ip classless** command in global configuration mode to enable it.

Restrictions

This task can only be performed on point-to-point (non-multi-access) WAN interfaces.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip classless**
4. **interface** *type number*
5. **no shutdown**
6. **ip address** *ip-address mask*
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`enable`** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**`configure terminal`** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**`ip classless`** **Example:** `Router(config)# ip classless`	(Optional) Enables IP classless (CIDR). **Note** This command is enabled by default in many versions of Cisco IOS. If you are not certain if it is enabled by default in the version of Cisco IOS that your networking device is running, enter the **ip classless** command as shown. When you are done with this task view the configuration. If the **ip classless** command does not appear in your configuration, it is enabled by default.
Step 4	**`interface`** *`type number`* **Example:** `Router(config)# interface serial 0/0`	Specifies a point-to-point WAN interface and enters interface configuration mode.
Step 5	**`no shutdown`** **Example:** `Router(config-if)# no shutdown`	Enables the interface.
Step 6	**`ip address`** *`ip-address mask`* **Example:** `Router(config-if)# ip address 192.168.100.4 255.255.255.254`	Configures the 31bit prefix IP address on the point-to-point WAN interface.
Step 7	**`end`** **Example:** `Router(config-if)# end`	Exits the current configuration mode and returns to privileged EXEC mode.

Troubleshooting Tips

The following commands can help troubleshoot IP addressing:

- **show ip interface**—Displays the IP parameters for the interface.
- **show ip route connected**—Displays the IP networks the networking device is connected to.

Maximizing the Number of Available IP Subnets by Allowing the use of IP Subnet Zero

If you using subnetting in your network and you are running out of network addresses, you can configure your networking device to allow the configuration of subnet zero. This adds one more usable network address for every subnet in your IP addressing scheme. Table 4 shows the IP subnets (including subnet 0) for 172.16.0.0/20.

Perform this task to enable the use of IP subnet zero on your networking device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip subnet-zero**
4. **interface** *type number*
5. **no shutdown**
6. **ip address** *ip-address mask*
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip subnet-zero** **Example:** `Router(config)# ip subnet-zero`	Enables the use of IP subnet zero.
Step 4	**interface** *type number* **Example:** `Router(config)# interface fastethernet 0/0`	Specifies an interface and enters interface configuration mode.
Step 5	**no shutdown** **Example:** `Router(config-if)# no shutdown`	Enables the interface.

	Command or Action	Purpose
Step 6	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 172.16.0.1 255.255.240.0`	Configures the subnet zero IP address on the interface.
Step 7	**end** **Example:** `Router(config-if)# end`	Exits the current configuration mode and returns to privileged EXEC mode.

Troubleshooting Tips

The following commands can help troubleshoot IP addressing:

- **show ip interface**—Displays the IP parameters for the interface.
- **show ip route connected**—Displays the IP networks the networking device is connected to.

Specifying the Format of Network Masks

By default, **show** commands display an IP address and then its netmask in dotted decimal notation. For example, a subnet would be displayed as 131.108.11.55 255.255.255.0.

You might find it more convenient to display the network mask in hexadecimal format or bit count format instead. The hexadecimal format is commonly used on UNIX systems. The previous example would be displayed as 131.108.11.55 0XFFFFFF00.

The bit count format for displaying network masks is to append a slash (/) and the total number of bits in the netmask to the address itself. The previous example would be displayed as 131.108.11.55/24.

- Specify the Format in Which Netmasks Appear for the Current Session
- Specify the Format in Which Netmasks Appear for an Individual Line

Specify the Format in Which Netmasks Appear for the Current Session

Perform this task to specify the format in which netmasks appear for the current session.

SUMMARY STEPS

1. **enable**
2. **term ip netmask-format {bitcount | decimal | hexadecimal}**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**term ip netmask-format** {**bitcount** | **decimal** | **hexadecimal**} **Example:** `Router# term ip netmask-format hexadecimal`	Specifies the format the router uses to display network masks.

Specify the Format in Which Netmasks Appear for an Individual Line

Perform this task to specify the format in which netmasks appear for an individual line.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **line vty** *first last*
4. **term ip netmask-format** {**bitcount** | **decimal** | **hexadecimal**}
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**line vty** *first last* **Example:** `Router(config)# line vty 0 4`	Enters line configuration mode for the range of lines specified by the *first* and *last* arguments.

	Command or Action	Purpose
Step 4	**term ip netmask-format** {**bitcount** | **decimal** | **hexadecimal**} **Example:** `Router(config-line)# ip netmask-format hexadecimal`	Specifies the format the router uses to display the network mask for an individual line.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits the current configuration mode and returns to privileged EXEC mode.

Configuration Examples for IP Addresses

This section provides the following configuration examples:

- Establishing IP Connectivity to a Network by Assigning an IP Address to an Interface: Example, page 23
- Increasing the Number of IP Hosts that are Supported on a Network by Using Secondary IP Addresses: Example, page 24
- Reducing the Number of IP Addresses Required to Establish IP Connectivity by Using IP Unnumbered on Point-to-Point WAN Interfaces: Example, page 24
- Reducing the Number of IP Addresses Required to Establish IP Connectivity by Using IP addresses with 31-Bit Prefixes on Point-to-Point WAN Interfaces: Example, page 24
- Maximizing the Number of Available IP Subnets by Allowing the use of IP Subnet Zero: Example, page 25

Establishing IP Connectivity to a Network by Assigning an IP Address to an Interface: Example

The following example configures an IP address on three interfaces:

```
!
interface FastEthernet0/0
 no shutdown
 ip address 172.16.16.1 255.255.240.0
!
interface FastEthernet0/1
 no shutdown
 ip address 172.16.32.1 255.255.240.0
!
interface FastEthernet0/2
 no shutdown
 ip address 172.16.48.1 255.255.240.0
!
```

Increasing the Number of IP Hosts that are Supported on a Network by Using Secondary IP Addresses: Example

The following example configures secondary IP addresses on three interfaces:

```
!
interface FastEthernet0/0
 no shutdown
 ip address 172.16.16.1 255.255.240.0
 ip address 172.16.32.1 255.255.240.0 secondary
!
!
interface FastEthernet0/1
 no shutdown
 ip address 172.17.16.1 255.255.240.0
 ip address 172.17.32.1 255.255.240.0 secondary
!
!
interface FastEthernet0/2
 no shutdown
 ip address 172.18.16.1 255.255.240.0
 ip address 172.18.32.1 255.255.240.0 secondary
!
```

Reducing the Number of IP Addresses Required to Establish IP Connectivity by Using IP Unnumbered on Point-to-Point WAN Interfaces: Example

The following example configures the unnumbered IP feature on three interfaces:

```
!
interface FastEthernet0/0
 no shutdown
 ip address 172.16.16.1 255.255.240.0
!
interface serial0/0
 no shutdown
 ip unnumbered fastethernet0/0
!
interface serial0/1
 no shutdown
 ip unnumbered fastethernet0/0
!
interface serial0/2
 no shutdown
 ip unnumbered fastethernet0/0
!
```

Reducing the Number of IP Addresses Required to Establish IP Connectivity by Using IP addresses with 31-Bit Prefixes on Point-to-Point WAN Interfaces: Example

The following example configures 31-bit prefixes on two interfaces:

```
!
ip classless
```

```
!
interface serial0/0
 no shutdown
 ip address 192.168.100.2 255.255.255.254
!
!
interface serial0/1
 no shutdown
 ip address 192.168.100.4 255.255.255.254
```

Maximizing the Number of Available IP Subnets by Allowing the use of IP Subnet Zero: Example

The following example enables subnet zero:

```
!
interface FastEthernet0/0
 no shutdown
 ip address 172.16.16.1 255.255.240.0
!
ip subnet-zero
!
```

Where to Go Next

If your network has two or more routers and you have not already configured a routing protocol, consult the *Cisco IOS IP Routing Protocols Configuration Guide*, Release 12.4, at this URL for information on configuring routing protocols:
http://www.cisco.com/en/US/products/ps6350/products_configuration_guide_book09186a0080437e22.html.

Additional References

The following sections provide references related to IP Addresses.

Related Documents

Related Topic	Document Title
IP addressing commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Addressing Services Command Reference*
Fundamental principles of IP addressing and IP routing	IP Routing Primer ISBN 1578701082

Standards

Standard	Title
No new or modified standards are supported, and support for existing standards has not been modified	—

MIBs

MIB	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified	—

RFCs

RFC[1]	Title
RFC 791	*Internet Protocol* http://www.ietf.org/rfc/rfc0791.txt
RFC 1338	*Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy* http://www.ietf.org/rfc/rfc1519.txt
RFC 1466	*Guidelines for Management of IP Address Space* http://www.ietf.org/rfc/rfc1466.txt
RFC 1716	*Towards Requirements for IP Routers* http://www.ietf.org/rfc/rfc1716.txt
RFC 1918	*Address Allocation for Private Internets* http://www.ietf.org/rfc/rfc1918.txt
RFC 3330	*Special-Use IP Addresses* http://www.ietf.org/rfc/rfc3330.txt

1. These references are only a sample of the many RFCs available on subjects related to IP addressing and IP routing. Refer to the IETF RFC site at http://www.ietf.org/rfc.html for a full list of RFCs.

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for IP Addresses

Table 10 lists the features in this module and provides links to specific configuration information.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 10 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 10 Feature Information for IP Addresses

Feature Name	Releases	Feature Information
Using 31-bit Prefixes on IP Point-to-Point Links	12.0(14)S 12.2(4)T	In order to conserve IP address space on the Internet, a 31-bit prefix length allows the use of only two IP addresses on a point-to-point link. Previously, customers had to use four IP addresses or unnumbered interfaces for point-to-point links. The following sections provide information about this feature: • Reducing the Number of IP Addresses Required to Establish IP Connectivity by Using IP addresses with 31-Bit Prefixes on Point-to-Point WAN Interfaces, page 17
IP Unnumbered Interfaces	10.0	In order to conserve IP address space, IP unnumbered interfaces use the IP address of another interface to enable IP connectivity. The following command was introduced or modified: **ip unnumbered**.

Table 10 Feature Information for IP Addresses (continued)

Feature Name	Releases	Feature Information
IP Subnet Zero	10.0	In order to conserve IP address space IP Subnet Zero allows the use of the all-zeros subnet as an IP address on an interface, such as configuring 172.16.0.1/24 on Fast Ethernet 0/0. The following command was introduced or modified: **ip subnet-zero**.
Classless Inter-Domain Routing	10.0	CIDR is a new way of looking at IP addresses that eliminates the concept of classes (class A, class B, and so on). For example, network 192.213.0.0, which is an illegal class C network number, is a legal supernet when it is represented in CIDR notation as 192.213.0.0/16. The /16 indicates that the subnet mask consists of 16 bits (counting from the left). Therefore, 192.213.0.0/16 is similar to 192.213.0.0 255.255.0.0. The following command was introduced or modified: **ip classless**.

ARP

Configuring Address Resolution Protocol Options

First Published: May 2, 2005
Last Updated: May 2, 2008

Address Resolution Protocol (ARP) performs a required function in IP routing. ARP finds the hardware address, also known as Media Access Control (MAC) address, of a host from its known IP address. ARP maintains a cache (table) in which MAC addresses are mapped to IP addresses. ARP is part of all Cisco IOS systems running IP.

This document explains ARP for IP routing and the optional ARP features you can configure, such as static ARP entries, time out for dynamic ARP entries, clearing the cache, and Proxy ARP.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for Configuring Address Resolution Protocol Options" section on page 20.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Information About Address Resolution Protocol Options

To configure the ARP options, you need to understand the following concepts:

Layer 2 and Layer 3 Addressing

IP addressing occurs at Layer 2 (data link) and Layer 3 (network) of the Open System Interconnection (OSI) reference model. OSI is an architectural network model developed by ISO and ITU-T that consists of seven layers, each of which specifies particular network functions such as addressing, flow control, error control, encapsulation, and reliable message transfer.

Layer 2 addresses are used for local transmissions between devices that are directly connected. Layer 3 addresses are used for indirectly connected devices in an internetwork environment. Each network uses addressing to identify and group devices so that transmissions can be sent and received. Ethernet (802.2, 802.3, Ethernet II, and Subnetwork Access Protocol [SNAP]), Token Ring, and Fiber Distributed Data Interface (FDDI) use Media Access Control (MAC) addresses that are "burned in" to the Network Interface Card (NIC). The most commonly used network types are Ehternet II and SNAP.

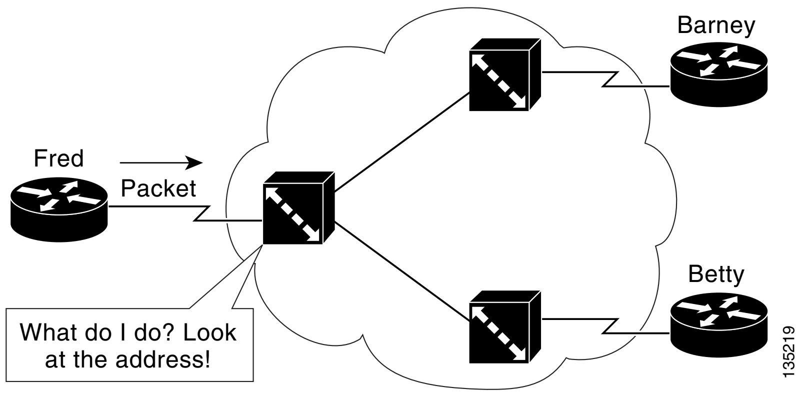
In order for devices to be able to communicate with each when they are not part of the same network, the 48-bit MAC address must be mapped to an IP address. Some of the Layer 3 protocols used to perform the mapping are:

- Address Resolution Protocol (ARP)
- Reverse ARP (RARP)
- Serial Line ARP (SLARP)
- Inverse ARP

For the purposes of IP mapping, Ethernet, Token Ring, and FDDI frames contain the destination and source addresses Frame Relay and Asynchronous Transfer Mode (ATM) networks, which are packet switched, data packets take different routes to reach the same destination. At the receiving end, the packet is reassembled in the correct order.

In a Frame Relay network, there is one physical link that has many logical circuits called virtual circuits (VCs). The address field in the frame contains a data-link connection identifier (DLCI) which identifies each VC. For example, in Figure 1, the Frame Relay switch to which router Fred is connected receives frames; the switch forwards the frames to either Barney or Betty based on the DLCI which identifies each VC. So Fred has one physical connection but multiple logical connections.

Figure 1 Frame Relay Network



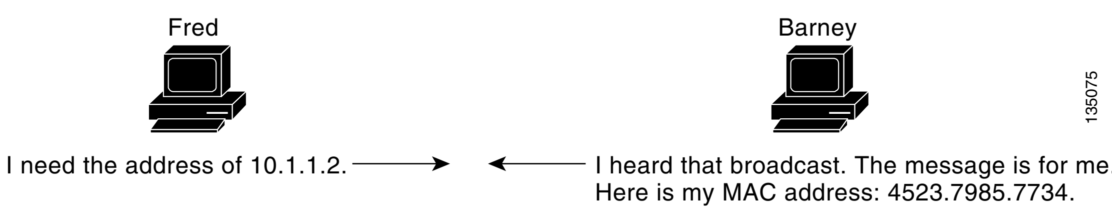
ATM networks use point-to-point serial links with the High-Level Data Link Control (HDLC) protocol. HDLC includes a meaningless address field included in five bytes of the frame header frame with the recipient implied since there can only be one.

AppleTalk is designed for Apple computers and has a special addressing scheme that uses 24-bit addresses and its own method for resolving addresses. Once the data reaches the internetwork, address resolution beyond the device connecting it to the internetwork operates the same as IP address resolution. For more information about AppleTalk networks, refer to Core Competence AppleTalk (white paper) at www.corecom.com/html/appletalk.html.

Address Resolution Protocol

Address Resolution Protocol (ARP) was developed to enable communications on an internetwork and is defined by RFC 826. Routers and Layer 3 switches need ARP to map IP addresses to MAC hardware addresses so that IP packets can be sent across networks. Before a device sends a datagram to another device, it looks in its own ARP cache to see if there is a MAC address and corresponding IP address for the destination device. If there is no entry, the source device sends a broadcast message to every device on the network. Each device compares the IP address to its own. Only the device with the matching IP address replies to the sending device with a packet containing the MAC address for the device. The source device adds the destination device MAC address to its ARP table for future reference, creates a data-link header and trailer that encapsulates the packet, and proceeds to transfer the data. Figure 2 illustrates the ARP broadcast and response process.

Figure 2 ARP Process



When the destination device lies on a remote network, one beyond another router, the process is the same except that the sending device sends an ARP request for the MAC address of the default gateway. After the address is resolved and the default gateway receives the packet, the default gateway broadcasts the destination IP address over the networks connected to it. The router on the destination device network uses ARP to obtain the MAC address of the destination device and delivers the packet.

Encapsulation of IP datagrams and ARP requests and replies on IEEE 802 networks other than Ethernet use Subnetwork Access Protocol (SNAP).

The ARP request message has the following fields:

- HLN—Hardware address length. Specifies how long the hardware addresses are in the message. For IEEE 802 MAC addresses (Ethernet) the value is 6.
- PLN—Protocol address length. Specifies how long the protocol (Layer 3) addresses are in the message. For IPv4, the value is 4.
- OP—Opcode. Specifies the nature of the message by code:
  - 1—ARP request.
  - 2—ARP reply.
  - 3 through 9—RARP and Inverse ARP requests and replies.
- SHA—Sender hardware address. Specifies the Layer 2 hardware address of the device sending the message.
- SPA—Sender protocol address. Specifies the IP address of the sending device.
- THA—Target hardware address. Specifies the Layer 2 hardware address of the receiving device.
- TPA—Target protocol address. Specifies the IP address of the receiving device.

ARP Caching

Because the mapping of IP addresses to MAC addresses occurs at each hop (router) on the network for every datagram sent over an internetwork, performance of the network could be compromised. To minimize broadcasts and limit wasteful use of network resources, ARP caching was implemented.

ARP caching is the method of storing network addresses and the associated data-link addresses in memory for a period of time as the addresses are learned. This minimizes the use of valuable network resources to broadcast for the same address each time a datagram is sent. The cache entries must be maintained because the information could become outdated, so it is critical that the cache entries are set to expire periodically. Every device on a network updates its tables as addresses are broadcast.

There are static ARP cache entries and dynamic ARP cache entries. Static entries are manually configured and kept in the cache table on a permanent basis. They are best for devices that have to communicate with other devices usually in the same network on a regular basis. Dynamic entries are added by the Cisco IOS software and kept for a period of time, then removed.

Static and Dynamic Entries in the ARP Cache

Static routing requires an administrator to manually enter IP addresses, subnet masks, gateways, and corresponding MAC addresses for each interface of each router into a table. Static routing enables more control but requires more work to maintain the table. The table must be updated each time routes are added or changed.

Dynamic routing uses protocols that enable the routers in a network to exchange routing table information with each other. The table is built and changed automatically. No administrative tasks are needed unless a time limit is added, so dynamic routing is more efficient than static routing. The default time limit is 4 hours, if the network is has a great many routes that are added and deleted from the cache, the time limit should be adjusted.

The routing protocols that dynamic routing uses to learn routes, such as distance-vector and link-state, is beyond the scope of this document. For more information, refer to *Cisco IOS IP Routing Protocols Configuration Guide*, Release 12.4.

Devices That Do Not Use ARP

When a network is divided into two segments, a bridge joins the segments and filters traffic to each segment based on MAC addresses. The bridge builds its own address table, which uses MAC addresses only, as opposed to a router, which has a ARP cache that contains both IP addresses and the corresponding MAC addresses.

Passive hubs are central-connection devices that physically connect other devices in a network. They send messages out all of their ports to the devices and operate at Layer 1, but do not maintain an address table.

Layer 2 switches determine which port is connected to a device to which the message is addressed and send only to that port, unlike a hub, which sends the message out all its ports. However, Layer 3 switches are routers that build an ARP cache (table).

For more information about bridges, refer to the *Cisco IOS Bridging and IBM Networking Configuration Guide*, Release 12.4. For more information about switches, refer to *Cisco IOS Switching Services Configuration Guide*, Release 12.4.

Inverse ARP

Inverse ARP, which is enabled by default in ATM networks, builds an ATM map entry and is necessary to send unicast packets to a server (or relay agent) on the other end of a connection. Inverse ARP is only supported for the **aal5snap** encapsulation type.

For multipoint interfaces, an IP address can be acquired using other encapsulation types because broadcast packets are used. However, unicast packets to the other end will fail because there is no ATM map entry and thus DHCP renewals and releases also fail.

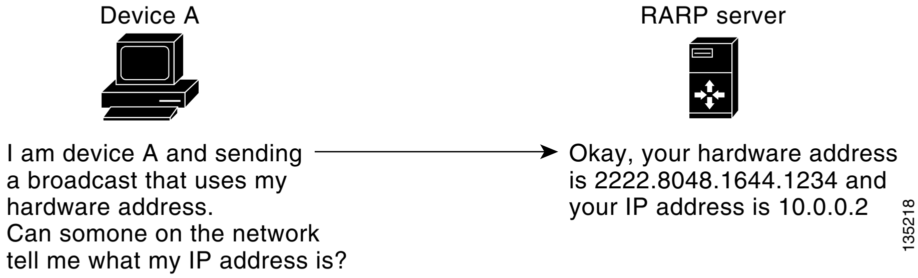
For more information about Inverse ARP and ATM networks, refer to the "Configuring ATM" chapter of the *Cisco IOS Wide-Area Networking Configuration Guide*, Release 12.4.

Reverse ARP

Reverse ARP (RARP) as defined by RFC 903 works the same way as ARP, except that the RARP request packet requests an IP address instead of a MAC address. RARP often is used by diskless workstations because this type of device has no way to store IP addresses to use when they boot. The only address that is known is the MAC address because it is burned into the hardware.

Use of RARP requires an RARP server on the same network segment as the router interface. Figure 3 illustrates how RARP works.

Figure 3 RARP Process



There are several limitations of RARP. Because of these limitations, most businesses use DHCP to assign IP addresses dynamically. DHCP is cost effective and requires less maintenance than RARP. The most important limitations are as follows:

- Since RARP uses hardware addresses, if the internetwork is large with many physical networks, a RARP server must be on every segment with an additional server for redundancy. Maintaining two servers for every segment is costly.
- Each server must be configured with a table of static mappings between the hardware addresses and IP addresses. Maintenance of the IP addresses is difficult.
- RARP only provides IP addresses of the hosts and not subnet masks or default gateways.

The Cisco IOS software attempts to use RARP if it does not know the IP address of an interface at startup to respond to RARP requests that they are able to answer. A feature of Cisco IOS software automates the configuration of Cisco devices and is called AutoInstall.

AutoInstall supports RARP and enables a network manager to connect a new router to a network, turn it on, and load a pre-existing configuration file automatically. The process begins when no valid configuration file is found in NVRAM. For more information about AutoInstall, refer to the *Cisco IOS Configuration Fundamentals Configuration Guide*, Release 12.4.

Proxy ARP

Proxy ARP, as defined in RFC 1027, was implemented to enable devices that are separated into physical network segments connected by a router in the same IP network or subnetwork to resolve the IP-to-MAC addresses. When devices are not in the same data link layer network but in the same IP network, they try to transmit data to each other as if they are on the local network. However, the router that separates the devices will not send a broadcast message because routers do not pass hardware-layer broadcasts. The addresses cannot be resolved.

Proxy ARP is enabled by default so the “proxy router” that resides between the local networks will respond with its MAC address as if it is the router to which the broadcast is addressed. When the sending device receives the MAC address of the proxy router, it sends the datagram to the proxy router that in turns sends the datagram to the designated device.

Proxy ARP is invoked by the following conditions:

- The target IP address is not on the same physical network (LAN) on which the request is received.
- The networking device has one or more routes to the target IP address.
- All of the routes to the target IP address go through interfaces other than the one on which the request is received.

When proxy ARP is disabled, a device will respond to ARP requests received on its interface only if the target IP address is the same as its IP address, or the target IP address in the ARP request has a statically configured ARP alias.

Serial Line Address Resolution Protocol

Serial Line ARP (SLARP) is used for serial interfaces that use High-Level Data Link Control (HDLC) encapsulation. A SLARP server, intermediate (staging) router, and another router providing a SLARP service may be required in addition to a TFTP server. If an interface is not directly connected to a server, the staging router is required to forward the address resolution requests to the server, otherwise a directly connected router with SLARP service is required. The Cisco IOS software attempts to use SLARP if it does not know the IP address of an interface at startup to respond to SLARP requests that software is able to answer.

A feature of Cisco IOS software automates the configuration of Cisco devices and is called AutoInstall. AutoInstall supports SLARP and enables a network manager to connect a new router to a network, turn it on, and load a pre-existing configuration file automatically. The process begins when no valid configuration file is found in NVRAM. For more information about AutoInstall, refer to the *Cisco IOS Configuration Fundamentals Configuration Guide*, Release 12.4.

Note Serial interfaces that use Frame Relay encapsulation are supported by AutoInstall.

Authorized ARP

Authorized ARP addresses a requirement of explicitly knowing when a user has logged off, either voluntarily or due to a failure of a network device. It is implemented for Public wireless LANs (WLANs) and DHCP. For more information about authorized ARP, refer to the "Configuring DHCP Services for Accounting and Security" chapter of the *DHCP Configuration Guide*, Cisco IOS Release 12.4.

How to Configure Address Resolution Protocol Options

ARP is enabled by default and is set to use Ethernet encapsulation by default. Perform the following tasks to change or verify ARP functionality:

- Enabling the Interface Encapsulation, page 8 (optional)
- Defining Static ARP Entries, page 9 (optional)
- Setting an Expiration Time for Dynamic Entries in the ARP Cache, page 12
- Globally Disabling Proxy ARP, page 13 (optional)
- Disabling Proxy ARP on an Interface, page 14 (optional)
- Verifying the ARP Configuration, page 15 (optional)

Enabling the Interface Encapsulation

Perform this task to support a type of encapsulation for a specific network, such as Ethernet, Frame Relay, FDDI, or Token Ring. When Frame Relay encapsulation is specified, the interface is configured for a Frame Relay subnetwork in which there is one physical link that has many logical circuits called virtual circuits (VCs). The address field in the frame contains a data-link connection identifier (DLCI) which identifies each VC. When SNAP encapsulation is specified, the interface is configured for FDDI or Token Ring networks.

Note The encapsulation type specified in this task should match the encapsulation type specified in the "Defining Static ARP Entries" section on page 9.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **arp** {**arpa** | **frame-relay** | **snap**}
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface ethernet0/0`	Enters interface configuration mode.
Step 4	**arp** {**arpa** | **frame-relay** | **snap**} **Example:** `Router(config-if)# arp arpa`	Specifies the encapsulation type for an interface by type of network, such as Ethernet, FDDI, Frame Relay, and Token Ring. The keywords are as follows: • **arpa**—Enables encapsulation for an Ethernet 802.3 network. • **frame-relay**—Enables encapsulation for a Frame Relay network. • **snap**—Enables encapsulation for FDDI and Token Ring networks.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.

Defining Static ARP Entries

Perform this task to define static mapping between IP addresses (32-bit address) and a MAC address (48-bit address) for hosts that do not support dynamic ARP. Because most hosts support dynamic address resolution, defining static ARP cache entries is usually not required. Performing this task installs a permanent entry in the ARP cache that never times out. The entries remain in the ARP table until they are removed using the **no arp** command or the **clear arp interface** command for each interface.

Note The encapsulation type specified in this task should match the encapsulation type specified in the "Enabling the Interface Encapsulation" section on page 8.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **arp** {*ip-address* | **vrf** *vrf-name*} *hardware-address encap-type* [*interface-type*]

4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**arp** {*ip-address* | **vrf** *vrf-name*} *hardware-address encap-type* [*interface-type*] **Example:** `Router(config)# arp 10.0.0.0 aabb.cc03.8200 arpa`	Globally associates an IP address with a MAC address in the ARP cache. The arguments and keyword are as follows: • *ip-address*—IP address in four-part dotted decimal format corresponding to the local data-link address. • **vrf** *vrf-name*—Virtual routing and forwarding instance for a Virtual Private Network (VPN). The *vrf-name* argument can be any name. • *hardware-address*—Local data-link address (a 48-bit address). • *encap-type*—Encapsulation type for the static entry. The keywords are as follows: – **arpa**—For Ethernet interfaces. – **sap**—For Hewlett Packard interfaces. – **smds**—For Switched Multimegabit Data Service (SMDS) interfaces. – **snap**—For FDDI and Token Ring interfaces. – **srp-a**—Switch route processor-side A (SRP-A) interfaces. – **srp-b**—Switch route processor-side B (SRP-B) interfaces. • *interface-type*—(Optional) Interface type. The keywords are as follows: – **ethernet**—IEEE 802.3 interface. – **loopback**—Loopback interface. – **null**—No interface. – **serial**—Serial interface – **alias**—Device responds to ARP requests as if it were the interface of the specified address.
Step 4	**exit** **Example:** `Router(config)# exit`	Exits to privileged EXEC mode.

Setting an Expiration Time for Dynamic Entries in the ARP Cache

Perform this task to set a time limit for dynamic entries in the ARP cache.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **arp timeout** *seconds*
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface ethernet0/0`	Enters interface configuration mode.
Step 4	**arp timeout** *seconds* **Example:** `Router(config-if)# arp timeout 30`	Sets the length of time, in seconds, an ARP cache entry will stay in the cache. A value of zero means that entries are never cleared from the cache. The default is 14400 seconds (4 hours). **Note** If the network has frequent changes to cache entries, the default should be changed to a shorter time period.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.

Globally Disabling Proxy ARP

Proxy ARP is enabled by default; perform this task to globally disable proxy ARP on all interfaces.

The Cisco IOS software uses proxy ARP (as defined in RFC 1027) to help hosts with no knowledge of routing determine the MAC addresses of hosts on other networks or subnets. For example, if hosts A and B are on different physical networks, host B will not receive the ARP broadcast request from host A and cannot respond to it. However, if the physical network of host A is connected by a gateway to the physical network of host B, the gateway will see the ARP request from host A.

Assuming that subnet numbers were assigned to correspond to physical networks, the gateway can also tell that the request is for a host that is on a different physical network. The gateway can then respond for host B, saying that the network address for host B is that of the gateway itself. Host A will see this reply, cache it, and send future IP packets for host B to the gateway.

The gateway will forward such packets to host B by using the configured IP routing protocols. The gateway is also referred to as a transparent subnet gateway or ARP subnet gateway.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip arp proxy disable**
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`enable`** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**`configure terminal`** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**`ip arp proxy disable`** **Example:** `Router(config)# ip arp proxy disable`	Disables proxy ARP on all interfaces. • The **ip arp proxy disable** command overrides any proxy ARP interface configuration. • To reenable proxy ARP, use the **no ip arp proxy disable** command. • You can also use the **default ip proxy arp** command to return to the default proxy ARP behavior, which is enabled.

Disabling Proxy ARP on an Interface

Proxy ARP is enabled by default; perform this task to disable proxy ARP on an interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no ip proxy-arp**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface ethernet0/0`	Enters interface configuration mode.
Step 4	**no ip proxy-arp** **Example:** `Router(config-if)# ip proxy-arp`	Disables proxy ARP on the interface. • To reenable proxy ARP, use the **ip proxy-arp** command. • You can also use the **default ip proxy-arp** command to return to the default proxy ARP behavior on the interface, which is enabled.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits to global configuration mode.

Clearing the ARP Cache

Perform the following tasks to clear the ARP cache of entries associated with an interface and to clear all dynamic entries from the ARP cache, the fast-switching cache, and the IP route cache.

SUMMARY STEPS

1. **enable**
2. **clear arp interface** *type number*
3. **clear arp-cache**
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`enable`** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**`clear arp interface`** *`type number`* **Example:** `Router# clear arp interface ethernet0/0`	Clears the entire ARP cache on the interface. The *type* and *number* arguments are the type of interface and the assigned number for the interface.
Step 3	**`clear arp-cache`** **Example:** `Router# clear arp-cache`	Clears all dynamic entries from the ARP cache, the fast-switching cache, and the IP route cache.
Step 4	**`exit`** **Example:** `Router# exit`	Exits to EXEC mode.

Verifying the ARP Configuration

To verify the ARP configuration, perform the following steps.

SUMMARY STEPS

1. **show interfaces**
2. **show arp**
3. **show ip arp**
4. **show processes cpu | include (ARP|PID)**

DETAILED STEPS

Step 1 **show interfaces**

To display the type of ARP being used on a particular interface and also display the ARP timeout value, use the **show interfaces** EXEC command.

```
Router# show interfaces

Ethernet 0 is up, line protocol is up
 Hardware is MCI Ethernet, address is 0000.0c00.750c (bia 0000.0c00.750c)
 Internet address is 10.108.28.8, subnet mask is 255.255.255.0
 MTU 1500 bytes, BW 10000 Kbit, DLY 100000 usec, rely 255/255, load 1/255
 Encapsulation ARPA, loopback not set, keepalive set (10 sec)
 ARP type: ARPA, ARP Timeout 4:00:00
 Last input 0:00:00, output 0:00:00, output hang never
 Last clearing of "show interface" counters 0:00:00
 Output queue 0/40, 0 drops; input queue 0/75, 0 drops
 Five minute input rate 0 bits/sec, 0 packets/sec
 Five minute output rate 2000 bits/sec, 4 packets/sec
 1127576 packets input, 447251251 bytes, 0 no buffer
 Received 354125 broadcasts, 0 runts, 0 giants, 57186* throttles
 0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  5332142 packets output, 496316039 bytes, 0 underruns
  0 output errors, 432 collisions, 0 interface resets, 0 restarts
```

Step 2 **show arp**

Use the **show arp** EXEC command to examine the contents of the ARP cache.

```
Router# show arp

Protocol    Address           Age (min)   Hardware Addr      Type    Interface
Internet    10.108.42.112     120         0000.a710.4baf     ARPA    Ethernet3
AppleTalk   4028.5            29          0000.0c01.0e56     SNAP    Ethernet2
Internet    110.108.42.114    105         0000.a710.859b     ARPA    Ethernet3
AppleTalk   4028.9            -           0000.0c02.a03c     SNAP    Ethernet2
Internet    10.108.42.121     42          0000.a710.68cd     ARPA    Ethernet3
Internet    10.108.36.9       -           0000.3080.6fd4     SNAP    TokenRing0
AppleTalk   4036.9            -           0000.3080.6fd4     SNAP    TokenRing0
Internet    10.108.33.9       -           0000.0c01.7bbd     SNAP    Fddi0
```

Step 3 **show ip arp**

Use the **show ip arp** EXEC command to show IP entries. To remove all nonstatic entries from the ARP cache, use the **clear arp-cache** privileged EXEC command.

```
Router# show ip arp

Protocol    Address           Age(min)      Hardware Addr      Type     Interface
Internet    171.69.233.22     9             0000.0c59.f892     ARPA     Ethernet0/0
Internet    171.69.233.21     8             0000.0c07.ac00     ARPA     Ethernet0/0
Internet    171.69.233.19     -             0000.0c63.1300     ARPA     Ethernet0/0
Internet    171.69.233.30     9             0000.0c36.6965     ARPA     Ethernet0/0
Internet    172.19.168.11     -             0000.0c63.1300     ARPA     Ethernet0/0
Internet    172.19.168.254    9             0000.0c36.6965     ARPA     Ethernet0/0
```

Step 4 **show processes cpu | include (ARP|PID)**

Use the **show processes cpu | include (ARP|PID)** command to display ARP and RARP processes.

```
Router# show processes cpu | include (ARP|PID)

PID      Runtime(ms) Invoked uSecs    5Sec  1Min  5Min    TTY Process
1          1736         58   29931     0%    0%    0%    Check heaps
```

```
2           68         585      116  1.00% 1.00%     0%    IP Input
3            0         744        0     0%    0%     0%    TCP Timer
4            0           2        0     0%    0%     0%    TCP Protocols
5            0           1        0     0%    0%     0%    BOOTP Server
6           16         130      123     0%    0%     0%    ARP Input
7            0           1        0     0%    0%     0%    Probe Input
8            0           7        0     0%    0%     0%    MOP Protocols
9            0           2        0     0%    0%     0%    Timers
10         692          64    10812     0%    0%     0%   Net Background
11           0           5        0     0%    0%     0%   Logger
12           0          38        0     0%    0%     0%   BGP Open
13           0           1        0     0%    0%     0%   Net Input
14         540        3466      155     0%    0%     0%   TTY Background
15           0           1        0     0%    0%     0%   BGP I/O
16        5100        1367     3730     0%    0%     0%   IGRP Router
17          88        4232       20  0.20% 1.00%     0%   BGP Router
18         152       14650       10     0%    0%     0%   BGP Scanner
19         224          99     2262     0%    0%  1.00%   Exec
```

Configuration Examples for Address Resolution Protocol Options

This section provides the following configuration examples:

Static ARP Entry Configuration: Example

The following example shows how to configure a static ARP entry in the cache and by using the **alias** keyword, Cisco IOS software can respond to ARP requests as if it were the interface of the specified address:

```
arp 10.0.0.0 aabb.cc03.8200 alias
interface ethernet0/0
```

Encapsulation Type Configuration: Example

The following example shows how to configure the encapsulation on the interface. The **snap** keyword indicates that interface Ethernet0/0 is connected to an FDDI or Token Ring network:

```
interface ethernet0/0
 ip address 10.108.10.1 255.255.255.0
 arp snap
```

Proxy ARP Configuration: Example

The following example shows how to configure proxy ARP because it was disabled for interface Ethernet0/0:

```
interface ethernet0/0
 ip proxy-arp
```

Clearing the ARP Cache: Example

The following example shows how to clear all of the entries in the ARP cache associated with an interface:

```
Router# clear arp interface ethernet0/0
```

The following example shows how to clear all of the dynamic entries in the ARP cache:

```
Router# clear arp-cache
```

Additional References

The following sections provide references related to configuring Address Resolution Protocol Options.

Related Documents

Related Topic	Document Title
ARP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Addressing Services Command Reference*
Monitoring and maintaining ARP tasks	"Monitoring and Maintaining ARP Information" module

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 826	*Address Resolution Protocol*
RFC 903	*Reverse Address Resolution Protocol*
RFC 1027	Proxy *Address Resolution Protocol*
RFC 1042	Standard for the Transmission of IP Datagrams over IEEE 802 Networks

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport

Feature Information for Configuring Address Resolution Protocol Options

Table 1 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Releases 12.2(1) or 12.0(3)S or later appear in the table.

Not all commands may be available in your Cisco IOS software release. For details on when support for specific commands was introduced, see the command reference documents.

Cisco IOS software images are specific to a Cisco IOS software release, a feature set, and a platform. Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Table 1 Feature Information for Configuring Address Resolution Protocol Options

Feature Name	Software Releases	Feature Configuration Information
ARP Optimization	12.2(15)T Cisco IOS XE Release 2.1	In previous versions of Cisco IOS software, the ARP table was organized for easy searching on an entry based on the IP address. However, there are cases such as interface flapping on the router and a topology change in the network where all related ARP entries need to be refreshed for correct forwarding. This situation could consume a substantial amount of CPU time in the ARP process to search and clean up all the entries. The ARP Optimization feature improves ARP performance by reducing the ARP searching time by using an improved data structure. The following sections provides information about this feature: • Clearing the ARP Cache • Clearing the ARP Cache: Example The following command was introduced by this feature: **clear arp interface**

DHCP

DHCP Features Roadmap

First Published: May 2, 2005
Last Updated: December 31, 2007

This roadmap lists the features documented in the Dynamic Host Configuration Protocol (DHCP) modules and maps the features to the modules in which they appear.

Feature and Release Support

Table 1 lists the DHCP feature support for the following Cisco IOS software release trains:

- Cisco IOS Releases 12.2T, 12.3, 12.3T, 12.4, and 12.4T
- Cisco IOS Release 12.2SB
- Cisco IOS Release 12.2SR

Only features that were introduced or modified in Cisco IOS Release 12.2(1)T, Cisco IOS Release 12.2(28)SB, Cisco IOS Releases 12.2(33)SRA, or a later release appear in the table. Not all features may be supported in your Cisco IOS software release.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Table 1 Supported DHCP Features

Release	Feature Name	Feature Description	Where Documented
Cisco IOS Releases 12.2T, 12.3, 12.3T, 12.4, and 12.4T			
12.4(15)T	DHCP Server Multiple Subnet	This feature enables multiple disjoint subnets to be configured under the same DHCP address pool. This functionality enables the DHCP server to manage additional IP addresses by adding the addresses to the existing DHCP address pool (instead of using a separate address pool). Multiple subnets in a DHCP address pool can occur along with or instead of managing individual client addresses.	Configuring the Cisco IOS DHCP Server
12.4(11)T	DHCP Class Support for Client Identification	The DHCP Class Support for Client Identification feature enhances the DHCP class mechanism to support options 60, 77, 124, and 125. These options identify the type of client sending the DHCP message. The DHCP relay agent can make forwarding decisions based on the content of the options in the DHCP message sent by the client.	Configuring the Cisco IOS DHCP Relay Agent
	DHCPv4 Relay per Interface VPN ID Support	The DHCPv4 Relay per Interface VPN ID Support feature allows the Cisco IOS DHCP Relay Agent to be configured per interface to override the global configuration of the **ip dhcp relay information option vpn** command. This feature allows subscribers with different relay information option VPN ID requirements on different interfaces to be reached from one Cisco router.	Configuring the Cisco IOS DHCP Relay Agent
12.4(6)T	DHCP Relay Option 82 per Interface Support	This feature enables support for the DHCP relay agent information option (option 82) on a per interface basis. The interface configuration allows different DHCP servers, with different DHCP option 82 requirements, to be reached from one Cisco router.	Configuring the Cisco IOS DHCP Relay Agent
	DHCP Relay Accounting	The DHCP Relay Accounting feature allows a Cisco IOS DHCP relay agent to send a RADIUS accounting start packet when an address is assigned to a client and a RADIUS accounting stop packet when the address is released.	Configuring DHCP Enhancements for Edge-Session Management
12.3(14)T	ARP Auto-logoff	The ARP Auto-logoff feature enhances DHCP authorized ARP by providing finer control and probing of authorized clients to detect a log off.	Configuring DHCP Services for Accounting and Security
	DHCP Enhancements for Edge-Session Management	The DHCP Enhancements for Edge-Session Management feature provides the capability of simultaneous service by multiple Internet Service Providers (ISPs) to customers using one network infrastructure. The end-user customer may change ISPs at any time.	Configuring DHCP Enhancements for Edge-Session Management
	DHCP Subscriber Identifier Suboption of Option 82	This feature enables an ISP to add a unique identifier to the subscriber-identifier suboption of the relay agent information option.	Configuring the Cisco IOS DHCP Relay Agent

Table 1 Supported DHCP Features (continued)

Release	Feature Name	Feature Description	Where Documented
12.3(11)T	DHCP Static Mapping	Configuring static mapping pools enables the DHCP server to read the static bindings from a separate text file (similar in format to the DHCP database file) that is stored in these special pools.	Configuring the Cisco IOS DHCP Server
12.3(8)T	Configurable DHCP Client	This feature provides the flexibility to include various configuration options for the DHCP client. A DHCP client is defined as an Internet host using DHCP to obtain configuration parameters such as an IP address.	Configuring the Cisco IOS DHCP Client
	DHCP Statically Configured Routes Using a DHCP Gateway	This feature enables the configuration of static routes that point to an assigned DHCP next hop router.	Configuring the Cisco IOS DHCP Server
12.3(4)T	DHCP Address Allocation Using Option 82	The Cisco IOS DHCP server can allocate dynamic IP addresses based on the relay information option (option 82) information sent by the relay agent.	Configuring the Cisco IOS DHCP Server
	DHCP Release and Renew CLI in EXEC Mode	This feature provides the ability to perform two independent operations from the CLI: (1) immediately release a DHCP lease for a DHCP client, and (2) force a DHCP renewal of a lease for a DHCP client.	Configuring the Cisco IOS DHCP Client
12.3(2)T	DHCP Authorized ARP	DHCP authorized ARP enhances the DHCP and ARP components of the Cisco IOS software to limit the leasing of IP addresses to mobile users to mobile users that are authorized. This feature enhances security in PWLANs by blocking ARP responses from unauthorized users at the DHCP server.	Configuring DHCP Services for Accounting and Security
	DHCP Lease Limit per ATM RBE Unnumbered Interface	This feature limits the number of DHCP leases per subinterface offered to DHCP clients connected from an ATM RBE unnumbered interface or serial unnumbered interface of the DHCP server or DHCP relay agent.	Configuring DHCP Services for Accounting and Security
12.2(15)T	DHCP Accounting	DHCP accounting introduces AAA and RADIUS support for DHCP configuration.	Configuring DHCP Services for Accounting and Security
	DHCP ODAP Server Support	This feature introduces the capability to configure a DHCP server (or router) as a subnet allocation server. This capability allows the Cisco IOS DHCP server to be configured with a pool of subnets for lease to ODAP clients.	Configuring the DHCP Server On-Demand Address Pool Manager
	DHCP Secured IP Address Assignment	DHCP secure IP address assignment provides the capability to secure ARP table entries to DHCP leases in the DHCP database.	Configuring DHCP Services for Accounting and Security
	DHCP Server On-Demand Address Pool Manager for Non-MPLS VPNs	This feature was enhanced to provide ODAP support for non-MPLS VPNs.	Configuring the DHCP Server On-Demand Address Pool Manager

Table 1 Supported DHCP Features (continued)

Release	Feature Name	Feature Description	Where Documented
12.2(8)T	DHCP Client on WAN Interfaces	This feature extends the DHCP to allow a DHCP client to acquire an IP address over PPP over ATM (PPPoA) and certain ATM interfaces.	Configuring the Cisco IOS DHCP Client
	DHCP Relay MPLS VPN Support	DHCP relay support for MPLS VPNs enables a network administrator to conserve address space by allowing overlapping addresses. The relay agent can support multiple clients on different VPNs, and many of these clients from different VPNs can share the same IP address.	Configuring the Cisco IOS DHCP Relay Agent
	DHCP Server On-Demand Address Pool Manager	The ODAP manager is used to centralize the management of large pools of addresses and simplify the configuration of large networks. ODAP provides a central management point for the allocation and assignment of IP addresses.	Configuring the DHCP Server On-Demand Address Pool Manager
	DHCP Server Option to Ignore all BOOTP Requests	This feature allows the Cisco IOS DHCP server to selectively ignore and not reply to received Bootstrap Protocol (BOOTP) request packets.	Configuring the Cisco IOS DHCP Server
Cisco IOS Release 12.2SB			
12.2(31)SB2	ISSU and SSO - DHCP High Availability Features	Cisco IOS Release 12.2(31)SB2 introduces the following series of DHCP High Availability features: • ISSU—DHCP Server • SSO—DHCP Server • ISSU—DHCP Relay on Unnumbered Interface • SSO—DHCP Relay on Unnumbered Interface • ISSU—DHCP Proxy Client • SSO—DHCP Proxy Client • ISSU—DHCP ODAP Client and Server • SSO—DHCP ODAP Client and Server These features are enabled by default when the redundancy mode of operation is set to Stateful Switchover (SSO).	ISSU and SSO - DHCP High Availability Features
	DHCP Relay Option 82 per Interface Support	This feature enables support for the DHCP relay agent information option (option 82) on a per interface basis. The interface configuration allows different DHCP servers, with different DHCP option 82 requirements, to be reached from one Cisco router.	Configuring the Cisco IOS DHCP Relay Agent

Table 1 Supported DHCP Features (continued)

Release	Feature Name	Feature Description	Where Documented
12.2(28)SB	Configurable DHCP Client	This feature provides the flexibility to include various configuration options for the DHCP client. A DHCP client is defined as an Internet host using DHCP to obtain configuration parameters such as an IP address.	Configuring the Cisco IOS DHCP Client
	DHCP Accounting	DHCP accounting introduces AAA and RADIUS support for DHCP configuration.	Configuring DHCP Services for Accounting and Security
	DHCP Address Allocation Using Option 82	The Cisco IOS DHCP server can allocate dynamic IP addresses based on the relay information option (option 82) information sent by the relay agent.	Configuring the Cisco IOS DHCP Server
	DHCP Client on WAN Interfaces	This feature extends the DHCP to allow a DHCP client to acquire an IP address over PPP over ATM (PPPoA) and certain ATM interfaces.	Configuring the Cisco IOS DHCP Client
	DHCP Lease Limit per ATM RBE Unnumbered Interface	This feature limits the number of DHCP leases per subinterface offered to DHCP clients connected from an ATM RBE unnumbered interface or serial unnumbered interface of the DHCP server or DHCP relay agent.	Configuring DHCP Services for Accounting and Security
	DHCP ODAP Server Support	This feature introduces the capability to configure a DHCP server (or router) as a subnet allocation server. This capability allows the Cisco IOS DHCP server to be configured with a pool of subnets for lease to ODAP clients.	Configuring the DHCP Server On-Demand Address Pool Manager
	DHCP Relay MPLS VPN Support	DHCP relay support for MPLS VPNs enables a network administrator to conserve address space by allowing overlapping addresses. The relay agent can support multiple clients on different VPNs, and many of these clients from different VPNs can share the same IP address.	Configuring the Cisco IOS DHCP Relay Agent
	DHCP Release and Renew CLI in EXEC Mode	This feature provides the ability to perform two independent operations from the CLI: (1) immediately release a DHCP lease for a DHCP client, and (2) force a DHCP renewal of a lease for a DHCP client.	Configuring the Cisco IOS DHCP Client
	DHCP Secured IP Address Assignment	DHCP secure IP address assignment provides the capability to secure ARP table entries to DHCP leases in the DHCP database.	Configuring DHCP Services for Accounting and Security
	DHCP Server On-Demand Address Pool Manager	The ODAP manager is used to centralize the management of large pools of addresses and simplify the configuration of large networks. ODAP provides a central management point for the allocation and assignment of IP addresses.	Configuring the DHCP Server On-Demand Address Pool Manager
	DHCP Server On-Demand Address Pool Manager for Non-MPLS VPNs	This feature was enhanced to provide ODAP support for non-MPLS VPNs.	Configuring the DHCP Server On-Demand Address Pool Manager

Table 1 ***Supported DHCP Features (continued)***

Release	Feature Name	Feature Description	Where Documented
	DHCP Server Option to Ignore all BOOTP Requests	This feature allows the Cisco IOS DHCP server to selectively ignore and not reply to received Bootstrap Protocol (BOOTP) request packets.	Configuring the Cisco IOS DHCP Server
	DHCP Statically Configured Routes Using a DHCP Gateway	This feature enables the configuration of static routes that point to an assigned DHCP next hop router.	Configuring the Cisco IOS DHCP Server
	DHCP Static Mapping	Configuring static mapping pools enables the DHCP server to read the static bindings from a separate text file (similar in format to the DHCP database file) that is stored in these special pools.	Configuring the Cisco IOS DHCP Server
	DHCP Subscriber Identifier Suboption of Option 82	This feature enables an ISP to add a unique identifier to the subscriber-identifier suboption of the relay agent information option.	Configuring the Cisco IOS DHCP Relay Agent
Cisco IOS Release 12.2SR			
12.2(33)SRC (cont)	DHCP Authorized ARP	DHCP authorized ARP enhances the DHCP and ARP components of the Cisco IOS software to limit the leasing of IP addresses to mobile users to mobile users that are authorized. This feature enhances security in PWLANs by blocking ARP responses from unauthorized users at the DHCP server.	Configuring DHCP Services for Accounting and Security
	DHCP Enhancements for Edge-Session Management	The DHCP Enhancements for Edge-Session Management feature provides the capability of simultaneous service by multiple Internet Service Providers (ISPs) to customers using one network infrastructure. The end-user customer may change ISPs at any time.	Configuring DHCP Enhancements for Edge-Session Management
	DHCP Relay MPLS VPN Support	DHCP relay support for MPLS VPNs enables a network administrator to conserve address space by allowing overlapping addresses. The relay agent can support multiple clients on different VPNs, and many of these clients from different VPNs can share the same IP address.	Configuring the Cisco IOS DHCP Relay Agent
	DHCP Relay Option 82 per Interface Support	This feature enables support for the DHCP relay agent information option (option 82) on a per interface basis. The interface configuration allows different DHCP servers, with different DHCP option 82 requirements, to be reached from one Cisco router.	Configuring the Cisco IOS DHCP Relay Agent
	DHCP Release and Renew CLI in EXEC Mode	This feature provides the ability to perform two independent operations from the CLI: (1) immediately release a DHCP lease for a DHCP client, and (2) force a DHCP renewal of a lease for a DHCP client.	Configuring the Cisco IOS DHCP Client
	DHCP Secured IP Address Assignment	DHCP secure IP address assignment provides the capability to secure ARP table entries to DHCP leases in the DHCP database.	Configuring DHCP Services for Accounting and Security

Table 1 Supported DHCP Features (continued)

Release	Feature Name	Feature Description	Where Documented
12.2(33)SRC (cont)	DHCP Server Import All Enhancement	The feature is an enhancement to the **import all** global configuration command. Before this feature was introduced, the options imported through the import all command were overwritten by those imported by another subsystem. Through this feature, options imported by multiple subsystems can co-exist in the DHCP address pool. When the session is terminated or the lease is released, the imported options are cleared.	Configuring the Cisco IOS DHCP Server
	DHCP Server On-Demand Address Pool Manager	The ODAP manager is used to centralize the management of large pools of addresses and simplify the configuration of large networks. ODAP provides a central management point for the allocation and assignment of IP addresses.	Configuring the DHCP Server On-Demand Address Pool Manager
	DHCP Server On-Demand Address Pool Manager for Non-MPLS VPNs	This feature was enhanced to provide ODAP support for non-MPLS VPNs.	Configuring the DHCP Server On-Demand Address Pool Manager
	DHCP ODAP Server Support	This feature introduces the capability to configure a DHCP server (or router) as a subnet allocation server. This capability allows the Cisco IOS DHCP server to be configured with a pool of subnets for lease to ODAP clients.	Configuring the DHCP Server On-Demand Address Pool Manager
	DHCP Per Interface Lease Limit and Statistics	This feature limits the number of DHCP leases offered to DHCP clients on an interface. DHCP server statistics reporting was enhanced to display interface-level statistics.	Configuring DHCP Services for Accounting and Security
	DHCP Server MIB	The DHCP Server MIB feature provides SNMP access to and control of Cisco IOS DHCP server software on a Cisco router by an external network management device.	DHCP Server MIB
	DHCP Statically Configured Routes Using a DHCP Gateway	This feature enables the configuration of static routes that point to an assigned DHCP next hop router.	Configuring the Cisco IOS DHCP Server
	DHCP Static Mapping	Configuring static mapping pools enables the DHCP server to read the static bindings from a separate text file (similar in format to the DHCP database file) that is stored in these special pools.	Configuring the Cisco IOS DHCP Server

Table 1 Supported DHCP Features (continued)

Release	Feature Name	Feature Description	Where Documented
12.2(33)SRC cont	ISSU and SSO - DHCP High Availability Features	Cisco IOS Release 12.2(33)SRC introduces the following series of DHCP High Availability features: • ISSU—DHCP ODAP Client/Server • SSO—DHCP ODAP Client/Server • ISSU—DHCP Relay on Unnumbered Interface • ISSU—DHCP Proxy Client • SSO—DHCP Proxy Client • ISSU—DHCP Server These features are enabled by default when the redundancy mode of operation is set to Stateful Switchover (SSO).	ISSU and SSO - DHCP High Availability Features
12.(33)SRB	DHCP Accounting	DHCP accounting introduces AAA and RADIUS support for DHCP configuration.	Configuring DHCP Services for Accounting and Security
	DHCP Address Allocation Using Option 82	The Cisco IOS DHCP server can allocate dynamic IP addresses based on the relay information option (option 82) information sent by the relay agent.	Configuring the Cisco IOS DHCP Server
	DHCP Server Multiple Subnet	This feature enables multiple disjoint subnets to be configured under the same DHCP address pool. This functionality enables the DHCP server to manage additional IP addresses by adding the addresses to the existing DHCP address pool (instead of using a separate address pool). Multiple subnets in a DHCP address pool can occur along with or instead of managing individual client addresses.	Configuring the Cisco IOS DHCP Server
	DHCP Subscriber Identifier Suboption of Option 82	This feature enables an ISP to add a unique identifier to the subscriber-identifier suboption of the relay agent information option.	Configuring the Cisco IOS DHCP Relay Agent
	SSO—DHCP Relay on Unnumbered Interface	The DHCP relay on unnumbered interface that is SSO aware adds high availability support for host routes to clients connected through unnumbered interfaces. The DHCP relay agent can now detect when a router is failing over to the standby route processor and keep the states related to unnumbered interfaces.	ISSU and SSO - DHCP High Availability Features
	SSO—DHCP Server	The DHCP server that is SSO aware is able to detect when a router is failing over to the standby route processor route processor and preserve the DHCP lease across a switchover event.	ISSU and SSO - DHCP High Availability Features

DHCP Overview

The *Dynamic Host Configuration Protocol* (DHCP) is based on the Bootstrap Protocol (BOOTP), which provides the framework for passing configuration information to hosts on a TCP/IP network. DHCP adds the capability to automatically allocate reusable network addresses and configuration options to Internet hosts. DHCP consists of two components: a protocol for delivering host-specific configuration parameters from a DHCP server to a host and a mechanism for allocating network addresses to hosts. DHCP is built on a client/server model, where designated DHCP server hosts allocate network addresses and deliver configuration parameters to dynamically configured hosts.

This module describes the concepts needed to understand Cisco IOS DHCP.

Module History

This module was first published on May 2, 2005, and last updated on February 27, 2006.

Contents

Information About DHCP

To configure DHCP, you should understand the following concepts:

- DHCP Overview, page 2
- Benefits of Using Cisco IOS DHCP, page 2
- DHCP Server, Relay Agent, and Client Operation, page 3
- DHCP Database, page 4
- DHCP Attribute Inheritance, page 4
- DHCP Options and Suboptions, page 4
- DHCP Server On-Demand Address Pool Management Overview, page 5

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

- DHCP Services for Accounting and Security Overview, page 6

DHCP Overview

Cisco routers running Cisco IOS software include DHCP server and relay agent software. The Cisco IOS DHCP server is a full DHCP server implementation that assigns and manages IP addresses from specified address pools within the router to DHCP clients. These address pools can also be configured to supply additional information to the requesting client such as the IP address of the DNS server, the default router, and other configuration parameters. If the Cisco IOS DHCP server cannot satisfy a DHCP request from its own database, it can forward the request to one or more secondary DHCP servers defined by the network administrator.

DHCP supports three mechanisms for IP address allocation:

- Automatic allocation—DHCP assigns a permanent IP address to a client.
- Dynamic allocation—DHCP assigns an IP address to a client for a limited period of time, which is called a lease (or until the client explicitly relinquishes the address).
  DHCP also supports on-demand address pools (ODAPs), which is a feature in which pools of IP addresses can be dynamically increased or reduced in size depending on the address utilization level. ODAPs support address assignment for customers using private addresses.
- Manual allocation—The network administrator assigns an IP address to a client and DHCP is used simply to convey the assigned address to the client.

The format of DHCP messages is based on the format of BOOTP messages, which ensures support for BOOTP relay agent functionality and interoperability between BOOTP clients and DHCP servers. BOOTP relay agents eliminate the need for deploying a DHCP server on each physical network segment. BOOTP is explained in RFC 951, *Bootstrap Protocol (BOOTP)*, and RFC 1542, *Clarifications and Extensions for the Bootstrap Protocol*.

The main advantage of DHCP compared to BOOTP is that DHCP does not require that the DHCP server be configured with all MAC addresses of all clients. DHCP defines a process by which the DHCP server knows the IP subnet in which the DHCP client resides, and it can assign an IP address from a pool of valid IP addresses in that subnet. Most of the other information that DHCP might supply, such as the default router IP address, is the same for all hosts in the subnet so DHCP servers can usually configure information per subnet rather than per host. This functionality reduces network administration tasks compared to BOOTP.

Benefits of Using Cisco IOS DHCP

The Cisco IOS DHCP implementation offers the following benefits:

- Reduced Internet access costs

  Using automatic IP address assignment at each remote site substantially reduces Internet access costs. Static IP addresses are considerably more expensive to purchase than are automatically allocated IP addresses.

- Reduced client configuration tasks and costs

  Because DHCP is easy to configure, it minimizes operational overhead and costs associated with device configuration tasks and eases deployment by nontechnical users.

- Centralized management

Because the DHCP server maintains configurations for several subnets, an administrator only needs to update a single, central server when configuration parameters change.

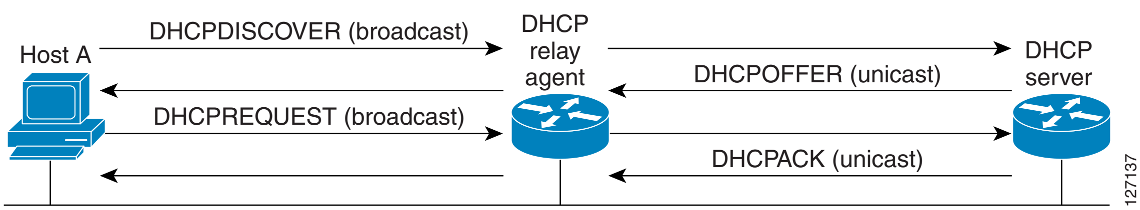
DHCP Server, Relay Agent, and Client Operation

DHCP provides a framework for passing configuration information dynamically to hosts on a TCP/IP network. A DHCP client is an Internet host using DHCP to obtain configuration parameters such as an IP address.

A DHCP relay agent is any host that forwards DHCP packets between clients and servers. Relay agents are used to forward requests and replies between clients and servers when they are not on the same physical subnet. Relay agent forwarding is distinct from the normal forwarding of an IP router, where IP datagrams are switched between networks somewhat transparently. By contrast, relay agents receive DHCP messages and then generate a new DHCP message to send on another interface.

Figure 1 shows the basic steps that occur when a DHCP client requests an IP address from a DHCP server. The client, Host A, sends a DHCPDISCOVER broadcast message to locate a DHCP server. A relay agent forwards the packets between the DHCP client and server. A DHCP server offers configuration parameters (such as an IP address, a MAC address, a domain name, and a lease for the IP address) to the client in a DHCPOFFER unicast message.

Figure 1 DHCP Request for an IP Address from a DHCP Server



A DHCP client may receive offers from multiple DHCP servers and can accept any one of the offers; however, the client usually accepts the first offer it receives. Additionally, the offer from the DHCP server is not a guarantee that the IP address will be allocated to the client; however, the server usually reserves the address until the client has had a chance to formally request the address.

The client returns a formal request for the offered IP address to the DHCP server in a DHCPREQUEST broadcast message. The DHCP server confirms that the IP address has been allocated to the client by returning a DHCPACK unicast message to the client.

The formal request for the offered IP address (the DHCPREQUEST message) that is sent by the client is broadcast so that all other DHCP servers that received the DHCPDISCOVER broadcast message from the client can reclaim the IP addresses that they offered to the client.

If the configuration parameters sent to the client in the DHCPOFFER unicast message by the DHCP server are invalid (a misconfiguration error exists), the client returns a DHCPDECLINE broadcast message to the DHCP server.

The DHCP server will send to the client a DHCPNAK denial broadcast message, which means the offered configuration parameters have not been assigned, if an error has occurred during the negotiation of the parameters or the client has been slow in responding to the DHCPOFFER message (the DHCP server assigned the parameters to another client) of the DHCP server.

DHCP Database

DHCP address pools are stored in non-volatile RAM (NVRAM). There is no limit on the number of address pools. An address binding is the mapping between the client's IP and hardware addresses. The client's IP address can be configured by the administrator (manual address allocation) or assigned from a pool by the DHCP server.

Manual bindings are stored in NVRAM. Manual bindings are just special address pools configured by a network administrator. There is no limit on the number of manual bindings.

Automatic bindings are IP addresses that have been automatically mapped to the MAC addresses of hosts that are found in the DHCP database. Automatic bindings are stored on a remote host called the database agent. A DHCP database agent is any host—for example, an FTP, TFTP, or RCP server—that stores the DHCP bindings database.The bindings are saved as text records for easy maintenance.

You can configure multiple DHCP database agents and you can configure the interval between database updates and transfers for each agent.

DHCP Attribute Inheritance

The DHCP server database is organized as a tree. The root of the tree is the address pool for natural networks, branches are subnetwork address pools, and leaves are manual bindings to clients. Subnetworks inherit network parameters and clients inherit subnetwork parameters. Therefore, common parameters, for example the domain name, should be configured at the highest (network or subnetwork) level of the tree.

Inherited parameters can be overridden. For example, if a parameter is defined in both the natural network and a subnetwork, the definition of the subnetwork is used.

Address leases are not inherited. If a lease is not specified for an IP address, by default, the DHCP server assigns a one-day lease for the address.

DHCP Options and Suboptions

Configuration parameters and other control information are carried in tagged data items that are stored in the options field of the DHCP message. Options provide a method of appending additional information. Vendors that want to provide additional information to their client not designed into the protocol can use options.

The Cisco IOS DHCP implementation also allows most DHCP server options to be customized. For example, the TFTP server, which stores the Cisco IOS image, can be customized with option 150 to support intelligent IP phones.

Virtual Private Networks (VPNs) allow the possibility that two pools in separate networks can have the same address space, with private network addresses, served by the same DHCP server. Cisco IOS software supports VPN-related options and suboptions such as the relay agent information option and VPN identification suboption. A relay agent can recognize these VPN-related options and suboptions and forward the client-originated DHCP packets to a DHCP server. The DHCP server can use this information to assign IP addresses and other parameters, distinguished by a VPN identifier, to help select the VPN to which the client belongs.

For more information on DHCP options and suboptions, see the "DHCP Options" appendix in the *Network Registrar User's Guide*, Release 6.2.

During lease negotiation, the DHCP server sends the options shown in Table 1 to the client.

Table 1 Default DHCP Server Options

DHCP Option Name	DHCP Option Code	Description
Subnet mask option	1	Specifies the client's subnet mask per RFC 950.
Router option	3	Specifies a list of IP addresses for routers on the client's subnet, usually listed in order of preference.
Domain name server option	6	Specifies a list of DNS name servers available to the client, usually listed in order of preference.
Hostname option	12	Specifies the name of the client. The name may or may not be qualified with the local domain name.
Domain name option	15	Specifies the domain name that the client should use when resolving hostnames via the Domain Name System.
NetBIOS over TCP/IP name server option	44	Specifies a list of RFC 1001/1002 NetBIOS name servers listed in order or preference.
NetBIOS over TCP/IP node type option	46	Enables NetBIOS over TCP/IP clients that are configurable to be configured as described in RFC 1001/1002.
IP address lease time option	51	Allows the client to request a lease for the IP address.
DHCP message type option	53	Conveys the type of the DHCP message.
Server identifier option	54	Identifies the IP address of the selected DHCP server.
Renewal (T1) time option	58	Specifies the time interval from address assignment until the client transitions to the renewing state.
Rebinding (T2) time option	59	Specifies the time interval from address assignment until the client transitions to the rebinding state.

DHCP Server On-Demand Address Pool Management Overview

The Cisco IOS DHCP server on-demand address pool (ODAP) manager is used to centralize the management of large pools of addresses and simplify the configuration of large networks. ODAP provides a central management point for the allocation and assignment of IP addresses. When a Cisco IOS router is configured as an ODAP manager, pools of IP addresses are dynamically increased or reduced in size depending on the address utilization level.

ODAPs support address assignment using DHCP for customers using private addresses. Each ODAP is configured and associated with a particular Multiprotocol Label Switching (MPLS) VPN. Cisco IOS software also provides ODAP support for non-MPLS VPN address pools by adding pool name support to the **peer default ip address dhcp-pool** *pool name* command.

DHCP server subnet allocation is a way of offering entire subnets (ranges of addresses) to relay agents so that remote access devices can provision IP addresses to DHCP clients. This functionality can occur along with or instead of managing individual client addresses. Subnet allocation can improve IP address provisioning, aggregation, characterization, and distribution by relying on the DHCP infrastructure to dynamically manage subnets.

This capability allows the DHCP server to be configured with a pool of subnets for lease to ODAP clients. Subnet pools can be configured for global ODAP clients or MPLS VPN ODAP clients on a per-client basis. The DHCP subnet allocation server creates bindings for the subnet leases and stores these leases in the DHCP database.

DHCP Services for Accounting and Security Overview

Cisco IOS software supports several new capabilities that enhance DHCP accounting, reliability, and security in Public Wireless LANs (PWLANs). This functionality can also be used in other network implementations.

DHCP accounting provides authentication, authorization, and accounting (AAA) and Remote Authentication Dial-In User Service (RADIUS) support for DHCP. The AAA and RADIUS support improves security by sending secure START and STOP accounting messages. The configuration of DHCP accounting adds a layer of security that allows DHCP lease assignment and termination to be triggered for the appropriate RADIUS START and STOP accounting records so that the session state is properly maintained by upstream devices, such as a Service Selection Gateway (SSG). This additional security can help to prevent unauthorized clients or hackers from gaining illegal entry to the network by spoofing authorized DHCP leases.

Three other features have been designed and implemented to address the security concerns in PWLANs. The first feature secures ARP table entries to DHCP leases in the DHCP database. The secure ARP functionality prevents IP spoofing by synchronizing the database of the DHCP server with the ARP table to avoid address hijacking. Secure ARP adds an entry to the ARP table for a client when an address is allocated that can be deleted by the DHCP server only when a binding expires.

The second feature is DHCP authorized ARP. This functionality provides a complete solution by addressing the need for DHCP to explicitly know when a user logs out. Before the introduction of DHCP authorized ARP, there was no mechanism to inform the DHCP server if a user had left the system ungracefully, which could result in excessive billing for a customer that had logged out but the system had not detected the log out. To prevent this problem, DHCP authorized ARP sends periodic ARP messages on a per-minute basis to determine if a user is still logged in. Only authorized users can respond to the ARP request. ARP responses from unauthorized users are blocked at the DHCP server providing an extra level of security.

In addition, DHCP authorized ARP disables dynamic ARP learning on an interface. The address mapping can be installed only by the authorized component specified by the **arp authorized** interface configuration command. DHCP is the only authorized component currently allowed to install ARP entries.

The third feature is ARP autologoff, which adds finer control for probing when authorized users log out. The **arp probe interval** command specifies when to start a probe (the timeout), how frequent a peer is probed (the interval), and the maximum number of retries (the count).

Additional References

The following sections provide references related to DHCP.

Related Documents

Related Topic	Document Title
DHCP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Addressing Services Command Reference*
DHCP server configuration	"Configuring the Cisco IOS DHCP Server" module
DHCP relay agent configuration	*"Configuring the Cisco IOS DHCP Relay Agent"* module

Related Topic	Document Title
DHCP client configuration	"Configuring the Cisco IOS DHCP Client" module
DHCP server on-demand address pools	"Configuring the DHCP Server On-Demand Address Pool Manager" module
DHCP advanced features	"Configuring DHCP Services for Accounting and Security" module
DHCP enhancements for edge-session management	"Configuring DHCP Enhancements for Edge-Session Management" module
DHCP options	"DHCP Options" appendix in the *Network Registrar User's Guide*, Release 6.1.1

Standards

Standards	Title
No new or modified standards are supported by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 951	*Bootstrap Protocol (BOOTP)*
RFC 2131	*Dynamic Host Configuration Protocol*
RFC 2132	*DHCP Options and BOOTP Vendor Extensions*
RFC 1542	*Clarifications and Extensions for the Bootstrap Protocol*

Technical Assistance

Description	Link
The Cisco Technical Support & Documentation website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport

Glossary

address binding—A mapping between the client's IP and hardware (MAC) addresses. The client's IP address may be configured by the administrator (manual address allocation) or assigned from a pool by the DHCP server (automatic address allocation). The binding also contains a lease expiration date. The default for the lease expiration date is one day.

address conflict—A duplication of use of the same IP address by two hosts. During address assignment, DHCP checks for conflicts using ping and gratuitous (ARP). If a conflict is detected, the address is removed from the pool. The address will not be assigned until the administrator resolves the conflict.

address pool—The range of IP addresses assigned by the DHCP server. Address pools are indexed by subnet number.

automatic address allocation—An address assignment method where a network administrator obtains an IP address for a client for a finite period of time or until the client explicitly relinquishes the address. Automatic allocation is particularly useful for assigning an address to a client that will be connected to the network only temporarily or for sharing a limited pool of IP addresses among a group of clients that do not need permanent IP addresses. Automatic allocation may also be a good choice for assigning an IP address to a new client being permanently connected to a network where IP addresses are sufficiently scarce that it is important to reclaim them when old clients are retired.

BOOTP—Bootstrap Protocol. A protocol that provides a method for a booting computer to find out its IP address and the location of the boot file with the rest of its parameters.

client—Any host requesting configuration parameters.

database—A collection of address pools and bindings.

database agent—Any host storing the DHCP bindings database, for example, a Trivial File Transfer Protocol (TFTP) server.

DHCP—Dynamic Host Configuration Protocol. A protocol that provides a mechanism for allocating IP addresses dynamically so that addresses can be reused when hosts no longer need them.

DNS—Domain Name System. A system used in the Internet for translating names of network nodes into addresses.

manual address allocation—An address assignment method that allocates an administratively assigned IP address to a host. Manual allocation allows DHCP to be used to eliminate the error-prone process of manually configuring hosts with IP addresses.

PWLAN—Public Wireless Local Area Network. A type of wireless LAN, often referred to as a hotspot, that anyone having a properly configured computer device can access.

relay agent—A router that forwards DHCP and BOOTP messages between a server and a client on different subnets.

server—Any host providing configuration parameters.

SSG—Service Selection Gateway. The Cisco IOS feature set that provides on-demand service enforcement within the Cisco network.

Note Refer to *Internetworking Terms and Acronyms* for terms not included in this glossary.

Configuring the Cisco IOS DHCP Server

First Published: May 2, 2005
Last Updated: May 2, 2008

Cisco routers running Cisco IOS software include Dynamic Host Configuration Protocol (DHCP) server and relay agent software. The Cisco IOS DHCP server is a full DHCP server implementation that assigns and manages IP addresses from specified address pools within the router to DHCP clients. The DHCP server can be configured to assign additional parameters such as the IP address of the domain name system (DNS) server and the default router.

This module describes the concepts and the tasks needed to configure the Cisco IOS DHCP server.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for the Cisco IOS DHCP Server" section on page 44.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Configuring the DHCP Server

Before you configure the Cisco IOS DHCP server, you should understand the concepts documented in the "DHCP Overview" module.

The Cisco IOS DHCP server and relay agent are enabled by default. You can verify if they have been disabled by checking your configuration file. If they have been disabled, the **no service dhcp** command will appear in the configuration file. Use the **service dhcp** command to reenable the functionality if necessary.

The Cisco IOS DHCP relay agent will be enabled on an interface only when the **ip helper-address** is configured. This command enables the DHCP broadcast to be forwarded to the configured DHCP server.

Information About the Cisco IOS DHCP Server

Before you configure the DHCP server, you should understand the following concepts:

- Overview of the DHCP Server, page 2
- DHCP Attribute Inheritance, page 2
- DHCP Server Address Allocation Using Option 82, page 2

Overview of the DHCP Server

The Cisco IOS DHCP server accepts address assignment requests and renewals and assigns the addresses from predefined groups of addresses contained within DHCP address pools. These address pools can also be configured to supply additional information to the requesting client such as the IP address of the DNS server, the default router, and other configuration parameters. The Cisco IOS DHCP server can accept broadcasts from locally attached LAN segments or from DHCP requests that have been forwarded by other DHCP relay agents within the network.

DHCP Attribute Inheritance

The DHCP server database is organized as a tree. The root of the tree is the address pool for natural networks, branches are subnetwork address pools, and leaves are manual bindings to clients. Subnetworks inherit network parameters and clients inherit subnetwork parameters. Therefore, common parameters, for example the domain name, should be configured at the highest (network or subnetwork) level of the tree.

Inherited parameters can be overridden. For example, if a parameter is defined in both the natural network and a subnetwork, the definition of the subnetwork is used.

Address leases are not inherited. If a lease is not specified for an IP address, by default, the DHCP server assigns a one-day lease for the address.

DHCP Server Address Allocation Using Option 82

The Cisco IOS DHCP server can allocate dynamic IP addresses based on the relay information option (option 82) information sent by the relay agent.

Automatic DHCP address allocation is typically based on an IP address, whether it be the gateway address (*giaddr* field of the DHCP packet) or the incoming interface IP address. In some networks, it is necessary to use additional information to further determine which IP addresses to allocate. By using option 82, the Cisco IOS relay agent has long been able to include additional information about itself when forwarding client-originated DHCP packets to a DHCP server. The Cisco IOS DHCP server can also use option 82 as a means to provide additional information to properly allocate IP addresses to DHCP clients.

How to Configure the Cisco IOS DHCP Server

This section contains the following tasks:

Configuring a DHCP Database Agent or Disabling Conflict Logging

Perform this task to configure a DHCP database agent.

Database Agents

A DHCP database agent is any host (for example, an FTP, TFTP, or rcp server) or storage media on the DHCP server (for example, disk0) that stores the DHCP bindings database. You can configure multiple DHCP database agents, and you can configure the interval between database updates and transfers for each agent.

Automatic bindings are IP addresses that have been automatically mapped to the MAC addresses of hosts that are found in the DHCP database. Automatic binding information (such as lease expiration date and time, interface index, and VPN routing and forwarding [VRF] name) is stored on a database agent. The bindings are saved as text records for easy maintenance.

Address Conflicts

An address conflict occurs when two hosts use the same IP address. During address assignment, DHCP checks for conflicts using ping and gratuitous Address Resolution Protocol (ARP). If a conflict is detected, the address is removed from the pool. The address will not be assigned until the administrator resolves the conflict.

Restrictions

We strongly recommend using database agents. However, the Cisco IOS server can run without them. If you choose not to configure a DHCP database agent, disable the recording of DHCP address conflicts on the DHCP server by using the **no ip dhcp conflict logging** command in global configuration mode. If there is conflict logging but no database agent configured, bindings are lost across router reboots. Possible false conflicts can occur causing the address to be removed from the address pool until the network administrator intervenes.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp database** *url* [**timeout** *seconds* | **write-delay** *seconds*]
   or
   no ip dhcp conflict logging

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`enable`** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**`configure terminal`** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**`ip dhcp database`** *`url`* `[`**`timeout`** *`seconds`* `|` **`write-delay`** *`seconds`*`]` or **`no ip dhcp conflict logging`** **Example:** `Router(config)# ip dhcp database ftp://user:password@172.16.1.1/router-dhcp timeout 80` or **Example:** `Router(config)# no ip dhcp conflict logging`	Configures a DHCP server to save automatic bindings on a remote host called a database agent. or Disables DHCP address conflict logging. • Choose this option only if you do not configure a DHCP database agent. See the "Restrictions" section for guidelines.

Excluding IP Addresses

Perform this task to specify IP addresses (excluded addresses) that the DHCP server should not assign to clients.

The IP address configured on the router interface is automatically excluded from the DHCP address pool. The DHCP server assumes that all other IP addresses in a DHCP address pool subnet are available for assigning to DHCP clients.

You need to exclude addresses from the pool if the DHCP server should not allocate those IP addresses. An example usage scenario is when two DHCP servers are set up to service the same network segment (subnet) for redundancy. If the two DHCP servers do not coordinate their services with each other using a protocol such as DHCP failover, then each DHCP server must be configured to allocate from a non-overlapping set of addresses in the shared subnet. See the "Configuring Manual Bindings: Example" for a configuration example.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp excluded-address** *low-address* [*high-address*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`enable`** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**`configure terminal`** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**`ip dhcp excluded-address`** *`low-address`* [*`high-address`*] **Example:** `Router(config)# ip dhcp excluded-address 172.16.1.100 172.16.1.103`	Specifies the IP addresses that the DHCP server should not assign to DHCP clients.

Configuring DHCP Address Pools

This section contains the following tasks:

- Configuring a DHCP Address Pool, page 6 (required)
- Configuring a DHCP Address Pool with Secondary Subnets, page 10 (optional)
- Verifying the DHCP Address Pool Configuration, page 15 (optional)

Configuring a DHCP Address Pool

Perform this task to configure a DHCP address pool. On a per-address pool basis, specify DHCP options for the client as necessary.

DHCP Address Pool Conventions

You can configure a DHCP address pool with a name that is a symbolic string (such as "engineering") or an integer (such as 0). Configuring a DHCP address pool also puts the router into DHCP pool configuration mode—identified by the (dhcp-config)# prompt—from which you can configure pool parameters (for example, the IP subnet number and default router list).

DHCP Address Pool Selection

DHCP defines a process by which the DHCP server knows the IP subnet in which the DHCP client resides, and it can assign an IP address from a pool of valid IP addresses in that subnet. The process by which the DHCP server identifies which DHCP address pool to use to service a client request is described in the "DHCP Address Pool Selection" section on page 6.

The DHCP server identifies which DHCP address pool to use to service a client request as follows:

- If the client is not directly connected (the giaddr field of the DHCPDISCOVER broadcast message is non-zero), the DHCP server matches the DHCPDISCOVER with a DHCP pool that has the subnet that contains the IP address in the giaddr field.

- If the client is directly connected (the giaddr field is zero), the DHCP server matches the DHCPDISCOVER with DHCP pool(s) that contain the subnet(s) configured on the receiving interface. If the interface has secondary IP addresses, the subnets associated with the secondary IP addresses are examined for possible allocation only after the subnet associated with the primary IP address (on the interface) is exhausted.

Cisco IOS DHCP server software supports advanced capabilities for IP address allocation. See the "Configuring DHCP Address Allocation Using Option 82" section for more information.

Prerequisites

Before you configure the DHCP address pool, you need to:

- Identify DHCP options for devices where necessary, including the following:
  - Default boot image name
  - Default routers
  - Domain Name System (DNS) servers
  - NetBIOS name server
  - Primary subnet
  - Secondary subnets and subnet-specific default router lists (See "Configuring a DHCP Address Pool with Secondary Subnets" for information on secondary subnets).
- Decide on a NetBIOS node type (b, p, m, or h).
- Decide on a DNS domain name.

Restrictions

You cannot configure manual bindings within the same pool that is configured with the **network** DHCP pool configuration command. To configure manual bindings, see the "Configuring Manual Bindings" section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp pool** *name*
4. **utilization mark high** *percentage-number* [**log**]
5. **utilization mark low** *percentage-number* [**log**]
6. **network** *network-number* [*mask* | /*prefix-length*]
7. **domain-name** *domain*
8. **dns-server** *address* [*address2 ... address8*]
9. **bootfile** *filename*
10. **next-server** *address* [*address2 ... address8*]
11. **netbios-name-server** *address* [*address2 ... address8*]
12. **netbios-node-type** *type*
13. **default-router** *address* [*address2 ... address8*]

14. **option** *code* [**instance** *number*] {**ascii** *string* | **hex** *string* | *ip-address*}
15. **lease** {*days* [*hours*] [*minutes*] | **infinite**}
16. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`configure terminal` **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**`ip dhcp pool`** *`name`* **Example:** `Router(config)# ip dhcp pool 1`	Creates a name for the DHCP server address pool and enters DHCP pool configuration mode.
Step 4	**`utilization mark high`** *`percentage-number`* [**`log`**] **Example:** `Router(dhcp-config)# utilization mark high 80 log`	(Optional) Configures the high utilization mark of the current address pool size. • The **log** keyword enables the logging of a system message. A system message will be generated for a DHCP pool when the pool utilization exceeds the configured high utilization threshold.
Step 5	**`utilization mark low`** *`percentage-number`* [**`log`**] **Example:** `Router(dhcp-config)# utilization mark low 70 log`	(Optional) Configures the low utilization mark of the current address pool size. • The **log** keyword enables the logging of a system message. A system message will be generated for a DHCP pool when the pool utilization falls below the configured low utilization threshold.
Step 6	**`network`** *`network-number`* [*`mask`* | **`/`***`prefix-length`*] **Example:** `Router(dhcp-config)# network 172.16.0.0 /16`	Specifies the subnet network number and mask of the DHCP address pool.
Step 7	**`domain-name`** *`domain`* **Example:** `Router(dhcp-config)# domain-name cisco.com`	Specifies the domain name for the client.
Step 8	**`dns-server`** *`address`* [*`address2 ... address8`*] **Example:** `Router(dhcp-config)# dns server 172.16.1.103 172.16.2.103`	Specifies the IP address of a DNS server that is available to a DHCP client. • One IP address is required; however, you can specify up to eight IP addresses in one command line. • Servers should be listed in order of preference.

	Command or Action	Purpose
Step 9	**bootfile** *filename* **Example:** `Router(dhcp-config)# bootfile xllboot`	(Optional) Specifies the name of the default boot image for a DHCP client. • The boot file is used to store the boot image for the client. The boot image is generally the operating system the client uses to load.
Step 10	**next-server** *address* [*address2 ... address8*] **Example:** `Router(dhcp-config)# next-server 172.17.1.103 172.17.2.103`	(Optional) Configures the next server in the boot process of a DHCP client. • If multiple servers are specified, DHCP assigns them to clients in round-robin order. The first client gets address 1, the next client gets address 2, and so on. • If this command is not configured, DHCP uses the server specified by the **ip helper address** command as the boot server.
Step 11	**netbios-name-server** *address* [*address2 ... address8*] **Example:** `Router(dhcp-config)# netbios-name-server 172.16.1.103 172.16.2.103`	(Optional) Specifies the NetBIOS Windows Internet Naming Service (WINS) server that is available to a Microsoft DHCP client. • One address is required; however, you can specify up to eight addresses in one command line. • Servers should be listed in order of preference.
Step 12	**netbios-node-type** *type* **Example:** `Router(dhcp-config)# netbios-node-type h-node`	(Optional) Specifies the NetBIOS node type for a Microsoft DHCP client.
Step 13	**default-router** *address* [*address2 ... address8*] **Example:** `Router(dhcp-config)# default-router 172.16.1.100 172.16.1.101`	(Optional) Specifies the IP address of the default router for a DHCP client. • The IP address should be on the same subnet as the client. • One IP address is required; however, you can specify a up to eight IP addresses in one command line. These default routers are listed in order of preference; that is, *address* is the most preferred router, address2 is the next most preferred router, and so on. • When a DHCP client requests an IP address, the router—acting as a DHCP server—accesses the default router list to select another router that the DHCP client is to use as the first hop for forwarding messages. After a DHCP client has booted, the client begins sending packets to its default router.
Step 14	**option** *code* [**instance** *number*] {**ascii** *string* | **hex** *string* | *ip-address*} **Example:** `Router(dhcp-config)# option 19 hex 01`	(Optional) Configures DHCP server options.

	Command or Action	Purpose
Step 15	**lease** {*days* [*hours*] [*minutes*]| **infinite**} **Example:** `Router(dhcp-config)# lease 30`	(Optional) Specifies the duration of the lease. • The default is a one-day lease. • The **infinite** keyword specifies that the duration of the lease is unlimited.
Step 16	**end** **Example:** `Router(config-dhcp-subnet-secondary)# end`	Returns to global configuration mode.

Configuring a DHCP Address Pool with Secondary Subnets

Perform this task to configure a DHCP address pool with secondary subnets.

DHCP Server Address Pool with Multiple Disjoint Subnets

For any DHCP pool, you can configure a *primary subnet* and any number of *secondary subnets*. Each subnet is a range of IP addresses that the router uses to allocate an IP address to a DHCP client. The DHCP server multiple subnet functionality enables a Cisco IOS DHCP server address pool to manage additional IP addresses by adding the addresses to a secondary subnet of an existing DHCP address pool (instead of using a separate address pool).

Secondary Subnet Conventions

Configuring a secondary DHCP subnetwork places the router in DHCP pool secondary subnet configuration mode—identified by the (config-dhcp-subnet-secondary)# prompt—from which you can configure a default address list that is specific to the secondary subnet. You can also specify the utilization rate of the secondary subnet, which allows pools of IP addresses to dynamically increase or reduce in size depending on the address utilization level. This setting overrides the global utilization rate.

IP Address Allocation from a DHCP Server Address Pool with Secondary Subnets

If the DHCP server selects an address pool that contains multiple subnets, the DHCP server allocates an IP address from the subnets as follows:

- When the DHCP server receives an address assignment request, it looks for a free address in the primary subnet.
- When the primary subnet is exhausted, the DHCP server automatically looks for a free address in any secondary subnets maintained by the DHCP server (even though the giaddr does not necessarily match the secondary subnet). The server inspects the subnets for address availability in the order in which the subnets were added to the pool.
- If the giaddr matches a secondary subnet in the pool, the DHCP server allocates an IP address from that secondary subnet (even if IP addresses are available in the primary subnet and irrespective of the order in which secondary subnets where added).

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **ip dhcp pool** *name*
4. **utilization mark high** *percentage-number* [**log**]
5. **utilization mark low** *percentage-number* [**log**]
6. **network** *network-number* [*mask* | /*prefix-length*]
7. **domain-name** *domain*
8. **dns-server** *address* [*address*2 ... *address8*]
9. **bootfile** *filename*
10. **next-server** *address* [*address*2 ... *address8*]
11. **netbios-name-server** *address* [*address*2 ... *address8*]
12. **netbios-node-type** *type*
13. **default-router** *address* [*address*2 ... *address8*]
14. **option** *code* [**instance** *number*] {**ascii** *string* | **hex** *string* | *ip-address*}
15. **lease** {*days* [*hours*] [*minutes*] | **infinite**}
16. **network** *network-number* [{*mask* | /*prefix-length*} [**secondary**]]
17. **override default-router** *address* [*address2* ... *address8*]
18. **override utilization high** *percentage-number*
19. **override utilization low** *percentage-number*
20. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`enable`** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**`configure terminal`** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**`ip dhcp pool`** *`name`* **Example:** `Router(config)# ip dhcp pool 1`	Creates a name for the DHCP server address pool and enters DHCP pool configuration mode.
Step 4	**`utilization mark high`** *`percentage-number`* [**`log`**] **Example:** `Router(dhcp-config)# utilization mark high 80 log`	(Optional) Configures the high utilization mark of the current address pool size. • The **log** keyword enables the logging of a system message. A system message will be generated for a DHCP pool when the pool utilization exceeds the configured high utilization threshold.

	Command or Action	Purpose
Step 5	**utilization mark low** *percentage-number* [**log**] **Example:** `Router(dhcp-config)# utilization mark low 70 log`	(Optional) Configures the low utilization mark of the current address pool size. • The **log** keyword enables the logging of a system message. A system message will be generated for a DHCP pool when the pool utilization falls below the configured low utilization threshold.
Step 6	**network** *network-number* [*mask* | **/***prefix-length*] **Example:** `Router(dhcp-config)# network 172.16.0.0 /16`	Specifies the subnet network number and mask of the DHCP address pool.
Step 7	**domain-name** *domain* **Example:** `Router(dhcp-config)# domain-name cisco.com`	Specifies the domain name for the client.
Step 8	**dns-server** *address* [*address2 ... address8*] **Example:** `Router(dhcp-config)# dns server 172.16.1.103 172.16.2.103`	Specifies the IP address of a DNS server that is available to a DHCP client. • One IP address is required; however, you can specify up to eight IP addresses in one command line. • Servers should be listed in order of preference.
Step 9	**bootfile** *filename* **Example:** `Router(dhcp-config)# bootfile xllboot`	(Optional) Specifies the name of the default boot image for a DHCP client. • The boot file is used to store the boot image for the client. The boot image is generally the operating system the client uses to load.
Step 10	**next-server** *address* [*address2 ... address8*] **Example:** `Router(dhcp-config)# next-server 172.17.1.103 172.17.2.103`	(Optional) Configures the next server in the boot process of a DHCP client. • If multiple servers are specified, DHCP assigns them to clients in round-robin order. The first client gets address 1, the next client gets address 2, and so on. • If this command is not configured, DHCP uses the server specified by the **ip helper address** command as the boot server.
Step 11	**netbios-name-server** *address* [*address2 ... address8*] **Example:** `Router(dhcp-config)# netbios-name-server 172.16.1.103 172.16.2.103`	(Optional) Specifies the NetBIOS Windows Internet Naming Service (WINS) server that is available to a Microsoft DHCP client. • One address is required; however, you can specify up to eight addresses in one command line. • Servers should be listed in order of preference.
Step 12	**netbios-node-type** *type* **Example:** `Router(dhcp-config)# netbios-node-type h-node`	(Optional) Specifies the NetBIOS node type for a Microsoft DHCP client.

	Command or Action	Purpose
Step 13	**default-router** *address* [*address2* ... *address8*] **Example:** `Router(dhcp-config)# default-router 172.16.1.100 172.16.1.101`	(Optional) Specifies the IP address of the default router for a DHCP client. • The IP address should be on the same subnet as the client. • One IP address is required; however, you can specify a up to eight IP addresses in one command line. These default routers are listed in order of preference; that is, *address* is the most preferred router, *address2* is the next most preferred router, and so on. • When a DHCP client requests an IP address, the router—acting as a DHCP server—accesses the default router list to select another router that the DHCP client is to use as the first hop for forwarding messages. After a DHCP client has booted, the client begins sending packets to its default router.
Step 14	**option** *code* [**instance** *number*] {**ascii** *string* | **hex** *string* | *ip-address*} **Example:** `Router(dhcp-config)# option 19 hex 01`	(Optional) Configures DHCP server options.
Step 15	**lease** {*days* [*hours*] [*minutes*]| **infinite**} **Example:** `Router(dhcp-config)# lease 30`	(Optional) Specifies the duration of the lease. • The default is a one-day lease. • The **infinite** keyword specifies that the duration of the lease is unlimited.
Step 16	**network** *network-number* [{*mask* | **/***prefix-length*} [**secondary**]] **Example:** `Router(dhcp-config)# network 10.10.0.0 255.255.0.0 secondary`	(Optional) Specifies the network number and mask of a secondary DHCP server address pool. Any number of secondary subnets can be added to the DHCP server address pool. • During execution of this command, the configuration mode changes to DHCP pool secondary subnet configuration mode, which is identified by the (config-dhcp-subnet-secondary)# prompt. In this mode, the administrator can configure a default router list that is specific to the subnet. • See "Troubleshooting Tips" if you are using secondary IP addresses under a loopback interface with DHCP secondary subnets.

	Command or Action	Purpose
Step 17	**override default-router** *address* [*address2* ... *address8*] **Example:** `Router(config-dhcp-subnet-secondary)# override default-router 10.10.0.100 10.10.0.101`	(Optional) Specifies the default router list that is used when an IP address is assigned to a DHCP client from this secondary subnet. • If this subnet-specific override value is configured, it is used when assigning an IP address from the subnet; the network-wide default router list is used only to set the gateway router for the primary subnet. • If this subnet-specific override value is not configured, the network-wide default router list is used when assigning an IP address from the subnet. • See "Configuring a DHCP Address Pool with Multiple Disjoint Subnets: Example" for an example configuration.
Step 18	**override utilization mark high** *percentage-number* **Example:** `Router(config-dhcp-subnet-secondary)# override utilization mark high 60`	(Optional) Sets the high utilization mark of the subnet size. • This command overrides the global default setting specified by the **utilization mark high** global configuration command.
Step 19	**override utilization mark low** *percentage-number* **Example:** `Router(config-dhcp-subnet-secondary)# override utilization mark low 40`	(Optional) Sets the low utilization mark of the subnet size. • This command overrides the global default setting specified by the **utilization mark low** global configuration command.
Step 20	**end** **Example:** `Router(config-dhcp-subnet-secondary)# end`	Returns to privileged EXEC mode.

Troubleshooting Tips

If you are using secondary IP addresses under a single loopback interface and using secondary subnets under a DHCP pool, use one DHCP pool to configure networks for all the secondary subnets instead of one pool per secondary subnet. The **network** *network-number* [{*mask* | /*prefix-length*} [**secondary**]] commands must be configured under a single DHCP address pool rather than multiple DHCP address pools.

The following is the correct configuration:

```
!
ip dhcp pool dhcp_1
 network 172.16.1.0 255.255.255.0
 network 172.16.2.0 255.255.255.0 secondary
 network 172.16.3.0 255.255.255.0 secondary
 network 172.16.4.0 255.255.255.0 secondary
!
interface Loopback111
 ip address 172.16.1.1 255.255.255.255 secondary
 ip address 172.16.2.1 255.255.255.255 secondary
 ip address 172.16.3.1 255.255.255.255 secondary
 ip address 172.16.4.1 255.255.255.255 secondary
```

The following is the incorrect configuration:

```
!
ip dhcp pool dhcp_1
 network 172.16.1.0 255.255.255.0
 lease 1 20 30
 accounting default
!
ip dhcp pool dhcp_2
 network 172.16.2.0 255.255.255.0
 lease 1 20 30
 accounting default
!
ip dhcp pool dhcp_3
 network 172.16.3.0 255.255.255.0
 lease 1 20 30
 accounting default
!
ip dhcp pool dhcp_4
 network 172.16.4.0 255.255.255.0
 lease 1 20 30
 accounting default
!
interface Loopback111
 ip address 172.16.1.1 255.255.255.255 secondary
 ip address 172.16.2.1 255.255.255.255 secondary
 ip address 172.16.3.1 255.255.255.255 secondary
 ip address 172.16.4.1 255.255.255.255 secondary
```

Verifying the DHCP Address Pool Configuration

Perform this task to verify the DHCP address pool configuration.

SUMMARY STEPS

1. **enable**
2. **show ip dhcp pool** [*name*]
3. **show ip dhcp binding** [*address*]
4. **show ip dhcp conflict** [*address*]
5. **show ip dhcp database** [*url*]
6. **show ip dhcp server statistics** [*type number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip dhcp pool** [*name*] **Example:** `Router# show ip dhcp pool`	(Optional) Displays information about DHCP address pools.
Step 3	**show ip dhcp binding** [*address*] **Example:** `Router# show ip dhcp binding`	(Optional) Displays a list of all bindings created on a specific DHCP server. • Use the **show ip dhcp binding** command to display the IP addresses that have already been assigned. Verify that the address pool has not been exhausted. If necessary, re-create the pool to create a larger pool of addresses. • Use the **show ip dhcp binding** command to display the lease expiration date and time of the IP address of the host.
Step 4	**show ip dhcp conflict** [*address*] **Example:** `Router# show ip dhcp conflict`	(Optional) Displays a list of all address conflicts.
Step 5	**show ip dhcp database** [*url*] **Example:** `Router# show ip dhcp database`	(Optional) Displays recent activity on the DHCP database.
Step 6	**show ip dhcp server statistics** [*type-number*] **Example:** `Router# show ip dhcp server statistics`	(Optional) Displays count information about server statistics and messages sent and received.

Configuring Manual Bindings

Perform this task to configure manual bindings.

Address Bindings

An address binding is a mapping between the IP address and MAC address of a client. The IP address of a client can be assigned manually by an administrator or assigned automatically from a pool by a DHCP server.

Manual bindings are IP addresses that have been manually mapped to the MAC addresses of hosts that are found in the DHCP database. Manual bindings are stored in NVRAM on the DHCP server. Manual bindings are just special address pools. There is no limit on the number of manual bindings, but you can only configure one manual binding per host pool.

Automatic bindings are IP addresses that have been automatically mapped to the MAC addresses of hosts that are found in the DHCP database. Because the bindings are stored in volatile memory on the DHCP server, binding information is lost in the event of a power failure or upon router reload for any other reason. To prevent the loss of automatic binding information in such an event, a copy of the automatic binding information can be stored on a remote host called a DHCP database agent. The bindings are periodically written to the database agent. If the router reloads, the bindings are read back from the database agent to the DHCP database on the DHCP server.

Note We strongly recommend using database agents. However, the Cisco IOS DHCP server can function without database agents.

All DHCP clients send a client identifier (DHCP option 61) in the DHCP packet. To configure manual bindings, you must enter the **client-identifier** DHCP pool configuration command with the appropriate hexadecimal values identifying the DHCP client.

Restrictions

You cannot configure manual bindings within the same pool that is configured with the **network** command in DHCP pool configuration mode. See the "Configuring DHCP Address Pools" section for information about DHCP address pools and the **network** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp pool** *pool-name*
4. **host** *address* [*mask* | /*prefix-length*]
5. **client-identifier** *unique-identifier*
6. **hardware-address** *hardware-address type*
7. **client-name** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip dhcp pool** *pool-name* **Example:** `Router(config)# ip dhcp pool pool1`	Creates a name for the DHCP server address pool and places you in DHCP pool configuration mode—identified by the (dhcp-config)# prompt.
Step 4	**host** *address* [*mask* | **/***prefix-length*] **Example:** `Router(dhcp-config)# host`	Specifies the IP address and subnet mask of the client. • There is no limit on the number of manual bindings but you can only configure one manual binding per host pool.
Step 5	**client-identifier** *unique-identifier* **Example:** `Router(dhcp-config)# client-identifier 01b7.0813.8811.66`	Specifies the unique identifier for DHCP clients. This command is used for DHCP requests. • DHCP clients require client identifiers. The unique identification of the client is specified in dotted hexadecimal notation, for example, 01b7.0813.8811.66, where 01 represents the Ethernet media type. • See "Troubleshooting Tips" below for information on how to determine the client identifier of the DHCP client.
Step 6	**hardware-address** *hardware-address type* **Example:** `Router(dhcp-config)# hardware-address b708.1388.f166 ieee802`	(Optional) Specifies a hardware address for the client. This command is used for BOOTP requests.
Step 7	**client-name** *name* **Example:** `Router(dhcp-config)# client-name client1`	(Optional) Specifies the name of the client using any standard ASCII character. • The client name should not include the domain name. For example, the name **mars** should not be specified as **mars.cisco.com**.

Troubleshooting Tips

You can determine the client identifier by using the **debug ip dhcp server packet** command. In the following example, the client is identified by the value 0b07.1134.a029.

```
Router# debug ip dhcp server packet

DHCPD:DHCPDISCOVER received from client 0b07.1134.a029 through relay 10.1.0.253.
```

```
DHCPD:assigned IP address 10.1.0.3 to client 0b07.1134.a029.
.
.
.
```

Configuring DHCP Static Mapping

The DHCP—Static Mapping feature enables assignment of static IP addresses without creating numerous host pools with manual bindings by using a customer-created text file that the DHCP server reads. The benefit of this feature is that it eliminates the need for a long configuration file and reduces the space required in NVRAM to maintain address pools.

DHCP Database

A DHCP database contains the mappings between a client IP address and hardware address, referred to as a binding. There are two types of bindings: manual bindings that map a single hardware address to a single IP address, and automatic bindings that dynamically map a hardware address to an IP address from a pool of IP addresses. Manual (also known as static) bindings can be configured individually directly on the router or, by using the DHCP—Static Mapping feature, these static bindings can be read from a separate static mapping text file. The static mapping text files are read when a router reloads or the DHCP service restarts. These files are read-only.

The read static bindings are treated just like the manual bindings, in that they are:

- Retained across DHCPRELEASEs from the clients.
- Not timed out.
- Deleted only upon deletion of the pool.
- Provided appropriate exclusions for the contained addresses, which are created at the time of the read.

Just like automatic bindings and manual bindings, the static bindings from the static mapping text file are also displayed by using the **show ip dhcp binding** command.

This section contains the following tasks:

- Creating the Static Mapping Text File (required)
- Configuring the DHCP Server to Read a Static Mapping Text File (required)

Creating the Static Mapping Text File

Perform this task to create the static mapping text file. You will input your addresses in the text file, which is stored in the DHCP database for the DHCP server to read. There is no limit on the number of addresses in the file. The file format has the following elements:

- Time the file was created
- Database version number
- IP address
- Hardware type
- Hardware address
- Lease expiration
- End-of-file designator

See Table 1 for more details about the format of the text file.

The following is a sample static mapping text file:

```
*time* Jan 21 2005 03:52 PM
*version* 2
!IP address    Type    Hardware address     Lease expiration
10.0.0.4 /24   1       0090.bff6.081e       Infinite
10.0.0.5 /28   id      00b7.0813.88f1.66    Infinite
10.0.0.2 /21   1       0090.bff6.081d       Infinite
*end*
```

Table 1 Static Mapping Text File Field Descriptions

Field	Description
time	Specifies the time the file was created. This field allows DHCP to differentiate between newer and older database versions when multiple agents are configured. The valid format of the time is Mmm dd yyyy hh:mm AM/PM.
version 2	Database version number.
IP address	Static IP address. If the subnet mask is not specified, a natural mask is assumed depending on the IP address. There must be a space between the IP address and mask.
Type	Specifies the hardware type. For example, type "1" indicates Ethernet. The type "id" indicates that the field is a DHCP client identifier. Legal values can be found online at http://www.iana.org/assignments/arp-parameters in the "Number Hardware Type" list.
Hardware address	Specifies the hardware address. When the type is numeric, it refers to the hardware media. Legal values can be found online at http://www.iana.org/assignments/arp-parameters in the "Number Hardware Type" list. When the type is "id," this means that we are matching on the client identifier. For more information about the client identifier, please see RFC 2132, *DHCP Options and BOOTP Vendor Extensions*, section 9.14, located at http://www.ietf.org/rfc/rfc2132.txt. or the **client-identifier** command reference page located at http://www.cisco.com/en/US/docs/ios/ipaddr/command/reference/iad_dhc1.html#wp1011901. If you are unsure what client identifier to match on, use the **debug dhcp detail** command to display the client identifier being sent to the DHCP server from the client.
Lease expiration	Specifies the expiration of the lease. "Infinite" specifies that the duration of the lease is unlimited.
end	End of file. DHCP uses the *end* designator to detect file truncation.

Configuring the DHCP Server to Read a Static Mapping Text File

Perform this task to configure the DHCP server to read the static mapping text file.

Prerequisites

The administrator should create the static mapping text file in the correct format and configure the address pools before performing this task.

Before editing the file, you must disable the DHCP server using the **no service dhcp** command.

Restrictions

The static bindings must not be deleted when a DHCPRELEASE is received or must not be timed out by the DHCP timer. The static bindings should be treated just like manual bindings created by using the **ip dhcp pool** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp pool** *name*
4. **origin file** *url*
5. **end**
6. **show ip dhcp binding** [*address*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`enable`** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**`configure terminal`** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**`ip dhcp pool`** *`name`* **Example:** `Router(config)# ip dhcp pool pool1`	Assigns a name to a DHCP pool and enters DHCP configuration mode. **Note** If you have already configured the IP DHCP pool name using the **ip dhcp pool** command and the static file URL using the **origin file** command, you must perform a fresh read using the **no service dhcp** command and **service dhcp** command.
Step 4	**`origin file`** *`url`* **Example:** `Router(dhcp-config)# origin file tftp://10.1.0.1/static-bindings`	Specifies the URL from which the DHCP server can locate the text file.
Step 5	**`end`** **Example:** `Router(dhcp-config)# end`	Returns to privileged EXEC mode.
Step 6	**`show ip dhcp binding`** [*`address`*] **Example:** `Router# show ip dhcp binding`	(Optional) Displays a list of all bindings created on a specific DHCP server.

Examples

The following example shows the address bindings that have been configured:

```
Router# show ip dhcp binding

00:05:14:%SYS-5-CONFIG_I: Configured from console by console

Bindings from all pools not associated with VRF:
IP address  Client-ID/            Ls expir   Type    Hw address            User name
10.9.9.4/8  0063.7363.2d30.3036.  Infinite   Static  302e.3762.2e39.3634.  632d.4574.8892.
10.9.9.1/24 0063.6973.636f.2d30.  Infinite   Static  3036.302e.3437.3165.  2e64.6462.342d.
```

The following sample shows each entry in the static mapping text file:

```
*time* Jan 21 2005 22:52 PM
!IP address     Type        Hardware address                            Lease expiration
10.19.9.1 /24    id          0063.6973.636f.2d30.3036.302e.3437
10.9.9.4         id          0063.7363.2d30.3036.302e.3762.2e39.3634.632d  Infinite
*end*
```

The following sample debug output shows the reading of the static mapping text file from the TFTP server:

```
Router# debug ip dhcp server

Loading abc/static_pool from 10.19.192.33 (via Ethernet0):
[OK - 333 bytes]

*May 26 23:14:21.259: DHCPD: contacting agent tftp://10.19.192.33/abc/static_pool (attempt
0)
*May 26 23:14:21.467: DHCPD: agent tftp://10.19.192.33/abc/static_pool is responding.
*May 26 23:14:21.467: DHCPD: IFS is ready.
*May 26 23:14:21.467: DHCPD: reading bindings from
                             tftp://10.19.192.33/abc/static_pool.
*May 26 23:14:21.707: DHCPD: read 333 / 1024 bytes.
*May 26 23:14:21.707: DHCPD: parsing text line
*time* Apr 22 2002 11:31 AM
*May 26 23:14:21.707: DHCPD: parsing text line ""
*May 26 23:14:21.707: DHCPD: parsing text line
!IP address Type Hardware address Lease expiration
*May 26 23:14:21.707: DHCPD: parsing text line
"10.9.9.1 /24 id 0063.6973.636f.2d30.3036.302e.3437"
*May 26 23:14:21.707: DHCPD: creating binding for 10.9.9.1
*May 26 23:14:21.707: DHCPD: Adding binding to radix tree (10.9.9.1)
*May 26 23:14:21.707: DHCPD: Adding binding to hash tree
*May 26 23:14:21.707: DHCPD: parsing text line
"10.9.9.4 id 0063.7363.2d30.3036.302e.3762.2e39.3634.632d"
*May 26 23:14:21.711: DHCPD: creating binding for 10.9.9.4
*May 26 23:14:21.711: DHCPD: Adding binding to radix tree (10.9.9.4)
*May 26 23:14:21.711: DHCPD: Adding binding to hash tree
*May 26 23:14:21.711: DHCPD: parsing text line "Infinite"
*May 26 23:14:21.711: DHCPD: parsing text line ""
*May 26 23:14:21.711: DHCPD: parsing text line
!IP address Interface-index Lease expiration VRF
*May 26 23:14:21.711: DHCPD: parsing text line "*end*"
*May 26 23:14:21.711: DHCPD: read static bindings from
tftp://10.19.192.33/abcemp/static_pool.
```

Customizing DHCP Server Operation

Perform this task to customize the behavior of the DHCP server.

Ping Packet Settings

By default, the DHCP server pings a pool address twice before assigning a particular address to a requesting client. If the ping is unanswered, the DHCP server assumes (with a high probability) that the address is not in use and assigns the address to the requesting client.

By default, the DHCP server waits 2 seconds before timing out a ping packet.

Option to Ignore All BOOTP Requests

You can configure the DHCP server to ignore and not reply to received Bootstrap Protocol (BOOTP) requests. This functionality is beneficial when there is a mix of BOOTP and DHCP clients in a network segment and there is a BOOTP server and a Cisco IOS DHCP server servicing the network segment. The BOOTP server is configured with static bindings for the BOOTP clients and the BOOTP clients are intended to obtain their addresses from the BOOTP server. However, because a DHCP server can also respond to a BOOTP request, an address offer may be made by the DHCP server causing the BOOTP clients to boot with the address from the DHCP server, instead of the address from the BOOTP server. Configuring the DHCP server to ignore BOOTP requests means that the BOOTP clients will receive address information from the BOOTP server and will not inadvertently accept an address from a DHCP server.

The Cisco IOS software can forward these ignored BOOTP request packets to another DHCP server if the **ip helper-address** interface configuration command is configured on the incoming interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp ping packets** *number*
4. **ip dhcp ping timeout** *milliseconds*
5. **ip dhcp bootp ignore**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`enable`** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**`configure terminal`** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**`ip dhcp ping packets`** *`number`* **Example:** `Router(config)# ip dhcp ping packets 5`	(Optional) Specifies the number of ping packets the DHCP server sends to a pool address before assigning the address to a requesting client. • The default is two packets. Setting the *number* argument to a value of 0 disables the DHCP server ping operation completely.
Step 4	**`ip dhcp ping timeout`** *`milliseconds`* **Example:** `Router(config)# ip dhcp ping timeout 850`	(Optional) Specifies the amount of time the DHCP server waits for a ping reply from an address pool.
Step 5	**`ip dhcp bootp ignore`** **Example:** `Router(config)# ip dhcp bootp ignore`	(Optional) Allows the DHCP server to selectively ignore and not reply to received BOOTP requests. • The **ip dhcp bootp ignore** command applies to all DHCP pools configured on the router. BOOTP requests cannot be selectively ignored on a per-DHCP pool basis.

Configuring a Remote Router to Import DHCP Server Options from a Central DHCP Server

The Cisco IOS DHCP server can dynamically configure options such as the DNS and WINS addresses to respond to DHCP requests from local clients behind the customer premises equipment (CPE). Previously, network administrators needed to manually configure the Cisco IOS DHCP server on each device. The Cisco IOS DHCP server was enhanced to allow configuration information to be updated automatically. Network administrators can configure one or more centralized DHCP servers to update specific DHCP options within the DHCP pools. The remote servers can request or "import" these option parameters from the centralized servers.

This section contains the following tasks:

- Configuring the Central DHCP Server to Update DHCP Options, page 25
- Configuring the Remote Router to Import DHCP Options, page 26

Configuring the Central DHCP Server to Update DHCP Options

Perform this task to configure the central DHCP server to update DHCP options.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp pool** *pool-name*
4. **network** *network-number* [*mask* | /*prefix-length*]
5. **dns-server** *address* [*address2 ... address8*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip dhcp pool** *name* **Example:** `Router(config)# ip dhcp pool 1`	Creates a name for the DHCP server address pool and enters DHCP pool configuration mode.
Step 4	**network** *network-number* [*mask* | **/***prefix-length*] **Example:** `Router(dhcp-config)# network 172.16.0.0 /16`	Specifies the subnet network number and mask of the DHCP address pool.
Step 5	**dns-server** *address* [*address2* ... *address8*] **Example:** `Router(dhcp-config)# dns server 172.16.1.103 172.16.2.103`	(Optional) Specifies the IP address of a DNS server that is available to a DHCP client. • One IP address is required; however, you can specify up to eight IP addresses in one command line. • Servers should be listed in order of preference.

Configuring the Remote Router to Import DHCP Options

Perform this task to configure the remote router to import DHCP options from a central DHCP server.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp pool** *pool-name*
4. **network** *network-number* [*mask* | **/***prefix-length*]
5. **import all**
6. **exit**
7. **interface** *type number*
8. **ip address dhcp**
9. **end**
10. **show ip dhcp import**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip dhcp pool** *pool-name* **Example:** `Router(config)# ip dhcp pool sanjose1`	Creates a name for the DHCP server address pool and enters DHCP pool configuration mode.
Step 4	**network** *network-number* [*mask* | **/***prefix-length*] **Example:** `Router(dhcp-config)# network 172.30.0.0 /16`	Specifies the subnet network number and mask of the DHCP address pool.
Step 5	**import all** **Example:** `Router(dhcp-config)# import all`	Imports DHCP option parameters into the DHCP server database.
Step 6	**exit** **Example:** `Router(dhcp-config)# exit`	Exits DHCP pool configuration mode.
Step 7	**interface** *type number* **Example:** `Router(config)# interface FastEthernet0/0`	Configures an interface and enters interface configuration mode.
Step 8	**ip address dhcp** **Example:** `Router(config-if)# ip address dhcp`	Specifies that the interface acquires an IP address through DHCP.
Step 9	**end** **Example:** `Router(dhcp-config)# end`	Returns to privileged EXEC mode.
Step 10	**show ip dhcp import** **Example:** `Router# show ip dhcp import`	Displays the options that have been imported from the central DHCP server.

Configuring DHCP Address Allocation Using Option 82

This section contains the following tasks:

- Enabling Option 82 for DHCP Address Allocation, page 29 (optional)
- Defining the DHCP Class and Relay Agent Information Patterns, page 30 (required)
- Defining the DHCP Address Pool, page 31 (required)

DHCP Address Allocation Using Option 82 Feature Design

DHCP provides a framework for passing configuration information to hosts on a TCP/IP network. Configuration parameters and other control information are carried in tagged data items that are stored in the options field of the DHCP message. The data items themselves are also called options. Option 82 is organized as a single DHCP option that contains information known by the relay agent.

This feature is designed to allow the Cisco IOS DHCP server to use option 82 information to help determine which IP addresses to allocate to clients. The information sent via option 82 will be used to identify which port the DHCP request came in on. This feature does not parse out the individual suboptions contained within option 82. Rather, the address allocation is done by matching a configured pattern byte by byte.

The feature introduces a new DHCP class capability, which is a method to group DHCP clients based on some shared characteristics other than the subnet in which the clients reside.

Usage Scenario for DHCP Address Allocation Using Option 82

In an example application, DHCP clients are connected to two ports of a single switch. Each port can be configured to be part of two VLANs: VLAN1 and VLAN2. DHCP clients belong to either VLAN1 or VLAN2 and it is assumed that the switch can differentiate the VLAN that a particular DHCP Discover message belongs to (possibly through Layer 2 encapsulation). Each VLAN has its own subnet and all DHCP messages from the same VLAN (same switch) will have the giaddr field set to the same value indicating the subnet of the VLAN.

The problem is that for a DHCP client connecting to port 1 of VLAN1, it must be allocated an IP address from one range within the VLAN’s subnet, whereas a DHCP client connecting to port 2 of VLAN1 must be allocated an IP address from another range. Both these two IP address ranges are part of the same subnet (and have the same subnet mask). In the normal DHCP address allocation, the DHCP server will look only at the giaddr field and thus will not be able to differentiate between the two ranges.

To solve this problem, a relay agent residing at the switch inserts the relay information option (option 82), which carries information specific to the port, and the DHCP server must inspect both the giaddr field and the inserted option 82 during the address selection process.

DHCP Class Capability

The Cisco IOS software will look up a pool based on IP address (giaddr or incoming interface IP address) and then match the request to a class or classes configured in the pool in the order the classes are specified in the DHCP pool configuration.

When a DHCP address pool has been configured with one or more DHCP classes, the pool becomes a restricted access pool, which means that no addresses will be allocated from the pool unless one or more of the classes in the pool is matched. This design allows DHCP classes to be used for either access control (no default class is configured on the pool) or to provide further address range partitions with the subnet of the pool.

Multiple pools can be configured with the same class, eliminating the need to configure the same pattern in multiple pools.

The following capabilities are currently supported for DHCP class-based address allocation:

- Specifying the full relay agent information option value as a raw hexadecimal string by using the **relay-information hex** command in the new relay agent information configuration mode.
- Support for bitmasking the raw relay information hexadecimal value.
- Support for a wildcard at the end of the hexadecimal string specified by the **relay-information hex** command.

Restrictions for DHCP Address Allocation Using Option 82

If the relay agent inserts option 82 but does not set the giaddr field in the DHCP packet, the DHCP server interface must be configured as a trusted interface by using the **ip dhcp relay information trusted** global configuration command. This configuration prevents the server from dropping the DHCP message.

Enabling Option 82 for DHCP Address Allocation

By default, the Cisco IOS DHCP server can use information provided by option 82 to allocate IP addresses. To reenable this capability if it has been disabled, perform the task described in this section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp use class**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip dhcp use class** **Example:** `Router(config)# ip dhcp use class`	Controls whether DHCP classes are used for address allocation. • This functionality is enabled by default. • Use the **no** form of this command to disable this functionality without deleting the DHCP class configuration.

Troubleshooting Tips

If DHCP classes are configured in the pool, but the DHCP server does not make use of the classes, verify if the **no ip dhcp use class** command was configured.

Defining the DHCP Class and Relay Agent Information Patterns

Perform this task to define the DHCP class and relay agent information patterns.

Prerequisites

You must know the hexadecimal value of each byte location in option 82 to be able to configure the **relay-information hex** command. The option 82 format may vary from product to product. Contact the relay agent vendor for this information.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp class** *class-name*
4. **relay agent information**
5. **relay-information hex** *pattern* [*] [**bitmask** *mask*]
6. Repeat Steps 3 through 5 for each DHCP class you need to configure.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip dhcp class** *class-name* **Example:** `Router(config)# ip dhcp class CLASS1`	Defines a DHCP class and enters DHCP class configuration mode.
Step 4	**relay agent information** **Example:** `Router(dhcp-class)# relay agent information`	Enters relay agent information option configuration mode. • If this step is omitted, then the DHCP class matches to any relay agent information option, whether it is present or not.
Step 5	**relay-information hex** *pattern* [*] [**bitmask** *mask*] **Example:** `Router(dhcp-class-relayinfo)# relay-information hex 01030a0b0c02050000000123`	(Optional) Specifies a hexadecimal value for the full relay information option. • The *pattern* argument creates a pattern that is used to match to the DHCP class. • If you omit this step, no pattern is configured and it is considered a match to any relay agent information option value, but the relay information option must be present in the DHCP packet. • You can configure multiple **relay-information hex** commands in a DHCP class.
Step 6	Repeat Steps 3 through 5 for each DHCP class you need to configure.	—

Troubleshooting Tips

You can enable the **debug ip dhcp server class** command to display the class matching results.

Defining the DHCP Address Pool

Perform this task to define the DHCP address pool.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp pool** *name*
4. **network** *network-number* [*mask* | /*prefix-length*]

5. **class** *class-name*
6. **address range** *start-ip end-ip*
7. Repeat Steps 5 and 6 for each DHCP class you need to associate to the DHCP pool.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`enable`** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**`configure terminal`** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**`ip dhcp pool`** *`name`* **Example:** `Router# ip dhcp pool ABC`	Configures a DHCP address pool on a Cisco IOS DHCP server and enters DHCP pool configuration mode. • Multiple pools can be configured with the same class, eliminating the need to configure the same pattern in multiple pools.
Step 4	**`network`** *`network-number`* `[`*`mask`* `|` **`/`***`prefix-length`*`]` **Example:** `Router(dhcp-config)# network 10.0.20.0`	Configures the subnet number and mask for a DHCP address pool on a Cisco IOS DHCP server.
Step 5	**`class`** *`class-name`* **Example:** `Router(dhcp-config)# class CLASS1`	Associates a class with a pool and enters DHCP pool class configuration mode. • This command will also create a DHCP class if the DHCP class is not yet defined.
Step 6	**`address range`** *`start-ip end-ip`* **Example:** `Router(dhcp-pool-class)# address range 10.0.20.1 10.0.20.100`	(Optional) Sets an address range for a DHCP class in a DHCP server address pool. • If this command is not configured for a class, the default value is the entire subnet of the pool.
Step 7	Repeat Steps 5 and 6 for each DHCP class you need to associate to the DHCP pool.	Each class in the DHCP pool will be examined for a match in the order configured.

Configuring a Static Route with the Next-Hop Dynamically Obtained Through DHCP

Perform this task to configure a static route to use a DHCP default gateway as the next-hop router.

This task enables static routes to be assigned using a DHCP default gateway as the next-hop router. This behavior was not possible before the introduction of this feature because the gateway IP address is not known until after the DHCP address assignment. A static route could not be configured with the command-line interface (CLI) that used that DHCP-supplied address.

The static routes are installed in the routing table when the default gateway is assigned by the DHCP server. The routes remain in the routing table until the DHCP lease expires at which time the routes are removed.

When a DHCP client releases an address, the corresponding static route (the route configured with the **ip route** command) is automatically removed from the routing table. If the DHCP router option (option 3 of the DHCP packet) changes during the client renewal, the DHCP default gateway changes to the new IP address supplied in the renewal.

This feature is particularly useful for VPN deployments such as Dynamic Multipoint VPNs (DMVPNs). This feature is useful when a non-physical interface like a multipoint generic routing encapsulation (mGRE) tunnel is configured on the router and certain traffic needs to be excluded from going to the tunnel interface.

Prerequisites

Verify all DHCP client and server configuration steps. Ensure that the DHCP client and server are properly defined to supply a DHCP router option 3.

Restrictions

- If the DHCP client is not able to obtain an IP address or default router IP address, the static route is not installed in the routing table.
- If the lease has expired and the DHCP client cannot renew the address, the DHCP IP address assigned to the client is released and any associated static routes are removed from the routing table.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip route** *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} **dhcp** [*distance*]
4. **end**
5. **show ip route**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip route** *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} **dhcp** [*distance*] **Example:** `Router(config)# ip route 209.165.200.225 255.255.255.255 ether1 dhcp` `Router(config)# ip route 209.165.200.226 255.255.255.255 ether2 dhcp 20`	Assigns a static route for the default next-hop router when the DHCP server is accessed for an IP address. • If more than one interface on a router is configured to obtain an IP address from a DHCP server, use the **ip route** *prefix mask interface-type interface-number* **dhcp** command for each interface. If the interface is not specified, the route is added to the routing table as soon as any of the interfaces obtain an IP address and default router.
Step 4	**end** **Example:** `Router(dhcp-config)# end`	Returns to global configuration mode.
Step 5	**show ip route** **Example:** `Router# show ip route`	(Optional) Displays the current state of the routing table. • Use this command to display assigned static routes once the DHCP client obtains an address and a default router address from the DHCP server.

Clearing DHCP Server Variables

Perform this task to clear DHCP server variables.

SUMMARY STEPS

1. **enable**
2. **clear ip dhcp binding** {*address* | *}
3. **clear ip dhcp conflict** {*address* | *}
4. **clear ip dhcp server statistics**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear ip dhcp binding** {*address* | *****} **Example:** `Router# clear ip dhcp binding *`	Deletes an automatic address binding from the DHCP database. • Specifying the *address* argument clears the automatic binding for a specific (client) IP address, whereas specifying an asterisk (*) clears all automatic bindings.
Step 3	**clear ip dhcp conflict** {*address* | *****} **Example:** `Router# clear ip dhcp conflict 172.16.1.103`	Clears an address conflict from the DHCP database. • Specifying the *address* argument clears the conflict for a specific IP address, whereas specifying an asterisk (*) clears conflicts for all addresses.
Step 4	**clear ip dhcp server statistics** **Example:** `Router# clear ip dhcp server statistics`	Resets all DHCP server counters to 0.

Configuration Examples for the Cisco IOS DHCP Server

This section provides the following configuration examples:

- Configuring the DHCP Database Agent: Example, page 35
- Excluding IP Addresses: Example, page 36
- Configuring DHCP Address Pools: Example, page 36
- Configuring a DHCP Address Pool with Multiple Disjoint Subnets: Example, page 37
- Configuring Manual Bindings: Example, page 38
- Configuring Static Mapping: Example, page 39
- Configuring the Option to Ignore all BOOTP Requests: Example, page 39
- Importing DHCP Options: Example, page 40
- Configuring DHCP Address Allocation Using Option 82: Example, page 42
- Configuring a Static Route with the Next-Hop Dynamically Obtained Through DHCP: Example, page 43

Configuring the DHCP Database Agent: Example

The following example shows how to store bindings on host 172.16.4.253. The file transfer protocol is FTP. The server should wait 2 minutes (120 seconds) before writing database changes.

```
ip dhcp database ftp://user:password@172.16.4.253/router-dhcp write-delay 120
```

Excluding IP Addresses: Example

In the following example, server A and server B service the subnet 10.0.20.0/24. Splitting the subnet equally between the two servers, server A is configured to allocate IP addresses 10.0.20.1 to 10.0.20.125 and server B is configured to allocate IP addresses 10.0.20.126 to 10.0.20.254.

Server A

```
ip dhcp excluded-address 10.0.20.126 10.0.20.255
!
ip dhcp pool A
 network 10.0.20.0 255.255.255.0
```

Server B

```
ip dhcp excluded-address 10.0.20.0 10.0.20.125
!
ip dhcp pool B
 network 10.0.20.0 255.255.255.0
```

Configuring DHCP Address Pools: Example

In the following example, three DHCP address pools are created: one in network 172.16.0.0, one in subnetwork 172.16.1.0, and one in subnetwork 172.16.2.0. Attributes from network 172.16.0.0—such as the domain name, DNS server, NetBIOS name server, and NetBIOS node type—are inherited in subnetworks 172.16.1.0 and 172.16.2.0. In each pool, clients are granted 30-day leases and all addresses in each subnetwork, except the excluded addresses, are available to the DHCP server for assigning to clients. Table 2 lists the IP addresses for the devices in three DHCP address pools.

Table 2 DHCP Address Pool Configuration Example

Pool 0 (Network 172.16.0.0)		Pool 1 (Subnetwork 172.16.1.0)		Pool 2 (Subnetwork 172.16.2.0)	
Device	**IP Address**	**Device**	**IP Address**	**Device**	**IP Address**
Default routers	–	Default routers	172.16.1.100 172.16.1.101	Default routers	172.16.2.100 172.16.2.101
DNS server	172.16.1.102 172.16.2.102	—	—	—	—
NetBIOS name server	172.16.1.103 172.16.2.103	—	—	—	—
NetBIOS node type	h-node	—	—	—	—

```
ip dhcp database ftp://user:password@172.16.4.253/router-dhcp write-delay 120
ip dhcp excluded-address 172.16.1.100 172.16.1.103
ip dhcp excluded-address 172.16.2.100 172.16.2.103
!
ip dhcp pool 0
 network 172.16.0.0 /16
 domain-name cisco.com
 dns-server 172.16.1.102 172.16.2.102
 netbios-name-server 172.16.1.103 172.16.2.103
 netbios-node-type h-node
```

```
!
ip dhcp pool 1
 network 172.16.1.0 /24
 default-router 172.16.1.100 172.16.1.101
 lease 30
!
ip dhcp pool 2
 network 172.16.2.0 /24
 default-router 172.16.2.100 172.16.2.101
 lease 30
```

Configuring a DHCP Address Pool with Multiple Disjoint Subnets: Example

Multiple disjoint subnets in a DHCP pool can be used in any of the following network topologies:

- IP address pooling—The DHCP client and server reside on the same subnet.
- DHCP relay—The DHCP client and DHCP server communicate through a DHCP relay agent where the relay interface is configured with secondary IP addresses.
- Hierarchical DHCP—The DHCP server is configured as the DHCP subnet allocation server, and the DHCP client and DHCP subnet allocation server communicate through an on-demand address pool (ODAP) router.

In the following example, one DHCP address pool named pool3 is created; the primary subnet is 172.16.0.0/16, one secondary subnet is 172.16.1.0/24, and another secondary subnet is 172.16.2.0/24.

- When the IP addresses in the primary subnet are exhausted, the DHCP server inspects the secondary subnets in the order in which the subnets were added to the pool.
- When the DHCP server allocates an IP address from the secondary subnet 172.16.1.0/24, the server uses the subnet-specific default router list that consists of IP addresses 172.16.1.100 and 172.16.1.101. When the DHCP server allocates an IP address from the subnet 172.16.2.0/24, however, the server uses the pool-wide list that consists of the four IP addresses from 172.16.0.100 to 172.16.0.103.
- Other attributes from the primary subnet 172.16.0.0/16—such as the domain name, DNS server, NetBIOS name server, and NetBIOS node type—are inherited in both of the secondary subnets.
- DHCP clients are granted 30-day leases on IP addresses in the pool. All addresses in each subnet, except the excluded addresses, are available to the DHCP server for assigning to clients.

Table 3 lists the IP addresses for the devices in the DHCP address pool that consists of three disjoint subnets.

Table 3 DHCP Address Pool Configuration with Multiple Disjoint Subnets Example

Primary Subnet (172.16.0.0/16)		**First Secondary Subnet (172.16.1.0/24)**		**Second Secondary Subnet (172.16.2.0/24)**	
Device	**IP Address**	**Device**	**IP Address**	**Device**	**IP Address**
Default routers	172.16.0.100 172.16.0.101 172.16.0.102 172.16.0.103	Default routers	172.16.1.100 172.16.1.101	Default routers	172.16.0.100 172.16.0.101 172.16.0.102 172.16.0.103
DNS server	172.16.1.102 172.16.2.102	—	—	—	—

Table 3 DHCP Address Pool Configuration with Multiple Disjoint Subnets Example (continued)

Primary Subnet (172.16.0.0/16)		First Secondary Subnet (172.16.1.0/24)		Second Secondary Subnet (172.16.2.0/24)	
Device	**IP Address**	**Device**	**IP Address**	**Device**	**IP Address**
NetBIOS name server	172.16.1.103 172.16.2.103	—	—	—	—
NetBIOS node type	h-node	—	—	—	—

```
ip dhcp database ftp://user:password@172.16.4.253/router-dhcp write-delay 120
ip dhcp excluded-address 172.16.0.100 172.16.1.103
ip dhcp excluded-address 172.16.1.100 172.16.1.101
!
ip dhcp pool pool3
 network 172.16.0.0 /16
 default-router 172.16.0.100 172.16.2.101 172.16.0.102 172.16.0.103
 domain-name cisco.com
 dns-server 172.16.1.102 172.16.2.102
 netbios-name-server 172.16.1.103 172.16.2.103
 netbios-node-type h-node
 lease 30
!
 network 172.16.1.0 /24 secondary
  override default-router 172.16.1.100 172.16.1.101
  exit
!
 network 172.16.2.0 /24 secondary
```

Configuring Manual Bindings: Example

The following example shows how to create a manual binding for a client named Mars.cisco.com. The MAC address of the client is 02c7.f800.0422 and the IP address of the client is 172.16.2.254.

```
ip dhcp pool Mars
 host 172.16.2.254
 hardware-address 02c7.f800.0422 ieee802
 client-name Mars
```

Because attributes are inherited, the previous configuration is equivalent to the following:

```
ip dhcp pool Mars
 host 172.16.2.254 mask 255.255.255.0
 hardware-address 02c7.f800.0422 ieee802
 client-name Mars
 default-router 172.16.2.100 172.16.2.101
 domain-name cisco.com
 dns-server 172.16.1.102 172.16.2.102
 netbios-name-server 172.16.1.103 172.16.2.103
 netbios-node-type h-node
```

Configuring Static Mapping: Example

The following example shows how to restart the DHCP server, configure the pool, and specify the URL at which the static mapping text file is stored:

```
no service dhcp
service dhcp
ip dhcp pool abcpool
 origin file tftp://10.1.0.1/staticfilename
```

Note The static mapping text file can be copied to flash memory on the router and served by the tftp process of the router. In this case, the IP address in the origin file line must be an address owned by the router and one additional line of configuration is required on the router:

tftp-server flash *staticfilename*

Configuring the Option to Ignore all BOOTP Requests: Example

The following example shows two DHCP pools that are configured on the router and that the router's DHCP server is configured to ignore all received BOOTP requests. If a BOOTP request is received from subnet 10.0.18.0/24, the request will be dropped by the router (because the **ip helper-address** command is not configured). If there is a BOOTP request from subnet 192.168.1.0/24, the request will be forwarded to 172.16.1.1 via the **ip helper-address** command.

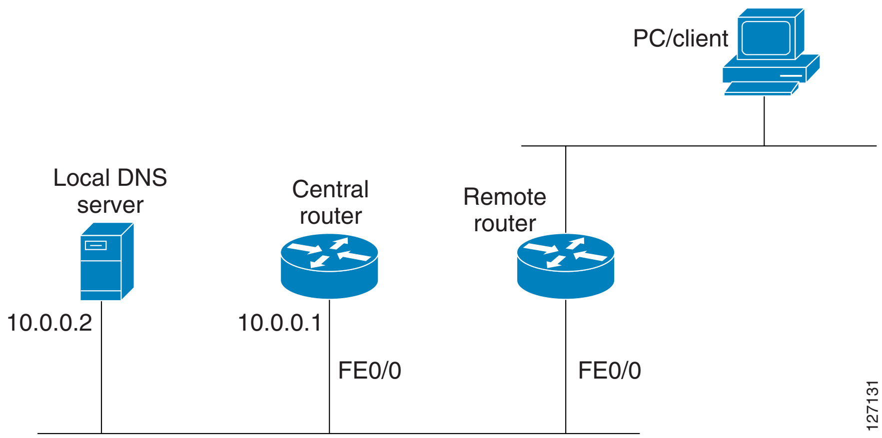
```
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Router
!
ip subnet-zero
!
ip dhcp bootp ignore
!
ip dhcp pool ABC
   network 192.168.1.0 255.255.255.0
   default-router 192.168.1.3
   lease 2
!
ip dhcp pool DEF
   network 10.0.18.0 255.255.255.0
!
ip cef
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface Ethernet1/0
 ip address 10.0.18.68 255.255.255.0
 duplex half
!
interface Ethernet1/1
 ip address 192.168.1.1 255.255.255.0
```

```
 ip helper-address 172.16.1.1
 duplex half
!
interface Ethernet1/2
 shutdown
 duplex half
!
interface Ethernet1/3
 no ip address
 shutdown
 duplex half
!
interface FastEthernet2/0
 no ip address
 shutdown
 duplex half
!
ip route 172.16.1.1 255.255.255.255 e1/0
no ip http server
no ip pim bidir-enable
!
call rsvp-sync
!
mgcp profile default
!
dial-peer cor custom
!
gatekeeper
 shutdown
!
line con 0
line aux 0
line vty 0 4
!
end
```

Importing DHCP Options: Example

The following example shows a remote and central server configured to support the importing of DHCP options. The central server is configured to automatically update DHCP options, such as DNS and WINs addresses, within the DHCP pools. In response to a DHCP request from a local client behind CPE equipment, the remote server can request or "import" these option parameters from the centralized server. See Figure 1 for a diagram of the network topology.

Figure 1 DHCP Example Network Topology



Central Router

```
!do not assign this range to DHCP clients
ip dhcp-excluded address 10.0.0.1 10.0.0.5
!
ip dhcp pool central
! Specifies network number and mask for DHCP clients
 network 10.0.0.0 255.255.255.0
! Specifies the domain name for the client
 domain-name central
! Specifies DNS server that will respond to DHCP clients when they need to correlate host
! name to ip address
 dns-server 10.0.0.2
!Specifies the NETBIOS WINS server
 netbios-name-server 10.0.0.2
!
interface FastEthernet0/0
 ip address 10.0.0.1 255.255.255.0
 duplex auto
 speed auto
```

Remote Router

```
ip dhcp pool client
! Imports DHCP option parameters into DHCP server database
 import all
 network 20.0.0.0 255.255.255.0
!
interface FastEthernet0/0
 ip address dhcp
 duplex auto
 speed auto
```

Configuring DHCP Address Allocation Using Option 82: Example

This example configures two DHCP classes. CLASS1 defines the group of DHCP clients whose address requests contain the relay agent information option with the specified hexadecimal values. CLASS2 defines the group of DHCP clients whose address requests contain the configured relay agent information suboptions. CLASS3 has no pattern configured and is treated as a "match to any" class. This type of class is useful for specifying a "default" class.

In the following example, the subnet of pool ABC has been divided into three ranges without further subnetting of the 10.0.20.0/24 subnet. If there is a DHCP Discover message from the 10.0.20.0/24 subnet with option 82 matching that of class CLASS1, an available address in the range from 10.0.20.1 to 10.0.20.100 will be allocated. If there is no free address in CLASS1's address range, the DHCP Discover message will be matched against CLASS2, and so on.

Thus, each class in the DHCP pool will be examined for a match in the order configured by the user. In pool ABC, the order of matching is CLASS1, CLASS2, and finally CLASS3. In pool DEF, class CLASS2 does not have any address range configured. By default, the address range for a particular class is the pool's entire subnet(s). Therefore, clients matching CLASS2 may be allocated addresses from 11.0.20.1 to 11.0.20.254.

Multiple pools can be configured with the same class, eliminating the need to configure the same patterns in multiple pools. In the future, further classification method may be implemented. For example, there may be a need to specify that one or more pools should only be used to service a particular class of devices (for example, cable modems and IP phones).

```
! Defines the DHCP classes and relay information patterns
ip dhcp class CLASS1
 relay agent information
  relay-information hex 01030a0b0c02050000000123
  relay-information hex 01030a0b0c02*
  relay-information hex 01030a0b0c02050000000000 bitmask 0000000000000000000000FF

ip dhcp class CLASS2
 relay agent information
  relay-information hex 01040102030402020102
  relay-information hex 01040101030402020102

ip dhcp class CLASS3
 relay agent information

! Associates the DHCP pool with DHCP classes
ip dhcp pool ABC
 network 10.0.20.0 255.255.255.0
 class CLASS1
  address range 10.0.20.1 10.0.20.100
class CLASS2
  address range 10.0.20.101 10.0.20.200
 class CLASS3
  address range 10.0.20.201 10.0.20.254

ip dhcp pool DEF
 network 11.0.20.0 255.255.255.0
 class CLASS1
  address range 11.0.20.1 11.0.20.64
 class CLASS2
```

Configuring a Static Route with the Next-Hop Dynamically Obtained Through DHCP: Example

The following example shows how to configure two Ethernet interfaces to obtain the next-hop router IP address from the DHCP server:

```
ip route 10.10.10.0 255.255.255.0 dhcp 200
ip route 10.10.20.1 255.255.255.255 ether 1 dhcp
```

Additional References

The following sections provide references related to the Cisco IOS DHCP server.

Related Documents

Related Topic	Document Title
DHCP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Addressing Services Command Reference*
DHCP conceptual information	"DHCP Overview" module
DHCP relay agent configuration	"Configuring the Cisco IOS DHCP Relay Agent" module
DHCP server on-demand address pools	"Configuring the DHCP Server On-Demand Address Pool Manager" module
DHCP client configuration	"Configuring the Cisco IOS DHCP Client" module
DHCP advanced features	"Configuring DHCP Services for Accounting and Security" module
DHCP enhancements for edge-session management	"Configuring DHCP Enhancements for Edge-Session Management" module
DHCP options	"DHCP Options" appendix in the *Network Registrar User's Guide*, Release 6.1.1

Standards

Standards	Title
No new or modified standards are supported by this functionality.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 951	*Bootstrap Protocol (BOOTP)*
RFC 1542	*Clarifications and Extensions for the Bootstrap Protocol*
RFC 2131	*Dynamic Host Configuration Protocol*
RFC 2132	*DHCP Options and BOOTP Vendor Extensions*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for the Cisco IOS DHCP Server

Table 4 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.2(1) or a later release appear in the table.

Not all commands may be available in your Cisco IOS software release. For details on when support for a specific command was introduced, see the command reference documentation.

For information on a feature in this technology that is not documented here, see the “DHCP Features Roadmap”.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 4 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 4 Feature Information for the Cisco IOS DHCP Server

Feature Name	Releases	Feature Configuration Information
DHCP Address Allocation Using Option 82	12.3(4)T 12.2(28)SB 12.2(33)SRB Cisco IOS XE Release 2.1	The Cisco IOS DHCP server can allocate dynamic IP addresses based on the relay information option (option 82) information sent by the relay agent. The following sections provides information about this feature: • DHCP Server Address Allocation Using Option 82 • Configuring DHCP Address Allocation Using Option 82 • Configuring DHCP Address Allocation Using Option 82: Example The following commands were introduced by this feature: **address range**, **class**, **ip dhcp class**, **ip dhcp use class**, **relay agent information**, **relay-information hex**.
DHCP Server Import All Enhancement	12.2(15)T 12.2(33)SRC	The feature is an enhancement to the **import all** global configuration command. Before this feature was introduced, the options imported through the **import all** command were overwritten by those imported by another subsystem. Through this feature, options imported by multiple subsystems can co-exist in the DHCP address pool. When the session is terminated or the lease is released, the imported options are cleared. The following sections provides information about this feature: • Configuring a Remote Router to Import DHCP Server Options from a Central DHCP Server • Importing DHCP Options: Example, page 40

Table 4 Feature Information for the Cisco IOS DHCP Server (continued)

Feature Name	Releases	Feature Configuration Information
DHCP Server Multiple Subnet	12.4(15)T 12.2(33)SRB	This feature enables multiple subnets to be configured under the same DHCP address pool. The following sections provides information about this feature: • Configuring DHCP Address Pools • Configuring a DHCP Address Pool with Multiple Disjoint Subnets: Example, page 37 The following command was introduced by this feature: **override default-router**. The following command was modified by this feature: **network (DHCP)**.
DHCP Server Option to Ignore all BOOTP Requests	12.2(8)T 12.2(28)SB	This feature allows the Cisco IOS DHCP server to selectively ignore and not reply to received Bootstrap Protocol (BOOTP) request packets. The following sections provides information about this feature: • Customizing DHCP Server Operation • Configuring the Option to Ignore all BOOTP Requests: Example The following command was introduced by this feature: **ip dhcp bootp ignore**.
DHCP Static Mapping	12.3(11)T 12.2(28)SB 12.2(33)SRC	Configuring static mapping pools enables the DHCP server to read the static bindings from a separate text file (similar in format to the DHCP database file) that is stored in these special pools. The following sections provides information about this feature: • Configuring DHCP Static Mapping • Configuring Static Mapping: Example The following command was modified by this feature: **origin**.
DHCP Statically Configured Routes Using a DHCP Gateway	12.3(8)T 12.2(28)SB 12.2(33)SRC Cisco IOS XE Release 2.1	This feature enables the configuration of static routes that point to an assigned DHCP next hop router. The following sections provides information about this feature: • Configuring a Static Route with the Next-Hop Dynamically Obtained Through DHCP • Configuring a Static Route with the Next-Hop Dynamically Obtained Through DHCP: Example The following commands were modified by this feature: **ip route**, **show ip route**.

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

Configuring the DHCP Server On-Demand Address Pool Manager

First Published: May 2, 2005
Last Updated:December 31, 2007

The Cisco IOS DHCP server on-demand address pool (ODAP) manager is used to centralize the management of large pools of addresses and simplify the configuration of large networks. ODAP provides a central management point for the allocation and assignment of IP addresses. When a Cisco IOS router is configured as an ODAP manager, pools of IP addresses are dynamically increased or reduced in size depending on the address utilization level. A DHCP pool configured in the router can also be used as an IP address pooling mechanism. The IP address pooling mechanism is configured in the router to specify the source of IP addresses for PPP peers.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. For the latest feature information and caveats, see the release notes for your Cisco IOS software release. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for the DHCP Server On-Demand Address Pool Manager" section on page 37.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Configuring the DHCP Server On-Demand Address Pool Manager

Before you configure the ODAP manager, you should understand the concepts documented in the "DHCP Overview" module.

You must configure standard Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs) unless you intend to use non-MPLS VPNs.

In order for the IP address pooling mechanism to work correctly, the VPN routing and forwarding instance (VRF) of the PPP session must match that configured on the pool. Typically this matching is done either by configuring the **ip vrf forwarding** *vrf-name* command on the virtual template interface, or if AAA is used to authorize the PPP user, it can be part of the user's profile configuration.

Restrictions for Configuring the DHCP Server On-Demand Address Pool Manager

- The **ip dhcp excluded-address** global configuration command cannot be used to exclude addresses from VRF associated pools.
- The **vrf** DHCP pool configuration command is currently not supported for host pools.
- Attribute inheritance is not supported on VRF pools.
- A router can be configured as a subnet allocation server and a DHCP server at the same time with one restriction: separate pools must be created for subnet allocation and IP address assignment. An address pool cannot be used by DHCP for both subnet allocation and IP address assignment.

Information About the DHCP Server On-Demand Address Pool Manager

Before you configure an ODAP, you should understand the following concepts:

ODAP Manager Operation

ODAPs enable pools of IP addresses to be dynamically increased or reduced in size depending on the address utilization level. Once configured, the ODAP is populated with one or more subnets leased from a source server and is ready to serve address requests from DHCP clients or from PPP sessions. The source server can be a remote DHCP server or a RADIUS server (via AAA). Currently, only the Cisco Access Registrar RADIUS server supports ODAPs. Subnets can be added to the pool when a certain utilization level (high utilization mark) is achieved. When the utilization level falls below a certain level (low utilization mark), a subnet can be returned to the server from which it was originally leased. Summarized routes for each leased subnet must be inserted or removed from the related VRF with each addition or removal of subnets into the ODAP.

ODAPs support address assignment using DHCP for customers using private addresses such as in MPLS VPNs. VPNs allow the possibility that two pools in separate networks can have the same address space, with private network addresses, served by the same DHCP server. These IP addresses can be distinguished by a VPN identifier to help select the VPN to which the client belongs.

Each ODAP is configured and associated with a particular MPLS VPN. Cisco IOS software also supports non-MPLS VPN address pools by adding pool name support to the **peer default ip address dhcp-pool** *pool-name* command.

For MPLS VPNs, each VPN is associated with one or more VRFs. The VRF is a key element in the VPN technology because it maintains the routing information that defines a customer VPN site. This customer site is attached to a provider edge (PE) router. A VRF consists of an IP routing table, a derived Cisco Express Forwarding (CEF) table, a set of interfaces that use the forwarding table, and a set of rules and routing protocol parameters that control the information that is included in the routing table.

A PPP session belonging to a specific VPN is only allocated an address from the ODAP associated with that VPN. These PPP sessions are terminated on a Virtual Home Gateway (VHG)/PE router where the ODAP is configured. The VHG/PE router maps the remote user to the corresponding MPLS VPNs.

For PPP sessions, individual address allocation from an ODAP follows a First Leased subnet First (FLF) policy. FLF searches for a free address beginning on the first leased subnet, followed by a search on the second leased subnet if no free address is available in the first subnet, and so on. This policy provides the benefit of grouping the leased addresses over time to a set of subnets, which allows an efficient subnet release and route summarization.

However, the FLF policy differs from the normal DHCP address selection policy. Normal DHCP address selection takes into account the IP address of the receiving interface or the gateway address if it is nonzero. To support both policies, the DHCP server needs to be able to distinguish between a normal DHCP address request and an address request for a PPP client. The ODAP manager uses an IP address pooling mechanism for PPP that allows the DHCP server to distinguish between a normal DHCP address request and a request from a PPP client.

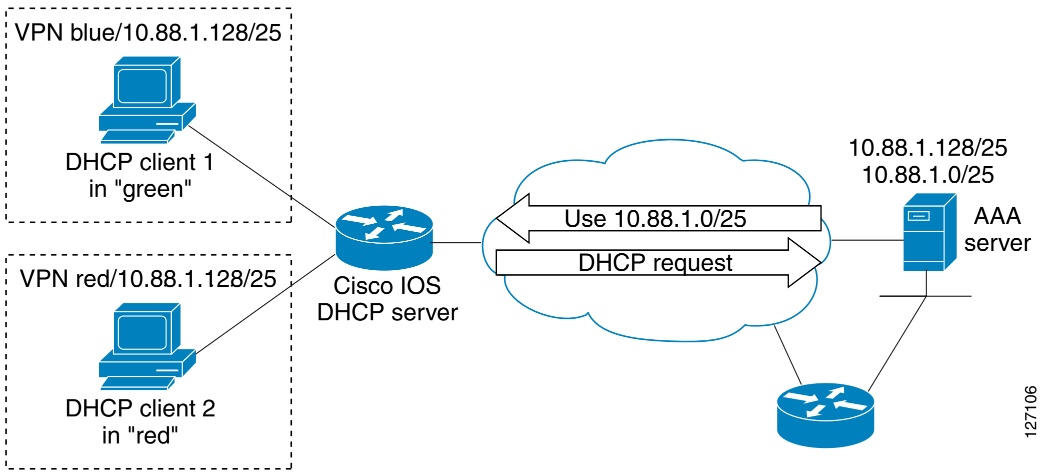
Subnet release from an ODAP follows a Last Leased subnet First (LLF) policy, which prefers the last leased subnet to be released first. This LLF policy searches for a releasable subnet (a subnet with no addresses currently being leased) starting with the last leased subnet. If a releasable subnet is found (candidate subnet), it is released, and the summarized route for that subnet is removed. If more than one releasable subnet exists at that time, only the most recently allocated is released. If there are no releasable subnets, no action is taken. If by releasing the candidate subnet, the high utilization mark is reached, the subnet is not released. The first leased subnet is never released (regardless of the instantaneous utilization level) until the ODAP is disabled.

When a DHCP pool receives multiple subnets from an upstream DHCP server, an address from each subnet is automatically configured on the client connected interface so that the addresses within the subnets can be requested by DHCP clients.

The first address in the first subnet is automatically assigned to the primary address on the interface. The first address of each subsequent subnet is assigned to secondary addresses on the interface. In addition, as client addresses are reclaimed, the count of lease addresses for that subnet is decremented. Once a lease counter for a subnet reaches zero (that is, lease expiry), the subnet is returned to the pool. The previous address on the interface is removed and the first secondary address on the interface is promoted as the primary address of the interface.

Figure 1 shows an ODAP manager configured on the Cisco IOS DHCP server. The ODAP requests an initial pool from the AAA server. Clients make DHCP requests and the DHCP server fulfills requests from the pool. When the utilization rate meets 90 percent, the ODAP manager requests an expansion and the AAA server allocates another subnet from which the ODAP manager can allocate addresses.

Figure 1 ODAP Address Pool Management for MPLS VPNs



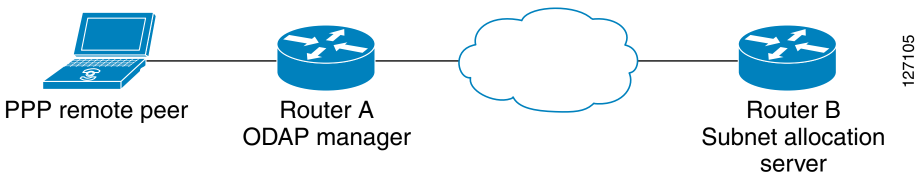
Subnet Allocation Server Operation

You can also configure the ODAP manager to allocate subnets instead of individual IP addresses.

This capability allows the network operator to configure a Cisco IOS router as a subnet allocation server. The operation of a subnet allocation server is similar to the operation of a DHCP server, except that pools of subnets are created and assigned instead of pools of IP addresses. Subnet allocation pools are created and configured by using the **subnet prefix-length** command in DHCP pool configuration mode. The size of each assigned or allocated subnet is set by the *prefix-length* argument, using standard Common InterDomain Routing (CIDR) bit count notation to determine the number of addresses that are configured in each subnet lease.

When a DHCP server is configured as a subnet allocation server, it provides subnet allocation pools for ODAP manager allocation. In Figure 2, Router B is the subnet allocation server and allocates subnets to the ODAP manager based on the demand for IP addresses and subnet availability. Router B is configured to allocate an initial amount of address space in the form of subnets to the ODAP manager. The size of the subnet allocated by the ODAP manager is determined by the subnet size that is configured on the subnet allocation server. The ODAP manager will then assign addresses to clients from these subnets and allocate more subnets as the need for address space increases.

Figure 2 Subnet Allocation Server Topology



When the ODAP manager allocates a subnet, the subnet allocation server creates a subnet binding. This binding is stored in the DHCP database for as long as the ODAP manager requires the address space. The binding is removed and the subnet is returned to the subnet pool only when the ODAP manager releases the subnet as address space utilization decreases.

The subnet allocation server can also be associated with a VRF. A VRF consists of an IP routing table, a derived CEF table, a set of interfaces that use the forwarding table, and a set of rules and routing protocol parameters that control the information that is included in the routing table.

Benefits of Using ODAPs

Efficient Address Management

The ODAP manager allows customers to optimize their use of IP addresses, thus conserving address space.

Efficient Route Summarization and Update

The ODAP manager inserts a summarized route when a subnet is added to the ODAP.

Multiple VRF and Independent Private Addressing Support

The ODAP manager automatically injects subnet routing information into the appropriate VRF.

How to Configure the DHCP Server On-Demand Address Pool Manager

This procedure contains the following tasks:

- Defining DHCP ODAPs as the Global Default Mechanism, page 6
- Defining DHCP ODAPs on an Interface, page 6
- Configuring the DHCP Pool as an ODAP, page 7
- Configuring ODAPs to Obtain Subnets Through IPCP Negotiation, page 9
- Configuring AAA, page 10
- Configuring RADIUS, page 12
- Disabling ODAPs, page 14
- Verifying ODAP Operation, page 14
- Monitoring and Maintaining the ODAP, page 17

Defining DHCP ODAPs as the Global Default Mechanism

Perform this task to specify that the global default mechanism to use is on-demand address pooling.

IP addressing allows configuration of a global default address pooling mechanism. The DHCP server needs to be able to distinguish between a normal DHCP address request and an address request for a PPP client.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip address-pool dhcp-pool**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`configure terminal` **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	`ip address-pool dhcp-pool` **Example:** `Router(config)# ip address-pool dhcp-pool`	Enables on-demand address pooling as the global default IP address mechanism. • For remote access (PPP) sessions into MPLS VPNs, IP addresses are obtained from locally configured VRF-associated DHCP pools.

Defining DHCP ODAPs on an Interface

Perform this task to configure on-demand address pools on an interface.

The interface on-demand address pooling configuration overrides the global default mechanism on that interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **peer default ip address dhcp-pool** [*pool-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface Virtual-Template1`	Specifies the interface and enters interface configuration mode.
Step 4	**peer default ip address dhcp-pool** [*pool-name*] **Example:** `Router(config)# peer default ip address dhcp-pool mypool`	Specifies an IP address from an on-demand address pool to be returned to a remote peer connecting to this interface. • The *pool-name* argument supports non-MPLS VPNs and is mandatory if the session is not associated with any VRF. Multiple pool names can be accepted but must be separated by white space.

Configuring the DHCP Pool as an ODAP

Perform this task to configure a DHCP address pool as an ODAP pool.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp pool** *pool-name*
4. **vrf** *name*
5. **origin** {**dhcp** | **aaa** | **ipcp**} [**subnet size initial** *size* [**autogrow** *size*]]
6. **utilization mark low** *percentage-number*
7. **utilization mark high** *percentage-number*
8. **end**
9. **show ip dhcp pool** [*pool-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip dhcp pool** *pool-name* **Example:** `Router(config)# ip dhcp pool red-pool`	Configures a DHCP address pool on a Cisco IOS DHCP server and enters DHCP pool configuration mode.
Step 4	**vrf** *name* **Example:** `Router(dhcp-config)# vrf red`	(Optional) Associates the address pool with a VRF name. • Only use this command for MPLS VPNs.
Step 5	**origin** {**dhcp** | **aaa** | **ipcp**} [**subnet size initial** *size* [**autogrow** *size*]] **Example:** `Router(dhcp-config)# origin dhcp subnet size initial /16 autogrow /16`	Configures an address pool as an on-demand address pool. • If you do not configure the pool as an autogrow pool, the pool will not request additional subnets if one subnet is already in the pool. • You can enter size as either the subnet mask (nnnn.nnnn.nnnn.nnnn) or prefix size (/nn). The valid values are /0 and /4 to /30. • When a DHCP pool receives multiple subnets from an upstream DHCP server, an address from each subnet is automatically configured on the client connected interface so that the addresses within the subnets can be requested by DHCP clients. The first address in the first subnet is automatically assigned to the primary address on the interface. The first address of each subsequent subnet is assigned to secondary addresses on the interface. In addition, as client addresses are reclaimed, the count of lease addresses for that subnet is decremented. Once a lease counter for a subnet reaches zero (that is, lease expiry), the subnet is returned to the pool. The previous address on the interface is removed and the first secondary address on the interface is promoted as the primary address of the interface. • If the **origin aaa** option is configured, AAA must be configured.
Step 6	**utilization mark low** *percentage-number* **Example:** `Router(dhcp-config)# utilization mark low 40`	Sets the low utilization mark of the pool size. • This command cannot be used unless the **autogrow** *size* option of the **origin** command is configured. • The default value is 0 percent.

	Command or Action	Purpose
Step 7	**`utilization mark high`** *`percentage-number`* **Example:** `Router(dhcp-config)# utilization mark high 60`	Sets the high utilization mark of the pool size. • This command cannot be used unless the **autogrow** *size* option of the **origin** command is configured. • The default value is 100 percent.
Step 8	**`end`** **Example:** `Router(dhcp-config)# end`	Returns to global configuration mode.
Step 9	**`show ip dhcp pool`** `[`*`pool-name`*`]` **Example:** `Router# show ip dhcp pool`	(Optional) Displays information about DHCP address pools. • Information about the primary and secondary interface address assignment is also displayed.

Configuring ODAPs to Obtain Subnets Through IPCP Negotiation

Perform this task to configure your router to use subnets obtained through IP Control Protocol (IPCP) negotiation.

You can assign IP address pools to customer premises equipment (CPE) devices, which, in turn, assign IP addresses to the CPE and to a DHCP pool. This functionality has three requirements:

- The Cisco IOS CPE device must be able to request and use the subnet.
- The RADIUS server (via AAA) must be able to provide that subnet and insert the framed route into the proper VRF table.
- The PE router must be able to facilitate providing the subnet through (IPCP) negotiation.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp pool** *pool-name*
4. **import all**
5. **origin ipcp**
6. **exit**
7. **interface** *type number*
8. **ip address pool** *pool-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip dhcp pool** *pool-name* **Example:** `Router(config)# ip dhcp pool red-pool`	Configures a DHCP address pool on a Cisco IOS DHCP server and enters DHCP pool configuration mode.
Step 4	**import all** **Example:** `Router(dhcp-config)# import all`	Imports option parameters into the Cisco IOS DHCP server database.
Step 5	**origin ipcp** **Example:** `Router(dhcp-config)# origin ipcp`	Configures an address pool as an on-demand address pool using IPCP as the subnet allocation protocol.
Step 6	**exit** **Example:** `Router(dhcp-config)# exit`	Exits DHCP pool configuration mode.
Step 7	**interface** *type number* **Example:** `Router(config)# interface ethernet 0`	Specifies the interface and enters interface configuration mode.
Step 8	**ip address pool** *pool-name* **Example:** `Router(config-if)# ip address pool red-pool`	Specifies that the interface IP address will be automatically configured from the named pool, when the pool is populated with a subnet from IPCP.

Configuring AAA

Perform this task to configure AAA.

To allow ODAP to obtain subnets from the AAA server, the AAA client must be configured on the VHG/PE router.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **aaa new-model**
4. **aaa authorization configuration default group radius**
5. **aaa accounting network default start-stop group radius**
   or
   aaa accounting network default stop-only group radius
6. **aaa session-id common**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Router(config)# aaa new-model`	Enables AAA access control.
Step 4	**aaa authorization configuration default group radius** **Example:** `Router(config)# aaa authorization configuration default group radius`	Downloads static route configuration information from the AAA server using RADIUS.

	Command or Action	Purpose
Step 5	**aaa accounting network default start-stop group radius** or **aaa accounting network default stop-only group radius** **Example:** `Router(config)# aaa accounting network default start-stop group radius` or **Example:** `Router(config)# aaa accounting network default stop-only group radius`	Enables AAA accounting of requested services for billing or security purposes when you use RADIUS. Sends a "start" accounting notice at the beginning of a process. or Enables AAA accounting of requested services for billing or security purposes when you use RADIUS. Sends a "stop" accounting notice at the end of the requested user process.
Step 6	**aaa session-id common** **Example:** `Router(config)# aaa session-id common`	Ensures that the same session ID will be used for each AAA accounting service type within a call.

Configuring RADIUS

Perform this task to configure RADIUS.

ODAP AAA Profile

The AAA server sends the RADIUS Cisco AV pair attributes "pool-addr" and "pool-mask" to the Cisco IOS DHCP server in the access request and access accept. The pool-addr attribute is the IP address and the pool-mask attribute is the network mask (for example, **pool-addr=192.168.1.0** and **pool-mask=255.255.0.0**). Together, these attributes make up a network address (address/mask) that is allocated by the AAA server to the Cisco IOS DHCP server.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip radius source-interface** *subinterface-name*
4. **radius-server host** *ip-address* **auth-port** *port-number* **acct-port** *port-number*
5. **radius server attribute 32 include-in-access-req**
6. **radius server attribute 44 include-in-access-req**
7. **radius-server vsa send accounting**
8. **radius-server vsa send authentication**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip radius source-interface** *subinterface-name* **Example:** `Router(config)# ip radius source-interface Ethernet1/1`	Forces RADIUS to use the IP address of a specified interface for all outgoing RADIUS packets.
Step 4	**radius-server host** *ip-address* **auth-port** *port-number* **acct-port** *port-number* **Example:** `Router(config)# radius-server host 172.16.1.1 auth-port 1645 acct-port 1646`	Specifies a RADIUS server host. • The *ip-address* argument specifies the IP address of the RADIUS server host.
Step 5	**radius server attribute 32 include-in-access-req** **Example:** `Router(config)# radius server attribute 32 include-in-access-req`	Sends RADIUS attribute 32 (NAS-Identifier) in an access request or accounting request.
Step 6	**radius server attribute 44 include-in-access-req** **Example:** `Router(config)# radius server attribute 44 include-in-access-req`	Sends RADIUS attribute 44 (Accounting Session ID) in an access request or accounting request.
Step 7	**radius-server vsa send accounting** **Example:** `Router(config)# radius-server vsa send accounting`	Configures the network access server (NAS) to recognize and use vendor-specific accounting attributes.
Step 8	Router(config)# **radius-server vsa send authentication** **Example:** `Router(config)# radius-server vsa send authentication`	Configures the NAS to recognize and use vendor-specific authentication attributes.

Disabling ODAPs

This task shows how to disable an ODAP from a DHCP pool.

When an ODAP is disabled, all leased subnets are released. If active PPP sessions are using addresses from the released subnets, those sessions will be reset. DHCP clients leasing addresses from the released subnets will not be able to renew their leases.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp pool** *pool-name*
4. **no origin** {**dhcp** | **aaa** | **ipcp**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`configure terminal` **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	`ip dhcp pool` *pool-name* **Example:** `Router(config)# ip dhcp pool red-pool`	Configures a DHCP address pool on a Cisco IOS DHCP server and enters DHCP pool configuration mode.
Step 4	`no origin {dhcp | aaa | ipcp}` **Example:** `Router(dhcp-config)# no origin dhcp`	Disables the ODAP.

Verifying ODAP Operation

Perform this task to verify ODAP operation.

SUMMARY STEPS

1. **enable**
2. **show ip dhcp pool** [*pool-name*]
3. **show ip dhcp binding**

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode. Enter your password if prompted.

```
Router> enable
```

Step 2 **show ip dhcp pool** [*pool-name*]

The following output is for two DHCP pools: Green and Global. Pool Green is configured with a high utilization mark of 50 and a low utilization mark of 30. The pool is also configured to obtain more subnets when the high utilization mark is reached (autogrow). The Subnet size field indicates the values configured in the **origin** command as the initial and incremental subnet sizes that would be requested by the pool named Green. The Total addresses field is a count of all the usable addresses in the pool. The Leased addresses field is a total count of how many bindings were created from the pool. The Pending event field shows subnet request, which means that a subnet request is pending for the pool. The subnet request was scheduled because the Leased addresses count has exceeded the high utilization level of the pool. Subnets currently added to pool Green are shown in sequence. The Current index column shows the address that would be allocated next from this subnet. The IP address range column shows the range of usable addresses from the subnet. The Leased addresses column shows individual count of bindings created from each subnet. Three subnets are currently added to pool Green. The first two subnets have used all their addresses and thus the Current index is showing 0.0.0.0.

Notice that pool Green and pool Global can have the same subnet (172.16.0.1-172.16.0.6) because pool Green is configured to be in VRF Green, while pool Global is configured to be in the global address space.

```
Router# show ip dhcp pool

Pool Green :
 Utilization mark (high/low)    : 50 / 30
 Subnet size (first/next)       : 24 / 24 (autogrow)
 VRF name                       : Green
 Total addresses                : 18
 Leased addresses               : 13
 Pending event                  : subnet request
 3 subnets are currently in the pool :
 Current index        IP address range                    Leased addresses
 0.0.0.0              172.16.0.1       - 172.16.0.6        6
 0.0.0.0              172.16.0.9       - 172.16.0.14       6
 172.16.0.18          172.16.0.17      - 172.16.0.22       1

Pool Global :
 Utilization mark (high/low)    : 100 / 0
 Subnet size (first/next)       : 24 / 24 (autogrow)
 Total addresses                : 6
 Leased addresses               : 0
 Pending event                  : none
 1 subnet is currently in the pool :
 Current index        IP address range                    Leased addresses
 172.16.0.1           172.16.0.1       - 172.16.0.6        0
```

Step 3 **show ip dhcp binding**

The following output shows the bindings from pool Green. The Type field shows On-demand, which indicates that the address binding was created for a PPP session. The Lease expiration field shows

Infinite, which means that the binding is valid as long as the session is up. If a subnet must be released back to the leasing server while the session is still up, the session is reset so that it will be forced to obtain a new IP address. The Hardware address column for an On-demand entry shows the identifier for the session as detected by PPP. There are no bindings shown under the Bindings from all pools not associated with VRF field because the Global pool has not allocated any addresses.

```
Router# show ip dhcp binding

Bindings from all pools not associated with VRF:
IP address       Hardware address        Lease expiration        Type

Bindings from VRF pool Green:
IP address       Hardware address        Lease expiration        Type
172.16.0.1       5674.312d.7465.7374.    Infinite                On-demand
                 2d38.3930.39
172.16.0.2       5674.312d.7465.7374.    Infinite                On-demand
                 2d38.3839.31
172.16.0.3       5674.312d.7465.7374.    Infinite                On-demand
                 2d36.3432.34
172.16.0.4       5674.312d.7465.7374.    Infinite                On-demand
                 2d38.3236.34
172.16.0.5       5674.312d.7465.7374.    Infinite                On-demand
                 2d34.3331.37
172.16.0.6       5674.312d.7465.7374.    Infinite                On-demand
                 2d37.3237.39
172.16.0.9       5674.312d.7465.7374.    Infinite                On-demand
                 2d39.3732.36
172.16.0.10      5674.312d.7465.7374.    Infinite                On-demand
                 2d31.3637
172.16.0.11      5674.312d.7465.7374.    Infinite                On-demand
                 2d39.3137.36
172.16.0.12      5674.312d.7465.7374.    Infinite                On-demand
                 2d37.3838.30
172.16.0.13      5674.312d.7465.7374.    Infinite                On-demand
                 2d32.3339.37
172.16.0.14      5674.312d.7465.7374.    Infinite                On-demand
                 2d31.3038.31
172.16.0.17      5674.312d.7465.7374.    Infinite                On-demand
                 2d38.3832.38
172.16.0.18      5674.312d.7465.7374.    Infinite                On-demand
                 2d32.3735.31
```

Troubleshooting Tips

By default, the Cisco IOS DHCP server on which the ODAP manager is based attempts to verify an address availability by performing a ping operation to the address before allocation. The default DHCP ping configuration will wait for 2 seconds for an ICMP echo reply. This default configuration results in the DHCP server servicing one address request every 2 seconds. The number of ping packets being sent and the ping timeout are configurable. Thus, to reduce the address allocation time, you can reduce either the timeout or the number of ping packets sent. Reducing the timeout or the ping packets being sent will improve the address allocation time, at the cost of less ability to detect duplicate addresses.

Each ODAP will make a finite number of attempts (up to four retries) to obtain a subnet from DHCP or AAA. If these attempts are not successful, the subnet request from the pool automatically starts when there is another individual address request to the pool (for example, from a newly brought up PPP session). If a pool has not been allocated any subnets, you can force restarting the subnet request process by using the **clear ip dhcp pool** *pool-name* **subnet** * EXEC command.

Monitoring and Maintaining the ODAP

This task shows how to monitor and maintain the ODAP.

Note the following behavior for the **clear ip dhcp binding**, **clear ip dhcp conflict**, and **clear ip dhcp subnet** commands:

- If you do not specify the **pool** *pool-name* option and an IP address is specified, it is assumed that the IP address is an address in the global address space and will look among all the non-VRF DHCP pools for the specified binding/conflict/subnet.
- If you do not specify the **pool** *pool-name* option and the * option is specified, it is assumed that all automatic/ or on-demand bindings/conflicts/subnets in all VRF and non-VRF pools are to be deleted.
- If you specify both the **pool** *pool-name* option and the * option, all automatic or on-demand bindings/conflicts/subnets in the specified pool only will be cleared.
- If you specify the **pool** *pool-name* option and an IP address, the specified binding/conflict or the subnet containing the specified IP address will be deleted from the specified pool.

SUMMARY STEPS

1. **enable**
2. **clear ip dhcp** [**pool** *pool-name*] **binding** {* | *address*}
3. **clear ip dhcp** [**pool** *pool-name*] **conflict** {* | *address*}
4. **clear ip dhcp** [**pool** *pool-name*] **subnet** {* | *address*}
5. **debug dhcp details**
6. **debug ip dhcp server events**
7. **show ip dhcp import**
8. **show ip interface** [*type number*]
9. **show ip dhcp pool** *pool-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear ip dhcp** [**pool** *pool-name*] **binding** {***** | *address*} **Example:** `Router# clear ip dhcp binding *`	Deletes an automatic address binding or objects from a specific pool from the DHCP server database.
Step 3	**clear ip dhcp** [**pool** *pool-name*] **conflict** {***** | *address*} **Example:** `Router# clear ip dhcp conflict *`	Clears an address conflict or conflicts from a specific pool from the DHCP server database.
Step 4	**clear ip dhcp** [**pool** *pool-name*] **subnet** {***** | *address*} **Example:** `Router# clear ip dhcp subnet *`	Clears all currently leased subnets in the named DHCP pool or all DHCP pools if *name* is not specified.
Step 5	**debug dhcp details** **Example:** `Router# debug dhcp details`	Monitors the subnet allocation/releasing in the on-demand address pools.
Step 6	**debug ip dhcp server events** **Example:** `Router# debug ip dhcp server events`	Reports DHCP server events, like address assignments and database updates.
Step 7	**show ip dhcp import** **Example:** `Router# show ip dhcp import`	Displays the option parameters that were imported into the DHCP server database.
Step 8	**show ip interface** [*type number*] **Example:** `Router# show ip interface`	Displays the usability status of interfaces configured for IP.
Step 9	**show ip dhcp pool** *pool-name* **Example:** `Router# show ip dhcp pool green`	Displays DHCP address pool information.

How to Configure DHCP ODAP Subnet Allocation Server Support

This procedure contains the following tasks:

- Configuring a Global Pool on a Subnet Allocation Server, page 19 (required)
- Configuring a VRF Subnet Pool on a Subnet Allocation Server, page 20 (optional)
- Using a VPN ID to Configure a VRF Subnet Pool on a Subnet Allocation Server, page 22 (optional)
- Verifying the Subnet Allocation and DHCP Bindings, page 24 (optional)
- Troubleshooting the DHCP ODAP Subnet Allocation Server, page 25 (optional)

Configuring a Global Pool on a Subnet Allocation Server

Perform this task to configure a global subnet pool on a subnet allocation server.

Global Subnet Pools

Global subnet pools are created in a centralized network. The ODAP manager allocates subnets from the subnet allocation server based on subnet availability. When the ODAP manager allocates a subnet, the subnet allocation server creates a subnet binding. This binding is stored in the DHCP database for as long as the ODAP manager requires the address space. The binding is destroyed and the subnet is returned to the subnet pool only when the ODAP manager releases the subnet as address space utilization decreases.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp pool** *pool-name*
4. **network** *network-number [mask | /prefix-length]*
5. **subnet prefix-length** *prefix-length*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip dhcp pool** *pool-name* **Example:** `Router(config)# ip dhcp pool GLOBAL-POOL`	Enters DHCP pool configuration mode and specifies the subnet pool name.
Step 4	**network** *network-number* [*mask* | **/***prefix-length*] **Example:** `Router(dhcp-config)# network 10.0.0.0 255.255.255.0`	Configures the subnet number and mask for a DHCP address pool on a Cisco IOS DHCP server. • The subnet mask or the prefix length can be configured in this step. The values that can be configured for the *prefix-length* argument follow CIDR bit count notation. The forward slash character must be used when configuring the *prefix-length* argument.
Step 5	**subnet prefix-length** *prefix-length* **Example:** `Router(dhcp-config)# subnet prefix-length 8`	Configures the subnet prefix length. The range of the *prefix-length* argument is from 1 to 31. • This command configures the number of IP addresses that each subnet is configured to allocate from the subnet pool. The values that can be configured for the *prefix-length* argument follow CIDR bit count notation format.

Configuring a VRF Subnet Pool on a Subnet Allocation Server

This task shows how to configure a VRF subnet pool on a subnet allocation server.

VRF Subnet Pools

A subnet allocation server can be configured to assign subnets from VRF subnet allocation pools for MPLS VPN clients. VPN routes between the ODAP manager and the subnet allocation server are configured based on VRF name or VPN ID configuration. The VRF and VPN ID are configured to maintain routing information that defines customer VPN sites. The VPN customer site (or Customer Equipment [CE]) is attached to a provider edge (PE) router. The VRF is used to specify the VPN and consists of an IP routing table, a derived Cisco Express Forwarding (CEF) table, a set of interfaces that use the forwarding table, and a set of rules and routing protocol parameters that control the information that is included in the routing table.

Prerequisites

The VRF name and VPN ID can be configured on the ODAP manager and subnet allocation server prior to the configuration of the subnet allocation pool.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp pool** *pool-name*
4. ***vrf*** *vrf-name*
5. **network** *network-number* [*mask* | /*prefix-length*]
6. **subnet prefix-length** *prefix-length*

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`configure terminal` **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	`ip dhcp pool` *pool-name* **Example:** `Router(config)# ip dhcp pool VRF-POOL`	Enters DHCP pool configuration mode and specifies the subnet pool name.
Step 4	`vrf` *vrf-name* **Example:** `Router(dhcp-config)# vrf RED`	Associates the on-demand address pool with a VPN routing and forwarding (VRF) instance name (or tag). • The **vrf** keyword and *vrf-name* argument are used to specify the VPN for the VRF pool. The *vrf-name* argument must match the VRF name (or tag) that is configured for the client.

	Command or Action	Purpose
Step 5	**network** *network-number* [*mask* | */prefix-length*] **Example:** `Router(dhcp-config)# network 10.1.1.0 /24`	Configures the subnet number and mask for a DHCP address pool on a Cisco IOS DHCP server. • The subnet mask or the prefix length can be configured in this step. The values that can be configured for the *prefix-length* argument follow CIDR bit count notation. The forward slash character must be used when configuring the *prefix-length* argument.
Step 6	**subnet prefix-length** *prefix-length* **Example:** `Router(dhcp-config)# subnet prefix-length 16`	Configures the subnet prefix length. The range of the *prefix-length* argument is from 1 to 31. • This command configures the number of IP addresses that each subnet is configured to allocate from the subnet pool. The values that can be configured for the *prefix-length* argument follow CIDR bit count notation format.

Using a VPN ID to Configure a VRF Subnet Pool on a Subnet Allocation Server

Perform this task to configure a VRF subnet pool, using a VPN ID, on a subnet allocation server.

VRF Pools and VPN IDs

A subnet allocation server can also be configured to assign subnets from VPN subnet allocation pools based on the VPN ID of a client. The VPN ID (or Organizational Unique Identifier [OUI]) is a unique identifier assigned by the IEEE.

Prerequisites

The VRF name and VPN ID can be configured on the ODAP manager and subnet allocation server prior to the configuration of the subnet allocation pool.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf** ***vrf-name***
4. **rd** *route-distinguisher*
5. **route-target both** *route-target-number*
6. **vpn id** *vpn-id*
7. **exit**
8. **ip dhcp pool** *pool-name*
9. **vrf** *vrf-name*
10. **network** *network-number* [*mask* | */prefix-length*]
11. **subnet prefix-length** *prefix-length*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip vrf** ***vrf-name*** **Example:** `Router(config)#ip vrf RED`	Creates a VRF routing table and specifies the VRF name (or tag). • The *vrf-name* argument must match the VRF name that is configured for the client and VRF pool in Step 9.
Step 4	**rd** *route-distinguisher* **Example:** `Router(config-vrf)# rd 100:1`	Creates routing and forwarding tables for a VRF instance created in Step 3. • There are two formats for configuring the route distinguisher argument. It can be configured in the as-number:network number (ASN:nn) format, as shown in the example, or it can be configured in the IP address:network number format (IP-address:nn).
Step 5	**route-target both** *route-target-number* **Example:** `Router(config-vrf)# route-target both 100:1`	Creates a route-target extended community for the VRF instance that was created in Step 3. • The **both** keyword is used to specify which routes should be imported and exported to the target VPN extended community (or the ODAP manager in this configuration). • The *route-target-number* argument follows the same format as the *route-distinguisher* argument in Step 4. These two arguments must match.
Step 6	**vpn id** *vpn-id* **Example:** `Router(config-vrf)# vpn id 1234:123456`	Configures the VPN ID. • This command is only used if the client (ODAP manager) is also configured with or assigned a VPN ID.
Step 7	**exit** **Example:** `Router(config-vrf)# exit`	Exits VRF configuration mode and enters global configuration mode.
Step 8	**ip dhcp pool** *pool-name* **Example:** `Router(config)# ip dhcp pool VPN-POOL`	Enters DHCP pool configuration mode and specifies the subnet pool name. • The **VRF** keyword and *vrf-name* argument are used to specify the VPN for the VRF pool. The *vrf-name* argument must match the VRF name (or tag) that is configured for the client.

	Command or Action	Purpose
Step 9	**vrf** *vrf-name* **Example:** `Router(dhcp-config)#vrf RED`	Associates the on-demand address pool with a VRF instance name. • The *vrf-name* argument must match the *vrf-name* argument that was configured in Step 3.
Step 10	**network** *network-number* [*mask* | **/***prefix-length*] **Example:** `Router(dhcp-config)# network 192.168.0.0 /24`	Configures the subnet number and mask for a DHCP address pool on a Cisco IOS DHCP server. • The subnet mask or the prefix length can be configured in this step. The values that can be configured for the *prefix-length* argument follow CIDR bit count notation. The forward slash character must be used when configuring the *prefix-length* argument.
Step 11	**subnet prefix-length** *prefix-length* **Example:** `Router(dhcp-config)# subnet prefix-length 16`	Configures the subnet prefix length. • The range of the *prefix-length* argument is from 1 to 31. • This command configures the number of IP addresses that each subnet is configured to allocate from the subnet pool. The values that can be configured for the *prefix-length* argument follow CIDR bit count notation format.

Verifying the Subnet Allocation and DHCP Bindings

Perform this task to verify subnet allocation and DHCP bindings.

The **show ip dhcp pool** and **show ip dhcp binding** commands do not need to be issued together or even in the same session as there are differences in the information that is provided. These commands, however, can be used to display and verify subnet allocation and DHCP bindings. The **show running-config | begin dhcp** command is used to display the local configuration of DHCP and the configuration of the **subnet prefix-length** command.

SUMMARY STEPS

1. **enable**
2. **show running-config | begin dhcp**
3. **show ip dhcp pool**
4. **show ip dhcp binding**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show running-config** | begin dhcp **Example:** `Router# show running-config | begin dhcp`	Used to display the local configuration of the router. • The configuration of the **subnet prefix-length** command will be displayed under the DHCP pools, for which subnet lease allocation has been configured. The subnet allocation size will be shown, following this command, in CIDR bit count notation. • The sample output is filtered with the **begin** keyword to start displaying output at the DHCP section of the running configuration.
Step 3	**show ip dhcp pool** [*pool-name*] **Example:** `Router# show ip dhcp pool`	Displays information about DHCP pools. • This command can be used to verify subnet allocation pool configuration on both the subnet allocation server and the ODAP manager. • The output of this command displays specific address pool information, including the name of the pool, utilization of address space, subnet size, number of total addresses, number of leased address, and pending events.
Step 4	**show ip dhcp binding** [*ip-address*] **Example:** `Router# show ip dhcp binding`	Displays information about DHCP bindings. • This command can be used to display subnet allocation to DHCP binding mapping information. • The output from this command displays binding information for individual IP address assignment and allocated subnets. The output that is generated for DHCP IP address assignment and subnet allocation is almost identical, except that subnet leases display an IP address followed by the subnet mask (which shows the size of the allocated subnet). Bindings for individual IP address only display an IP address and are not followed by a subnet mask.

Troubleshooting the DHCP ODAP Subnet Allocation Server

Perform this task to troubleshoot the DHCP ODAP subnet allocation server.

SUMMARY STEPS

1. **enable**
2. **debug dhcp** [**detail**]

3. **debug ip dhcp server**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug dhcp** [**detail**] **Example:** `Router# debug dhcp detail`	Displays debugging information about DHCP client activities and monitors the status of DHCP packets. • This example is issued with the **detail** keyword on the ODAP manager. The **detail** keyword is used to display and monitor the lease entry structure of the client and the state transitions of lease entries. This command also displays the values of the op, htype, hlen, hops, server identifier option, xid, secs, flags, ciaddr, yiaddr, siaddr, and giaddr fields of the DHCP packet that are shown in addition to the length of the options field.
Step 3	**debug ip dhcp server** {**events** | **packets** | **linkage**} **Example:** `Router# debug ip dhcp server packets` `Router# debug ip dhcp server events`	Enables DHCP server debugging. • This example is issued with the **packets** and **events** keywords on the subnet allocation server. The output displays lease transition and reception, as well as database information.

Configuration Examples for DHCP Server On-Demand Address Pool Manager

This section provides the following configuration examples:

Defining DHCP ODAPs as the Global Default Mechanism: Example

The following example shows how to configure the on-demand address pooling mechanism to be used to serve an address request from a PPP client.

```
ip address-pool dhcp-pool
!
ip dhcp pool Green-pool
```

Defining DHCP ODAPs on an Interface: Example

The following example shows how to configure an interface to retrieve an IP address from an on-demand address pool:

```
interface Virtual-Template1
 ip vrf forwarding green
 ip unnumbered loopback1
 ppp authentication chap
 peer default ip address dhcp-pool
!
```

Configuring the DHCP Pool as an ODAP: Example

The following example shows two ODAPs configured to obtain their subnets from an external DHCP server:

```
Router# show run

Building configuration...

Current configuration : 3943 bytes
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Router
!
no logging console
enable password lab
!
username vpn_green_net1 password 0 lab
username vpn_red_net1 password 0 lab
ip subnet-zero
!
ip dhcp pool green_pool
   vrf Green
   utilization mark high 60
   utilization mark low 40
   origin dhcp subnet size initial /24 autogrow /24
!
ip dhcp pool red_pool
   vrf Red
   origin dhcp
!
ip vrf Green
 rd 200:1
 route-target export 200:1
```

```
 route-target import 200:1
!
ip vrf Red
 rd 300:1
 route-target export 300:1
 route-target import 300:1
ip cef
ip address-pool dhcp-pool
!
no voice hpi capture buffer
no voice hpi capture destination
!
interface Loopback0
 ip address 1.1.1.1 255.255.255.255
!
interface Loopback1
 ip vrf forwarding Green
 ip address 100.10.10.1 255.255.255.255
!
interface Loopback2
 ip vrf forwarding Red
 ip address 110.10.10.1 255.255.255.255
!
interface ATM2/0
 no ip address
 shutdown
 no atm ilmi-keepalive
!
interface ATM3/0
 no ip address
 no atm ilmi-keepalive
!
interface Ethernet4/0
 ip address 10.0.105.12 255.255.255.224
 duplex half
!
interface Ethernet4/1
 ip address 150.10.10.1 255.255.255.0
 duplex half
!
interface Ethernet4/2
 ip address 120.10.10.1 255.255.255.0
 duplex half
 tag-switching ip
!
interface Virtual-Template1
 ip vrf forwarding Green
 ip unnumbered Loopback1
 ppp authentication chap
!
interface Virtual-Template2
 ip vrf forwarding Green
 ip unnumbered Loopback1
 ppp authentication chap
!
interface Virtual-Template3
 ip vrf forwarding Green
 ip unnumbered Loopback1
 ppp authentication chap
!
interface Virtual-Template4
 ip vrf forwarding Red
 ip unnumbered Loopback2
 ppp authentication chap
```

```
!
interface Virtual-Template5
 ip vrf forwarding Red
 ip unnumbered Loopback2
 ppp authentication chap
!
interface Virtual-Template6
 ip vrf forwarding Red
 ip unnumbered Loopback2
 ppp authentication chap
!
router ospf 100
 log-adjacency-changes
 redistribute connected
 network 1.1.1.1 0.0.0.0 area 0
 network 120.10.10.0 0.0.0.255 area 0
 network 150.10.10.0 0.0.0.255 area 0
!
router bgp 100
 no synchronization
 bgp log-neighbor-changes
 neighbor 3.3.3.3 remote-as 100
 neighbor 3.3.3.3 update-source Loopback0
 !
 address-family ipv4 vrf Red
 redistribute connected
 redistribute static
 no auto-summary
 no synchronization
 network 110.0.0.0
 exit-address-family
 !
 address-family ipv4 vrf Green
 redistribute connected
 redistribute static
 no auto-summary
 no synchronization
 network 100.0.0.0
 exit-address-family
 !
 address-family vpnv4
 neighbor 3.3.3.3 activate
 neighbor 3.3.3.3 send-community extended
 exit-address-family
!
ip classless
ip route 172.19.0.0 255.255.0.0 10.0.105.1
no ip http server
ip pim bidir-enable
!
call rsvp-sync
!
mgcp profile default
!
dial-peer cor custom
!
gatekeeper
 shutdown
!
line con 0
 exec-timeout 0 0
line aux 0
line vty 0 4
 password lab
```

```
 login
!
end
```

Configuring the DHCP Pool as an ODAP for Non-MPLS VPNs: Example

The following example shows how to configure an interface to retrieve an IP address from an on-demand address pool. In this example, two non-VRF ODAPs are configured. There are two virtual-templates and two DHCP address pools, usergroup1 and usergroup2. Each virtual-template interface is configured to obtain IP addresses for the peer from the associated address pool.

```
!
ip dhcp pool usergroup1
 origin dhcp subnet size initial /24 autogrow /24
 lease 0 1
!
ip dhcp pool usergroup2
 origin dhcp subnet size initial /24 autogrow /24
 lease 0 1
!
interface virtual-template1
 ip unnumbered loopback1
 peer default ip address dhcp-pool usergroup1
!
interface virtual-template2
 ip unnumbered loopback1
 peer default ip address dhcp-pool usergroup2
```

IPCP Subnet Mask Delivery: Example

The following example shows a Cisco 827 router configured to use IPCP subnet masks:

```
Router# show run

 Building configuration...

 Current configuration :1479 bytes
 !
 version 12.2
 no service single-slot-reload-enable
 no service pad
 service timestamps debug datetime msec
 service timestamps log uptime
 no service password-encryption
 !
 hostname Router
 !
 no logging buffered
 logging rate-limit console 10 except errors
 !
 username 6400-nrp2 password 0 lab
 ip subnet-zero
 ip dhcp smart-relay
 !
 ip dhcp pool IPPOOLTEST
    import all
    origin ipcp
 !
 no ip dhcp-client network-discovery
```

```
!
interface Ethernet0
 ip address pool IPPOOLTEST
 ip verify unicast reverse-path
 hold-queue 32 in
!
interface ATM0
 no ip address
 atm ilmi-keepalive
 bundle-enable
 dsl operating-mode auto
 hold-queue 224 in
!
interface ATM0.1 point-to-point
 pvc 1/40
  no ilmi manage
  encapsulation aal5mux ppp dialer
  dialer pool-member 1
 !
!
interface Dialer0
 ip unnumbered Ethernet0
 ip verify unicast reverse-path
 encapsulation ppp
 dialer pool 1
 dialer-group 1
 no cdp enable
 ppp authentication chap callin
 ppp chap hostname Router
 ppp chap password 7 12150415
 ppp ipcp accept-address
 ppp ipcp dns request
 ppp ipcp wins request
 ppp ipcp mask request
!
ip classless
ip route 0.0.0.0 0.0.0.0 Dialer0
no ip http server
!
dialer-list 1 protocol ip permit
line con 0
 exec-timeout 0 0
 transport input none
 stopbits 1
line vty 0 4
 login
!
scheduler max-task-time 5000
end
```

Configuring AAA and RADIUS: Example

The following example shows one pool “Green” configured to obtain its subnets from the AAA (RADIUS) server located at IP address 172.16.1.1:

```
!
aaa new-model
!
aaa authorization configuration default group radius
aaa accounting network default start-stop group radius
```

```
aaa session-id common
!
ip subnet-zero
!
ip dhcp ping packets 0
!
ip dhcp pool Green
   vrf Green
   utilization mark high 50
   utilization mark low 30
   origin aaa subnet size initial /28 autogrow /28
!
ip vrf Green
 rd 300:1
 route-target export 300:1
 route-target import 300:1
!
interface Ethernet1/1
 ip address 172.16.1.12 255.255.255.0
 duplex half
!
interface Virtual-Template1
 ip vrf forwarding Green
 no ip address
!
ip radius source-interface Ethernet1/1
!
!IP address of the RADIUS server host
radius-server host 172.16.1.1 auth-port 1645 acct-port 1646
radius-server retransmit 3
radius-server attribute 32 include-in-access-req
radius-server attribute 44 include-in-access-req
radius-server key cisco
radius-server vsa send accounting
radius-server vsa send authentication
```

Configuring a Global Pool for a Subnet Allocation Server: Example

The following example shows how to configure a router to be a subnet allocation server and create a global subnet allocation pool named “GLOBAL-POOL” that allocates subnets from the 10.0.0.0/24 network. The configuration of the **subnet prefix-length** command in this example configures the size of each subnet that is allocated from the subnet pool to support 254 host IP addresses.

```
ip dhcp pool GLOBAL-POOL
 network 10.0.0.0 255.255.255.0
 subnet prefix-length 24
!
```

Configuring a VRF Pool for a Subnet Allocation Server: Example

The following example shows how to configure a router to be a subnet allocation server and create a VRF subnet allocation pool named “VRF-POOL” that allocates subnets from the 172.16.0.0/16 network and configures the VPN to match the VRF named “RED.” The configuration of the **subnet prefix-length** command in this example configures the size of each subnet that is allocated from the subnet pool to support 62 host IP addresses.

```
 ip dhcp pool VRF-POOL
 vrf RED
 network 172.16.0.0 /16
```

```
 subnet prefix-length 26
!
```

Using a VPN ID to Configure a VRF Pool on a Subnet Allocation Server: Example

The following example shows how to configure a router to be a subnet allocation server and create a VRF subnet allocation pool named "VRF-POOL" that allocates subnets from the 192.168.0.0/24 network and configures the VRF named "RED." The VPN ID must match the unique identifier that is assigned to the client site. The route target and route distinguisher are configured in the as-number:network-number format. The route target and route distinguisher must match. The configuration of the **subnet prefix-length** command in this example configures the size of each subnet that is allocated from the subnet pool to support 30 host IP addresses.

```
ip vrf RED
 rd 100:1
 route-target both 100:1
 vpn id 1234:123456
 exit
 ip dhcp pool VPN-POOL
  vrf RED
  network 192.168.0.0 /24
  subnet prefix-length /27
  exit
```

Verifying Local Configuration on a Subnet Allocation Server: Example

The following example is output from the **show running-config** command. This command can be used to verify the local configuration on a subnet allocation server. The output from this command displays the configuration of the **subnet prefix-length** command under the DHCP pool named "GLOBAL-POOL." The total size of the subnet allocation pool is set to 254 addresses with the **network** command. The configuration of the **subnet prefix-length** command configures this pool to allocate a subnet that will support 254 host IP addresses. Because the total pool size supports only 254 addresses, only one subnet can be allocated from this pool.

```
Router# show running-config | begin dhcp
ip dhcp pool GLOBAL-POOL
   network 10.0.0.0 255.255.255.0
   subnet prefix-length 24
!
```

Verifying Address Pool Allocation Information: Example

The following examples are output from the **show ip dhcp pool** command. This command can be used to verify subnet allocation pool configuration on the subnet allocation server and the ODAP manager. The output from this command displays information about the address pool name, utilization level, configured subnet size, total number of addresses (from subnet), pending events, and specific subnet lease information.

The following sample output shows that the configured subnet allocation size is /24 (254 IP addresses), that there is a pending subnet allocation request, and there are no subnets in the pool:

```
Router> show ip dhcp pool ISP-1
Pool ISP-1 :
 Utilization mark (high/low)    :100 / 0
 Subnet size (first/next)       :24 / 24 (autogrow)
```

```
 Total addresses                :0
 Leased addresses               :0
 Pending event                  :subnet request
 0 subnet is currently in the pool
```

The next example shows that the configured subnet allocation size is /24 (254 IP address), the configured VRF name is “RED”, and a subnet containing 254 IP addresses has been allocated but no IP addresses have been leased from the subnet:

```
Router> show ip dhcp pool SUBNET-ALLOC
Pool SUBNET-ALLOC :
 Utilization mark (high/low)    :100 / 0
 Subnet size (first/next)       :24 / 24 (autogrow)
 VRF name                       :RED
 Total addresses                :254
 Leased addresses               :0
 Pending event                  :none
 1 subnet is currently in the pool :
 Current index        IP address range                    Leased addresses
 10.0.0.1              10.0.0.1          - 10.0.0.254        0
```

Verifying Subnet Allocation and DHCP Bindings: Example

The following example is from the **show ip dhcp binding** command. This command can be used to display subnet allocation to DHCP binding mapping information. The output of this command shows the subnet lease to MAC address mapping, the lease expiration, and the lease type (subnet lease bindings are configured to be automatically created and released by default). The output that is generated for DHCP IP address assignment and subnet allocation is almost identical, except that subnet leases display an IP address followed by the subnet mask (which shows the size of the allocated subnet) in CIDR bit count notation. Bindings for individual IP address only display an IP address and are not followed by a subnet mask.

```
Router# show ip dhcp binding
Bindings from all pools not associated with VRF:
IP address          Client-ID/              Lease expiration        Type
                    Hardware address/
                    User name
10.0.0.0/26         0063.6973.636f.2d64.    Mar 29 2003 04:36 AM    Automatic
                    656d.6574.6572.2d47.
                    4c4f.4241.4c
```

Additional References

The following sections provide references related to configuring the DHCP ODAP manager.

Related Documents

Related Topic	Document Title
DHCP commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Addressing Services Command Reference*
DHCP conceptual information	“DHCP Overview” module
DHCP server configuration	“Configuring the Cisco IOS DHCP Server” module

Related Topic	Document Title
DHCP client configuration	"Configuring the Cisco IOS DHCP Client" module
DHCP relay agent configuration	"Configuring the Cisco IOS DHCP Relay Agent" module
DHCP advanced features	"Configuring DHCP Services for Accounting and Security" module
DHCP enhancements for edge-session management configuration	"Configuring DHCP Enhancements for Edge-Session Management" module
DHCP options	"DHCP Options" appendix in the *Network Registrar User's Guide*, Release 6.1.1

Standards

Standards	Title
No new or modified standards are supported by this functionality.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 951	*Bootstrap Protocol (BOOTP)*
RFC 1542	*Clarifications and Extensions for the Bootstrap Protocol*
RFC 2131	*Dynamic Host Configuration Protocol*
RFC 2685	*Virtual Private Networks Identifier*
RFC 3046	*DHCP Relay Information Option*

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Glossary

AAA—authentication, authorization, and accounting. Suite of network security services that provide the primary framework through which access control can be set up on your Cisco router or access server.

Cisco Access Registrar—A RADIUS server that supports service provider deployment of access services by centralizing AAA information and simplifying provisioning and management.

client—A host trying to configure its interface (obtain an IP address) using DHCP or BOOTP protocols.

DHCP—Dynamic Host Configuration Protocol.

incremental subnet size—The desired size of the second and subsequent subnets requested for an on-demand pool.

initial subnet size—The desired size of the first subnet requested for an on-demand pool.

IPCP—IP Control Protocol. Protocol that establishes and configures IP over PPP.

MPLS—Multiprotocol Label Switching. Emerging industry standard upon which tag switching is based.

ODAP—on-demand address pool.

PE router—provider edge router.

PPP—Point-to-Point Protocol.

RADIUS— Remote Authentication Dial-In User Service. Database for authenticating modem and ISDN connections and for tracking connection time.

relay agent—A router that forwards DHCP and BOOTP messages between a server and a client on different subnets.

releasable subnet—A leased subnet that has no address leased from it.

server—DHCP or BOOTP server.

VHG—Virtual Home Gateway. A Cisco IOS software component that terminates PPP sessions. It is owned and managed by the service provider on behalf of its customer to provide access to remote users of that customers network. A single service provider device (router) can host multiple VHGs of different customers. A VHG can be dynamically brought up and down based on the access pattern of the remote users. Note that there is no single IOS feature called the VHG; it is a collection of function and features.

VHG/PE router—A device that terminates PPP sessions and maps the remote users to the corresponding MPLS VPNs.

VPN—Virtual Private Network. Enables IP traffic to use tunneling to travel securely over a public TCP/IP network.

VPN information—In this document, VPN information refers to VRF name or VPN ID.

VRF—VPN routing and forwarding instance. A VRF consists of an IP routing table, a derived forwarding table, a set of interfaces that use the forwarding table, and a set of rules and routing protocols that determine what goes into the forwarding table. In general, a VRF includes the routing information that defines a customer VPN site that is attached to a PE router. Each VPN instantiated on the PE router has its own VRF.

Note See *Internetworking Terms and Acronyms* for terms not included in this glossary.

Feature Information for the DHCP Server On-Demand Address Pool Manager

Table 1 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Releases 12.2(1) or a later release appear in the table.

Not all commands may be available in your Cisco IOS software release. For details on when support for specific commands was introduced, see the command reference documents.

For information on a feature in this technology that is not documented here, see the "DHCP Features Roadmap".

Cisco IOS software images are specific to a Cisco IOS software release, a feature set, and a platform. Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for the DHCP On-Demand Address Pool Manager

Feature Name	Releases	Feature Configuration Information
DHCP Server On-Demand Address Pool Manager for Non-MPLS VPNs	12.2(15)T 12.2(28)SB 12.2(33)SRC	This feature was enhanced to provide ODAP support for non-MPLS VPNs. The following section provides information about this feature: • How to Configure the DHCP Server On-Demand Address Pool Manager The following command was modified by this feature: **peer default ip address**

Table 1 Feature Information for the DHCP On-Demand Address Pool Manager (continued)

Feature Name	Releases	Feature Configuration Information
DHCP ODAP Server Support	12.2(15)T 12.2(28)SB 12.2(33)SRC	This feature introduces the capability to configure a DHCP server (or router) as a subnet allocation server. This capability allows the Cisco IOS DHCP server to be configured with a pool of subnets for lease to ODAP clients. The following section provides information about this feature: • How to Configure DHCP ODAP Subnet Allocation Server Support The following commands were introduced or modified by this feature: **subnet prefix-length** and **show ip dhcp binding**
DHCP Server On-Demand Address Pool Manager	12.2(8)T 12.28(SB) 12.2(33)SRC	The ODAP manager is used to centralize the management of large pools of addresses and simplify the configuration of large networks. ODAP provides a central management point for the allocation and assignment of IP addresses. When a Cisco IOS router is configured as an ODAP manager, pools of IP addresses are dynamically increased or reduced in size depending on the address utilization level. The following section provides information about this feature: • How to Configure the DHCP Server On-Demand Address Pool Manager The following commands were introduced by this feature: **aaa session-id**, **clear ip dhcp subnet**, **ip address pool**, **ip dhcp aaa default username**, **origin**, **show ip dhcp pool**, **utilization mark high**, **utilization mark low**, **vrf**. The following commands were modified by this feature: **clear ip dhcp binding**, **clear ip dhcp conflict**, **ip address-pool**, **peer default ip address**.

Configuring the Cisco IOS DHCP Relay Agent

First Published: May 2, 2005
Last Updated: May 2, 2008

Cisco routers running Cisco IOS software include Dynamic Host Configuration Protocol (DHCP) server and relay agent software. A DHCP relay agent is any host that forwards DHCP packets between clients and servers. This module describes the concepts and tasks needed to configure the Cisco IOS DHCP relay agent.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. For the latest feature information and caveats, see the release notes for your Cisco IOS software release. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for the Cisco IOS DHCP Relay Agent" section on page 24.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Configuring the Cisco IOS DHCP Relay Agent

Before you configure the DHCP relay agent, you should understand the concepts documented in the "DHCP Overview" module.

The Cisco IOS DHCP server and relay agent are enabled by default. You can verify if they have been disabled by checking your configuration file. If they have been disabled, the **no service dhcp** command will appear in the configuration file. Use the **service dhcp** command to reenable the functionality if necessary.

The Cisco IOS DHCP relay agent will be enabled on an interface only when the **ip helper-address** is configured. This command enables the DHCP broadcast to be forwarded to the configured DHCP server.

Information About the DHCP Relay Agent

Before you configure the DHCP relay agent, you should understand the following concept:

- DHCP Relay Agent Overview, page 2

DHCP Relay Agent Overview

A DHCP relay agent is any host that forwards DHCP packets between clients and servers. Relay agents are used to forward requests and replies between clients and servers when they are not on the same physical subnet. Relay agent forwarding is distinct from the normal forwarding of an IP router, where IP datagrams are switched between networks somewhat transparently. By contrast, relay agents receive DHCP messages and then generate a new DHCP message to send out on another interface. The relay agent sets the gateway address (giaddr field of the DHCP packet) and, if configured, adds the relay agent information option (option82) in the packet and forwards it to the DHCP server. The reply from the server is forwarded back to the client after removing option 82.

The Cisco IOS DHCP relay agent supports the use of unnumbered interfaces. For DHCP clients connected though the unnumbered interfaces, the DHCP relay agent automatically adds a static host route once the DHCP client obtains an address, specifying the unnumbered interface as the outbound interface. The route is automatically removed once the lease time expires or when the client releases the address.

How to Configure the DHCP Relay Agent

This section contains the following tasks:

- Specifying the Packet Forwarding Address, page 3 (required)
- Configuring Relay Agent Information Option Support, page 4 (optional)
- Configuring Relay Agent Information Option Support per Interface, page 8 (optional)
- Configuring the Subscriber Identifier Suboption of the Relay Agent Information Option, page 11 (optional)
- Configuring DHCP Relay Class Support for Client Identification, page 12 (optional)
- Configuring DHCP Relay Agent Support for MPLS VPNs, page 15 (optional)

- Setting the Gateway Address of the DHCP Broadcast to a Secondary Address Using Smart Relay Agent Forwarding, page 19 (optional)
- Troubleshooting the DHCP Relay Agent, page 19 (optional)

Specifying the Packet Forwarding Address

Perform this task to configure the DHCP relay agent to forward packets to a DHCP server.

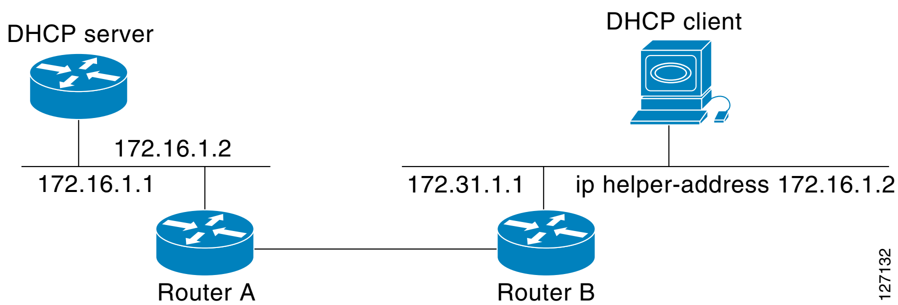
DHCP clients need to use User Datagram Protocol (UDP) broadcasts to send their initial DHCPDISCOVER messages because they don't have information about the network to which they are attached. If the client is on a network segment that does not include a server, UDP broadcasts normally are not forwarded because most routers are configured to not forward broadcast traffic.

You can remedy this situation by configuring the interface of your router that is receiving the broadcasts to forward certain classes of broadcasts to a helper address. You can use more than one helper address per interface.

When a router forwards these address assignment/parameter requests, it is acting as a DHCP relay agent. The Cisco router implementation of the DHCP relay agent is provided via the **ip helper-address** interface configuration command.

In Figure 1, the DHCP client broadcasts a request for an IP address and additional configuration parameters on its local LAN. Router B, acting as a DHCP relay agent, picks up the broadcast and generates a new DHCP message to send out on another interface. As part of this DHCP message, the relay agent inserts the IP address of the interface containing the **ip helper-address** command into the gateway IP address (giaddr) field of the DHCP packet. This IP address enables the DHCP server to determine which subnet should receive the offer and identify the appropriate IP address range to offer. The DHCP relay agent sends the local broadcast, via IP unicast, to the DHCP server address 172.16.1.2 specified by the **ip helper-address** interface configuration command.

Figure 1 Forwarding UDP Broadcasts to a DHCP Server Using a Helper Address



SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip helper-address** *address*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface FastEthernet0/0`	Configures an interface and enters interface configuration mode.
Step 4	**ip helper-address** *address* **Example:** `Router(config-if)# ip helper-address 172.16.1.2`	Forwards UPD broadcasts, including BOOTP and DHCP. • The *address* argument can be a specific DHCP server address, or it can be the network address if other DHCP servers are on the destination network segment. Using the network address enables other servers to respond to DHCP requests. • If you have multiple servers, you can configure one helper address for each server.

Configuring Relay Agent Information Option Support

Perform this task to enable support for the DHCP relay agent information option.

Relay Agent Information Option

Automatic DHCP address allocation is typically based on an IP address, whether it be the gateway IP address (giaddr field of the DHCP packet) or the incoming interface IP address. In some networks, it is necessary to use additional information to further determine which IP addresses to allocate. By using the relay agent information option (option 82), the Cisco IOS relay agent can include additional information about itself when forwarding client-originated DHCP packets to a DHCP server.

Cisco IOS supports this functionality by using the **ip dhcp relay information option** command. The relay agent will automatically add the circuit identifier suboption and the remote ID suboption to the relay agent information option and forward them to the DHCP server.

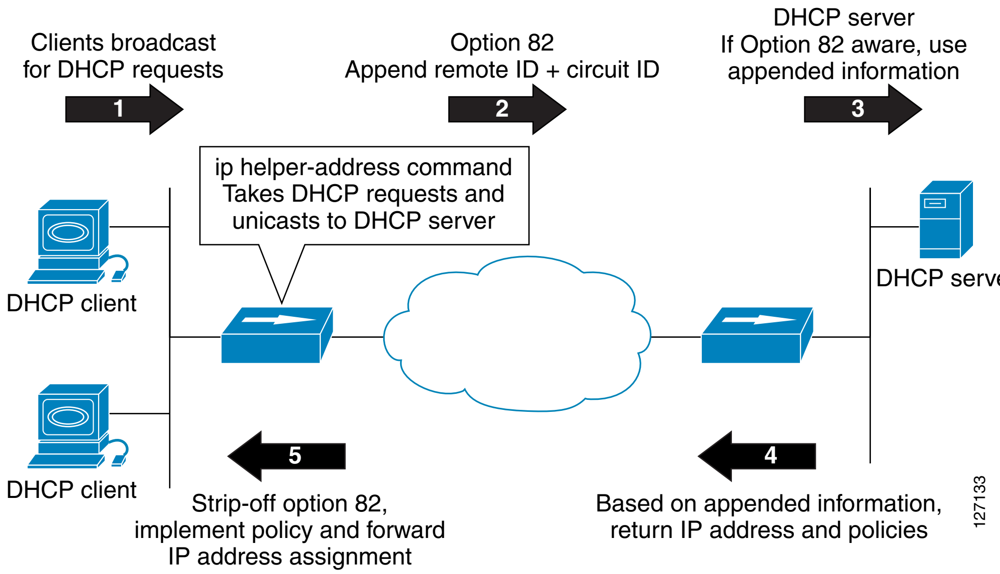
The DHCP server can use this information to assign IP addresses, perform access control, and set quality of service (QoS) and security policies (or other parameter-assignment policies) for each subscriber of a service provider network.

Figure 2 shows how the relay agent information option is inserted into the DHCP packet as follows:

1. The DHCP client generates a DHCP request and broadcasts it on the network.
2. The DHCP relay agent intercepts the broadcast DHCP request packet and inserts the relay agent information option (option 82) in the packet. The relay agent information option contains the related suboptions.

3. The DHCP relay agent unicasts the DHCP packet to the DHCP server.
4. The DHCP server receives the packet and uses the suboptions to assign IP addresses and other configuration parameters and forwards them back to the client.
5. The suboption fields are stripped off of the packet by the relay agent while forwarding to the client.

Figure 2 Relay Agent Information Option Operation



Relay Agent Information Reforwarding Policy

A DHCP relay agent may receive a message from another DHCP relay agent that already contains relay information. By default, the relay information from the previous relay agent is replaced. If this behavior is not suitable for your network, you can use the **ip dhcp relay information policy** {**drop** | **keep** | **replace**} global configuration command to change it.

To ensure the correct operation of the reforwarding policy, make sure to disable the relay agent information check by using the **no ip dhcp relay information check** global configuration command.

Prerequisites

It is important to understand how DHCP options work. See the "DHCP Overview" module for more information.

Restrictions

- If an **ip dhcp relay information** command is configured in global configuration mode but not configured in interface configuration mode, the global configuration is applied to all interfaces.
- If an **ip dhcp relay information** command is configured in both global configuration mode and interface configuration mode, the interface configuration command takes precedence over the global configuration command. However, the global configuration is applied to interfaces without the interface configuration.

- If an **ip dhcp relay information** command is not configured in global configuration mode but is configured in interface configuration mode, only the interface with the configuration option applied is affected. All other interfaces are not impacted by the configuration.

See the "Configuring Relay Agent Information Option Support per Interface" section for more information on per-interface support for the relay agent information option.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp relay information option**
4. **ip dhcp relay information check**
5. **ip dhcp relay information policy** {**drop** | **keep** | **replace**}
6. **ip dhcp relay information trust-all**
7. **end**
8. **show ip dhcp relay information trusted-sources**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip dhcp relay information option** **Example:** `Router(config)# ip dhcp relay information option`	Enables the system to insert the DHCP relay agent information option (option-82 field) in forwarded BOOTREQUEST messages to a DHCP server. • This function is disabled by default.
Step 4	**ip dhcp relay information check** **Example:** `Router(config)# ip dhcp relay information check`	(Optional) Configures DHCP to check that the relay agent information option in forwarded BOOTREPLY messages is valid. • By default, DHCP checks that the option-82 field in DHCP reply packets it receives from the DHCP server is valid. If an invalid message is received, the relay agent drops it. If a valid message is received, the relay agent removes the option-82 field and forwards the packet. Use the **ip dhcp relay information check** command to reenable this functionality if it has been disabled.
Step 5	**ip dhcp relay information policy** {**drop** | **keep** | **replace**} **Example:** `Router(config)# ip dhcp relay information policy replace`	(Optional) Configures the reforwarding policy for a DHCP relay agent (what a relay agent should do if a message already contains relay information). • See the "Relay Agent Information Reforwarding Policy" section for more information.

	Command or Action	Purpose
Step 6	**ip dhcp relay information trust-all** **Example:** `Router(config)# ip dhcp relay information trust-all`	(Optional) Configures all interfaces on a router as trusted sources of the DHCP relay information option. • By default, if the gateway address is set to all zeros in the DHCP packet and the relay agent information option is already present in the packet, the DHCP relay agent will discard the packet. Use the **ip dhcp relay information trust-all** command to override this behavior and accept the packets. • This command is useful if there is a switch in between the client and the relay agent that may insert option 82. Use this command to ensure that these packets do not get dropped. • You can configure an individual interface as a trusted source of the DHCP relay information option by using the **ip dhcp relay information trusted** interface configuration mode command.
Step 7	**end** **Example:** `Router(config)# end`	Returns to privileged EXEC mode.
Step 8	**show ip dhcp relay information trusted-sources** **Example:** `Router# show ip dhcp relay information trusted-sources`	(Optional) Displays all interfaces configured to be a trusted source for the DHCP relay information option.

Configuring Relay Agent Information Option Support per Interface

Perform this task to enable support for the DHCP relay agent information option (option 82) on a per interface basis.

The interface configuration allows the subscribers with different DHCP option 82 requirements on different interfaces to be reached from one Cisco router.

Prerequisites

It is important to understand how DHCP options work. See the “DHCP Overview” module for more information.

Read the “Relay Agent Information Option” and “Relay Agent Information Reforwarding Policy” sections to understand how DHCP processes the relay agent information option for global configurations.

Restrictions

- If an **ip dhcp relay information** command is configured in global configuration mode but not configured in interface configuration mode, the global configuration is applied to all interfaces.

- If an **ip dhcp relay information** command is configured in both global configuration mode and interface configuration mode, the interface configuration command takes precedence over the global configuration command. However, the global configuration is applied to interfaces without the interface configuration.
- If an **ip dhcp relay information** command is not configured in global configuration mode but is configured in interface configuration mode, only the interface with the configuration option applied is affected. All other interfaces are not impacted by the configuration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip dhcp relay information option-insert** [**none**]
5. **ip dhcp relay information check-reply** [**none**]
6. **ip dhcp relay information policy-action** {**drop** | **keep** | **replace**}
7. **exit**
8. Repeat Steps 3 through 7 to configure relay agent information option settings on different interfaces.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface FastEthernet0/0`	Configures an interface and enters interface configuration mode.
Step 4	**ip dhcp relay information option-insert** [**none**] **Example:** `Router(config-if)# ip dhcp relay information option-insert`	Enables the system to insert the DHCP relay agent information option (option-82 field) in forwarded BOOTREQUEST messages to a DHCP server. • This function is disabled by default. However, if support for the relay agent information option is configured in global configuration mode, but not in interface configuration mode, the interface inherits the global configuration. • The **ip dhcp relay information option-insert none** interface configuration command is saved in the running configuration. This command takes precedence over any global relay agent information configuration.
Step 5	**ip dhcp relay information check-reply** [**none**] **Example:** `Router(config-if)# ip dhcp relay information check-reply`	Configures a DHCP server to validate the relay information option in forwarded BOOTREPLY messages. • By default, DHCP checks that the option-82 field in DHCP reply packets it receives from the DHCP server is valid. If an invalid message is received, the relay agent drops it. If a valid message is received, the relay agent removes the option-82 field and forwards the packet. Use the **ip dhcp relay information check-reply** command to reenable this functionality if it has been disabled. • The **ip dhcp relay information check-reply none** interface configuration command option is saved in the running configuration. This command takes precedence over any global relay agent information configuration.
Step 6	**ip dhcp relay information policy-action** {**drop** | **keep** | **replace**} **Example:** `Router(config-if)# ip dhcp relay information policy-action replace`	Configures the information reforwarding policy for a DHCP relay agent (what a relay agent should do if a message already contains relay information). • See the "Relay Agent Information Reforwarding Policy" section on page 5 for more information.

	Command or Action	Purpose
Step 7	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 8	Repeat Steps 3 through 7 to configure relay agent information settings on different interfaces.	(Optional)

Configuring the Subscriber Identifier Suboption of the Relay Agent Information Option

Perform this task to enable an Internet service provider (ISP) to add a unique identifier to the subscriber-identifier suboption of the relay agent information option.

The unique identifier enables an ISP to identify a subscriber, to assign specific actions to that subscriber (for example, assignment of host IP address, subnet mask, and domain name system DNS), and to trigger accounting.

Before the introduction of this feature, if a subscriber moved, each ISP had to be informed of the change and all ISPs had to reconfigure the DHCP settings for the affected customers at the same time. Even if the service was not changed, every move involved administrative changes in the ISP environment. With the introduction of this feature, if a subscriber moves from one Network Access Server to another, there is no need for a change in the configuration on the part of the DHCP server or ISP.

Prerequisites

You should configure the unique identifier for each subscriber.

The new configurable subscriber-identifier option should be configured on the interface connected to the client. When a subscriber moves from one interface to the other, the interface configuration should also be changed.

The server should be able to recognize the new suboption.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp relay information option**
4. **interface** *type number*
5. **ip dhcp relay information option subscriber-id** *string*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip dhcp relay information option** **Example:** `Router(config)# ip dhcp relay information option`	Enables the system to insert the DHCP relay agent information option (option-82 field) in forwarded BOOTREQUEST messages to a DHCP server. • This function is disabled by default.
Step 4	**interface** *type number* **Example:** `Router(config)# interface atm4/0.1`	Configures an interface and enters interface configuration mode.
Step 5	**ip dhcp relay information option subscriber-id** *string* **Example:** `Router(config-if)# ip dhcp relay information option subscriber-id newsubscriber123`	Specifies that a DHCP relay agent add a subscriber identifier suboption to the relay information option. • The *string* argument can be up to a maximum of 50 characters and can be alphanumeric. **Note** If more than 50 characters are configured, the string is truncated. **Note** The **ip dhcp relay information option subscriber-id** command is disabled by default to ensure backward capability.

Configuring DHCP Relay Class Support for Client Identification

Perform this task to configure DHCP relay class support for client identification.

Relay Class Support Overview

DHCP relay class support for client identification allows the Cisco IOS relay agent to forward client-generated DHCP messages to different DHCP servers based on the content of the following four options:

- Option 60: vendor class identifier
- Option 77: user class
- Option 124: vendor-identifying vendor class
- Option 125: vendor-identifying vendor-specific information

Each option identifies the type of client sending the DHCP message.

Relay pools provide a method to define DHCP pools that are not used for address allocation. These relay pools can specify that DHCP messages from clients on a specific subnet should be forwarded to a specific DHCP server. These relay pools can be configured with relay classes inside the pool that help determine the forwarding behavior.

For example, after receiving the option in the DHCP DISCOVER message, the relay agent will match and identify the relay class from the relay pool and then direct the DHCP DISCOVER message to the DHCP server associated with that identified relay class.

Relay Class Support Usage Scenario

In an example application, a Cisco router acting as a DHCP relay agent receives DHCP requests from two VoIP services (H323 and SIP). The requesting devices are identified by option 60.

Both VoIP services have a different back-office infrastructure so they cannot be serviced by the same DHCP server. Requests for H323 devices must be forwarded to the H323 server and requests from the SIP devices must be forwarded to the SIP server.

The solution is to configure the relay agent with relay classes that are configured to match option 60 values sent by the client devices. Based on the option value, the relay agent will match and identify the relay class, and forward the DHCP DISCOVER message to the DHCP server associated with that identified relay class.

Prerequisites

It is important to understand how DHCP options work. See the “DHCP Overview” module for more information.

You must know the hexadecimal value of each byte location in the options to be able to configure the **option hex** command. The format may vary from product to product. Contact the relay agent vendor for this information.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp class** *class-name*
4. **option** *code* **hex** *hex-pattern* [*] [**mask** *bit-mask-pattern*]
5. **exit**
6. Repeat Steps 3 through 5 for each DHCP class you need to configure.
7. **ip dhcp pool** *name*
8. **relay source** *ip-address subnet-mask*
9. **class** *class-name*
10. **relay target** [**vrf** *vrf-name* | **global**] *ip-address*
11. **exit**
12. Repeat Steps 9 through 11 for each DHCP class you need to configure.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`enable`** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**`configure terminal`** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**`ip dhcp class`** *`class-name`* **Example:** `Router(config)# ip dhcp class SIP`	Defines a DHCP class and enters DHCP class configuration mode.
Step 4	**`option`** *`code`* **`hex`** *`hex-pattern`* `[`**`*`**`][`**`mask`** *`bit-mask-pattern`*`]` **Example:** `Router(dhcp-class)# option 60 hex 010203`	Enables the relay agent to make forwarding decisions based on DHCP options inserted in the DHCP message.
Step 5	**`exit`** **Example:** `Router(dhcp-class)# exit`	Exits DHCP class configuration mode.
Step 6	Repeat Steps 3 through 5 for each DHCP class you need to configure.	—
Step 7	**`ip dhcp pool`** *`name`* **Example:** `Router(config)# ip dhcp pool ABC`	Configures a DHCP pool on a DHCP server and enters DHCP pool configuration mode.
Step 8	**`relay source`** *`ip-address subnet-mask`* **Example:** `Router(dhcp-config)# relay source 10.2.0.0 255.0.0.0`	Configures the relay source. The *ip-address* and *subnet-mask* arguments are the IP address and subnet mask for the relay source. • This command is similar to the **network** command in a normal DHCP network pool, because it restricts the use of the address pool to packets arriving on the interface whose configured IP address and mask matches the relay source configuration.
Step 9	**`class`** *`class-name`* **Example:** `Router(dhcp-config)# class SIP`	Associates a class with a DHCP pool and enters DHCP pool class configuration mode.
Step 10	**`relay target`** `[`**`vrf`** *`vrf-name`* `|` **`global`**`]` *`ip-address`* **Example:** `Router(config-dhcp-pool-class)# relay target 10.21.3.1`	Configures an IP address for a DHCP server to which packets are forwarded.

	Command or Action	Purpose
Step 11	**exit** **Example:** `Router(dhcp-class)# exit`	Exits DHCP pool class configuration mode.
Step 12	Repeat Steps 9 through 11 for each DHCP class you need to configure	—

Configuring DHCP Relay Agent Support for MPLS VPNs

Perform this task to configure DHCP relay agent support for MPLS VPNs.

DHCP Relay Agent Support for MPLS VPNs

DHCP relay support for Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs) enables a network administrator to conserve address space by allowing overlapping addresses. The relay agent can support multiple clients on different VPNs, and many of these clients from different VPNs can share the same IP address.

Configuring VPNs involves an adjustment to the usual DHCP host IP address designation. VPNs use private address spaces that might not be unique across the Internet.

In some environments, a relay agent resides in a network element that also has access to one or more MPLS VPNs. A DHCP server that provides service to DHCP clients on those different VPNs must locate the VPN in which each client resides. The network element that contains the relay agent typically captures the VPN association of the DHCP client and includes this information in the relay agent information option of the DHCP packet.

DHCP relay support for MPLS VPNs allows the relay agent to forward this necessary VPN-related information to the DHCP server using the following three suboptions of the DHCP relay agent information option:

- VPN identifier
- Subnet selection
- Server identifier override

The VPN identifier suboption is used by the relay agent to tell the DHCP server the VPN for every DHCP request it passes on to the DHCP server, and it is also used to properly forward any DHCP reply that the DHCP server sends back to the relay agent. The VPN identifier suboption contains the VPN ID configured on the incoming interface to which the client is connected. If you configure the VRF name but not the VPN ID, the VRF name is used as the VPN identifier suboption. If the interface is in global routing space, the VPN suboptions are not added.

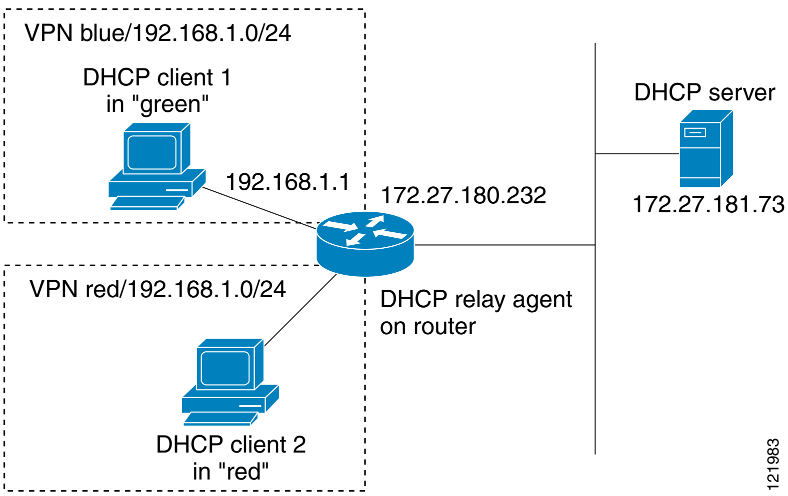
The subnet selection suboption allows the separation of the subnet where the client resides from the IP address used to communicate with the relay agent. In typical DHCP processing, the gateway address specifies both the subnet on which a DHCP client resides and the IP address that the server can use to communicate with the relay agent. Situations exist where the relay agent needs to specify the subnet on which a DHCP client resides that is different from the IP address the server can use to communicate with the relay agent. The subnet selection suboption is included in the relay agent information option and passed on to the DHCP server. The gateway address is changed to the outgoing interface of the relay agent toward the DHCP server. The DHCP server uses this gateway address to send reply packets back to the relay agent.

The server identifier override suboption value is copied in the reply packet from the DHCP server instead of the normal server ID address. The server identifier override suboption contains the incoming interface IP address, which is the IP address on the relay agent that is accessible from the client. Using this information, the DHCP client sends all renew and release packets to the relay agent. The relay agent adds all of the VPN suboptions and then forwards the renew and release packets to the original DHCP server.

After adding these suboptions to the DHCP relay agent information option, the gateway address is changed to the outgoing interface of the relay agent toward the DHCP server. When the packets are returned from the DHCP server, the relay agent removes the relay agent information options and forwards the packets to the DHCP client on the correct VPN.

Figure 3 shows a VPN scenario where the DHCP relay agent and DHCP server can recognize the VPN that each client resides within. DHCP client 1 is part of VPN *green* and DHCP client 2 is part of VPN *red* and both have the same private IP address 192.168.1.0/24. Because the clients have the same IP address, the DHCP relay agent and DHCP server use the VPN identifier, subnet selection, and server identifier override suboptions of the relay agent information option to distinguish the correct VPN of the client.

Figure 3 Virtual Private Network DHCP Configuration



Prerequisites

Before configuring DHCP relay support for MPLS VPNs, you must configure standard MPLS VPNs.

Restrictions

- If the **ip dhcp relay information option vpn** global configuration command is configured and the **ip dhcp relay information option vpn-id** interface configuration command is not configured, the global configuration is applied to all interfaces.
- If the **ip dhcp relay information option vpn** global configuration command is configured and the **ip dhcp relay information option vpn-id** interface configuration command is also configured, the interface configuration command takes precedence over the global configuration command. However, the global configuration is applied to interfaces without the interface configuration.

- If the **ip dhcp relay information option vpn** global configuration command is not configured and the **ip dhcp relay information option vpn-id** interface configuration command is configured, only the interface with the configuration option applied is affected. All other interfaces are not impacted by the configuration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp relay information option vpn**
4. **interface** *type number*
5. **ip helper-address vrf** *name* [**global**] *address*
6. **ip dhcp relay information option vpn-id** [**none**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`enable`** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**`configure terminal`** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**`ip dhcp relay information option vpn`** **Example:** `Router(config)# ip dhcp relay information option vpn`	Enables the system to insert VPN suboptions into the DHCP relay agent information option in forwarded BOOTREQUEST messages to a DHCP server and sets the gateway address to the outgoing interface toward the DHCP server. • The VPN suboptions are also added to the BOOTP broadcast packets when the command is configured.
Step 4	**`interface`** *`type number`* **Example:** `Router(config)# interface FastEthernet0/0`	Configures an interface and enters interface configuration mode.
Step 5	**`ip helper-address vrf`** *`name`* [**`global`**] *`address`* **Example:** `Router(config-if)# ip helper-address vrf blue 172.27.180.232`	Forwards UDP broadcasts, including BOOTP, received on an interface. • If the DHCP server resides in a different VPN or global space that is different from the VPN, then the **vrf** *name* or **global** options allow you to specify the name of the VRF or global space in which the DHCP server resides.
Step 6	**`ip dhcp relay information option vpn-id`** [**`none`**] **Example:** `Router(config-if)# ip dhcp relay information option vpn-id`	(Optional) Enables the system to insert VPN suboptions into the DHCP relay agent information option in forwarded BOOTREQUEST messages to a DHCP server and sets the gateway address to the outgoing interface toward the DHCP server. • The VPN suboptions are also added to the BOOTP broadcast packets when the command is configured. • The **ip dhcp relay information option vpn-id none** command allows you to disable the VPN functionality on the interface. The only time you need to use this command is when the **ip dhcp relay information option vpn** global configuration command is configured and you want to override the global configuration. • The **no ip dhcp relay information option vpn-id** command removes the configuration from the running configuration. In this case, the interface inherits the global configuration, which may or may not be configured to insert VPN suboptions.

Setting the Gateway Address of the DHCP Broadcast to a Secondary Address Using Smart Relay Agent Forwarding

Perform this task to configure smart relay agent forwarding.

You only need to configure helper addresses on the interface where the UDP broadcasts that you want to forward to the DHCP server are being received, and you only need the **ip dhcp smart-relay** command configured if you have secondary addresses on that interface and you want the router to step through each IP network when forwarding DHCP requests. Without the smart relay agent configured, all requests are forwarded using the primary IP address on the interface.

If the **ip dhcp smart-relay** command is configured, the relay agent counts the number of times the client retries sending a request to the DHCP server when there is no DHCPOFFER message from the DHCP server. After three retries, the relay agent sets the gateway address to the secondary address. If the DHCP server still does not respond after three more retries, then the next secondary address is used as the gateway address.

This functionality is useful when the DHCP server cannot be configured to use secondary pools.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp smart-relay**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip dhcp smart-relay** **Example:** `Router(config)# ip dhcp smart-relay`	Allows the DHCP relay agent to switch the gateway address (giaddr field of a DHCP packet) to secondary addresses when there is no DHCPOFFER message from a DHCP server.

Troubleshooting the DHCP Relay Agent

Perform this task to troubleshoot the DHCP relay agent.

The **show ip route dhcp** command is useful to help you understand any problems with the DHCP relay agent adding routes to clients from unnumbered interfaces. All routes added to the routing table by the DHCP server and relay agent are displayed.

SUMMARY STEPS

1. **enable**
2. **show ip route dhcp**
3. **show ip route dhcp** *ip-address*
4. **show ip route vrf** *vrf-name* **dhcp**
5. **clear ip route** [**vrf** *vrf-name*] **dhcp** [*ip-address*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip route dhcp** **Example:** `Router# show ip route dhcp`	Displays all routes added by the Cisco IOS DHCP server and relay agent.
Step 3	**show ip route dhcp** *ip-address* **Example:** `Router# show ip route dhcp 172.16.1.3`	Displays all routes added by the Cisco IOS DHCP server and relay agent associated with an IP address.
Step 4	**show ip route vrf** *vrf-name* **dhcp** **Example:** `Router# show ip route vrf red dhcp`	Displays all routes added by the Cisco IOS DHCP server and relay agent associated with the named VRF.
Step 5	**clear ip route** [**vrf** *vrf-name*] **dhcp** [*ip-address*] **Example:** `Router# clear ip route dhcp`	Removes routes from the routing table added by the DHCP server and relay agent for the DHCP clients on unnumbered interfaces.

Configuration Examples for the Cisco IOS DHCP Relay Agent

This section provides the following configuration examples:

- Configuring the DHCP Relay Agent and Relay Agent Information Option Support: Example, page 21
- Configuring the DHCP Relay Agent and Relay Agent Information Option Support per Interface: Example, page 21
- Configuring the Subscriber Identifier Suboption: Example, page 21
- Configuring DHCP Relay Class Support for Client Identification: Example, page 22
- Configuring DHCP Relay Agent Support for MPLS VPNs: Example, page 22
- Configuring DHCP Smart Relay Agent Forwarding: Example, page 22

Configuring the DHCP Relay Agent and Relay Agent Information Option Support: Example

The following example shows how to enable the DHCP server, the relay agent, and the insertion and removal of the DHCP relay information option (option 82). Note that the Cisco IOS DHCP server is enabled by default. In this example, the DHCP server was disabled:

```
!reenables the DHCP server
service dhcp
ip dhcp relay information option
!
interface ethernet0/0
 ip address 192.168.100.1 255.255.255.0
 ip helper-address 10.55.11.3
```

Configuring the DHCP Relay Agent and Relay Agent Information Option Support per Interface: Example

The following example shows that for subscribers being serviced by the same aggregation router, the relay agent information option needs to be processed differently for Asynchronous Transfer Mode (ATM) subscribers than for Ethernet digital subscribers. For ATM subscribers, the relay agent information option is configured to be removed from the packet by the relay agent before forwarding to the client. For Ethernet subscribers, the connected device provides the relay agent information option, and it is configured to remain in the packet and be forwarded to the client.

```
ip dhcp relay information trust-all
interface Loopback0
 ip address 10.16.0.1 255.255.255.0
!
interface ATM3/0
 no ip address
!
interface ATM3/0.1
 ip helper-address 10.16.1.2
 ip unnumbered loopback0
 ip dhcp relay information option-insert
!
interface Loopback1
 ip address 10.18.0.1 255.255.255.0
!
interface Ethernet4
 no ip address
!
interface Ethernet4/0.1
 encap dot1q 123
 ip unnumbered loopback1
 ip helper-address 10.18.1.2
 ip dhcp relay information policy-action keep
```

Configuring the Subscriber Identifier Suboption: Example

The following example shows how to add a unique identifier to the subscriber-identifier suboption of the relay agent information option.

```
ip dhcp relay information option
!
```

```
interface Loopback0
 ip address 10.1.1.129 255.255.255.192
!
interface ATM4/0
 no ip address
!
interface ATM4/0.1 point-to-point
 ip helper-address 10.16.1.2
 ip unnumbered Loopback0
 ip dhcp relay information option subscriber-id newperson123
 atm route-bridged ip
 pvc 88/800
 encapsulation aal5snap
```

Configuring DHCP Relay Class Support for Client Identification: Example

In the following example, DHCP messages are received from DHCP clients on subnet 10.2.2.0. The relay agent will match and identify the relay class from the relay pool and forward the DHCP message to the appropriate DHCP server identified by the **relay target** command.

```
!
ip dhcp class H323
 option 60 hex 010203
!
ip dhcp class SIP
 option 60 hex 040506
!
! The following is the relay pool
ip dhcp pool red
 relay source 10.2.2.0 255.255.255.0
 class H323
  relay target 172.16.2.1
  relay target 172.17.2.1
!
 class SIP
  relay target 172.18.2.1
```

Configuring DHCP Relay Agent Support for MPLS VPNs: Example

In the following example, the DHCP relay agent receives a DHCP request on Ethernet interface 0/1 and sends the request to the DHCP server located at IP helper address 10.44.23.7, which is associated with the VRF named red:

```
ip dhcp relay information option vpn
!
interface ethernet 0/1
 ip helper-address vrf red 10.44.23.7
!
```

Configuring DHCP Smart Relay Agent Forwarding: Example

In the following example, the router will forward the DHCP broadcast received on Ethernet interface 0/0 to the DHCP server (10.55.11.3), inserting 192.168.100.1 in the giaddr field of the DHCP packet. If the DHCP server has a scope or pool configured for the 192.168.100.0/24 network, it will respond; otherwise it will not respond.

Because the **ip dhcp smart-relay** global configuration command is configured, if the router sends three requests using 192.168.100.1 in the giaddr field, and doesn't get a response, it will move on and start using 172.16.31.254 in the giaddr field instead. Without the smart relay functionality, the route only uses 192.168.100.1 in the giaddr field.

```
ip dhcp smart-relay
!
interface ethernet0/0
 ip address 192.168.100.1 255.255.255.0
 ip address 172.16.31.254 255.255.255.0
 ip helper-address 10.55.11.3
!
```

Additional References

The following sections provide references related to configuring the Cisco IOS DHCP relay agent.

Related Documents

Related Topic	Document Title
DHCP commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Addressing Services Command Reference*
DHCP conceptual information	"DHCP Overview" module
DHCP server configuration	"Configuring the Cisco IOS DHCP Server" module
DHCP client configuration	"Configuring the Cisco IOS DHCP Client" module
DHCP server on-demand address pool manager configuration	"Configuring the DHCP Server On-Demand Address Pool Manager" module
DHCP advanced features	"Configuring DHCP Services for Accounting and Security" module
DHCP enhancements for edge-session management configuration	"Configuring DHCP Enhancements for Edge-Session Management" module
DHCP options	"DHCP Options" appendix in the *Network Registrar User's Guide*, Release 6.1.1

Standards

Standards	Title
No new or modified standards are supported by this functionality.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 951	*Bootstrap Protocol (BOOTP)*
RFC 1542	*Clarifications and Extensions for the Bootstrap Protocol*
RFC 2131	*Dynamic Host Configuration Protocol*
RFC 2685	*Virtual Private Networks Identifier*
RFC 3046	*DHCP Relay Information Option*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for the Cisco IOS DHCP Relay Agent

Table 1 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.2(1) or a later release appear in the table.

For information on a feature in this technology that is not documented here, see the "DHCP Features Roadmap."

Not all commands may be available in your Cisco IOS software release. For details on when support for a specific command was introduced, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for the Cisco IOS DHCP Relay Agent

Feature Name	Releases	Feature Configuration Information
DHCP Class Support for Client Identification	12.4(11)T	This feature enhances the DHCP class mechanism to support options 60, 77, 124, and 125. These options identify the type of client sending the DHCP message. The DHCP relay agent can make forwarding decisions based on the content of the options in the DHCP message sent by the client. The following sections provide information about this feature: • Configuring DHCP Relay Class Support for Client Identification • Configuring DHCP Relay Class Support for Client Identification: Example The following command was introduced by this feature: **option hex**
DHCPv4 Relay per Interface VPN ID Support	12.4(11)T Cisco IOS XE Release 2.1	The DHCPv4 Relay per Interface VPN ID Support feature allows the Cisco IOS DHCP relay agent to be configured per interface to override the global configuration of the **ip dhcp relay information option vpn** command. This feature allows subscribers with different relay information option VPN ID requirements on different interfaces to be reached from one Cisco router. The following sections provide information about this feature: • Configuring DHCP Relay Agent Support for MPLS VPNs • Configuring DHCP Relay Agent Support for MPLS VPNs: Example The following command was introduced by this feature: **ip dhcp relay information option vpn-id**

Table 1 Feature Information for the Cisco IOS DHCP Relay Agent

Feature Name	Releases	Feature Configuration Information
DHCP Relay Option 82 per Interface Support	12.4(6)T 12.2(31)SB2 12.2(33)SRC Cisco IOS XE Release 2.1	This feature enables support for the DHCP relay agent information option (option 82) on a per interface basis. The interface configuration allows different DHCP servers, with different DHCP option 82 requirements to be reached from one Cisco router. The following sections provide information about this feature: • Configuring Relay Agent Information Option Support per Interface • Configuring the DHCP Relay Agent and Relay Agent Information Option Support per Interface: Example The following commands were introduced by this feature: **ip dhcp relay information check-reply**, **ip dhcp relay information option-insert**, and **ip dhcp relay information policy-action**
DHCP Subscriber Identifier Suboption of Option 82	12.3(14)T 12.2(28)SB 12.2(33)SRB Cisco IOS XE Release 2.1	This feature enables an ISP to add a unique identifier to the subscriber-identifier suboption of the relay agent information option. The following sections provide information about this feature: • Configuring the Subscriber Identifier Suboption of the Relay Agent Information Option • Configuring the Subscriber Identifier Suboption: Example The following command was introduced by this feature: **ip dhcp relay information option subscriber-id**
DHCP Relay MPLS VPN Support	12.2(8) 12.2(28)SB 12.2(33)SRC Cisco IOS XE Release 2.1	DHCP relay support for MPLS VPNs enables a network administrator to conserve address space by allowing overlapping addresses. The relay agent can support multiple clients on different VPNs, and many of these clients from different VPNs can share the same IP address. The following sections provide information about this feature: • Configuring DHCP Relay Agent Support for MPLS VPNs • Configuring DHCP Relay Agent Support for MPLS VPNs: Example The following commands were modified by this feature: **ip dhcp relay information option** and **ip helper address**

Glossary

client—A host trying to configure its interface (obtain an IP address) using DHCP or BOOTP protocols.

DHCP—Dynamic Host Configuration Protocol.

giaddr—Gateway address. The giaddr field of the DHCP message provides the DHCP server with information about the IP address subnet on which the client is to reside. It also provides the DHCP server with an IP address where the response messages are to be sent.

MPLS—Multiprotocol Label Switching. Emerging industry standard upon which tag switching is based.

relay agent—A router that forwards DHCP and BOOTP messages between a server and a client on different subnets.

server—DHCP or BOOTP server.

VPN—Virtual Private Network. Enables IP traffic to use tunneling to travel securely over a public TCP/IP network.

VRF—VPN routing and forwarding instance. A VRF consists of an IP routing table, a derived forwarding table, a set of interfaces that use the forwarding table, and a set of rules and routing protocols that determine what goes into the forwarding table. In general, a VRF includes the routing information that defines a customer VPN site that is attached to a PE router. Each VPN instantiated on the PE router has its own VRF.

Note See *Internetworking Terms and Acronyms* for terms not included in this glossary.

Configuring the Cisco IOS DHCP Client

Cisco IOS Dynamic Host Configuration Protocol (DHCP) client software provides the flexibility to include various configuration options for the DHCP client. A DHCP client is defined as an Internet host using DHCP to obtain configuration parameters such as an IP address. This module describes the concepts and tasks needed to configure the Cisco IOS DHCP client.

Module History

This module was first published on May 2, 2005, and last updated on December 31, 2007.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. For the latest feature information and caveats, see the release notes for your Cisco IOS software release. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for the Cisco IOS DHCP Client" section on page 13.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Restrictions for Configuring the DHCP Client

The DHCP client can be configured on Ethernet interfaces and on PPPoA and certain ATM interfaces. The DHCP client works with ATM point-to-point interfaces and will accept any encapsulation type. For ATM multipoint interfaces, the DHCP client is only supported using the aal5snap encapsulation type combined with Inverse ARP. Inverse ARP, which builds an ATM map entry, is necessary to send unicast packets to the server (or relay agent) on the other end of the connection. Inverse ARP is only supported for the aal5snap encapsulation type.

For multipoint interfaces, an IP address can be acquired using other encapsulation types because broadcast packets are used. However, unicast packets to the other end will fail because there is no ATM map entry and thus DHCP renewals and releases also fail.

Information About the DHCP Client

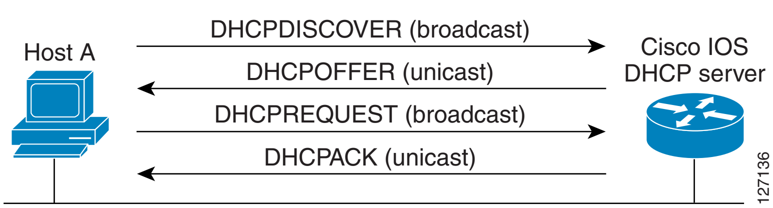
To configure the DHCP client, you must understand the following concepts:

- DHCP Client Operation, page 2
- DHCP Client Overview, page 3
- DHCP Client on WAN Interfaces, page 3

DHCP Client Operation

DHCP provides a framework for passing configuration information to hosts on a TCP/IP network. A DHCP client is an Internet host using DHCP to obtain configuration parameters such as an IP address. Figure 1 shows the basic steps that occur when a DHCP client requests an IP address from a DHCP server. The client, Host A, sends a DHCPDISCOVER broadcast message to locate a DHCP server. A DHCP server offers configuration parameters (such as an IP address, a MAC address, a domain name, and a lease for the IP address) to the client in a DHCPOFFER unicast message.

Figure 1 DHCP Request for an IP Address from a DHCP Server



A DHCP client may receive offers from multiple DHCP servers and can accept any one of the offers; however, the client usually accepts the first offer it receives. Additionally, the offer from the DHCP server is not a guarantee that the IP address will be allocated to the client; however, the server usually reserves the address until the client has had a chance to formally request the address.

The client returns a formal request for the offered IP address to the DHCP server in a DHCPREQUEST broadcast message. The DHCP server confirms that the IP address has been allocated to the client by returning a DHCPACK unicast message to the client.

DHCP Client Overview

The configurable DHCP client functionality allows a DHCP client to use a user-specified client identifier, class identifier, or suggested lease time when requesting an address from a DHCP server.

Configuration parameters and other control information are carried in tagged data items that are stored in the options field of the DHCP message. The DHCP client provides flexibility by allowing the following options to be configured for a DHCP client:

- Option 12—This option specifies the name of the client. The name may or may not be qualified with the local domain.
- Option 51—This option is used in a client request (DHCPDISCOVER or DHCPREQUEST) to allow the client to request a lease time for the IP address.
- Option 55—This option allows the DHCP client to request certain options from the DHCP server. The **ip dhcp client request** command allows the system administrator to turn off some of the requested options, thus removing them from the request list.
- Option 60—This option allows the user to configure the vendor class identifier string to use in the DHCP interaction.
- Option 61—This option is used by DHCP clients to specify their unique identifier. DHCP servers use this value to index their database of address bindings. This value is expected to be unique for all clients in an administrative domain.

DHCP Client on WAN Interfaces

The DHCP client on WAN interfaces allows a DHCP client to acquire an IP address over PPP over ATM (PPPoA) and certain ATM interfaces. By using DHCP rather than the IP Control Protocol (IPCP), a DHCP client can acquire other useful information such as DNS addresses, the DNS default domain name, and the default route.

The configuration of PPPoA and Classical IP and ARP over ATM already allows for a broadcast capability over the interface (using the **broadcast** keyword on the ATM interface). Most changes in this feature are directed at removing already existing restrictions on what types of interfaces are allowed to send out DHCP packets (previously, dialer interfaces have not been allowed). This feature also ensures that DHCP RELEASE messages are sent out the interface before a connection is allowed to be broken.

How to Configure the DHCP Client

This section contains the following tasks:

- Configuring the DHCP Client, page 3
- Forcing a Release or Renewal of a DHCP Lease for a DHCP Client, page 6

Configuring the DHCP Client

Perform this task to configure the DHCP client.

DHCP Client Default Behavior

Cisco routers running Cisco IOS software include DHCP server and relay agent software, which are enabled by default. Your router can act as both the DHCP client and DHCP server. Use the **ip address dhcp** interface command to obtain IP address information for the configured interface.

Prerequisites

You must configure the **ip dhcp client** commands before entering the **ip address dhcp** command on an interface to ensure that the DHCPDISCOVER messages that are generated contain the correct option values. The **ip dhcp client** commands are checked only when an IP address is acquired from DHCP. If any of the **ip dhcp client** commands are entered after an IP address has been acquired from DHCP, it will not take effect until the next time the router acquires an IP address from DHCP. This means that the new configuration will only take effect after either the **ip address dhcp** command or the **release dhcp** and **renew dhcp** EXEC commands have been configured.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip dhcp client client-id** {*interface-name* | **ascii** *string* | **hex** *string*}
5. **ip dhcp client class-id** {*string* | **hex** *string*}
6. **ip dhcp client lease** *days* [*hours*] [*minutes*]
7. **ip dhcp client hostname** *host-name*
8. [**no**] **ip dhcp client request** *option-name*
9. **ip address dhcp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`configure terminal` **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	`interface` *type number* **Example:** `Router(config)# interface Ethernet 1`	Configures an interface type and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**ip dhcp client client-id** {*interface-name* | **ascii** *string* | **hex** *string*} **Example:** `Router(config-if)# ip dhcp client client-id ascii mytest1`	(Optional) Specifies the client identifier. • When you specify the **no** form of this command, the configuration is removed and the system returns to using the default form. It is not possible to configure the system to not include a client identifier.
Step 5	**ip dhcp client class-id** {*string* | **hex** *string*} **Example:** `Router(config-if)# ip dhcp client class-id my-class-id`	(Optional) Specifies the class identifier.
Step 6	**ip dhcp client lease** *days* [*hours*] [*minutes*] **Example:** `Router(config-if)# ip dhcp client lease 2`	(Optional) Configures the duration of the lease for an IP address that is requested from a DHCP client to a DHCP server.
Step 7	**ip dhcp client hostname** *host-name* **Example:** `Router(config-if)# ip dhcp client hostname router1`	(Optional) Specifies or modifies the host name sent in the DHCP message.
Step 8	[**no**] **ip dhcp client request** *option-name* **Example:** `Router(config-if)# no ip dhcp client request tftp-server-address`	(Optional) Configures a DHCP client to request an option from a DHCP server. • The option name can be **tftp-server-address**, **netbios-nameserver**, **vendor-specific**, **static-route**, **domain-name**, **dns-nameserver**, or **router**. By default, all these options are requested. The **no** form of the command instructs the system to not request certain options.
Step 9	**ip address dhcp** **Example:** `Router(config-if)# ip address dhcp`	Acquires an IP address on an interface from DHCP.

Troubleshooting Tips

To verify the configuration, you can use the **debug dhcp detail** EXEC command to display the DHCP packets that were sent and received. To display the server side of the DHCP interaction, use the **debug ip dhcp server packets** command.

The following are troubleshooting tips for DHCP clients on WAN interfaces:

- An ATM primary interface is always multipoint.
- An ATM subinterface can be multipoint or point-to-point.
- If you are using a point-to-point interface, the routing table determines when to send a packet to the interface and ATM map entries are not needed. Consequently, Inverse ARP, which builds ATM map entries, is not needed.

- If you are using a multipoint interface you must use Inverse ARP to discover the IP address of the other side of the connection.
- You can specify Inverse ARP through the **protocol ip inarp** interface configuration command. You must use the aal5snap encapsulation type when using Inverse ARP because it is the only encapsulation type that supports Inverse ARP.

Forcing a Release or Renewal of a DHCP Lease for a DHCP Client

Perform this task to force a release or renewal of a DHCP lease for a DHCP client.

Forcing a release or renewal of a DHCP lease for a DHCP client provides the ability to perform two independent operations from the command-line interface (CLI) in EXEC mode:

- Immediately release a DHCP lease for a DHCP client.
- Force a DHCP renewal of a lease for a DHCP client.

This functionality provides the following benefits:

- Eliminates the need to go into the configuration mode to reconfigure the router to release or renew a DHCP lease.
- Simplifies the release and renewal of a DHCP lease.
- Reduces the amount of time spent performing DHCP IP release and renewal configuration tasks.

DHCP Release and Renew CLI Operation

Release a DHCP Lease

The **release dhcp** command starts the process to immediately release a DHCP lease for the specified interface. After the lease is released, the interface address is deconfigured. The **release dhcp** command does not deconfigure the **ip address dhcp** command specified in the configuration file for the interface. During a write memory or show running configuration file action, or if the router is rebooted, the **ip address dhcp** command executes to acquire a DHCP address for the interface.

The original IP address for the interface must be assigned by the DHCP server. If the interface is not assigned an IP address by the DHCP server, the **release dhcp** command fails and displays the following error message:

```
Interface does not have a DHCP originated address
```

Renew a DHCP Lease

The **renew dhcp** command advances the DHCP lease timer to the next stage, at which point one of the following occurs:

- If the lease is currently in a BOUND state, the lease is advanced to the RENEW state and a DHCP RENEW request is sent.
- If the lease is currently in a RENEW state, the timer is advanced to the REBIND state and a DHCP REBIND request is sent.

If there is no response to the RENEW request, the interface remains in the RENEW state. In this case, the lease timer will advance to the REBIND state and subsequently send a REBIND request.

If a NAK response is sent in response to the RENEW request, the interface is deconfigured.

The original IP address for the interface must be assigned by the DHCP server. If the interface is not assigned an IP address by the DHCP server, the **renew dhcp** command fails and displays the following error message:

```
Interface does not have a DHCP originated address
```

Prerequisites

The DHCP client must be assigned an IP address by the DHCP server.

Restrictions

If the DHCP client is not assigned an IP address by the DHCP server, the DHCP release and renew CLI commands will fail.

SUMMARY STEPS

1. **enable**
2. **release dhcp** *type number*
3. **renew dhcp** *type number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**release dhcp** *type number* **Example:** `Router# release dhcp ethernet 3/1`	Performs an immediate release of the DHCP lease for the interface and deconfigures the IP address for the interface.
Step 3	**renew dhcp** *type number* **Example:** `Router# renew dhcp ethernet 3/1`	Forces the DHCP timer to advance to the next stage, at which point a subsequent action is taken: a DHCP REQUEST packet is sent to renew or rebind the lease.

Configuration Examples for the DHCP Client

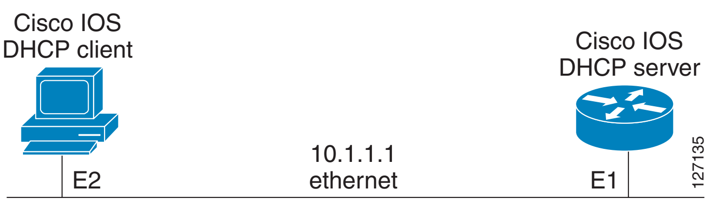
This section provides the following configuration examples:

- Configuring the DHCP Client: Example, page 8
- Customizing the DHCP Client Configuration: Example, page 8
- Configuring an ATM Primary Interface (Multipoint) Using aal5snap Encapsulation and Inverse ARP: Example, page 9
- Configuring an ATM Point-to-Point Subinterface Using aa15snap Encapsulation: Example, page 9

Configuring the DHCP Client: Example

Figure 2 shows a simple network diagram of a DHCP client on an Ethernet LAN.

Figure 2 Topology Showing DHCP Client with Ethernet Interface



On the DHCP server, the configuration is as follows:

```
ip dhcp pool 1
 network 10.1.1.0 255.255.255.0
 lease 1 6
```

On the DHCP client, the configuration is as follows on interface E2:

```
interface Ethernet2
 ip address dhcp
```

This configuration allows the DHCP client to acquire an IP address from the DHCP server through an Ethernet interface.

Customizing the DHCP Client Configuration: Example

The following example shows how to customize the DHCP client configuration with various options on Ethernet interface 1:

```
interface Ethernet 1
 ip dhcp client client-id ascii my-test1
 ip dhcp client class-id my-class-id
 ip dhcp client lease 0 1 0
 ip dhcp client hostname sanfran
 no ip dhcp client request tftp-server-address
 ip address dhcp
```

Configuring an ATM Primary Interface (Multipoint) Using aal5snap Encapsulation and Inverse ARP: Example

In the following example, the **protocol ip 255.255.255.255 broadcast** configuration is needed because there must be an ATM map entry to recognize the broadcast flag on the permanent virtual circuit (PVC). You can use any ATM map entry. The **protocol ip inarp** configuration is needed so the ATM Inverse ARP can operate on the interface such that the system on the other side can be pinged once an address is assigned by DHCP.

```
interface atm0
 ip address dhcp
 pvc 1/100
  encapsulation aal5snap
  broadcast
  protocol ip 255.255.255.255 broadcast
   protocol ip inarp
```

Configuring an ATM Point-to-Point Subinterface Using aa15snap Encapsulation: Example

The following example shows an ATM point-to-point subinterface configuration using aa15snap encapsulation:

```
interface atm0.1 point-to-point
 ip address dhcp
 pvc 1/100
  encapsulation aal5snap
  broadcast
```

Configuring an ATM Point-to-Point Subinterface Using aa15nlpid Encapsulation: Example

The following example shows an ATM point-to-point subinterface configuration using aa15nlpid encapsulation:

```
interface atm0.1 point-to-point
 ip address dhcp
 pvc 1/100
  encapsulation aal5nlpid
  broadcast
```

Configuring an ATM Point-to-Point Subinterface Using aa15mux PPP Encapsulation: Example

The following example shows an ATM point-to-point subinterface configuration using aa15mux PPP encapsulation:

```
interface atm0.1 point-to-point
 pvc 1/100
  encapsulation aal5mux ppp virtual-template1
  broadcast
!
interface virtual-template1
```

```
ip address dhcp
```

Releasing a DHCP Lease: Example

In the following example, a DHCP release is performed on an interface that was originally assigned an IP address by the DHCP server.

```
Router# release dhcp ethernet 3/1
```

In the following example, an attempt is made to release the DHCP lease on an interface that was not originally assigned an IP address by the DHCP server.

```
Router# release dhcp ethernet 3/1
Interface does not have a DHCP originated address
```

In the following example, the **release dhcp** command is executed without specifying the *type* and *number* arguments.

```
Router# release dhcp
Incomplete command.
```

Renewing a DHCP Lease: Example

In the following example, the DHCP lease is renewed on an interface that was originally assigned an IP address by the DHCP server.

```
Router# renew dhcp ethernet 3/1
```

In the following example, an attempt is made to renew the DHCP lease on an interface that was not originally assigned an IP address by the DHCP server.

```
Router# renew dhcp ethernet 3/1
Interface does not have a DHCP originated address
```

In the following example, the **renew dhcp** command is executed without specifying the *type* and *number* arguments.

```
Router# renew dhcp
Incomplete command.
```

Additional References

The following sections provide references related to the DHCP client.

Related Documents

Related Topic	Document Title
DHCP commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples.	*Cisco IOS IP Addressing Services Command Reference*
DHCP conceptual information	"DHCP Overview" module
DHCP server configuration	"Configuring the Cisco IOS DHCP Server" module

Related Topic	Document Title
DHCP server on-demand address pools	"Configuring the DHCP Server On-Demand Address Pool Manager" module
DHCP relay agent configuration	"Configuring the Cisco IOS DHCP Relay Agent" module
DHCP advanced features	"Configuring DHCP Services for Accounting and Security"module
DHCP enhancements for edge-session management	"Configuring DHCP Enhancements for Edge-Session Management" module

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

RFCs	Title
RFC 2131	Dynamic Host Configuration Protocol
RFC 2132	DHCP Options and BOOTP Vendor Extensions

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for the Cisco IOS DHCP Client

Table 1 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.2(1) or a later release appear in the table.

For information on a feature in this technology that is not documented here, see the "DHCP Features Roadmap".

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for the Cisco IOS DHCP Client

Feature Name	Releases	Feature Configuration Information
Configurable DHCP Client	12.3(8)T 12.2(28)SB	The Configurable DHCP Client feature provides the flexibility to include various configuration options for the DHCP client. A DHCP client is defined as an Internet host using DHCP to obtain configuration parameters such as an IP address. The following sections provide information about this feature: • Configuring the DHCP Client The following commands were introduced by this feature: **ip dhcp client class-id**, **ip dhcp client client-id**, **ip dhcp client hostname**, **ip dhcp client lease**, **ip dhcp client request**

Table 1 Feature Information for the Cisco IOS DHCP Client (continued)

Feature Name	Releases	Feature Configuration Information
DHCP Release and Renew CLI in EXEC Mode	12.3(4)T 12.2(28)SB 12.2(33)SRC	This feature provides the ability to perform two independent operations from the CLI: (1) immediately release a DHCP lease for a DHCP client, and (2) force a DHCP renewal of a lease for a DHCP client. The following section provides information about this feature: • Forcing a Release or Renewal of a DHCP Lease for a DHCP Client The following commands were introduced by this feature: **release dhcp** and **renew dhcp.**
DHCP Client on WAN Interfaces	12.2(8)T 12.2(28)SB	The DHCP Client on WAN Interfaces feature extends the DHCP to allow a DHCP client to acquire an IP address over PPP over ATM (PPPoA) and certain ATM interfaces. The following section provides information about this feature: • DHCP Client on WAN Interfaces No commands were introduced or modified by this feature.

Configuring DHCP Services for Accounting and Security

Cisco IOS software supports several capabilities that enhance DHCP security, reliability, and accounting in Public Wireless LANs (PWLANs). This functionality can also be used in other network implementations. This module describes the concepts and tasks needed to configure DHCP services for accounting and security.

Module History

This module was first published on May 2, 2005, and last updated on May 16, 2008.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. For the latest feature information and caveats, see the release notes for your Cisco IOS software release. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for DHCP Services for Accounting and Security" section on page 23.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Configuring DHCP Services for Accounting and Security

Before you configure DHCP services for accounting and security, you should understand the concepts documented in the "DHCP Overview" module.

Information About DHCP Services for Accounting and Security

Before you configure DHCP services for accounting and security, you should understand the following concepts:

- DHCP Operation in Public Wireless LANs, page 2
- Security Vulnerabilities in Public Wireless LANs, page 2
- DHCP Services for Security and Accounting Overview, page 3
- DHCP Lease Limits, page 3

DHCP Operation in Public Wireless LANs

The configuration of DHCP in a public wireless LAN (PWLAN) simplifies the configuration of wireless clients and reduces the overhead necessary to maintain the network. DHCP clients are leased IP addresses by the DHCP server and then authenticated by the Service Selection Gateway (SSG), which allows the clients to access network services. The DHCP server and client exchange DHCP messages for IP address assignments. When a DHCP server assigns an IP address to a client, a DHCP binding is created. The IP address is leased to the client until the client explicitly releases the IP address and disconnects from the network. If the client disconnects without releasing the address, the server terminates the lease after the lease time is over. In either case, the DHCP server removes the binding and the IP address is returned to the pool.

Security Vulnerabilities in Public Wireless LANs

As more people start using PWLANs, security becomes an important concern. Most implementations of PWLANs rely on DHCP for users to obtain an IP address while in a hot spot (such as a coffee shop, airport terminal, hotel, and so on) and use this IP address provided by the DHCP server throughout their session.

IP spoofing is a common technique used by hackers to spoof IP addresses. For example, customer A obtains an IP address from DHCP and has already been authenticated to use the PWLAN, but a hacker spoofs the IP address of customer A and uses this IP address to send and receive traffic. Customer A will still be billed for the service even though he or she is not using the service.

Address Resolution Protocol (ARP) table entries are dynamic by design. Request and reply ARP packets are sent and received by all the networking devices in a network. In a DHCP network, the DHCP server stores the leased IP address to the MAC address or the client-identifier of the client in the DHCP binding. But as ARP entries are learned dynamically, an unauthorized client can spoof the IP address given by the DHCP server and start using that IP address. The MAC address of this unauthorized client will replace the MAC address of the authorized client in the ARP table allowing the unauthorized client to freely use the spoofed IP address.

DHCP Services for Security and Accounting Overview

DHCP security and accounting features have been designed and implemented to address the security concerns in PWLANs but also can be used in other network implementations.

DHCP accounting provides authentication, authorization, and accounting (AAA) and Remote Authentication Dial-In User Service (RADIUS) support for DHCP. The AAA and RADIUS support improves security by sending secure START and STOP accounting messages. The configuration of DHCP accounting adds a layer of security that allows DHCP lease assignment and termination to be triggered for the appropriate RADIUS START and STOP accounting records so that the session state is properly maintained by upstream devices, such as an SSG. This additional security can help to prevent unauthorized clients or hackers from gaining illegal entry to the network by spoofing authorized DHCP leases.

Three other features have been designed and implemented to address the security concerns in PWLANs. The first feature secures ARP table entries to DHCP leases in the DHCP database. The secure ARP functionality prevents IP spoofing by synchronizing the database of the DHCP server with the ARP table to avoid address hijacking. Secure ARP adds an entry to the ARP table for a client when an address is allocated that can be deleted by the DHCP server only when a binding expires.

The second feature is DHCP authorized ARP. This functionality provides a complete solution by addressing the need for DHCP to explicitly know when a user logs out. Before the introduction of DHCP authorized ARP, there was no mechanism to inform the DHCP server if a user had left the system ungracefully, which could result in excessive billing for a customer that had logged out but the system had not detected the log out. To prevent this problem, DHCP authorized ARP sends periodic ARP messages on a per-minute basis to determine if a user is still logged in. Only authorized users can respond to the ARP request. ARP responses from unauthorized users are blocked at the DHCP server providing an extra level of security.

In addition, DHCP authorized ARP disables dynamic ARP learning on an interface. The address mapping can be installed only by the authorized component specified by the **arp authorized** interface configuration command. DHCP is the only authorized component currently allowed to install ARP entries.

The third feature is ARP Auto-logoff, which adds finer control for probing when authorized users log out. The **arp probe interval** command specifies when to start a probe (the timeout), how frequent a peer is probed (the interval), and the maximum number of retries (the count).

DHCP Lease Limits

You can control the number of subscribers globally or on a per-interface basis by configuring a DHCP lease limit. This functionality allows an Internet service provider (ISP) to limit the number of leases available to clients per household or connection.

How to Configure DHCP Services for Accounting and Security

This section contains the following tasks:

- Configuring DHCP Authorized ARP, page 11
- Configuring a DHCP Lease Limit to Globally Control the Number of Subscribers, page 13
- Configuring a DHCP Lease Limit to Control the Number of Subscribers on an Interface, page 14

Configuring AAA and RADIUS for DHCP Accounting

Perform this task to configure AAA and RADIUS for DHCP accounting.

RADIUS provides the accounting capability for the transmission of secure START and STOP messages. AAA and RADIUS are enabled prior to the configuration of DHCP accounting but can also be enabled to secure an insecure DHCP network. The configuration steps in this section are required for configuring DHCP accounting in a new or existing network.

RADIUS Accounting Attributes

DHCP accounting introduces the attributes shown in Table 1. These attributes are processed directly by the RADIUS server when DHCP accounting is enabled. These attributes can be monitored in the output of the **debug radius** command. The output will show the status of the DHCP leases and specific configuration details about the client. The **accounting** keyword can be used with the **debug radius** command to filter the output and display only DHCP accounting messages.

Table 1 RADIUS Accounting Attributes

Attribute	Description
Calling-Station-ID	The output from this attribute displays the MAC address of the client.
Framed-IP-Address	The output from this attribute displays the IP address that is leased to the client.
Acct-Terminate-Cause	The output from this attribute displays the message “session-timeout” if a client does not explicitly disconnect.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa group server radius** *group-name*
5. **server** *ip-address* **auth-port** *port-number* **acct-port** *port-number*
6. **exit**
7. **aaa accounting** {**system** | **network** | **exec** | **connection** | **commands** *level*} {**default** | *list-name*} {**start-stop** | **stop-only** | **none**} [**broadcast**] **group** *group-name*
8. **aaa session-id** {**common** | **unique**}
9. **ip radius source-interface** *type-number* [**vrf** *vrf-name*]
10. **radius-server host** {*hostname* | *ip-address*} [**auth-port** *port-number*] [**acct-port** *port-number*]
11. **radius-server retransmit** *number-of-retries*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Router(config)# aaa new-model`	Enables the AAA access control model. • DHCP accounting functions only in the access control model. **Note** TACACS and extended TACACS commands are not available after this command is configured and are not supported by DHCP accounting.
Step 4	**aaa group server radius** *group-name* **Example:** `Router(config)# aaa group server radius RGROUP-1`	Creates a server group for AAA or TACACS+ services and enters server group configuration mode. • The server group is created in this step so that accounting services can be applied.
Step 5	**server** *ip-address* **auth-port** *port-number* **acct-port** *port-number* **Example:** `Router(config-sg-radius)# server 10.0.0.1 auth-port 1645 acct-port 1646`	Specifies the servers that are members of the server group that was created in Step 4. • You must open port numbers for authorization and accounting. 1645 is the default port number for authorization, and 1646 is the default port number for accounting. The range of port numbers that can be specified is from 0 to 65535. • The values entered for the **auth-port** *port-number* and **acct-port** *port-number* keywords and arguments must match the port numbers that will be configured in Step 10.
Step 6	**exit** **Example:** `Router(config-sg-radius)# exit`	Exits server group configuration mode and enters global configuration mode.

	Command or Action	Purpose
Step 7	**aaa accounting** {**system** | **network** | **exec** | **connection** | **commands** *level*} {**default** | *list-name*} {**start-stop** | **stop-only** | **none**} [**broadcast**] **group** *group-name* **Example:** `Router(config)# aaa accounting network RADIUS-GROUP1 start-stop group RGROUP-1`	Configures RADIUS accounting for the specified server group. • The RADIUS accounting server is specified in the first *list-name* argument (RADIUS-GROUP1), and the target server group is specified in the second *group-name* argument (RGROUP-1). • This command enables start and stop accounting for DHCP accounting. The **start-stop** keyword enables the transmission of both START and STOP accounting messages. The **stop-only** keyword will enable the generation and verification of STOP accounting messages only.
Step 8	**aaa session-id** {**common** | **unique**} **Example:** `Router(config)# aaa session-id common`	Specifies whether the same session ID will be used for each AAA accounting service type within a call or whether a different session ID will be assigned to each accounting service type.
Step 9	**ip radius source-interface** *type-number* [**vrf** *vrf-name*] **Example:** `Router(config)# ip radius source-interface Ethernet 0`	Forces RADIUS to use the IP address of the specified interface for all outgoing RADIUS packets.
Step 10	**radius-server host** {*hostname* | *ip-address*} [**auth-port** *port-number*] [**acct-port** *port-number*] **Example:** `Router(config)# radius-server host 10.1.1.1 auth-port 1645 acct-port 1646`	Specifies the radius server host. • The values entered for the **auth-port** *port-number* and **acct-port** *port-number* keywords and arguments must match the port numbers that were configured in Step 5.
Step 11	**radius-server retransmit** *number-of-retries* **Example:** `Router(config)# radius-server retransmit 3`	Specifies the number of times that Cisco IOS software will look for RADIUS server hosts.

Troubleshooting Tips

To monitor and troubleshoot the configuration of RADIUS accounting, use the following command:

Command	Purpose
debug radius accounting **Example:** `Router# debug radius accounting`	The **debug radius** command is used to display RADIUS events on the console of the router. These events provide information about RADIUS processes. DHCP accounting information can be filtered with the **accounting** keyword. START and STOP accounting message information will also be displayed.

Configuring DHCP Accounting

Perform this task to configure DHCP accounting.

DHCP Accounting

DHCP accounting is enabled with the **accounting** DHCP pool configuration command. This command configures DHCP to operate with AAA and RADIUS to enable secure START and STOP accounting messages. This configuration adds a layer of security that allows DHCP lease assignment and termination to be triggered for the appropriate RADIUS START and STOP accounting records so that the session state is properly maintained by upstream devices, such as the SSG.

DHCP accounting is configured on a per-client or per-lease basis. Separate DHCP accounting processes can be configured on a per-pool basis.

Prerequisites

You must configure an SSG for client authentication. AAA and RADIUS must be enabled before DHCP accounting will operate.

Restrictions

The following restrictions apply to DHCP accounting:

- DHCP accounting can be configured only for DHCP network pools in which bindings are created automatically and destroyed upon lease termination or when the client sends a DHCPRELEASE message.
- DHCP bindings are destroyed when the **clear ip dhcp binding** or **no service dhcp** commands are entered, which also triggers an accounting STOP message. You should exercise caution when entering these commands if a pool is configured with DHCP accounting, as these commands will clear active leases.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp pool** *pool-name*
4. **accounting** *method-list-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip dhcp pool** *pool-name* **Example:** `Router(config)# ip dhcp pool WIRELESS-POOL`	Configures a DHCP address pool and enters DHCP pool configuration mode.
Step 4	**accounting** *method-list-name* **Example:** `Router(dhcp-config)# accounting RADIUS-GROUP1`	Enables DHCP accounting if the specified server group is configured to run RADIUS accounting. • The example configures DHCP accounting START and STOP messages to be sent if RADIUS-GROUP1 is configured as a start-stop group. STOP messages will only be sent if RADIUS-GROUP1 is configured as a stop-only group. See Step 7 in the Configuring AAA and RADIUS for DHCP Accounting configuration task table for more details.

Verifying DHCP Accounting

Perform this task to verify the DHCP accounting configuration.

The **debug radius**, **debug ip dhcp server events**, **debug aaa accounting**, **debug aaa id** commands do not need to be issued together or in the same session as there are differences in the information that is provided. These commands, however, can be used to display DHCP accounting start and stop events, AAA accounting messages, and information about AAA and DHCP hosts and clients. See the "RADIUS Accounting Attributes" section of this module for a list of AAA attributes that have been introduced by DHCP accounting. The **show running-config | begin dhcp** command can be used to display the local DHCP configuration including the configuration of DHCP accounting.

SUMMARY STEPS

1. **enable**
2. **debug radius accounting**
3. **debug ip dhcp server events**
4. **debug aaa accounting**
5. **debug aaa id**
6. **show running-config | begin dhcp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug radius accounting** **Example:** `Router# debug radius accounting`	Displays RADIUS events on the console of the router. • These events provide information about RADIUS processes. DHCP accounting information can be filtered with the **accounting** keyword. START and STOP accounting messages will be displayed in the output.
Step 3	**debug ip dhcp server events** **Example:** `Router# debug ip dhcp server events`	Displays DHCP IP address assignments, DHCP lease expirations, and DHCP database changes.
Step 4	**debug aaa accounting** **Example:** `Router# debug aaa accounting`	Displays AAA accounting events. • START and STOP accounting messages will be displayed in the output.
Step 5	**debug aaa id** **Example:** `Router# debug aaa id`	Displays AAA events as they relate to unique AAA session IDs.
Step 6	**show running-config | begin dhcp** **Example:** `Router# show running-config | begin dhcp`	The **show running-config** command is used to display the local configuration of the router. The sample output is filtered with the **begin** keyword to start displaying output at the DHCP section of the running configuration.

Securing ARP Table Entries to DHCP Leases

Perform this task to secure ARP table entries to DHCP leases in the DHCP database.

When the **update arp** command is used, ARP table entries and their corresponding DHCP leases are secured automatically for all new leases and DHCP bindings. However, existing active leases are not secured. These leases are still insecure until they are renewed. When the lease is renewed, it is treated as a new lease and will be secured automatically. If this command is disabled on the DHCP server, all existing secured ARP table entries will automatically change to dynamic ARP entries.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp pool** *pool-name*
4. **update arp**

5. **renew deny unknown**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip dhcp pool** *pool-name* **Example:** `Router(config)# ip dhcp pool WIRELESS-POOL`	Configures a DHCP address pool and enters DHCP pool configuration mode.
Step 4	**update arp** **Example:** `Router(dhcp-config)# update arp`	Secures insecure ARP table entries to the corresponding DHCP leases. • Existing active DHCP leases will not be secured until they are renewed. Using the **no update arp** command will change secured ARP table entries back to dynamic ARP table entries.
Step 5	**renew deny unknown** **Example:** `Router(dhcp-config)# renew deny unknown`	(Optional) Configures the renewal policy for unknown clients. • See the "Troubleshooting Tips" section for information about when to use this command.

Troubleshooting Tips

In some usage scenarios, such as a wireless hotspot, where both DHCP and secure ARP are configured, a connected client device might go to sleep or suspend for a period of time. If the suspended time period is greater than the secure ARP timeout (default of 91 seconds), but less than the DHCP lease time, the client can awake with a valid lease, but the secure ARP timeout has caused the lease binding to be removed because the client has been inactive. When the client awakes, the client still has a lease on the client side but is blocked from sending traffic. The client will try to renew its IP address but the DHCP server will ignore the request because the DHCP server has no lease for the client. The client must wait for the lease to expire before being able to recover and send traffic again.

To remedy this situation, use the **renew deny unknown** command in DHCP pool configuration mode. This command forces the DHCP server to reject renewal requests from clients if the requested address is present at the server but is not leased. The DHCP server sends a DHCPNAK denial message to the client, which forces the client back to its initial state. The client can then negotiate for a new lease immediately, instead of waiting for its old lease to expire.

Configuring DHCP Authorized ARP

Perform this task to configure DHCP authorized ARP, which disables dynamic ARP learning on an interface.

ARP Probing Behavior

DHCP authorized ARP has a limitation in supporting accurate one-minute billing. DHCP authorized ARP probes for authorized users once or twice, 30 seconds apart. In a busy network the possibility of missing reply packets increases, which can cause a premature log off. If you need a more accurate and finer control for probing of the authorized user, configure the **arp probe interval** command. This command specifies when to start a probe, the interval between unsuccessful probes, and the maximum number of retries before triggering an automatic log off.

Restrictions

If both static and authorized ARP are installing the same ARP entry, static configuration overrides authorized ARP. You can install a static ARP entry by using the **arp** global configuration command. You can only remove a nondynamic ARP entry by the same method in which it was installed.

The ARP timeout period should not be set to less than 30 seconds. The feature is designed to send out an ARP message every 30 seconds, beginning 90 seconds before the ARP timeout period specified by the **arp timeout** command. This behavior allows probing for the client at least three times before giving up on the client. If the ARP timeout is set to 60 seconds, an ARP message is sent twice, and if it is set to 30 seconds, an ARP message is sent once. An ARP timeout period set to less than 30 seconds can yield unpredictable results.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask*

5. **arp authorized**
6. **arp timeout** *seconds*
7. **arp probe interval** *seconds* **count** *number*
8. **end**
9. **show arp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface ethernet 1`	Configures an interface type and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 168.71.6.23 255.255.255.0`	Sets a primary IP address for an interface.
Step 5	**arp authorized** **Example:** `Router(config-if)# arp authorized`	Disables dynamic ARP learning on an interface. • The IP address to MAC address mapping can only be installed by the authorized subsystem.
Step 6	**arp timeout** *seconds* **Example:** `Router(config-if)# arp timeout 60`	Configures how long an entry remains in the ARP cache. • Do not set the timeout period to less than 30 seconds as discussed in the "Restrictions" section.
Step 7	**arp probe interval** *seconds* **count** *number* **Example:** `Router(config-if)# arp probe interval 5 count 30`	(Optional) Specifies an interval, in seconds, and number of probe retries. The arguments are as follows: • *seconds*—Interval, in seconds, after which the next probe will be sent to see if a peer is present. The range is from 1 to 10. • *count-number*—Number of probe retries. If there is no reply after the count has been reached, the peer has logged off. The range is from 1 to 60. **Note** You must use the **no** form of the command to stop the probing process.

	Command or Action	Purpose
Step 8	**end** **Example:** `Router(config-if)# end`	Exits the configuration mode and returns to privileged EXEC mode.
Step 9	**show arp** **Example:** `Router# show arp`	(Optional) Displays the entries in the ARP table.

Configuring a DHCP Lease Limit to Globally Control the Number of Subscribers

Perform this task to globally control the number of DHCP leases allowed for clients behind an ATM RBE unnumbered interface or serial unnumbered interface.

This feature allows an ISP to globally limit the number of leases available to clients per household or connection.

If this feature is enabled on a Cisco IOS DHCP relay agent connected to clients through unnumbered interfaces, the relay agent keeps information about the DHCP leases offered to the clients per subinterface. When a DHCPACK message is forwarded to the client, the relay agent increments the number of leases offered to clients on that subinterface. If a new DHCP client tries to obtain an IP address and the number of leases has already reached the configured lease limit, DHCP messages from the client will be dropped and will not be forwarded to the DHCP server.

If this feature is enabled on the Cisco IOS DHCP server directly connected to clients through unnumbered interfaces, the server allocates addresses and increments the number of leases per subinterface. If a new client tries to obtain an IP address, the server will not offer an IP address if the number of leases on the subinterface has already reached the configured lease limit.

Restrictions for the DHCP Lease Limit

This feature is not supported on numbered interfaces. The lease limit can be applied only to ATM with RBE unnumbered interfaces or serial unnumbered interfaces.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp limit lease log**
4. **ip dhcp limit lease per interface** *lease-limit*
5. **end**
6. **show ip dhcp limit lease** [*type number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`enable`** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**`configure terminal`** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**`ip dhcp limit lease log`** **Example:** `Router(config)# ip dhcp limit lease log`	(Optional) Enables DHCP lease violation logging when a DHCP lease limit threshold is exceeded. • If this command is configured, any lease limit violations will display in the output of the **show ip dhcp limit lease** command.
Step 4	**`ip dhcp limit lease per interface`** *`lease-limit`* **Example:** `Router(config)# ip dhcp limit lease per interface 2`	Limits the number of leases offered to DHCP clients behind an ATM RBE unnumbered or serial unnumbered interface.
Step 5	**`end`** **Example:** `Router(config)# interface FastEthernet0/0`	Exits the configuration mode and returns to privileged EXEC mode.
Step 6	**`show ip dhcp limit lease`** [*`type number`*] **Example:** `Router# show ip dhcp limit lease`	(Optional) Displays the number of times the lease limit threshold has been violated. • You can use the **clear ip dhcp limit lease** privileged EXEC command to manually clear the stored lease violation entries.

Troubleshooting Tips

You can use the **debug ip dhcp server packet** and **debug ip server events** commands to troubleshoot the DHCP lease limit.

Configuring a DHCP Lease Limit to Control the Number of Subscribers on an Interface

Perform this task to limit the number of DHCP leases allowed on an interface.

This feature allows an ISP to limit the number of leases available to clients per household or connection on an interface.

If this feature is enabled on the Cisco IOS DHCP server directly connected to clients through unnumbered interfaces, the server allocates addresses and increments the number of leases per subinterface. If a new client tries to obtain an IP address, the server will not offer an IP address if the number of leases on the subinterface has already reached the configured lease limit.

Restrictions

This feature is not supported on numbered interfaces. The lease limit can be applied only to ATM with RBE unnumbered interfaces or serial unnumbered interfaces.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp limit lease log**
4. **interface** *type number*
5. **ip dhcp limit lease** *lease-limit*
6. **end**
7. **show ip dhcp limit lease** [*type number*]
8. **show ip dhcp server statistics** [*type number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`enable`** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**`configure terminal`** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**`ip dhcp limit lease log`** **Example:** `Router(config)# ip dhcp limit lease log`	(Optional) Enables DHCP lease violation logging when a DHCP lease limit threshold is exceeded. • If this command is configured, any lease limit violations will display in the output of the **show ip dhcp limit lease** command.
Step 4	**`interface`** *`type number`* **Example:** `Router(config)# interface Serial0/0`	Enters interface configuration mode.
Step 5	**`ip dhcp limit lease`** *`lease-limit`* **Example:** `Router(config-if)# ip dhcp limit lease 6`	Limits the number of leases offered to DHCP clients per interface. • The interface configuration will override any global setting specified by the **ip dhcp limit lease per interface** global configuration command.
Step 6	**`end`** **Example:** `Router(config-if)# end`	Exits the configuration mode and returns to privileged EXEC mode.
Step 7	**`show ip dhcp limit lease`** `[`*`type number`*`]` **Example:** `Router# show ip dhcp limit lease Serial0/0`	(Optional) Displays the number of times the lease limit threshold has been violated. • You can use the **clear ip dhcp limit lease** privileged EXEC command to manually clear the stored lease violation entries.
Step 8	**`show ip dhcp server statistics`** `[`*`type number`*`]` **Example:** `Router# show ip dhcp server statistics Serial0/0`	(Optional) Displays DHCP server statistics. • This command was modified in Cisco IOS Release 12.2(33)SRC to display interface-level DHCP statistics.

Troubleshooting Tips

You can use the **debug ip dhcp server packet** and **debug ip server events** commands to troubleshoot the DHCP lease limit.

Configuration Examples for DHCP Services for Accounting and Security

This section provides the following configuration examples:

Configuring AAA and RADIUS for DHCP Accounting: Example

The following example shows how to configure AAA and RADIUS for DHCP accounting:

```
aaa new-model
aaa group server radius RGROUP-1
 server 10.1.1.1 auth-port 1645 acct-port 1646
 exit
aaa accounting network RADIUS-GROUP1 start-stop group RGROUP-1
aaa session-id common
ip radius source-interface Ethernet0
radius-server host 10.1.1.1 auth-port 1645 acct-port 1646
radius-server retransmit 3
exit
```

Configuring DHCP Accounting: Example

DHCP accounting is configured on a per-client or per-lease basis. Separate DHCP accounting processes can be configured on a per-pool basis. The following example shows how to configure DHCP accounting START and STOP messages to be sent if RADIUS-GROUP1 is configured as a start-stop group.

```
ip dhcp pool WIRELESS-POOL
 accounting RADIUS-GROUP1
 exit
```

Verifying DHCP Accounting: Example

DHCP accounting is enabled after both RADIUS and AAA for DHCP are configured. DHCP START and STOP accounting generation information can be monitored with the **debug radius accounting** and **debug ip dhcp server events** commands. See the "RADIUS Accounting Attributes" section on page 4 of this module for a list of AAA attributes that have been introduced by DHCP accounting.

The following is sample output from the **debug radius accounting** command. The output shows the DHCP lease session ID, the MAC address, and the IP address of the client interface.

```
00:00:53: RADIUS: Pick NAS IP for uid=2 tableid=0 cfg_addr=10.0.18.3 best_addr=0.0.0.0
00:00:53: RADIUS(00000002): sending
00:00:53: RADIUS(00000002): Send to unknown id 21645/1 10.1.1.1 :1646, Accounting-Request,
len 76
00:00:53: RADIUS: authenticator C6 FE EA B2 1F 9A 85 A2 - 9A 5B 09 B5 36 B5 B9 27
```

```
00:00:53: RADIUS: Acct-Session-Id [44] 10 "00000002"
00:00:53: RADIUS: Framed-IP-Address [8] 6 10.0.0.10
00:00:53: RADIUS: Calling-Station-Id [31] 16 "00000c59df76"
00:00:53: RADIUS: Acct-Status-Type [40] 6 Start [1]
00:00:53: RADIUS: Service-Type [6] 6 Framed [2]
00:00:53: RADIUS: NAS-IP-Address [4] 6 10.0.18.3
00:00:53: RADIUS: Acct-Delay-Time [41] 6 0
```

The following is sample output from the **debug ip dhcp server events** command. The output was generated on a DHCP server and shows an exchange of DHCP messages between the client and server to negotiate a DHCP lease. The acknowledgment that confirms to the DHCP server that the client has accepted the assigned IP address triggers the accounting START message. It is shown in the last line of the following output:

```
00:45:50:DHCPD:DHCPDISCOVER received from client
0063.6973.636f.2d30.3030.312e.3432.6339.2e65.6337.352d.4574.31 on
interface Ethernet0.

00:45:52:DHCPD:assigned IP address 10.10.10.16 to client
0063.6973.636f.2d30.3030.312e.3432.6339.2e65.6337.352d.4574.31.

00:45:52:DHCPD:Sending DHCPOFFER to client
0063.6973.636f.2d30.3030.312e.3432.6339.2e65.6337.352d.4574.31(10.10.10.16)

00:45:52:DHCPD:broadcasting BOOTREPLY to client 0001.42c9.ec75.

00:45:52:DHCPD:DHCPREQUEST received from client
0063.6973.636f.2d30.3030.312e.3432.6339.2e65.6337.352d.4574.31.

00:45:52:DHCPD:Sending DHCPACK to client
0063.6973.636f.2d30.3030.312e.3432.6339.2e65.6337.352d.4574.31
(10.10.10.16).

00:45:52:DHCPD:broadcasting BOOTREPLY to client 0001.42c9.ec75.

00:45:52:DHCPD:triggered Acct Start for 0001.42c9.ec75 (10.10.10.16).
```

The following is sample output from the **debug ip dhcp server events** command. The output was generated on a DHCP server and shows the receipt of an explicit release message from the DHCP client. The DHCP server triggers an accounting STOP message and then returns the IP address to the DHCP pool. Information about the accounting STOP message is shown in the third line of the following output:

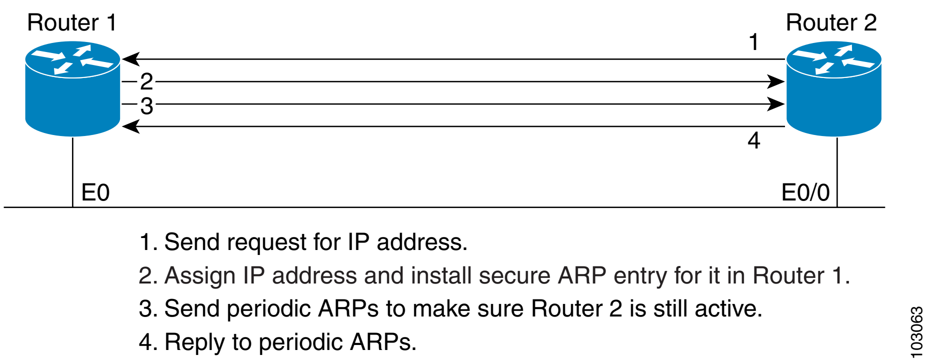
```
00:46:26:DHCPD:DHCPRELEASE message received from client
0063.6973.636f.2d30.3030.312e.3432.6339.2e65.6337.352d.4574.31 (10.10.10.16)

00:46:26:DHCPD:triggered Acct Stop for (10.10.10.16).

00:46:26:DHCPD:returned 10.10.10.16 to address pool WIRELESS-POOL.
```

Configuring DHCP Authorized ARP: Example

Router 1 is the DHCP server that assigns IP addresses to the routers that are seeking IP addresses, and Router 2 is the DHCP client configured to obtain its IP address through the DHCP server. Because the **update arp** DHCP pool configuration command is configured on Router 1, it will install a secure ARP entry in its ARP table. The **arp authorized** command stops any dynamic ARP on that interface. Router 1 will send periodic ARPs to Router 2 to make sure that the client is still active. Router 2 responds with an ARP reply. Unauthorized clients cannot respond to these periodic ARPs. The unauthorized ARP responses are blocked at the DHCP server. The timer for the entry is refreshed on Router 1 upon receiving the response from the authorized client.

See Figure 1 for an example topology.

Figure 1 Example Topology for DHCP Authorized ARP



Router 1 (DHCP Server)

```
ip dhcp pool name1
 network 10.0.0.0 255.255.255.0
 lease 0 0 20
 update arp
!
interface Ethernet0
 ip address 10.0.0.1 255.255.255.0
 half-duplex
 arp authorized
 arp timeout 60
! optional command to adjust the periodic ARP probes sent to the peer
 arp probe interval 5 count 15
```

Router 2 (DHCP Client)

```
interface Ethernet0/0
 ip address dhcp
 half-duplex
```

Verifying DHCP Authorized ARP: Example

The following is sample output for the **show arp** command on Router 1:

```
Router1# show arp

Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  10.0.0.3                 0  0004.dd0c.ffcb  ARPA   Ethernet01
Internet  10.0.0.1                 -  0004.dd0c.ff86  ARPA   Ethernet0
```

The following is the output for the **show arp** command on Router 2:

```
Router2# show arp

Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  10.0.0.3                 -  0004.dd0c.ffcb  ARPA   Ethernet0/02
Internet  10.0.0.1                 0  0004.dd0c.ff86  ARPA   Ethernet0/0
```

Configuring a DHCP Lease Limit: Examples

In the following example, if more than three clients try to obtain an IP address from interface ATM4/0.1, the DHCPDISCOVER packets will not be forwarded to the DHCP server. If the DHCP server resides on the same router, DHCP will not reply to more than three clients.

```
ip dhcp limit lease per interface 3
!
interface loopback0
 ip address 10.1.1.129 255.255.255.192
!
interface ATM4/0.1
 no ip address
!
interface ATM4/0.1 point-to-point
 ip helper-address 172.16.1.2
 ip unnumbered loopback0
 atm route-bridged ip
  pvc 88/800
  encapsulation aal5snap
```

In the following example, 5 DHCP clients are allowed to receive IP addresses. If a sixth client tries to obtain an IP address, the DHCPDISCOVER messages will not be forwarded to the DHCP server and a trap will be sent to the SNMP manager.

```
ip dhcp limit lease log
!
ip dhcp pool pool1
 network 10.1.1.0 255.255.255.0
!
interface loopback0
 ip address 10.1.1.1 255.255.255.0
!
interface serial 0/0.2 point-to-point
 ip dhcp limit lease 5
 ip unnumbered loopback0
 exit
snmp-server enable traps dhcp interface
```

Additional References

The following sections provide references related to configuring DHCP services for accounting and security.

Related Documents

Related Topic	Document Title
ARP commands: complete command syntax, command modes, defaults, usage guidelines, and examples	*Cisco IOS IP Addressing Services Command Reference*
DHCP commands: complete command syntax, command modes, defaults, usage guidelines, and examples	*Cisco IOS IP Addressing Services Command Reference*
DHCP conceptual information	"DHCP Overview" module

Related Topic	Document Title
DHCP server configuration	"Configuring the Cisco IOS DHCP Server" module
DHCP ODAP configuration	"Configuring the DHCP Server On-Demand Address Pool Manager" module
DHCP client configuration	"Configuring the Cisco IOS DHCP Client" module
DHCP relay agent configuration	"Configuring the Cisco IOS DHCP Relay Agent" module
DHCP enhancements for edge-session management	"Configuring DHCP Enhancements for Edge-Session Management" module
AAA and RADIUS configuration tasks	*Cisco IOS Security Configuration Guide*
AAA and RADIUS commands: complete command syntax, command mode, defaults, usage guidelines, and examples	*Cisco IOS Security Command Reference*

Standards

Standards	Title
No new or modified standards are supported by this functionality.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for DHCP Services for Accounting and Security

Table 2 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Releases 12.2(1) or later appear in the table.

For information on a feature in this technology that is not documented here, see the "DHCP Features Roadmap".

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 2 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 2 Feature Information for DHCP Services for Accounting and Security

Feature Name	Releases	Feature Configuration Information
DHCP Per Interface Lease Limit and Statistics	12.2(33)SRC Cisco IOS XE Release 2.1	This feature limits the number of DHCP leases offered to DHCP clients on an interface. DHCP server statistics reporting was enhanced to display interface-level statistics. The following sections provide information about this feature: • Configuring a DHCP Lease Limit to Control the Number of Subscribers on an Interface • Configuring a DHCP Lease Limit: Examples The following commands were introduced or modified by this feature: **ip dhcp limit lease**, **ip dhcp limit lease log**, **clear ip dhcp limit lease**, **show ip dhcp limit lease**, and **show ip dhcp server statistics**.
DHCP Lease Limit per ATM RBE Unnumbered Interface	12.3(2)T 12.2(28)SB	This feature limits the number of DHCP leases per subinterface offered to DHCP clients connected from an ATM RBE unnumbered interface or serial unnumbered interface of the DHCP server or DHCP relay agent. The following section provides information about this feature: • Configuring a DHCP Lease Limit to Globally Control the Number of Subscribers The following command was introduced by this feature: **ip dhcp limit lease per interface**.

Table 2 Feature Information for DHCP Services for Accounting and Security

Feature Name	Releases	Feature Configuration Information
ARP Auto-logoff	12.3(14)T	The ARP Auto-logoff feature enhances DHCP authorized ARP by providing finer control and probing of authorized clients to detect a log off. The following sections provide information about this feature: • DHCP Services for Security and Accounting Overview • Configuring DHCP Authorized ARP • Configuring DHCP Authorized ARP: Example The following command was introduced by this feature: **arp probe interval**.
DHCP Authorized ARP	12.3(4)T 12.2(33)SRC	DHCP authorized ARP enhances the DHCP and ARP components of the Cisco IOS software to limit the leasing of IP addresses to mobile users to authorized users. This feature enhances security in PWLANs by blocking ARP responses from unauthorized users at the DHCP server. The following sections provide information about this feature: • DHCP Services for Security and Accounting Overview • Configuring DHCP Authorized ARP • Configuring DHCP Authorized ARP: Example The following command was introduced by this feature: **arp authorized**.

Table 2 Feature Information for DHCP Services for Accounting and Security

Feature Name	Releases	Feature Configuration Information
DHCP Accounting	12.2(15)T 12.2(28)SB 12.2(33)SRB	DHCP accounting introduces AAA and RADIUS support for DHCP configuration. The following sections provide information about this feature: • DHCP Services for Security and Accounting Overview • Configuring DHCP Accounting The following command was introduced by this feature: **accounting**.
DHCP Secured IP Address Assignment	12.2(15)T 12.2(28)SB 12.2(33)SRC	DHCP secure IP address assignment provides the capability to secure ARP table entries to DHCP leases in the DHCP database. This feature secures and synchronizes the MAC address of the client to the DHCP binding, preventing unauthorized clients or hackers from spoofing the DHCP server and taking over a DHCP lease of an authorized client. The following sections provide information about this feature: • DHCP Services for Security and Accounting Overview • Securing ARP Table Entries to DHCP Leases The following command was introduced by this feature: **update arp**. The following command was modified by this feature: **show ip dhcp server statistics**.

Configuring DHCP Enhancements for Edge-Session Management

The DHCP Enhancements for Edge-Session Management feature provides the capability of simultaneous service by multiple Internet Service Providers (ISPs) to customers using one network infrastructure. The end-user customer may change ISPs at any time.

The DHCP enhancements evolved out of the Service Gateways (SGs) requirement to receive information from the DHCP server about when client DISCOVER packets (session initiation) are received, when an address has been allocated to a client, and when a client has released a DHCP lease or the lease has expired (session termination).

Module History

This module was first published on March 29, 2005, and last updated on December 31, 2007.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. For the latest feature information and caveats, see the release notes for your Cisco IOS software release. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for DHCP Enhancements for Edge-Session Management" section on page 22.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Information About DHCP Enhancements for Edge-Session Management

To configure the DHCP Enhancements for Edge-Session Management feature, you should understand the following concepts:

DHCP Servers and Relay Agents

DHCP provides a framework for passing configuration information dynamically to hosts on a TCP/IP network. A DHCP client is an Internet host using DHCP to obtain configuration parameters such as an IP address.

A DHCP relay agent is any host that forwards DHCP packets between clients and servers. Relay agents are used to forward requests and replies between clients and servers when they are not on the same physical subnet. Relay agent forwarding is distinct from the normal forwarding of an IP router, where IP datagrams are switched between networks somewhat transparently. By contrast, relay agents receive DHCP messages and then generate a new DHCP message to send on another interface.

For more information, refer to the DHCP modules in the *Cisco IOS IP Addressing Services Configuration Guide*, Release 12.4.

On-Demand Address Pool Management

An On-Demand Address Pool (ODAP) is used to centralize the management of large pools of addresses and simplifies the configuration of large networks. ODAP provides a central management point for the allocation and assignment of IP addresses.

When a Cisco router is configured as an ODAP manager, pools of IP addresses are dynamically increased or reduced in size depending on the address utilization level. The ODAP manager is supported by centralized Remote Authentication Dial-In User Service (RADIUS) or DHCP servers and is configured to request an initial pool of addresses from either the RADIUS or DHCP server.

The ODAP manager controls IP address assignment and will allocate additional IP addresses as necessary. This method of address allocation and assignment optimizes the use of available address space and simplifies the configuration of medium and large-sized networks.

For more information, see the "Configuring the DHCP Server On-Demand Address Pool Manager" module.

Design of the DHCP Enhancements for Edge-Session Management Feature

With the DHCP Enhancements for Edge-Session Management feature, a DHCP server and relay agent are separate, but closely coupled. The basic design of the feature encompasses two types of configuration at the edge of an ISP network as follows:

- DHCP server and an SG that are co-resident (in the same device)
- DHCP relay agent and an SG that are co-resident

DHCP Server Co-Resident with the SG

With this configuration, the DHCP server is in the same device as the SG and allocates addresses from locally configured address pools or acquires a subnet of addresses to allocate from some other system in the network. There are no changes to the server address allocation function to support the configuration.

This configuration enables the DHCP server to notify the SG that it has received a broadcast sent by the end-user DHCP client. The SG passes the MAC address and other information to the DHCP server. The SG also passes a class name (for example, the name of the ISP), which is used by the DHCP server to match a pool-class definition.

Lease-state notifications are always made by the DHCP server to the SG, because the information is already present.

Note The local configuration may also be performed by an ODAP that acquires subnets for the address pools from another DHCP server or a RADIUS server.

DHCP Relay Agent Co-Resident with the SG

With this configuration, the relay agent is in the same device as the SG and intercedes in DHCP sessions to appear as the DHCP server to the DHCP client. As the server, the relay agent may obtain enough information about the DHCP session to notify the SG of all events (for example, lease termination).

Appearing to be the DHCP server is performed by using the DHCP functionality that is currently in use on unnumbered interfaces. This functionality enables the relay agent to substitute its own IP address for the server.

The packet is passed by the relay agent to the DHCP server and the SG is notified of the receipt. Following the notification, an inquiry is made by the relay agent to the SG about which DHCP class name to use. Then, the packet is passed by the relay agent to the selected DHCP server.

The end-user DHCP client MAC address and other pertinent information is passed to the SG. The SG returns the DHCP class name to use when matching a DHCP pool if the SG is configured to do so. If the DHCP relay agent is not acting as a server, it relays the packet to the DHCP server.

Note An address pool may have one DHCP class defined to specify one central DHCP server to which the relay agent passes the packet, or it may have multiple DHCP classes defined to specify a different DHCP server for each client.

Benefits of the DHCP Enhancements for Edge-Session Management

The benefits of the DHCP Enhancements for Edge-Session Management feature are as follows:

- Allows the full DHCP server system to be located farther inside the network, while only running a relatively simple DHCP relay agent at the edge.
- Simplifies the DHCP configuration at the edge.
- Allows all DHCP server administration to occur closer to the middle of the network on one centralized DHCP server, or on separate DHCP servers (one for each ISP).
- Allows each ISP full control over all DHCP options and lease times.
- Allows both the DHCP server and client configurations to be used on the same edge system simultaneously.

How to Configure DHCP Enhancements for Edge-Session Management

This section contains the following procedures:

- Configuring the DHCP Address Pool and a Class Name, page 4 (optional)
- Configuring a Relay Pool with a Relay Source and Destination, page 6 (required)
- Configuring a Relay Pool for a Remote DHCP Server, page 9 (required)
- Configuring Other Types of Relay Pools, page 12 (optional)

Configuring the DHCP Address Pool and a Class Name

Perform this task to configure a DHCP server that assigns addresses from an address pool for a specific class name that has been assigned by an SG that is co-resident with the DHCP server at the edge.

If a DHCP server is resident in the same device as an SG and both are at the edge, a class name and address pool should be configured. In this case, the DHCP server notifies an SG of a DISCOVER broadcast received from a client and the SG returns a class name. The returned class name designates an address range of an address pool. The DHCP server sends the MAC address and IP address of the incoming interface or the specified relay-agent address to the SG.

Note If the DHCP server has its address pools defined locally or retrieves the subnets from ISP DHCP servers or AAA servers using ODAP, additional DHCP server configuration on behalf of the SG is not required.

If dynamic allocation of the address pool is required using ODAP, the **origin** command is specified.

Prerequisites

The specification of the class name is required in the DHCP address-pool configuration and in the SG system itself to designate each DHCP client class name. A default class name should be configured if a user does not have one.

Each address pool should be associated with one or more DHCP classes (address-provider ISPs). When the DHCP client selects an ISP, the selection becomes the class name designated by the SG.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp pool** *name*
4. **origin** {**dhcp** | **file** *url*}
5. **network** *network-number* [*mask* | *prefix-length*]
6. **class** *class-name*
7. **address range** *start-ip end-ip*
8. Repeat Steps 3, 5, and 6.
9. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip dhcp pool** *name* **Example:** `Router(config)# ip dhcp pool abc-pool`	Configures a DHCP address pool on a Cisco IOS DHCP server and enters DHCP pool configuration mode. The *name* argument is the name of the pool and may either be a symbolic string (such as engineering) or an integer (such as 0).
Step 4	**origin** {**dhcp** | **file** *url*} **Example:** `Router(dhcp-config)# origin dhcp`	(Optional) Configures an address pool as an On-Demand Address Pool (ODAP) or static mapping pool. The argument and keywords are as follows:

	Command or Action	Purpose
Step 5	**network** *network-number* [*mask* | *prefix-length*] **Example:** `Router(dhcp-config)# network 10.10.0.0 255.255.0.0`	Configures the subnet number and mask for a DHCP address pool on a Cisco IOS DHCP server. The arguments are as follows: • *network-number*—The IP address of the DHCP address pool. Use this argument if ODAP is not the IP address assignment method. • *mask*—(Optional) The bit combination that renders which portion of the address of the DHCP address pool refers to the network or subnet and which part refers to the host. • *prefix-length*—(Optional) The number of bits that comprise the address prefix. The prefix is an alternative way of specifying the network mask of the client. The prefix length must be preceded by a forward slash (/).
Step 6	**class** *class-name* **Example:** `Router(dhcp-config)# class abc-pool`	Associates a class with a DHCP address pool and enters DHCP pool-class configuration mode. The *class-name* argument is the name of the class. It should match the DHCP address pool name. Repeat this step to specify a default class name if required by the SG.
Step 7	**address range** *start-ip end-ip* **Example:** `Router(config-dhcp-pool-class)# address range 10.10.5.0 10.99.99.99`	(Optional) Configures an IP address range from which the DHCP server would allocate the IP addresses. If an SG returned an IP address that is not configured, no action is taken. This step enables the allocation of an address from a range for the class name specified in the previous step. **Note** The **address range** command cannot be used with a relay pool that is configured with the **relay destination** command. Further, if no address range is assigned to a class name, the address is specified with the **network** command.
Step 8	Repeat Steps 3, 5, and 6.	If there is an interface configured with multiple subnets and different ISPs, repeat this step to match the number of subnets. See the "Multiple DHCP Pools and Different ISPs Configuration: Example" section on page 18.
Step 9	**exit** **Example:** `Router(config-dhcp-pool-class)# exit`	Exits to DHCP pool configuration mode.

Configuring a Relay Pool with a Relay Source and Destination

Perform this task to configure a relay pool when the DHCP relay and SG are resident in the same device at the edge, and all end users will obtain addresses from one pool. This task replaces the IP helper-address interface configuration.

If the SG notifies the relay agent that DHCP session notifications are required for a particular DHCP client, the relay agent will retain enough information about the DHCP session to notify the SG of all events (for example, lease termination). The relay intercedes DHCP sessions and assumes the role of the DHCP server. The IP address configuration becomes a dynamically changing value depending on the DHCP client information and the SG device policy information.

Restrictions

If a relay agent is interceding in DHCP sessions and assuming the role of the DHCP server, the use of DHCP authentication is not possible.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp pool** *name*
4. **update arp**
5. **relay source** *ip-address subnet-mask*
6. **relay destination** [**vrf** *vrf-name* | **global**] *ip-address*
7. **accounting** *method-list-name*
8. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip dhcp pool** *name* **Example:** `Router(config)# ip dhcp pool abc-pool`	Configures a DHCP address pool on a Cisco IOS DHCP server and enters DHCP pool configuration mode. The *name* argument is the name of the pool and may either be a symbolic string (such as engineering) or an integer (such as 0). More than one name may be configured.
Step 4	**update arp** **Example:** `Router(dhcp-config)# update arp`	(Optional) Configures secure and dynamic Address Resolution Protocol (ARP) entries in the ARP table to their corresponding DHCP bindings. **Note** If the system is allocating an address from an address pool, it will add secure ARP. If the system is relaying a packet using an address pool, it will also add secure ARP.
Step 5	**relay source** *ip-address subnet-mask* **Example:** `Router(dhcp-config)# relay source 10.0.0.0 255.0.0.0`	Configures the relay source. The *ip-address* and *subnet-mask* arguments are the IP address and subnet mask for the relay source. **Note** This command is similar to the **network** command in a normal DHCP network pool, because it restricts the use of the address pool to packets arriving on the interface whose configured IP address and mask matches the relay source configuration.

	Command or Action	Purpose
Step 6	**relay destination** [**vrf** *vrf-name* | **global**] *ip-address* **Example:** `Router(dhcp-config)# relay destination 10.5.5.0`	Configures the IPv4 address of a remote DHCP server to which DHCP client packets are sent. The arguments and keywords are as follows: • **vrf**—(Optional) Virtual routing and forwarding (VRF). The *vrf-name* argument is the name of the VRF associated with the relay destination IP address. • **global**—(Optional) Global IP address. Use the this keyword when the relay agent is in the global address space and the relay source is in a VRF. • *ip-address*—IP address of the relay destination. **Note** When using the **relay destination** command, the *ip-address* argument is assumed to be in the same VRF as the address pool under which the command was configured. If the relay destination IP address is in a different VRF, or in the global address space, then the **vrf** *vrf-name* or **global** keywords need to be specified.
Step 7	**accounting** *method-list-name* **Example:** `Router(dhcp-config)# accounting RADIUS-GROUP1`	(Optional) Enables DHCP accounting if the specified server group is configured to run RADIUS accounting. • AAA and RADIUS must be enabled before DHCP accounting will operate. • The example configures DHCP accounting START and STOP messages to be sent if RADIUS-GROUP1 is configured as a start-stop group. STOP messages will only be sent if RADIUS-GROUP1 is configured as a stop-only group. See “Configuring DHCP Services for Accounting and Security” module for more information on DHCP accounting.
Step 8	**exit** **Example:** `Router(dhcp-config)# exit`	Exits to global configuration mode.

Configuring a Relay Pool for a Remote DHCP Server

Perform this task to use an SG-supplied class name when selecting the remote DHCP server in a configured relay pool, which is used to specify how DHCP client packets should be relayed. Multiple configurations of relay targets may appear in a pool-class definition in which case all addresses are used for relay purposes.

Restrictions

The **relay source** command cannot be used with the **network** command or **origin** command since those commands implicitly designate the incoming interface and are used to define a different type of pool. It associates the relay only with an interface in the same way that the **ip helper-address** command does by its presence as an interface configuration command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp pool** *name*
4. **relay source** *ip-address subnet-mask*
5. **relay destination** [**vrf** *vrf-name* | **global**] *ip-address*
6. **accounting** *method-list-name*
7. **class** *class-name*
8. **relay target** [**vrf** *vrf-name* | **global**] *ip-address*
9. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`enable`** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**`configure terminal`** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**`ip dhcp pool`** *`name`* **Example:** `Router(config)# ip dhcp pool abc-pool`	Configures a DHCP address pool on a Cisco IOS DHCP server and enters DHCP pool configuration mode. The *name* argument is the name of the pool and may either be a symbolic string (such as engineering) or an integer (such as 0). You may specify more than one DHCP address pool.
Step 4	**`relay source`** *`ip-address subnet-mask`* **Example:** `Router(dhcp-config)# relay source 10.0.0.0 255.0.0.0`	Configures the relay source. The *ip-address* and *subnet-mask* arguments are the IP address and subnet mask for the relay source. **Note** This command is similar to the **network** command in a normal DHCP network pool, because it restricts the use of the address pool to packets arriving on the interface whose configured IP address and mask matches the relay source configuration.

	Command or Action	Purpose
Step 5	**relay destination** [**vrf** *vrf-name* | **global**] *ip-address* **Example:** `Router(dhcp-config)# relay destination 10.5.5.0`	Configures the IPv4 address of a remote DHCP server to which DHCP client packets are sent. The arguments and keywords are as follows: • **vrf**—(Optional) Virtual routing and forwarding (VRF). The *vrf-name* argument is the name of the VRF associated with the relay destination IP address. • **global**—(Optional) Global IP address. Use the this keyword when the relay agent is in the global address space and the relay source is in a VRF. • *ip-address*—IP address of the relay destination. **Note** When using the **relay destination** command, the *ip-address* argument is assumed to be in the same VRF as the address pool under which the command was configured. If the relay destination IP address is in a different VRF, or in the global address space, then the **vrf** *vrf-name* or **global** keywords need to be specified.
Step 6	**accounting** ***method-list-name*** **Example:** `Router(dhcp-config)# accounting RADIUS-GROUP1`	(Optional) Enables DHCP accounting if the specified server group is configured to run RADIUS accounting. • AAA and RADIUS must be enabled before DHCP accounting will operate. • The example configures DHCP accounting START and STOP messages to be sent if RADIUS-GROUP1 is configured as a start-stop group. STOP messages will only be sent if RADIUS-GROUP1 is configured as a stop-only group. See "Configuring DHCP Services for Accounting and Security" module for more information on DHCP accounting.
Step 7	**class** *class-name* **Example:** `Router(dhcp-config)# class abc-pool`	Associates a class with a DHCP address pool and enters DHCP pool-class configuration mode. The *class-name* argument is the name of the class. You may configure more than one class name.

	Command or Action	Purpose
Step 8	`relay target [vrf vrf-name | global] ip-address` **Example:** `Router(config-dhcp-pool-class)# relay target 10.0.0.0`	Configures the relay target IP address. The arguments and keywords are as follows: • **vrf**—(Optional) Virtual routing and forwarding (VRF). The *vrf-name* argument is the name of VRF associated with the relay target IP address and more than one target may be specified. • **global**—(Optional) Global IP address space. • *ip-address*—IP address of the relay target. More than one target IP address may be specified. **Note** This command specifies the destination for the relay function in the same manner as the **ip helper-address** command. **Note** When using the **relay target** command, the *ip-address* argument is assumed to be in the same VRF as the address pool under which the command was configured. If the relay target IP address is in a different VRF, or in the global address space, then the **vrf** *vrf-name* or **global** keywords need to be specified.
Step 9	`exit` **Example:** `Router(config-dhcp-pool-class)# exit`	Exits to DHCP pool configuration mode.

Configuring Other Types of Relay Pools

This section contains the following procedures:

- Configuring Relay Information for an Address Pool, page 12 (required)
- Configuring Multiple Relay Sources for a Relay Pool, page 14 (required)

Configuring Relay Information for an Address Pool

Perform this task to configure relay information for an address pool. In this configuration, the SG sends one class name that results in the DISCOVER packet being relayed to a server at the IP address configured using the **relay target** command. If the SG sends a class name that is not configured as being associated with the address pool, then no action is taken.

Restrictions

Specifying the **address range** command and **relay target** command in a pool-class definition is not possible, because this would allocate an address and relay for the same packet.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **ip dhcp pool** *name*
4. **network** *network-number* [*mask* | *prefix-length*]
5. **class** *class-name*
6. **relay target** [**vrf** *vrf-name* | **global**] *ip-address*
7. **exit**
8. Repeat Steps 5 through 7 for each DHCP class you need to configure.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip dhcp pool** *name* **Example:** `Router(config)# ip dhcp pool abc-pool`	Configures a DHCP address pool on a Cisco IOS DHCP server and enters DHCP pool configuration mode. The *name* argument is the name of the pool and may either be a symbolic string (such as engineering) or an integer (such as 0).
Step 4	**network** *network-number* [*mask* | *prefix-length*] **Example:** `Router(dhcp-config)# network 10.0.0.0 255.0.0.0`	Configures the subnet number and mask for a DHCP address pool on a Cisco IOS DHCP server. The arguments are as follows: • *network-number*—The IP address of the DHCP address pool. • *mask*—(Optional) The bit combination that renders which portion of the address of the DHCP address pool refers to the network or subnet and which part refers to the host. • *prefix-length*—(Optional) The number of bits that comprise the address prefix. The prefix is an alternative way of specifying the network mask of the client. The prefix length must be preceded by a forward slash (/).
Step 5	**class** *class-name* **Example:** `Router(dhcp-config)# class abc-pool`	Associates a class with a DHCP address pool and enters DHCP pool-class configuration mode. The *class-name* argument is the name of the class. More than one class name may be configured. **Note** If no relay target or address range is configured for a DHCP pool class name, the DHCP pool configuration is used as the class by default.

	Command or Action	Purpose
Step 6	**relay target** [**vrf** *vrf-name* | **global**] *ip-address* **Example:** `Router(config-dhcp-pool-class)# relay target 10.0.0.0`	Configures the relay target IP address. The arguments and keywords for the **relay target** command are as follows: • **vrf**—(Optional) Virtual routing and forwarding (VRF). The *vrf-name* argument is the name of VRF associated with the relay target IP address and more than one target may be specified. • **global**—(Optional) Global IP address space. • *ip-address*—IP address of the relay target. More than one target IP address may be specified. **Note** When using the **relay target** command, the *ip-address* argument is assumed to be in the same VRF as the address pool under which the command was configured. If the relay target IP address is in a different VRF, or in the global address space, then the **vrf** *vrf-name* or **global** keywords need to be specified.
Step 7	**exit** **Example:** `Router(config-dhcp-pool-class)# exit`	Exits to DHCP pool configuration mode.
Step 8	Repeat Steps 5 through 7 for each DHCP class you need to configure.	—

Configuring Multiple Relay Sources for a Relay Pool

Perform this task to configure multiple relay sources for a relay pool. The configuration is similar to configuring an IP helper address on multiple interfaces. Pools are matched to the IP addresses on an incoming interface in the order in which the interfaces display when the **show running-config** command is used. Once a relay is found or an address allocation is found, the search stops.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask* [**secondary**]
5. **exit**
6. **ip dhcp pool** *name*
7. **relay source** *ip-address subnet-mask*
8. **relay destination** [**vrf** *vrf-name* | **global**] *ip-address*
9. **accounting** *method-list-name*
10. Repeat Steps 6 and 7 for each configured DHCP pool.
11. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface ethernet1`	Configures an interface and enters interface configuration mode. The arguments are as follows:
Step 4	**ip address** *ip-address mask* [**secondary**] **Example:** `Router(config-if)# ip address 10.0.0.0 255.0.0.0`	Sets a primary or secondary IP address for an interface.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits to global configuration mode.
Step 6	**ip dhcp pool** *name* **Example:** `Router(config)# ip dhcp pool abc-pool1`	Configures a DHCP address pool on a DHCP server and enters DHCP pool configuration mode. The *name* argument is the name of the pool and may either be a symbolic string (such as engineering) or an integer (such as 0). More than one pool may be assigned.
Step 7	**relay source** *ip-address subnet-mask* **Example:** `Router(dhcp-config)# relay source 10.0.0.0 255.0.0.0`	Configures the relay source. The *ip-address* and *subnet-mask* arguments are the IP address and subnet mask for the relay source. **Note** This command is similar to the **network** command in a normal DHCP network pool, because it restricts the use of the address pool to packets arriving on the interface whose configured IP address and mask matches the relay source configuration.

	Command or Action	Purpose
Step 8	**relay destination** [**vrf** *vrf-name* | **global**] *ip-address* **Example:** `Router(dhcp-config)# relay destination 10.5.5.0`	Configures the IPv4 address of a remote DHCP server to which DHCP client packets are sent. The arguments and keywords are as follows: • **vrf**—(Optional) Virtual routing and forwarding (VRF). The *vrf-name* argument is the name of the VRF associated with the relay destination IP address. • **global**—(Optional) Global IP address. Use the this keyword when the relay agent is in the global address space and the relay source is in a VRF. • *ip-address*—IP address of the relay destination. **Note** When using the **relay destination** command, the *ip-address* argument is assumed to be in the same VRF as the address pool under which the command was configured. If the relay destination IP address is in a different VRF, or in the global address space, then the **vrf** *vrf-name* or **global** keywords need to be specified.
Step 9	**accounting** *method-list-name* **Example:** `Router(dhcp-config)# accounting RADIUS-GROUP1`	(Optional) Enables DHCP accounting if the specified server group is configured to run RADIUS accounting. • AAA and RADIUS must be enabled before DHCP accounting will operate. • The example configures DHCP accounting START and STOP messages to be sent if RADIUS-GROUP1 is configured as a start-stop group. STOP messages will only be sent if RADIUS-GROUP1 is configured as a stop-only group. See “Configuring DHCP Services for Accounting and Security” module for more information on DHCP accounting.
Step 10	Repeat Steps 6 and 7 for each configured DHCP pool.	—
Step 11	**exit** **Example:** `Router(dhcp-config)# exit`	Exits to global configuration mode.

Configuration Examples for DHCP Enhancements for Edge Session Management

This section provides the following configuration examples:

- DHCP Address Range and Class Name Configuration: Example, page 17
- DHCP Server Co-Resident with SG Configuration: Example, page 17
- DHCP Relay Agent Co-Resident with SG Configuration: Example, page 17
- Multiple DHCP Pools and Different ISPs Configuration: Example, page 18

DHCP Address Range and Class Name Configuration: Example

The following example shows how to configure an address range for a particular network and class name for a DHCP pool.

```
ip dhcp pool abc-pool
 network 10.10.0.0 255.255.0.0
 class abc-pool
  address range 10.10.5.0 10.10.5.99
```

DHCP Server Co-Resident with SG Configuration: Example

In the following example, the ISPs are ABC and DEF companies. The ABC company has its addresses assigned from an address pool that is dynamically allocated using ODAP. The DEF company has its customer addresses assigned from the address pool 10.100.0.0/16. Customers not associated with any ISP will have an address allocated from the address pool 10.1.0.0/16 and the lease time is set to 10 minutes.

```
!Interface configuration

interface ethernet1
 ip address 10.20.0.1. 255.255.0.0
 ip address 10.1.0.1 255.255.0.0 secondary
 ip address 10.100.0.1 255.255.0.0 secondary

!Address pool for ABC customers

ip dhcp pool abc-pool
 network 20.1.0.0 255.255.0.0
 class abc
!
!Address pool for DEF customers

ip dhcp pool def-pool
 network 10.100.0.0 255.255.0.0
 class def

!Address pool for customers without an ISP

ip dhcp pool temp
 network 10.1.0.0 255.255.0.0
 lease 0 0 10
 class default
```

DHCP Relay Agent Co-Resident with SG Configuration: Example

In the following example, there are two ISPs: abcpool and defpool. The abcpool ISP and its customers are allowed to have addresses in the ranges 10.1.0.0/16 and 30.1.0.0/16 and are relayed to the DHCP server at 10.55.10.1. The defpool ISP and its customers are allowed to have addresses in the ranges 20.1.0.0/16 and 40.4.0.0/16 and are relayed to the DHCP server at 12.10.2.1.

```
!Address ranges:
```

```
interface ethernet1
 ip address 10.1.0.0 255.255.0.0
 ip address 10.2.0.0 255.255.0.0 secondary

interface ethernet2
 ip address 10.3.0.0 255.255.0.0
 ip address 10.4.0.0 255.255.0.0 secondary

!Address pools for abcpool1 and abcpool2:

ip dhcp pool abcpool1
 relay source 10.1.0.0 255.255.0.0
 class abcpool
  relay target 10.5.10.1

!Address pool for abcpool2:

ip dhcp pool abcpool2
 relay source 10.1.0.0 255.255.0.0
 class abcpool
  relay target 10.55.10.1

!Address pools for defpool1 and defpool2:

ip dhcp pool defpool1
 relay source 10.1.0.0 255.255.0.0
 class defpool
  relay target 10.10.2.1

ip dhcp pool defpool2
 relay source 10.4.0.0 255.255.0.0
 class defpool
  relay target 10.10.2.1
```

Multiple DHCP Pools and Different ISPs Configuration: Example

The following example shows how to configure one interface and multiple DHCP pools that have different ISPs by using the **network** command.

```
interface ethernet1
 ip address 10.0.0.1 255.0.0.0
 ip address 10.1.0.1 255.0.0.0
!
ip dhcp pool x
 network 10.0.0.0 255.0.0.0
 class ISP1
!
ip dhcp pool y
 network 10.1.0.0 255.0.0.0
 class ISP2
```

Multiple Relay Sources and Destinations Configuration: Example

In the following example, multiple relay sources and destinations may be configured for a relay pool. This is similar the ip helper-address configuration on multiple interfaces. Pools are matched to the (possibly multiple) IP addresses on an incoming interface in the order in which they appear when using the **show running-config** command to display information about that interface. Once either a relay is found or an address allocation is found, the search stops. For example, given the following configuration:

```
interface ethernet1
 ip address 10.0.0.1 255.0.0.0
 ip address 10.0.0.5 255.0.0.0 secondary

ip dhcp pool x
 relay source 10.0.0.0 255.0.0.0
 relay destination 10.0.0.1

ip dhcp pool y
 relay source 10.0.0.0 255.0.0.0
 relay destination 10.0.0.1
```

In the following example, the DHCP client packet would be relayed to 10.0.0.1, if the SG specified ISP1 as the class name, and would be relayed to 10.0.0.5, if the SG specified ISP2 as the class name.

```
interface ethernet1
 ip address 10.0.0.1 255.0.0.0
 ip address 10.0.0.5 255.0.0.0 secondary

ip dhcp pool x
 relay source 10.0.0.0 255.0.0.0
 relay destination 10.2.0.0 255.0.0.0
 class ISP1
  relay target 10.0.0.1
 class ISP2
  relay target 10.0.0.5
```

SG-Supplied Class Name Configuration: Example

In the following example, an SG-supplied class name is to be used in selecting the remote DHCP server to which packets should be relayed.

```
ip dhcp pool abc-pool-1
 relay source 10.1.0.0 255.255.0.0
 relay destination 10.1.0.0
 class classname1
  relay target 10.20.10.1
 class classname2
  relay target 10.0.10.1
 class classname3
```

In the example above, an SG-supplied class name, called classname1, would relay the DHCP DISCOVER packet to the server at the relay target IP address 10.20.10.1, while SG classname2 would relay the DHCP DISCOVER packet to the server at the relay target IP address 10.0.10.1. This configuration relays the packet to destination IP address 10.0.0.1, because the pool matches the first configured address on the interface. If the SG returns a classname3, then the default pool is the default address specified as the relay destination. If the SG returns any class name other than classname1, classname2, or classname3, then no relay action is taken.

Additional References

The following sections provide references related to configuring DHCP Enhancements for Edge-Session Management.

Related Documents

Related Topic	Document Title
DHCP commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Addressing Services Command Reference*
DHCP conceptual information	"DHCP Overview" module
DHCP server configuration	"Configuring the Cisco IOS DHCP Server" module
DHCP client configuration	"Configuring the Cisco IOS DHCP Client" module
DHCP relay agent configuration	"Configuring the Cisco IOS DHCP Relay Agent" module
DHCP server on-demand address pool manager configuration	"Configuring the DHCP Server On-Demand Address Pool Manager" module
DHCP advanced features	"Configuring DHCP Services for Accounting and Security" module
DHCP options	"DHCP Options" appendix in the *Network Registrar User's Guide*, Release 6.1.1

Standards

Standards	Title
No new or modified standards are supported by this functionality.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 951	*Bootstrap Protocol (BOOTP)*
RFC 1542	*Clarifications and Extensions for the Bootstrap Protocol*
RFC 2131	*Dynamic Host Configuration Protocol*
RFC 2685	*Virtual Private Networks Identifier*
RFC 3046	*DHCP Relay Information Option*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for DHCP Enhancements for Edge-Session Management

Table 1 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.2(1) or a later release appear in the table.

For information on a feature in this technology that is not documented here, see the "DHCP Features Roadmap".

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for DHCP Enhancements for Edge-Session Management

Feature Name	Releases	Feature Configuration Information
DHCP Relay Accounting	12.4(6)T	The DHCP Relay Accounting feature allows a Cisco IOS DHCP relay agent to send a RADIUS accounting start packet when an address is assigned to a client and a RADIUS accounting stop packet when the address is released. This feature is enabled by using the **accounting** command with relay pools that use the **relay destination** command in DHCP pool configuration mode. The following sections provide information about this feature: • Configuring a Relay Pool with a Relay Source and Destination • Configuring a Relay Pool for a Remote DHCP Server No new commands were introduced by this feature.
DHCP Enhancements for Edge-Session Management	12.3(14)T 12.2(28)SB 12.2(33)SRC	The DHCP Enhancements for Edge-Session Management feature provides the capability of simultaneous service by multiple ISPs to customers using one network infrastructure. The end-user customer may change ISPs at any time. All sections in this module provide information about this feature. The following commands were introduced by this feature: **relay destination**, **relay source**, and **relay target**.

DNS

Configuring DNS

The Domain Name System (DNS) is a distributed database in which you can map host names to IP addresses through the DNS protocol from a DNS server. Each unique IP address can have an associated host name. The Cisco IOS software maintains a cache of host name-to-address mappings for use by the **connect**, **telnet**, and **ping** EXEC commands, and related Telnet support operations. This cache speeds the process of converting names to addresses.

Module History

This module was first published on May 2, 2005, and last updated on March 15, 2007.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all features. To find information about feature support and configuration, use the "Feature Information for DNS" section on page 15.

Contents

Prerequisites for Configuring DNS

To use DNS, you must have a DNS name server on your network.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Information About DNS

To configure DNS, you should understand the following concept:

- DNS Overview, page 2

DNS Overview

If your network devices require connectivity with devices in networks for which you do not control name assignment, you can assign device names that uniquely identify your devices within the entire internetwork. The global naming scheme of the Internet, the DNS, accomplishes this task. This service is enabled by default. The following sections summarize DNS concepts and function:

Host Names for Network Devices

Each unique IP address can have an associated host name. DNS uses a hierarchical scheme for establishing host names for network nodes. This allows local control of the segments of the network through a client-server scheme. The DNS system can locate a network device by translating the host name of the device into its associated IP address.

Domains Names for Groups of Networks

IP defines a naming scheme that allows a device to be identified by its location in the IP. This is a hierarchical naming scheme that provides for *domains*. On the Internet, a domain is a portion of the naming hierarchy tree that refers to general groupings of networks based on organization type or geography. Domain names are pieced together with periods (.) as the delimiting characters. For example, Cisco is a commercial organization that the IP identifies by a *com* domain name, so its domain name is *cisco.com*. A specific device in this domain, the File Transfer Protocol (FTP) system, for example, is identified as *ftp.cisco.com*.

Name Servers

To keep track of domain names, IP has defined the concept of a *name server*. Name servers are programs that have complete information about their namespace portion of the domain tree and may also contain pointers to other name servers that can be used to lead to information from any other part of the domain tree. Name servers know the parts of the domain tree for which they have complete information. A name server may also store information about other parts of the domain tree. To map domain names to IP addresses, you must first identify the host names, then specify a name server, and enable the DNS service.

Cache

To speed the process of converting names to addresses, the name server maintains a database, called a *cache*, of host name-to-address mappings for use by the **connect**, **telnet**, and **ping** EXEC commands, and related Telnet support operations. The cache stores the results from previous responses. Upon receiving a client-issued DNS query, it will check this local storage to see if the answer is available locally.

Name Resolvers

Name resolvers are programs that extract information from name servers in response to client requests. Resolvers must be able to access at least one name server. The resolver either uses that name server's information to answer a query directly or pursues the query using referrals to other names servers. A resolver will typically be a system routine that is directly accessible to user programs. Therefore, no protocol is necessary between the resolver and the user program.

Zones

The domain namespace is divided into areas called *zones* that are points of delegation in the DNS tree. A zone contains all domains from a certain point downward, except those for which other zones are authoritative.

Authoritative Name Servers

A name server is said to be an *authority* for the parts of the domain tree for which it has complete information. A zone usually has an authoritative name server, often more than one. An *authoritative name server* has been configured with host table information or has acquired host table information though a *zone transfer* (the action that occurs when a secondary DNS server starts up and updates itself from the primary server).

DNS Operation

Within an organization, you can have many name servers, but Internet clients can query only those that the root name servers know. The other name servers answer internal queries only.

A name server handles client-issued queries to the DNS server for locally defined hosts within a particular zone as follows:

- An authoritative name server responds to DNS user queries for a domain name that is under its zone of authority by using the permanent and cached entries in its own host table. If the query is for a domain name that is under its zone of authority but for which it does not have any configuration information, the authoritative name server simply replies that no such information exists..
- A name server that is not configured as the authoritative name server responds to DNS user queries by using information that it has cached from previously received query responses. If no router is configured as the authoritative name server for a zone, queries to the DNS server for locally defined hosts will receive nonauthoritative responses.

Name servers answer DNS queries (forward incoming DNS queries or resolve internally generated DNS queries) according to the forwarding and lookup parameters configured for the specific domain.

How to Configure DNS

This section contains the following procedures:

Mapping Host Names to IP Addresses

Perform this task to associate host names with IP addresses.

Host Name-to-Address Mappings

A *name server* is used to keep track of information associated with domain names. A name server can maintain a database of host name-to-address mappings. Each name can map to one or more IP addresses. In order to use this service to map domain names to IP addresses, you must specify a name server.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip host** *name* [*tcp-port-number*] *address1* [*address2* ... *address8*]
4. **ip domain name** *name*
   or
   ip domain list *name*
5. **ip name-server** *server-address1* [*server-address2* ... *server-address6*]
6. **ip domain lookup**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`enable`** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**`configure terminal`** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**`ip host`** *`name`* `[`*`tcp-port-number`*`]` *`address1`* `[`*`address2 ... address8`*`]` **Example:** `Router(config)# ip host cisco-rtp 192.168.0.148`	Defines a static host name-to-address mapping in the host name cache. • Typically, it is easier to refer to network devices by symbolic names rather than numerical addresses (services such as Telnet can use host names or addresses). Host names and IP addresses can be associated with one another through static or dynamic means. • Manually assigning host names to addresses is useful when dynamic mapping is not available.

	Command or Action	Purpose
Step 4	**ip domain name** *name* or **ip domain list** *name* **Example:** `Router(config)# ip domain name cisco.com` or **Example:** `Router(config)# ip domain list cisco1.com`	(Optional) Defines a default domain name that the Cisco IOS software will use to complete unqualified host names. or (Optional) Defines a list of default domain names to complete unqualified host names. • You can specify a default domain name that the Cisco IOS software will use to complete domain name requests. You can specify either a single domain name or a list of domain names. Any host name that does not contain a complete domain name will have the default domain name you specify appended to it before the name is looked up. **Note** If there is no domain list, the domain name that you specified with the **ip domain name** global configuration command is used. If there is a domain list, the default domain name is not used. The **ip domain list** command is similar to the **ip domain name** command, except that with the **ip domain list** command you can define a list of domains, each to be tried in turn until the system finds a match.
Step 5	**ip name-server** *server-address1* [*server-address2 ... server-address6*] **Example:** `Router(config)# ip name-server 172.16.1.111 172.16.1.2`	Specifies one or more hosts that supply name information. • Specifies one or more hosts (up to six) that can function as a name server to supply name information for DNS.
Step 6	**ip domain lookup** **Example:** `Router(config)# ip domain lookup`	(Optional) Enables DNS-based address translation. • DNS is enabled by default. Use this command if DNS has been disabled.

The name lookup system can be statically configured using the commands described in this task. Some other functions in Cisco IOS, such as DHCP can dynamically modify the state of the name lookup system. Use the **show hosts** command to display the cached host names and the DNS configuration.

Customizing DNS

Perform this task to customize your DNS configuration.

DNS Round-Robin Operation

In a multiple server configuration without the DNS round-robin functionality, many programs will use the first host server/IP address for the whole time to live (TTL) of the cache while using the second and third host servers/IP addresses only in the event of host failure. This behavior presents a problem when a high volume of users all arrive at the first host during the TTL time. For example, the network access server (NAS) sends out a DNS query; the DNS servers reply with a list of the configured IP addresses to the NAS. The NAS then caches these IP addresses for a given time (for example, five minutes). All users that dial in during the five minute TTL time will land on one host, the first IP address in the list.

In a multiple server configuration with the DNS round-robin functionality, the DNS server returns the IP address of all hosts to rotate between the cache of host names. During the TTL of the cache, users are distributed among the hosts. This functionality distributes calls across the configured hosts and reduces the amount of DNS queries.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip domain timeout** *seconds*
4. **ip domain retry** *number*
5. **ip domain round-robin**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip domain timeout** *seconds* **Example:** `Router(config)# ip domain timeout 17`	(Optional) Specifies the amount of time to wait for a response to a DNS query. • If the **ip domain timeout** command is not configured, the Cisco IOS software will wait 3 seconds for a response to a DNS query.
Step 4	**ip domain retry** *number* **Example:** `Router(config)# ip domain retry 10`	(Optional) Specifies the number of times to retry sending DNS queries. • If the **ip domain retry** command is not configured, the Cisco IOS software will retry DNS queries twice.
Step 5	**ip domain round-robin** **Example:** `Router(config)# ip domain round-robin`	(Optional) Enables round-robin functionality on DNS servers.

Configuring DNS Spoofing

Perform this task to enable DNS spoofing.

DNS spoofing is designed to allow a router to act as a proxy DNS server and “spoof” replies to any DNS queries using either the configured IP address in the **ip dns spoofing** *ip-address* command or the IP address of the incoming interface for the query. This feature is useful for devices where the interface toward the Internet service provider (ISP) is not up. Once the interface to the ISP is up, the router forwards DNS queries to the real DNS servers.

This feature turns on DNS spoofing and is functional if any of the following conditions are true:

- The **no ip domain lookup** command is configured.
- IP name server addresses are not configured.
- There are no valid interfaces or routes for sending to the configured name server addresses.

If these conditions are removed, DNS spoofing will not occur.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dns server**

4. **ip dns spoofing** [*ip-address*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip dns server** **Example:** `Router(config)# ip dns server`	Activates the DNS server on the router.
Step 4	**ip dns spoofing** [*ip-address*] **Example:** `Router(config)# ip dns spoofing 192.168.15.1`	Enables DNS spoofing. • The router will respond to the DNS query with the configured *ip-address* when queried for any host name other than its own. • The router will respond to the DNS query with the IP address of the incoming interface when queried for its own host name.

Configuring the Router as a DNS Server

Perform this task to configure the router as a DNS server.

A Cisco IOS router can provide service to DNS clients, acting as both a caching name server and as an authoritative name server for its own local host table.

When configured as a caching name server, the router relays DNS requests to other name servers that that resolve network names into network addresses. The caching name server caches information learned from other name servers so that it can answer requests quickly, without having to query other servers for each transaction.

When configured as an authoritative name server for its own local host table, the router listens on port 53 for DNS queries and then answers DNS queries using the permanent and cached entries in its own host table.

Role of an Authoritative Name Server

An authoritative name server usually issues zone transfers or responds to zone transfer requests from other authoritative name servers for the same zone. However, the Cisco IOS DNS server does not perform zone transfers.

When it receives a DNS query, an authoritative name server handles the query as follows:

- If the query is for a domain name that is not under its zone of authority, the authoritative name server determines whether to forward the query to specific back-end name servers based on whether IP DNS-based hostname-to-address translation has been enabled via the **ip domain lookup** command.
- If the query is for a domain name that is under its zone of authority and for which it has configuration information, the authoritative name server answers the queriy using the permanent and cached entries in its own host table.
- If the query is for a domain name that is under its zone of authority but for which it does not have any configuration information, the authoritative name server does not forward the query elsewhere for a response; instead the authoritative name server simply replies that no such information exists.

Restrictions

Unless Distributed Director is enabled, the TTL on locally defined resource records will always be ten seconds, regardless of any authority record parameters that may have been specified for the DNS name server by the use of the **ip dns primary** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dns server**
4. **ip name-server** *server-address1* [*server-address2*...*server-address6*]
5. **ip host** [**vrf** *vrf-name*] [**view** *view-name*] *hostname* {*address1* [*address2* ... *address8*] | **additional** *address9* [*address10* ... *addressn*]}
6. **ip dns primary** *domain-name* **soa** *server-name mailbox-name* [*refresh-interval* [*retry-interval* [*expire-ttl* [*minimum-ttl*]]]]
7. **ip host** *domain-name* **ns** *server-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip dns server** **Example:** `Router(config)# ip dns server`	Enables the DNS server.

	Command or Action	Purpose
Step 4	**ip name-server** *server-address1* [*server-address2...server-address6*] **Example:** `Router(config)# ip name-server 192.168.2.120 192.168.2.121`	(Optional) Configures other DNS servers: • IOS resolver name servers • DNS server forwarders **Note** If the IOS name server is being configured to respond only to domain names for which it is authoritative, there is no need to configure other DNS servers.
Step 5	**ip host** [**vrf** *vrf-name*] [**view** *view-name*] *hostname* {address1 [*address2* ... *address8*] | **additional** *address9* [*address10* ... *addressn*]} **Example:** `Router(config)# ip host user1.example.com 192.168.201.5 192.168.201.6`	(Optional) Configures local hosts.
Step 6	**ip dns primary** *domain-name* **soa** *primary-server-name mailbox-name* [*refresh-interval* [*retry-interval* [*expire-ttl* [*minimum-ttl*]]]] **Example:** `Router(config)# ip dns primary example.com soa ns1.example.com mb1.example.com`	Configures the router as the primary DNS name server for a domain (zone) and as the start of authority (SOA) record source (which designates the start of a zone). **Note** Unless Distributed Director is enabled, the TTL on locally defined resource records will always be ten seconds.
Step 7	**ip host** *domain-name* **ns** *server-name* **Example:** `Router(config)# ip host example.com ns ns1.example.com`	(Optional) Configures the router to create an NS resource record to be returned when the DNS server is queried for the associated domain. This configuration is needed only if the zone for which the system is authoritative will also be served by other name servers.

Example Debugging Output

This section provides examples of debugging output that is logged when a router is configured as an authoritative name server for its own local host table and the **debug domain** command is in effect:

- Debugging Output for Relaying a DNS Query to Another Name Server: Example, page 10
- Debugging Output for Servicing a DNS Query from the Local Host Table: Example, page 11

Note For DNS-based X.25 routing, the **debug x25 events** command supports functionality to describe the events that occur while the X.25 address is being resolved to an IP address using a DNS server. The **debug domain** command can be used along with **debug x25 events** to observe the whole DNS-based X.25 routing data flow.

Debugging Output for Relaying a DNS Query to Another Name Server: Example

The following is sample output from the **debug domain** command that corresponds to relaying a DNS query to another name server when the router is configured as an authoritative name server for its own local host table:

```
Apr  4 22:18:32.183: DNS: Incoming UDP query (id#18713)
```

```
Apr  4 22:18:32.183: DNS: Type 1 DNS query (id#18713) for host 'ns1.example.com' from
192.0.2.120(1283)
Apr  4 22:18:32.183: DNS: Re-sending DNS query (type 1, id#18713) to 192.0.2.121
Apr  4 22:18:32.211: DNS: Incoming UDP query (id#18713)
Apr  4 22:18:32.211: DNS: Type 1 response (id#18713) for host <ns1.example.com> from
192.0.2.121(53)
Apr  4 22:18:32.215: DOM: dom2cache: hostname is ns1.example.com, RR type=1, class=1,
ttl=86400, n=4
Apr  4 22:18:32.215: DNS: Forwarding back A response - no director required
Apr  4 22:18:32.215: DNS: Finished processing query (id#18713) in 0.032 secs
Apr  4 22:18:32.215: DNS: Forwarding back reply to 192.0.2.120/1283
```

Debugging Output for Servicing a DNS Query from the Local Host Table: Example

The following is sample output from the **debug domain** command that corresponds to servicing a DNS query from the local host table when the router is configured as an authoritative name server for its own local host table:

```
Apr  4 22:16:35.279: DNS: Incoming UDP query (id#8409)
Apr  4 22:16:35.279: DNS: Type 1 DNS query (id#8409) for host 'ns1.example.com' from
192.0.2.120(1279)
Apr  4 22:16:35.279: DNS: Finished processing query (id#8409) in 0.000 secs
```

Disabling DNS Queries for ISO CLNS Addresses

Perform this task to disable DNS queries for ISO CLNS addresses.

If your router has both IP and ISO Connectionless Network Service (ISO CLNS) enabled and you want to use ISO CLNS network service access point (NSAP) addresses, you can use the DNS to query these addresses, as documented in RFC 1348. This feature is enabled by default.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no ip domain lookup nsap**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`enable`** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**`configure terminal`** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**`no ip domain lookup nsap`** **Example:** `Router(config)# no ip domain lookup nsap`	Disables DNS queries for ISO CLNS addresses.

Verifying DNS

Perform this task to verify your DNS configuration.

1. **enable**
2. **ping** *hosts*
3. **show hosts**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**ping** *hosts* **Example:** `Router# ping cisco-rtp`	Diagnoses basic network connectivity. • After the DNS configuration is set, you can verify the DNS server by using a hostname to ping or telnet to a device.
Step 3	**show hosts** **Example:** `Router# show hosts`	Displays the default domain name, the style of name lookup service, a list of name server hosts, and the cached list of host names and addresses. • After a name is resolved using DNS, use the **show hosts** command to view the cached hostnames and the DNS configuration.

Configuration Examples for DNS

This section provides the following configuration examples:

- IP Domains: Example, page 13
- Dynamic Lookup: Example, page 13
- Customizing DNS: Example, page 14
- DNS Spoofing: Example, page 14

IP Domains: Example

The following example establishes a domain list with several alternate domain names:

```
ip domain list csi.com
ip domain list telecomprog.edu
ip domain list merit.edu
```

Dynamic Lookup: Example

The following example configures the host name-to-address mapping process. IP DNS-based translation is specified, the addresses of the name servers are specified, and the default domain name is given.

```
! IP DNS-based host name-to-address translation is enabled
ip domain lookup
! Specifies hosts 192.168.1.111 and 192.168.1.2 as name servers
ip name-server 192.168.1.111 192.168.1.2
! Defines cisco.com as the default domain name the router uses to complete
! Set the name for unqualified host names
ip domain name cisco.com
```

Customizing DNS: Example

The following example allows a Telnet to company.example.com to connect to each of the three IP addresses specified in the following order: the first time the hostname is referenced, it would connect to 10.0.0.1; the second time the hostname is referenced, it would connect to 10.1.0.1; and the third time the hostname is referenced, it would connect to 10.2.0.1. In each case, the other two addresses would also be tried if the first one failed; this is the normal operation of the Telnet command.

```
Router(config)# ip host company.example.com 10.0.0.1 10.1.0.1 10.2.0.1
Router(config)# ip domain round-robin
```

DNS Spoofing: Example

In the following example, the router is configured to spoof replies to any DNS queries:

```
ip dns server
ip dns spoofing
no ip domain lookup
interface e3/1
 ip address 10.1.1.1 255.255.255.0
```

Additional References

The following sections provide references related to DNS.

Related Documents

Related Topic	Document Title
DNS commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Addressing Services Command Reference*

Standards

Standards	Title
No new or modified standards are supported by this functionality.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Feature Information for DNS

Table 1 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Releases 12.2(1) or later appear in the table.

Not all commands may be available in your Cisco IOS software release. For details on when support for specific commands was introduced, see the command reference documents.

Cisco IOS software images are specific to a Cisco IOS software release, a feature set, and a platform. Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for DNS

Feature Name	Releases	Feature Configuration Information
DNS Spoofing	12.3(2)T	This feature is designed to allow a router to act as a proxy DNS server and "spoof" replies to any DNS queries using either the configured IP address in the **ip dns spoofing** *ip-address* command or the IP address of the incoming interface for the query. The following section provides information about this feature: • Configuring DNS Spoofing The following command was introduced by this feature: **ip dns spoofing.**

Dynamic DNS Support for Cisco IOS Software

The Dynamic DNS Support for Cisco IOS Software feature enables Cisco IOS software devices to perform Dynamic Domain Name System (DDNS) updates to ensure that an IP host DNS name is correctly associated with its IP address.

It provides two mechanisms to generate or perform DDNS: the IETF standard as defined by RFC 2136 and a generic HTTP using various DNS services. With this feature, you can define a list of hostnames and IP addresses that will receive updates, specify an update method, and specify a configuration for Dynamic Host Configuration Protocol (DHCP) triggered updates.

History for the Dynamic DNS Support for Cisco IOS Software Feature

Release	Modification
12.3(8)YA	This feature was introduced.
12.3(14)T	This feature was integrated into Cisco IOS Release 12.3(14)T.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Restrictions for Dynamic DNS Support for Cisco IOS Software

The performance of the DHCP client can be impacted when the Dynamic DNS Support for Cisco IOS Software feature is enabled, because of sending DDNS update packets and waiting for responses from the server (before sending the ACK to the client REQUEST) and the client (immediately after receiving the ACK and assigning the address to the interface). The default for the client is two attempts with a 5-second wait time between attempts.

The DHCP server continues to process DHCP client DISCOVER and REQUEST packets while waiting for the DDNS updates to complete. Even if the update is done before sending the ACK to the client, it does not delay processing of other DHCP requests. The DHCP server could be impacted minimally because of the time and memory needed in order to set up the DDNS update and get things started.

Reloading the system may take a little longer in some cases, such as, if there are outstanding DDNS updates that need to complete.

Information About Dynamic DNS Support for Cisco IOS Software

To configure the Dynamic DNS Support for Cisco IOS Software, you should understand the following concepts:

- Domain Name System and Dynamic Updates, page 2
- DDNS Updates for HTTP-Based Protocols, page 2
- DHCP Support for DDNS Updates, page 3
- Feature Design of Dynamic DNS Support for Cisco IOS Software, page 3

Domain Name System and Dynamic Updates

The DNS was designed to support queries of a statically configured database. The data was expected to change, but minimally. All updates were made as external edits to a zone master file. The domain name identifies a node within the domain name space tree structure. Each node has a set (possibly empty) of Resource Records (RRs). All RRs having the same NAME, CLASS, and TYPE are called a Resource Record Set (RRset).

There are address (A) or forward RRs and pointer (PTR) or reverse RRs. The DDNS update can specify additions or deletions of hostnames and IP addresses. The two mechanisms to update this information are by using HTTP-based protocols such as DynDNS.org or by using the IETF standard.

DDNS Updates for HTTP-Based Protocols

The Dynamic DNS Support for Cisco IOS Software feature provides the capability of a proprietary HTTP-based protocol to generate or perform DDNS updates. The most notable HTTP-based protocol is DynDNS.org, but there are many others.

Since most of these protocols consist of a simple HTTP command that specifies parameters such as hostname and IP address in the URL portion of the command, this feature takes the same generic approach. You can specify the hostname and IP address in a URL. Configuration of a maximum interval between updates is also allowed.

DHCP Support for DDNS Updates

Before the Dynamic DNS Support for Cisco IOS Software feature, a DHCP server assigned IP addresses to DHCP clients and any DNS information was static. In a network that uses a DHCP server, there are many cases in which DNS hostnames should be associated with the IP addresses that are being assigned. There is an existing method for dynamically updating DNS for DHCP by using information in the fully qualified domain name (FQDN) DHCP option (if it is supplied by the client).

The Dynamic DNS Support for Cisco IOS Software feature enables the DHCP server to support a new FQDN DHCP option. In addition, when the address on an interface is configured, the client can pass the new FQDN option to the server so that name-to-address and address-to-name translations can be updated for the DHCP client as well.

Feature Design of Dynamic DNS Support for Cisco IOS Software

The Dynamic DNS Support for Cisco IOS Software feature enables the tracking of the FQDN DHCP option. If dynamic updates are enabled for the DHCP server, the server updates the PTR RR. The PTR RRs are used for reverse mapping (translation of addresses to names). PTRs use official names not aliases. The name in a PTR record is the local IP address portion of the reverse name.

If the client requests the server to update A RRs as well, the server will attempt to do it. The A RR provides the name-to-address mapping for a DNS zone. The server may be configured to override the client suggestion and always update PTR and A RRs.

The DHCP client can specify whether or not it wants to allow dynamic updates (include the FQDN option), instruct the server to allow the client to update both A and PTR RRs (normally only the A RR is updated by the client), and optionally instruct the server not to update any DNS information (either because the client will be updating both or simply because the client does not want the server to do any updates at all).

There are three basic components of the Dynamic DNS Support for Cisco IOS Software feature that are as follows:

- Definition of the hostname list and IP addresses that will receive updates using a new command that specifies a group of hostnames. Each configured list can consist of any number of IPv4 addresses or hostnames. If a hostname is configured, the name is translated to an IPv4 address at the time at which it is used.
- Specification of an update method. The options are HTTP, DDNS, or an internal Cisco IOS name cache. If the HTTP option is specified, the configuration will include a URL. The username and password must be explicitly written into the URL string and the entire “GET” operation must be specified on one line. The specification will be stored in a linked list. If the update method is DDNS, the configuration will include the update of the IP address.

Events that trigger updates can be as follows:

- IP address that is assigned by a DHCP server for an IP device
- IP address assigned to a router using a DHCP client
- Forwarding of the fully qualified domain name (FQDN) of a user or router hostname from the DHCP client to the server
- Point-to-Point Protocol (PPP)/IP Control Protocol (IPCP) obtaining an IP address for a router interface
- Forced update using a timer to verify a router IP address

Associated with each update method is a value specifying the maximum number of seconds between updates. If left unspecified, then the update is performed only when the address is changed. If specified, the update is performed automatically if the specified number of seconds have passed since the last update.

How to Configure Dynamic DNS Support for Cisco IOS Software

This section contains the following procedures:

- Configuring a Host List, page 4 (optional)
- Verifying the Host-List Configuration, page 6 (optional)
- Configuring DHCP Support of DDNS Updates, page 9 (optional)
- Configuring DDNS Update Support on Interfaces, page 11 (required)
- Configuring a Pool of DHCP Servers to Support DDNS Updates, page 13 (optional)
- Configuring the Update Method and Interval, page 15 (required)
- Verifying DDNS Updates, page 19 (optional)

Note The internal Cisco IOS name cache does not require any configuration.

Configuring a Host List

Perform this task to configure a host list if you are going to use a host list in your configuration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip host-list** *host-list-name*
4. **host** [**vrf** *vrf-name] {host-ip-address* | *hostname}*
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip host-list** *host-list-name* **Example:** `Router(config)# ip host-list abc`	Specifies a list of hosts and enters host-list configuration mode. The *host-list-name argument* assigns a name to the list of hosts.
Step 4	**host** [**vrf** *vrf-name*] {*host-ip-address* | *hostname*} **Example:** `Router(host-list)# host 10.1.1.1 10.2.2.2 10.3.3.3 a.com b.com 10.4.4.4 10.5.5.5 d.com host 10.6.6.6 f.com host vrf abc a.com b.com c.com host vrf def 10.1.1.1 10.2.2.2 10.3.3.3`	Configures one or more hosts. The arguments and keyword are as follows: • **vrf** *vrf-name*—Associates a hostname with a virtual private network (VPN) routing and forwarding instance (VRF) name. **Note** All hostnames or IP addresses specified after the **vrf** keyword are associated with that VRF. • *host-ip-address*—Specifies an IP address for a host in the host list. You can specify more than one host using this argument by listing the hostname and IP addresses on the same line. • *hostname*—Specifies a hostname.
Step 5	exit **Example:** `Router(config) exit`	Exits to global configuration mode.

Examples

The following example shows how to configure several hosts with VRF:

```
ip host-list abc
 host 10.1.1.1 10.2.2.2 10.3.3.3 a.com b.com 10.4.4.4 10.5.5.5 d.com
 host 10.6.6.6 f.com
 host vrf abc a.com b.com c.com
 host vrf def 10.1.1.1 10.2.2.2 10.3.3.3
```

Verifying the Host-List Configuration

To verify the host-list configuration, perform the following steps.

SUMMARY STEPS

1. **show ip host-list**
2. **show running-config | inc host-list**
3. **show running-config | inc host**
4. **debug ip ddns update**

DETAILED STEPS

Step 1 **show ip host-list**

Use this command to verify that the IP addresses and hostnames have been assigned to a host list, for example:

```
Router# show ip host-list abc

Host list: abc
 ddns.abc
 10.2.3.4
 ddns2.abc
 10.3.4.5
 ddns3.com
 10.3.3.3
 d.org
 e.org
 1.org.2.org
 3.com
 10.2.2.2 (VRF: test)
 10.5.5.5 (VRF: test)
 a.net (VRF: test)
 b.net (VRF: test)
```

Step 2 **show running-config | inc host-list**

Use this command to verify the configuration of a host list, for example:

```
Router# show running-config | inc host-list

ip host-list a
ip host-list b
ip host-list c
ip host-list abc
```

Step 3 **show running-config | inc host**

Use this command to verify the configuration of a hostname, for example:

```
Router# show running-config | inc host

hostname who
ip host who 10.0.0.2
ip host-list a
 host 10.1.1.1 a.com b.com 10.2.2.3 10.2.2.2 c.com. 10.3.3.3 10.4.4.4
 host d.com
 host vrf abc 10.10.10.4 10.10.10.8
 host vrf def 10.2.3.4 10.6.7.8
```

```
ip host-list b
 host a.com b.com c.com 10.1.1.1 10.2.2.2 10.3.3.3
 host vrf ppp 10.2.1.0
ip host-list c
 host 10.1.1.1 10.2.2.2 10.3.3.3 a.com b.com 10.4.4.4 10.5.5.5 d.com
 host 10.6.6.6 f.com
 host vrf zero a.com b.com c.om
 host vrf one 10.1.1.1 10.2.2.2 10.3.3.3
ip host-list unit-test
 host ddns.unit.test 10.2.3.4 ddns2.unit.test 10.3.4.5 ddns3.com 10.3.3.3 d.org e.org
 host 1.org.2.org 3.com
 host vrf ZERO 10.2.2.2 10.5.5.5 a.net b.net
 ip ddns update hostname use-this.host.name
 ip ddns update this-method host 10.2.3.4
 ip ddns update this-method host this-host
 ip ddns update this-method host-group this-list
 ip ddns update this-method host 10.3.4.5
 ip ddns update test host 10.19.192.32
 ip ddns update test host 10.19.192.32
 ip ddns update a host-group a
 ip ddns update a host-group ab
 ip ddns update aa host-group ab
 ip ddns update method host 10.33.44.55
```

Step 4 **debug ip ddns update**

Use the **debug ip ddns update** command for the following configuration to verify the configuration of the hosts. Two servers are configured in the host list. A DHCP client is configured for IETF DDNS updating of both A and DNS RRs and requesting the DHCP server to update neither. The DHCP client is configured to include an FQDN DHCP option that instructs the DHCP server not to update either A or PTR Resource Records. This is configured using the interface version of the command. The DHCP server is configured to allow the DHCP client to update whatever RRs it chooses.

```
!Configure the DHCP Client

ip host-list servers
 host 10.19.192.32 10.0.0.1

ip ddns update method testing
 ddns

interface Ethernet1
 ip dhcp client update dns server none
 ip ddns update testing host-group servers
 ip address dhcp
end

!Configure the DHCP Server

ip dhcp pool test
 network 10.0.0.0 255.0.0.0
 update dns

!Enable Debugging

debug ip ddns update

!The update to the server 10.0.0.1 fails in this example

00:18:58:%DHCP-6-ADDRESS_ASSIGN: Interface Ethernet1 assigned DHCP address 10.0.0.8, mask
255.0.0.0, hostname canada_reserved
00:18:58: DYNDNSUPD: Adding DNS mapping for canada_reserved.hacks <=> 10.0.0.8 server
10.19.192.32
```

```
00:18:58: DYNDNSUPD: Sleeping for 3 seconds waiting for interface Ethernet1 configuration
to settle
00:19:01: DDNS: Enqueuing new DDNS update 'canada_reserved.hacks' <=> 10.0.0.8 server
10.19.192.32
00:19:01: DYNDNSUPD: Adding DNS mapping for canada_reserved.hacks <=> 10.0.0.8 server
10.0.0.1
00:19:01: DDNS: Enqueuing new DDNS update 'canada_reserved.hacks' <=> 10.0.0.8 server
10.0.0.1
00:19:01: DYNDNSUPD: Adding DNS mapping for canada_reserved.hacks <=> 10.0.0.8 server
10.0.0.1
00:19:01: DDNS: Enqueuing new DDNS update 'canada_reserved.hacks' <=> 10.0.0.8 server
10.0.0.1
00:19:01: DDNS: Zone name for '10.0.0.11.in-addr.arpa.' is '10.in-addr.arpa'
00:19:01: DDNS: Using server 10.19.192.32
00:19:01: DDNS: Dynamic Update 1: (sending to server 10.19.192.32)
00:19:01: DDNS: Zone = 10.in-addr.arpa
00:19:01: DDNS: Prerequisite: 10.0.0.11.in-addr.arpa. not in use
00:19:01: DDNS: Update: add 10.0.0.11.in-addr.arpa. IN PTR canada_reserved.hacks
00:19:01: DDNS: Zone name for '10.0.0.11.in-addr.arpa.' is '10.in-addr.arpa'
00:19:01: DDNS: Using server 10.0.0.1
00:19:01: DDNS: Dynamic Update 1: (sending to server 10.0.0.1)
00:19:01: DDNS: Zone = 10.in-addr.arpa
00:19:01: DDNS: Prerequisite: 10.0.0.11.in-addr.arpa. not in use
00:19:01: DDNS: Update: add 10.0.0.11.in-addr.arpa. IN PTR canada_reserved.hacks
00:19:01: DDNS: Zone name for '10.0.0.11.in-addr.arpa.' is '10.in-addr.arpa'
00:19:01: DDNS: Using server 10.0.0.1
00:19:01: DDNS: Dynamic Update 1: (sending to server 10.0.0.1)
00:19:01: DDNS: Zone = 10.in-addr.arpa
00:19:01: DDNS: Prerequisite: 10.0.0.11.in-addr.arpa. not in use
00:19:01: DDNS: Update: add 10.0.0.11.in-addr.arpa. IN PTR canada_reserved.hacks
00:19:01: DDNS: Dynamic DNS Update 1 (PTR) for host canada_reserved.hacks returned 6
(YXDOMAIN)
00:19:01: DDNS: Dynamic Update 2: (sending to server 10.19.192.32)
00:19:01: DDNS: Zone = 10.in-addr.arpa
00:19:01: DDNS: Update: delete 10.0.0.11.in-addr.arpa. all PTR RRs
00:19:01: DDNS: Update: add 10.0.0.11.in-addr.arpa. IN PTR canada_reserved.hacks
00:19:01: DDNS: Dynamic DNS Update 2 (PTR) for host canada_reserved.hacks returned 0
(NOERROR)
00:19:01: DDNS: Zone name for 'canada_reserved.hacks' is 'hacks'
00:19:01: DDNS: Using server 10.19.192.32
00:19:01: DDNS: Dynamic Update 1: (sending to server 10.19.192.32)
00:19:01: DDNS: Zone = hacks
00:19:01: DDNS: Prerequisite: canada_reserved.hacks not in use
00:19:01: DDNS: Update: add canada_reserved.hacks IN A 10.0.0.8
00:19:01: DDNS: Dynamic DNS Update 1 (A) for host canada_reserved.hacks returned 0
(NOERROR)
00:19:01: DDNS: Update of 'canada_reserved.hacks' <=> 10.0.0.8 finished
00:19:01: DYNDNSUPD: Another update completed (total outstanding=2)
00:19:11: DDNS: Dynamic DNS Update 1 (PTR) for host canada_reserved.hacks returned 0
(NOERROR)
00:19:11: DDNS: Dynamic DNS Update 1 (PTR) for host canada_reserved.hacks returned 0
(NOERROR)
00:19:11: DDNS: Zone name for 'canada_reserved.hacks' is 'hacks'
00:19:11: DDNS: Using server 10.0.0.1
00:19:11: DDNS: Dynamic Update 1: (sending to server 10.0.0.1)
00:19:11: DDNS: Zone = hacks
00:19:11: DDNS: Prerequisite: canada_reserved.hacks not in use
00:19:11: DDNS: Update: add canada_reserved.hacks IN A 10.0.0.8
00:19:11: DDNS: Zone name for 'canada_reserved.hacks' is 'hacks'
00:19:11: DDNS: Using server 10.0.0.1
00:19:11: DDNS: Dynamic Update 1: (sending to server 10.0.0.1)
00:19:11: DDNS: Zone = hacks
00:19:11: DDNS: Prerequisite: canada_reserved.hacks not in use
00:19:11: DDNS: Update: add canada_reserved.hacks IN A 10.0.0.8
```

```
00:19:21: DDNS: Dynamic DNS Update 1 (A) for host canada_reserved.hacks returned 0
(NOERROR)
00:19:21: DDNS: Update of 'canada_reserved.hacks' <=> 10.0.0.8 failed
00:19:21: DYNDNSUPD: Another update completed (total outstanding=1)
00:19:21: DDNS: Dynamic DNS Update 1 (A) for host canada_reserved.hacks returned 0
(NOERROR)
00:19:21: DDNS: Update of 'canada_reserved.hacks' <=> 10.0.0.8 failed
00:19:21: DYNDNSUPD: Another update completed (total outstanding=0)
```

Configuring DHCP Support of DDNS Updates

DDNS updates contain information about A or forward RRs for a particular IP address. The IP address is in dotted decimal form, and there must be at least one A record for each host address. The name specified is the hostname expressed as an FQDN (ns.example.com). The PTR or reverse RRs map a domain name to another domain name and is used for reverse mapping (IP address to domain name).

The updates are performed using messages. In general, you will probably want DDNS updates done by the server *after* the server has sent the ACK response to the DHCP client. Performing the DDNS updates *before* sending the ACK response will delay the response to the client. Both methods are supported. The default is to do the updates *after* sending the response.

When looking for a client hostname to use in the update, the server will take the hostname from the FQDN option, if such exists, first. If there is no FQDN option, the server will look for a HOSTNAME option and take the name from there.

If the FQDN or HOSTNAME option is included in subsequent RENEWAL messages, the server will attempt to perform the DDNS update each time the lease is renewed. This process gives the opportunity for the client to change the name specified after the lease has been granted and have the server do the appropriate updates. Although the server has this capability, the DHCP client will continue to use the same hostname throughout the duration of a lease.

The IP address of the server to update is discovered by sending a DNS query for records associated with the hostname to update. If such a record exists, the hostname of the master DNS server is extracted from this information. If no such record exists, the record, which should be included in the response, is used as the authoritative record for the zone where the hostname exists. In either case, once the master DNS server hostname is found, another query for A RRs is sent in order to discover the IP address of this server. The resulting IP address is used for sending updates.

Perform this task to configure the DDNS updates.

Prerequisites

In order for DDNS updates to discover the DNS server, in cases in which the user did not configure the server, the **ip name-server** command should be configured. This name server should be reachable by the system, and the **ip domain lookup** command should be configured (which is the default anyway). In cases in which the configured hostname does not include a period (is not a fully qualified domain name [FQDN]), an IP domain name should be configured.

Restrictions

Note DHCP server-pool configuration commands and interface configurations have precedence over global configurations.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp update dns** [**both**] [**override**] [**before**]
4. **ip dhcp-client update dns** [**server** {**both** | **none**}]
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`configure terminal` **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	`ip dhcp update dns [both] [override] [before]` **Example:** `Router(config)# ip dhcp update dns both override`	Enables DDNS updates of PTR RRs for all address pools except those configured with the per-pool **update dns** command, which overrides global configuration. The keywords are as follows: • **both**—(Optional) Enables the DHCP server to perform DDNS updates for A and PTR RRs, unless the DHCP client has specified in the FQDN option that the server should not perform the updates. • **override**—(Optional) Enables the DHCP server to perform DDNS updates for PTR RRs even if the DHCP client has specified in the FQDN option that the server should not perform the updates. **Note** If you specify the **both** and **override** keywords together, this enables the DHCP server to perform DDNS updates for A and PTR RRs overriding anything the DHCP client specified in the FQDN option to the contrary. • **before**—(Optional) Enables the DHCP server to perform DDNS updates before sending the DHCP ACK back to the client. The default is to perform updates after sending the DHCP ACK.

	Command or Action	Purpose
Step 4	**ip dhcp-client update dns** [**server** {**both** | **none**}] **Example:** `Router(config)# ip dhcp-client update dns server both`	Enables DDNS updates of PTR RRs. The optional **server** keyword enables the server to perform DDNS updates for A and PTR RRs. The keywords are as follows: • **both**—Enables the DHCP server to perform DDNS updates for A and PTR RRs, unless the DHCP client specifies in the FQDN option that the server should not perform the updates. • **none**—Enables the DHCP client to perform DDNS updates and the server will not perform any updates. The server can override this action. **Note** The **ip dhcp-client update dns server none** command instructs the server not to perform any updates. If configured to do so, the server can override the client. **Note** The **ip dhcp-client update dns server both** command instructs the server to update both the A and PTR RRs.
Step 5	**exit** **Example:** `Router(config)# exit`	Exits to privileged EXEC mode.

Examples

The following example shows how to configure A and PTR RR updates that are performed by the server only:

```
ip dhcp-client update dns server both
ip dhcp update dns both override
```

Configuring DDNS Update Support on Interfaces

Perform this task to configure your interfaces for DDNS update capability.

Note The interface configuration overrides the global configuration.

Prerequisites

In order for DDNS updates to discover the DNS server, in cases in which the user did not configure the server, the **ip name-server** command should be configured. This name server should be reachable by the system, and the **ip domain lookup** command should be configured (which is the default anyway). In cases in which the configured hostname does not include a period (is not a fully qualified domain name [FQDN]), an IP domain name should be configured.

Restrictions

The changes will not take effect until any current lease on the interface is released and a new lease is requested that uses a new DHCP DISCOVER packet. This means configuring the **ip address dhcp** command or using the **release dhcp** EXEC command followed by the **renew dhcp** EXEC command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-type number*
4. **ip dhcp client update dns** [**server** {**both** | **none**}]
5. ip address dhcp
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`configure terminal` **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	`interface` *interface-type number* **Example:** `Router(config)# interface ethernet1`	Specifies an interface type and number and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**ip dhcp client update dns** [**server** {**both** | **none**}] **Example:** `Router(config-if)# ip dhcp client update dns server both`	Configures the DHCP client to include an FQDN option when sending packets to the DHCP server. The keywords are as follows: • **both**—(Optional) Enables the DHCP server to perform DDNS updates for A and PTR RRs, unless the DHCP client specifies in the FQDN option that the server should not perform the updates. • **none**—(Optional) Enables the DHCP client to perform DDNS updates and the server will not perform any updates. The server can override this action. **Note** The **ip dhcp client update dns server none** command instructs the server not to perform any updates. If configured to do so, the server can override the client. **Note** The **ip dhcp client update dns server both** command instructs the server to update both the A and PTR RRs.
Step 5	**ip address dhcp** **Example:** `Router(config-if)# ip address dhcp`	Releases any current lease on the interface and enables the configuration. **Note** You can also release any lease by using the **release dhcp** EXEC command followed by the **renew dhcp** EXEC command.
Step 6	`exit` **Example:** `Router(config-if)# exit`	Exits to privileged EXEC mode.

Configuring a Pool of DHCP Servers to Support DDNS Updates

There are two parts to the DDNS update configuration on the client side. First, if the **ip ddns update method** command is configured on the client, which specifies the DDNS-style updates, then the client will be trying to generate or perform A updates. If the **ip ddns update method ddns both** command is configured, then the client will be trying to update both A and PTR RRs.

Second, the only way for the client to communicate with the server, with reference to what updates it is generating or expecting the server to generate, is to include an FQDN option when communicating with the server. Whether or not this option is included is controlled on the client side by the **ip dhcp-client update dns** command in global configuration mode or the **ip dhcp client update dns** command in interface configuration mode.

If the FQDN option is included in the DHCP interaction, then the client may instruct the server to update “reverse” (the default), “both”, or “none.” Obviously, if the **ip ddns update method** command is configured with the **ddns** and **both** keywords, then the FQDN option configuration should reflect an IP DHCP client update DNS server none, but you have to configure the system correctly.

Finally, even if the client instructs the server to update both or update none, the server can override the client request and do whatever it was configured to do anyway. If there is an FQDN option in the DHCP interaction as above, then server can communicate to the client that it was overridden, in which case the client will not perform the updates because it knows that the server has done the updates. Even if the server is configured to perform the updates after sending the ACK (the default), it can still use the FQDN option to instruct the client what updates it will be performing and thus the client will not do the same types of updates.

If the server is configured with the **update dns** command with or without any keywords, and if the server does not see an FQDN option in the DHCP interaction, then it will assume that the client does not understand DDNS and will automatically act as though it were configured to update both A and PTR RRs on behalf of the client.

Perform this task to configure a pool of DHCP servers to support DDNS updates.

Prerequisites

In order for DDNS updates to discover the DNS server, in cases in which the user did not configure the server, the **ip name-server** command should be configured. This name server should be reachable by the system, and the **ip domain lookup** command should be configured (which is the default anyway). In cases in which the configured hostname does not include a period (is not a fully qualified domain name [FQDN]), an IP domain name should be configured.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp pool** *pool-name*
4. **update dns [both | never] [override] [before]**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`configure terminal` **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	`ip dhcp pool` *`pool-name`* **Example:** `Router(config)# ip dhcp pool test`	Assigns a name to a DHCP pool and enters DHCP configuration mode.

	Command or Action	Purpose
Step 4	**update dns** [**both** | **never**] [**override**] [**before**] **Example:** `Router(dhcp-config)# update dns never`	Enables DDNS update capability for a pool of DHCP servers for any addresses assigned from this address pool. If the server is configured using this command with or without any of the other keywords, and if the server does not see an FQDN option in the DHCP interaction, then it will assume that the client does not understand DDNS and act as though it were configured to update both A and PTR records on behalf of the client. The keywords are as follows: • **both**—(Optional) Perform forward and reverse updates. If the **before** optional keyword is specified along with the **both** keyword, the server can perform DDNS updates before sending the ACK back to the client. If the **override** optional keyword is specified with the **both** keyword, the server can override the client and update forward and reverse RRs. If the **override** and **before** optional keywords are specified with the **both** keyword, the server can override the client (forward and reverse updates) and perform the updates before sending the ACK. • **never—**(Optional) Never perform updates for this pool. • **override**—(Optional) Override the client FQDN flags. If the **before** optional keyword is specified, the updates will be performed before sending the ACK. • **before**—(Optional) Perform updates before sending the ACK.
Step 5	`exit` **Example:** `Router(dhcp-config)# exit`	Exits to global configuration mode.

Examples

The following example shows how to configure a pool of DHCP servers to perform updates for A and PTR RRs before the ACK is sent:

```
ip dhcp pool test
 update dns both before
```

Configuring the Update Method and Interval

Perform this task to specify the update method and interval maximum.

Prerequisites

In order for DDNS updates to discover the DNS server, in cases in which the user did not configure the server, the **ip name-server** command should be configured. This name server should be reachable by the system, and the **ip domain lookup** command should be configured (which is the default anyway). In cases in which the configured hostname does not include a period (is not a fully qualified domain name [FQDN]), an IP domain name should be configured.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip ddns update method** *method-name*
4. **interval minimum** *days hours minutes seconds*
5. **interval maximum** *days hours minutes seconds*
6. **ddns** [**both**]
7. **internal**
8. **http**
9. **add** *url*
10. **remove** *url*
11. **exit**
12. **exit**
13. **interface** *interface-type number*
14. **ip ddns update hostname** *hostname*
15. **ip ddns update** *method-name*
16. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip ddns update method** *method-name* **Example:** `Router(config)# ip ddns update method myupdate`	Specifies the update method name and enters DDNS update method configuration mode.
Step 4	**interval minimum** *days hours minutes seconds* **Example:** `Router(DDNS-update-method)# interval minimum 1 0 0 0`	Configures a minimum update interval. The arguments are as follows: • *days*—Range is from 0 to 365. • *hours*—Range is from 0 to 23. • *minutes*—Range is from 0 to 59. • *seconds*—Range is from 0 to 59.
Step 5	**interval maximum** *days hours minutes seconds* **Example:** `Router(DDNS-update-method)# interval maximum 1 0 0 0`	Configures a maximum update interval. The arguments are as follows: • *days*—Range is from 0 to 365. • *hours*—Range is from 0 to 24. • *minutes*—Range is from 0 to 60. • *seconds*—Range is from 0 to 60.
Step 6	**ddns** [**both**] **Example:** `Router(DDNS-update-method)# ddns`	Configures DDNS as the update method. The **both** keyword specifies that both A and PTR RRs will be updated. **Note** You can specify DDNS or HTTP but not both in one step. If you have specified DDNS, you must disable it by using the **no ddns** command before you can configure HTTP. For the HTTP configuration, see Steps 7,8, and 9.
Step 7	**internal** **Example:** `Router(DDNS-update-method)# internal`	Specifies that an internal cache will be used as the update method.
Step 8	http **Example:** `Router(DDNS-update-method)# http`	Configures HTTP as the update method and enters DDNS-HTTP configuration mode.

	Command or Action	Purpose
Step 9	**add** *url* **Example:** Router(DDNS-HTTP)# add http://test:test@members.dyndns.org/nic/update?system=dyndns&hostname=<h>&myip=<a>	Configures a URL that should be invoked in order to add or change a mapping between a hostname and an IP address. The following example configures the URL to be invoked to add or change the mapping information using DynDNS.org: • http://userid:password@members.dyndns.org/nic/update?system=dyndns&hostname=<h>&myip=<a>. You have to enter the URL string above. Userid is your userid and password is your password at the DynDNS.org website. The special character strings **<h>** and **<a>** will be substituted with the hostname to update and the IP address with which that hostname should be associated, respectively. **Note** Before entering the question mark (?) character, press the control (Ctrl) key and the v key together on your keyboard. This will allow you to enter the ? without the software interpreting the ? as a help query.
Step 10	**remove** *url* **Example:** Router(DDNS-HTTP)# remove http://test:test@members.dyndns.org/nic/update?system=dyndns&hostname=<h>&myip=<a>	Configures a URL that should be invoked in order to remove a mapping between a hostname and an IP address. The URL takes the same form as the **add** keyword in Step 8.
Step 11	exit **Example:** Router(DDNS-HTTP)# exit	Exits to update-method configuration mode.
Step 12	**exit** **Example:** Router(DDNS-update-method)# exit	Exits to global configuration mode.
Step 13	**interface** *interface-type number* **Example:** Router(config)# interface ether1	Enters interface configuration mode.
Step 14	**ip ddns update hosthame** *hostname* **Example:** Router(config-if)# ip ddns update hostname abc.dyndns.org	Specifies a host to be used for the updates. The update will associate this hostname with the configured IP address of the interface. The *hostname* argument specifies the hostname that will receive the updates (for example, DynDNS.org).

	Command or Action	Purpose
Step 15	`ip ddns update` *`name`* **Example:** `Router(config-if) ip ddns update myupdate`	Specifies the name of the update method to use for sending Dynamic DNS updates associated with address changes on this interface.
Step 16	**`exit`** **Example:** `Router(config)# exit`	Exits to privileged EXEC mode.

Examples

The following example shows how to configure the update method, the maximum interval of the updates (globally), and configure the hostname on the interface:

```
ip ddns update method mytest
ddns
 http

!Before entering the question mark (?) character in the add http CLI, press the control
(Ctrl) key and the v key together on your keyboard. This will allow you to enter the ?
without the software interpreting the ? as a help query.

 add http://test:test@members.dyndns.org/nic/update?system=dyndns&hostname=<h>&myip=<a>
 interval maximum 1 0 0 0
 exit
interface ether1
 ip ddns update hostname abc.dyndns.org
 ip ddns update mytest
```

Verifying DDNS Updates

Use the **debug ip ddns update** command to verify that DDNS updates are being performed. There are several sample configurations and the debug output that would display for that scenario.

Sample Configuration #1

The following scenario has a client configured for IETF DDNS updating of A DNS RRs during which a DHCP server is expected to update the PTR DNS RR. The DHCP client discovers the DNS server to update using an SOA RR lookup since the IP address to the server to update is not specified. The DHCP client is configured to include an FQDN DHCP option and notifies the DHCP server that it will be updating the A RRs.

```
!Configure the DHCP Client

ip ddns update method testing
 ddns

interface Ethernet1
 ip dhcp client update dns
 ip ddns update testing
 ip address dhcp
end

!Configure the DHCP Server
```

```
ip dhcp pool test
 network 10.0.0.0 255.0.0.0
 update dns

!Enable Debugging

Router# debug ip ddns update

00:14:39:%DHCP-6-ADDRESS_ASSIGN: Interface Ethernet1 assigned DHCP address 10.0.0.4, mask
255.0.0.0, hostname canada_reserved
00:14:39: DYNDNSUPD: Adding DNS mapping for canada_reserved.hacks <=> 10.0.0.4
00:14:39: DYNDNSUPD: Sleeping for 3 seconds waiting for interface Ethernet1 configuration
to settle
00:14:42: DHCPC: Server performed PTR update
00:14:42: DDNS: Enqueuing new DDNS update 'canada_reserved.hacks' <=> 10.0.0.4
00:14:42: DDNS: Zone name for 'canada_reserved.hacks' is 'hacks'
00:14:42: DDNS: Dynamic Update 1: (sending to server 10.19.192.32)
00:14:42: DDNS:   Zone = hacks
00:14:42: DDNS:   Prerequisite: canada_reserved.hacks not in use
00:14:42: DDNS:   Update: add canada_reserved.hacks IN A 10.0.0.4
00:14:42: DDNS: Dynamic DNS Update 1 (A) for host canada_reserved.hacks returned 0
(NOERROR)
00:14:42: DDNS: Update of 'canada_reserved.hacks' <=> 10.0.0.4 finished
00:14:42: DYNDNSUPD: Another update completed (total outstanding=0)
```

Sample Configuration #2

The following scenario has the client configured for IETF DDNS updating of both A and DNS RRs and requesting that the DHCP server update neither. The DHCP client discovers the DNS server to update using an SOA RR lookup since the IP address to the server to update is not specified. The DHCP client is configured to include an FQDN DHCP option that instructs the DHCP server not to update either A or PTR RRs. This is configured using the global version of the command.

```
!Configure the DHCP Client

ip dhcp-client update dns server none

ip ddns update method testing
 ddns both

interface Ethernet1
 ip ddns update testing
 ip address dhcp
end

!Configure the DHCP Server

ip dhcp pool test
 network 10.0.0.0 255.0.0.0
 update dns

!Enable Debugging

Router# debug ip ddns update

00:15:33:%DHCP-6-ADDRESS_ASSIGN: Interface Ethernet1 assigned DHCP address 10.0.0.5, mask
255.0.0.0, hostname canada_reserved
00:15:33: DYNDNSUPD: Adding DNS mapping for canada_reserved.hacks <=> 10.0.0.5
00:15:33: DYNDNSUPD: Sleeping for 3 seconds waiting for interface Ethernet1 configuration
to settle
00:15:36: DDNS: Enqueuing new DDNS update 'canada_reserved.hacks' <=> 10.0.0.5
00:15:36: DDNS: Zone name for '10.0.0.11.in-addr.arpa.' is '10.in-addr.arpa'
00:15:36: DDNS: Dynamic Update 1: (sending to server 10.19.192.32)
```

```
00:15:36: DDNS:   Zone = 10.in-addr.arpa
00:15:36: DDNS:   Prerequisite: 10.0.0.11.in-addr.arpa. not in use
00:15:36: DDNS:   Update: add 10.0.0.11.in-addr.arpa. IN PTR canada_reserved.hacks
00:15:36: DDNS: Dynamic DNS Update 1 (PTR) for host canada_reserved.hacks returned 0 
(NOERROR)
00:15:36: DDNS: Zone name for 'canada_reserved.hacks' is 'hacks'
00:15:36: DDNS: Dynamic Update 1: (sending to server 10.19.192.32)
00:15:36: DDNS:   Zone = hacks
00:15:36: DDNS:   Prerequisite: canada_reserved.hacks not in use
00:15:36: DDNS:   Update: add canada_reserved.hacks IN A 10.0.0.5
00:15:36: DDNS: Dynamic DNS Update 1 (A) for host canada_reserved.hacks returned 0 
(NOERROR)
00:15:36: DDNS: Update of 'canada_reserved.hacks' <=> 10.0.0.5 finished
00:15:36: DYNDNSUPD: Another update completed (total outstanding=0)
```

Sample Configuration #3

The following scenario the client is configured for IETF DDNS updating of both A and DNS RRs and requesting that the DHCP server update neither. The DHCP client explicitly specifies the server to update. The DHCP client is configured to include an FQDN DHCP option which instructs the DHCP server not to update either A or PTR RRs. This is configured using the global version of the command. The DHCP server is configured to override the client request and update both A and PTR RR anyway.

```
!Configure the DHCP Client

ip dhcp client update dns server non

ip ddns update method testing
 ddns both

interface Ethernet1
 ip dhcp client update dns server none
 ip ddns update testing
 ip address dhcp
end

!Configure the DHCP Server

ip dhcp pool test
 network 10.0.0.0 255.0.0.0
 update dns both override

!Enable Debugging on the DHCP Client

Router# debug ip ddns update

00:16:30:%DHCP-6-ADDRESS_ASSIGN: Interface Ethernet1 assigned DHCP address 10.0.0.6, mask 
255.0.0.0, hostname canada_reserved
00:16:30: DYNDNSUPD: Adding DNS mapping for canada_reserved.hacks <=> 10.0.0.6
00:16:30: DYNDNSUPD: Sleeping for 3 seconds waiting for interface Ethernet1 configuration 
to settle
00:16:33: DHCPC: Server performed both updates
```

Sample Configuration #4

In the following scenario the client is configured for IETF DDNS updating of both A and DNS RRs and requesting the DHCP server to update neither. The DHCP client explicitly specifies the server to update. The DHCP client is configured to include an FQDN DHCP option which instructs the DHCP server not to update either A or PTR RRs. This is configured using the global version of the command. The DHCP server is configured to allow the client to update whatever RR it chooses.

```
!Configure the DHCP Client
```

```
ip dhcp client update dns server non

ip ddns update method testing
 ddns both

interface Ethernet1
 ip dhcp client update dns server none
 ip ddns update testing host 172.19.192.32
 ip address dhcp
end

!Configure the DHCP Server

ip dhcp pool test
 network 10.0.0.0 255.0.0.0
 update dns

!Enable Debugging on the DHCP Client

Router# debug ip ddns update

00:17:52:%DHCP-6-ADDRESS_ASSIGN: Interface Ethernet1 assigned DHCP address 10.0.0.7, mask
255.0.0.0, hostname canada_reserved
00:17:52: DYNDNSUPD: Adding DNS mapping for canada_reserved.hacks <=> 10.0.0.6
00:17:52: DYNDNSUPD: Sleeping for 3 seconds waiting for interface Ethernet1 configuration
to settle
00:17:55: DDNS: Enqueuing new DDNS update 'canada_reserved.hacks' <=> 10.0.0.7
00:17:55: DYNDNSUPD: Adding DNS mapping for canada_reserved.hacks <=> 10.0.0.7 server
10.19.192.32
00:17:55: DDNS: Enqueuing new DDNS update 'canada_reserved.hacks' <=> 10.0.0.7 server
10.19.192.32
00:17:55: DDNS: Zone name for '10.0.0.11.in-addr.arpa.' is '11.in-addr.arpa'
00:17:55: DDNS: Dynamic Update 1: (sending to server 10.19.192.32)
00:17:55: DDNS: Zone = 10.in-addr.arpa
00:17:55: DDNS: Prerequisite: 10.0.0.11.in-addr.arpa. not in use
00:17:55: DDNS: Update: add 10.0.0.11.in-addr.arpa. IN PTR canada_reserved.hacks
00:17:55: DDNS: Zone name for '10.0.0.11.in-addr.arpa.' is '10.in-addr.arpa'
00:17:55: DDNS: Using server 10.19.192.32
00:17:55: DDNS: Dynamic Update 1: (sending to server 10.19.192.32)
00:17:55: DDNS: Zone = 10.in-addr.arpa
00:17:55: DDNS: Prerequisite: 10.0.0.11.in-addr.arpa. not in use
00:17:55: DDNS: Update: add 10.0.0.11.in-addr.arpa. IN PTR canada_reserved.hacks
00:17:55: DDNS: Dynamic DNS Update 1 (PTR) for host canada_reserved.hacks returned 0
(NOERROR)
00:17:55: DDNS: Dynamic DNS Update 1 (PTR) for host canada_reserved.hacks returned 6
(YXDOMAIN)
00:17:55: DDNS: Dynamic Update 2: (sending to server 10.19.192.32)
00:17:55: DDNS: Zone = 10.in-addr.arpa
00:17:55: DDNS: Update: delete 10.0.0.11.in-addr.arpa. all PTR RRs
00:17:55: DDNS: Update: add 10.0.0.11.in-addr.arpa. IN PTR canada_reserved.hacks
00:17:55: DDNS: Dynamic DNS Update 2 (PTR) for host canada_reserved.hacks returned 0
(NOERROR)
00:17:55: DDNS: Zone name for 'canada_reserved.hacks' is 'hacks'
00:17:55: DDNS: Dynamic Update 1: (sending to server 10.19.192.32)
00:17:55: DDNS: Zone = hacks
00:17:55: DDNS: Prerequisite: canada_reserved.hacks not in use
00:17:55: DDNS: Update: add canada_reserved.hacks IN A 10.0.0.7
00:17:55: DDNS: Dynamic DNS Update 1 (A) for host canada_reserved.hacks returned 0
(NOERROR)
00:17:55: DDNS: Update of 'canada_reserved.hacks' <=> 10.0.0.7 finished
00:17:55: DYNDNSUPD: Another update completed (total outstanding=1)
00:17:55: DDNS: Zone name for 'canada_reserved.hacks' is 'hacks'
00:17:55: DDNS: Using server 10.19.192.32
00:17:55: DDNS: Dynamic Update 1: (sending to server 10.19.192.32)
```

```
00:17:55: DDNS: Zone = hacks
00:17:55: DDNS: Prerequisite: canada_reserved.hacks not in use
00:17:55: DDNS: Update: add canada_reserved.hacks IN A 10.0.0.7
00:17:55: DDNS: Dynamic DNS Update 1 (A) for host canada_reserved.hacks returned 6
(YXDOMAIN)
00:17:55: DDNS: Dynamic Update 2: (sending to server 10.19.192.32)
00:17:55: DDNS: Zone = hacks
00:17:55: DDNS: Update: delete canada_reserved.hacks all A RRs
00:17:55: DDNS: Update: add canada_reserved.hacks IN A 10.0.0.7
00:17:55: DDNS: Dynamic DNS Update 2 (A) for host canada_reserved.hacks returned 0
(NOERROR)
00:17:55: DDNS: Update of 'canada_reserved.hacks' <=> 10.0.0.7 finished
00:17:55: DYNDNSUPD: Another update completed (total outstanding=0)
```

Sample Configuration #5

In the following scenario, the debug output is displaying internal host table updates when the default domain name is "hacks." The "test" update method specifies that the internal Cisco IOS host table should be updated. Configuring the update method as "test" should be used when the address on the Ethernet 0/0 interface changes. The hostname is configured for the update on this interface.

```
ip domain name hacks

ip ddns update method test
 internal

interface ethernet0/0
 ip ddns update test hostname test2
 ip addr dhcp

!Enable Debugging

Router# debug ip ddns update

*Jun 4 03:11:10.591:%DHCP-6-ADDRESS_ASSIGN: Interface Ethernet0/0 assigned DHCP address
10.0.0.5, mask 255.0.0.0, hostname test2
*Jun 4 03:11:10.591: DYNDNSUPD: Adding DNS mapping for test2.hacks <=> 10.0.0.5
*Jun 4 03:11:10.591: DYNDNSUPD: Adding internal mapping test2.hacks <=> 10.0.0.5
```

Using the **show hosts** command displays the newly added host table entry.

```
Router# show hosts

Default domain is hacks
Name/address lookup uses domain service
Name servers are 255.255.255.255

Codes: UN - unknown, EX - expired, OK - OK,?? - revalidate
       temp - temporary, perm - permanent
       NA - Not Applicable None - Not defined

Host                      Port Flags      Age Type   Address(es)
test2.hacks               None (perm, OK) 0   IP    10.0.0.5
```

Shutting down the interface removes the host table entry.

```
interface ethernet0/0
 shutdown

*Jun 4 03:14:02.107: DYNDNSUPD: Removing DNS mapping for test2.hacks <=> 10.0.0.5
*Jun 4 03:14:02.107: DYNDNSUPD: Removing mapping test2.hacks <=> 10.0.0.5
```

The **show hosts** command output shows the entry has been removed.

```
Router# show hosts

Default domain is hacks
Name/address lookup uses domain service
Name servers are 255.255.255.255

Codes: UN - unknown, EX - expired, OK - OK,?? - revalidate
       temp - temporary, perm - permanent
       NA - Not Applicable None - Not defined

Host                      Port Flags      Age Type   Address(es)
```

Sample Configuration #6

In the following scenario, the debug output shows the HTTP-style DDNS updates. The sample configuration defines a new IP DDNS update method named dyndns that configures a URL to use when adding or changing an address. No URL has been defined for use when removing an address since DynDNS.org does not use such a URL for free accounts. A maximum update interval of 28 days has been configured, so specifying that updates should be sent at least every 28 days. Configuring the new dyndns update method should be used for Ethernet interface .

Note Before entering the question mark (?) character in the "add http" configuration after the **update** keyword, press the control (Ctrl) key and the "v" key together on your keyboard. This will allow you to enter the ? without the software interpreting it as a help query.

```
!Configure the DHCP Client

ip ddns update method dyndns
 http
    add http://test:test@<s>/nic/update?system=dyndns&hostname=<h>&myip=<a>
    interval max 28 0 0 0

interface ethernet1
 ip ddns update hostname test.dyndns.org
 ip ddns update dyndns host members.dyndns.org
 ip addr dhcp

!Enable Debugging

Router# debug ip ddns update

00:04:35:%DHCP-6-ADDRESS_ASSIGN: Interface Ethernet1 assigned DHCP address 10.32.254.187,
mask 255.255.255.240, hostname test.dyndns.org
00:04:35: DYNDNSUPD: Adding DNS mapping for test.dyndns.org <=> 10.32.254.187 server
10.208.196.94
00:04:35: DYNDNSUPD: Sleeping for 3 seconds waiting for interface Ethernet1 configuration
to settle
00:04:38: HTTPDNS: Update add called for test.dyndns.org <=> 10.32.254.187
00:04:38: HTTPDNS: Update called for test.dyndns.org <=> 10.32.254.187
00:04:38: HTTPDNS: init
00:04:38: HTTPDNSUPD: Session ID = 0x7
00:04:38: HTTPDNSUPD: URL =
'http://test:test@10.208.196.94/nic/update?system=dyndns&hostname=test.dyndns.org&myip=10.
32.254.187'
00:04:38: HTTPDNSUPD: Sending request
00:04:40: HTTPDNSUPD: Response for update test.dyndns.org <=> 10.32.254.187
00:04:40: HTTPDNSUPD: DATA START
good 10.32.254.187
00:04:40: HTTPDNSUPD: DATA END, Status is Response data received, successfully
00:04:40: HTTPDNSUPD: Call returned SUCCESS for update test.dyndns.org <=> 10.32.254.187
```

```
00:04:40: HTTPDNSUPD: Freeing response
00:04:40: DYNDNSUPD: Another update completed (outstanding=0, total=0)
00:04:40: HTTPDNSUPD: Clearing all session 7 info

!28 days later, the automatic update happens.

00:05:39: DYNDNSUPD: Adding DNS mapping for test.dyndns.org <=> 10.32.254.187 server
10.208.196.94
00:05:39: HTTPDNS: Update add called for test.dyndns.org <=> 10.32.254.187
00:05:39: HTTPDNS: Update called for test.dyndns.org <=> 10.32.254.187
00:05:39: HTTPDNS: init
00:05:39: HTTPDNSUPD: Session ID = 0x8
00:05:39: HTTPDNSUPD: URL =
'http://test:test@10.208.196.94/nic/update?system=dyndns&hostname=test.dyndns.org&myip=10.
32.254.187'
00:05:39: HTTPDNSUPD: Sending request
00:05:39: HTTPDNSUPD: Response for update test.dyndns.org <=> 10.32.254.187
00:05:39: HTTPDNSUPD: DATA START
nochg 10.32.254.187
00:05:39: HTTPDNSUPD: DATA END, Status is Response data received, successfully
00:05:39: HTTPDNSUPD: Call returned SUCCESS for update test.dyndns.org <=> 10.32.254.187
00:05:39: HTTPDNSUPD: Freeing response
00:05:39: DYNDNSUPD: Another update completed (outstanding=0, total=0)
00:05:39: HTTPDNSUPD: Clearing all session 8 info
```

Configuration Examples for Dynamic DNS Support for Cisco IOS Software

The section contains the following configuration examples:

- Configuration of the DHCP Client: Example, page 25
- Configuration of the DHCP Server: Example, page 25
- Configuration of the HTTP Updates: Example, page 26

Configuration of the DHCP Client: Example

The following example shows that no DDNS updates will be performed for addresses assigned from the address pool “abc.” Addresses allocated from the address pool “def” will have both forward (A) and reverse (PTR) updates performed. This configuration has precedence over the global server configurations.

```
ip dhcp update dns both override
 ip dhcp pool abc
  network 10.1.0.0 255.255.0.0
!
 update dns never
!
 ip dhcp pool def
  network 10.10.0.0 255.255.0.0
```

Configuration of the DHCP Server: Example

The following example shows how to configure A and PTR RR updates that are performed by the server only:

```
ip dhcp-client update dns server both
ip dhcp update dns both override
```

Configuration of the HTTP Updates: Example

The following example shows how to configure a PPPoE server for HTTP DDNS:

```
!Username and Password for PPP Authentication Configuration
!
username user1 password 0 cisco
!
!DHCP Pool Configuration

ip dhcp pool mypool
 network 10.10.10.0 255.255.255.0
 default-router 10.10.10.1
!
!VPDN configuration for PPPoE

vpdn enable
!
vpdn-group pppoe
accept-dialin
protocol pppoe
virtual-template 1
!
interface Loopback0
ip address 10.10.10.1 255.255.255.0
!
!Port used to connect to the Internet, it can be the same port that is under test, but to
make the test clear and simple these two are separated.
!
interface FastEthernet0/0
 ip address 10.0.58.71 255.255.255.0
!
!Port under test.
!
interface FastEthernet0/1
 no ip address
 pppoe enable
!
!Virtual template and address pool config for PPPoE.

interface Virtual-Template1
 ip unnumbered Loopback0
 ip mtu 1492
 peer default ip address dhcp-pool mypool
 ppp authentication chap
```

The following example shows how to configure a DHCP client for IETF DDNS:

```
!Default hostname of the router.

hostname mytest
!
!Default domain name on the router.

ip domain name test.com
!
!Port under test.
!
interface FastEthernet0/1
```

```
 no ip address (configured to "ip address dhcp")
```

The following example shows how to configure the method of update and the maximum interval of the updates (globally) and configure the hostname on the interface:

Note Before entering the question mark (?) character in the “add http” configuration after the **update** keyword, press the control (Ctrl) key and the “v” key together on your keyboard. This will allow you to enter the ? without the software interpreting it as a help query.

```
ip ddns update method mytest
ddns
 http
 add http://test:test@members.dyndns.org/nic/update?system=dyndns&hostname=<h>&myip=<a>
 interval maximum 1 0 0 0
 exit
interface ether1
 ip ddns update hostname abc.dyndns.org
 ip ddns update mytest
```

The following are examples of URLs that can be used to update some HTTP DNS update services. These URLs are correct to the best of the knowledge of Cisco but have not been tested in all cases. Where the word “USERNAME:” appears in the URL, the customer account username at the HTTP site should be used.

Where the word “PASSWORD” appears in the URL, the customer password for that account should be used:

Note Before entering the question mark (?) character in the “add http” configuration after the **update** keyword, press the control (Ctrl) key and the “v” key together on your keyboard. This will allow you to enter the ? without the software interpreting it as a help query.

DDNS

```
http://USERNAME:PASSWORD@members.dyndns.org/nic/update?system=dyndns&hostname=<h>&myip=<a>
!Requires "interval max 28 0 0 0" in the update method definition.
```

TZO

```
http://cgi.tzo.com/webclient/signedon.html?TZOName=<h>&Email=USERNAME&TZOKey=PASSWORD&IPAd
dress=<a>
```

EASYDNS

```
http://USERNAME:PASSWORD@members.easydns.com/dyn/ez-ipupdate.php?action=edit&myip=<a>&host
_id=<h>
```

JUSTLINUX

```
http://USERNAME:PASSWORD@www.justlinux.com/bin/controlpanel/dyndns/jlc.pl?direst=1&usernam
e=USERNAME&password=PASSWORD&host=<h>&ip=<a>
```

DYNS

```
http://USERNAME:PASSWORD@www.dyns.cx/postscript.php?username=USERNAME&password=PASSWORD&ho
st=<h>&ip=<a>
```

HN

```
http://USERNAME:PASSWORD@dup.hn.org/vanity/update?ver=1&IP=<a>
```

ZONEEDIT

```
http://USERNAME:PASSWORD@www.zoneedit.com/auth/dynamic.html?host=<h>&dnsto=<a>
```

Note Because these services are provided by the respective companies, the URLs may be subject to change or the service could be discontinued at any time. Cisco takes no responsibility for the accuracy or use of any of this information. The URLs were obtained using an application called "ez-ipupdate," which is available for free on the Internet.

Additional References

The following sections provide references related to the Dynamic DNS Support for Cisco IOS Software feature.

Related Documents

Related Topic	Document Title
DNS Configuration Tasks	*Configuring DNS* module
DNS commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Addressing Services Command Reference*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 2136	*Dynamic Updates in the Domain Name System (DNS Update)*
RFC 3007	*Secure Domain Name System (DNS) Dynamic Update*

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS IP Addressing Command Reference* at http://www.cisco.com/en/US/docs/ios/ipaddr/command/reference/iad_book.html. For information about all Cisco IOS commands, go to the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or to the *Cisco IOS Master Commands List*.

- **ddns (DDNS-update-method)**
- **debug ip ddns update**
- **host (host-list)**
- **http (DDNS-update-method)**
- **internal (DDNS-update-method)**
- **interval maximum**
- **ip ddns update hostname**
- **ip ddns update method**
- **ip dhcp client update dns**
- **ip dhcp-client update dns**
- **ip dhcp update dns**
- **ip host-list**
- **show ip ddns update**
- **show ip ddns update method**
- **show ip host-list**
- **update dns**

NHRP

Configuring NHRP

First Released: April 3, 2007
Last Updated: May 2, 2008

The purpose of this module is to describe how to configure the Next Hop Resolution Protocol (NHRP) for use in a nonbroadcast multiaccess (NBMA) network. NHRP is an Address Resolution Protocol (ARP)-like protocol that dynamically maps an NBMA network. With NHRP, systems attached to an NBMA network can dynamically learn the NBMA (physical) address of the other systems that are part of that network, allowing these systems to directly communicate.

NHRP is a client and server protocol where the hub is the Next Hop Server (NHS) and the spokes are the Next Hop Clients (NHCs). The hub maintains an NHRP database of the public interface addresses of each spoke. Each spoke registers its real address when it boots and queries the NHRP database for real addresses of the destination spokes to build direct tunnels.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for Configuring NHRP" section on page 38.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Information About NHRP

To configure NHRP, you should understand the following concepts:

How NHRP and NBMA Networks Interact

Most WAN networks are a collection of point-to-point links. Virtual tunnel networks, for example GRE tunnels, are also a collection of point-to-point links. To effectively scale the connectivity of these point-to-point links, they are usually grouped into a single or multi-layer hub-and-spoke network. Multipoint interfaces (for example, GRE tunnel interfaces) can be used to reduce the configuration on a hub router in such a network. This resulting network is a Non-Broadcast Multi-Access (NBMA) network.

Because there are multiple tunnel endpoints reachable through the single multipoint interface and in order to forward packets out the multipoint GRE (mGRE) tunnel interfaces over this NBMA network, there has to be a mapping from the logical tunnel endpoint IP address to the physical tunnel endpoint IP address. This mapping could be statically configured but it is preferable if the mapping can be discovered or learned dynamically.

NHRP is an ARP-like protocol that alleviates these NBMA network problems. With NHRP, systems attached to an NBMA network dynamically learn the NBMA address of the other systems that are part of that network, allowing these systems to directly communicate without requiring traffic to use an intermediate hop.

Routers, access servers, and hosts can use the NHRP to discover the addresses of other routers and hosts connected to an NBMA network. Partially meshed NBMA networks typically have multiple logical networks behind the NBMA network. In such configurations, packets traversing the NBMA network might have to make several hops over the NBMA network before arriving at the exit router (the router nearest the destination network). With NHRP and when NHRP is combined with IPsec, the NBMA network is basically a collection of point-to-point logical tunnel links over a physical IP network.

NHRP allows two functions to help support these NBMA networks:

1. NHRP Registration. NHRP is an ARP-like protocol that allows Next Hop Clients (NHCs) to dynamically register with Next Hop Servers (NHSs). This allows the NHCs to join the NBMA network without configuration changes on the NHSs, especially in cases where the NHC has a dynamic physical IP address or is behind a Network Address Translation (NAT) router that dynamically changes the physical IP address. In these cases it would be impossible to preconfigure the logical virtual private network (VPN IP) to physical (NBMA IP) mapping for the NHC on the NHS. This function is called NHRP registration. See the "NHRP Registration" section on page 4 for more information.
2. NHRP Resolution. NHRP is a resolution protocol that allows one NHC (spoke) to dynamically discover the logical VPN IP to physical NBMA IP mapping for another NHC (spoke) within the same NBMA network. Without this discovery, IP packets traversing from hosts behind one spoke to hosts behind another spoke would have to traverse by way of the NHS (hub) router. This would increase the utilization of the hub's physical bandwidth and CPU to process these packets that come into the hub on the multipoint interface and go right back out the multipoint interface. With NHRP, systems attached to an NBMA network dynamically learn the NBMA address of the other systems

that are part of that network, allowing these systems to directly communicate without requiring traffic to use an intermediate hop. This function alleviates the load on the intermediate hop (NHS) and can increase the overall bandwidth of the NBMA network to be greater than the bandwidth of the hub router.

Dynamically Built Hub-and-Spoke Networks

With NHRP, the NBMA network is initially laid out as a hub-and-spoke network that can be multiple hierarchical layers of NHCs as spokes and NHSs as hubs. The NHCs are configured with static mapping information to reach their NHSs and will connect to their NHS and send an NHRP registration to the NHS. This configuration allows the NHS to dynamically learn the mapping information for the spoke, reducing the configuration needed on the hub and allowing the spoke to obtain a dynamic NBMA (physical) IP address.

Once the base hub-and-spoke network is dynamically built out, then NHRP resolution requests and responses can be used to dynamically discover spoke-to-spoke mapping information, allowing spokes to contact each other directly, bypassing the hub. This allows a dynamic mesh of connections between spokes to be built based on data traffic patterns without requiring a preconfigured static fully meshed network. Using a dynamic-mesh network allows smaller spoke routers to participate up to their capability in a large NBMA network when these smaller spoke routers do not have the resources to participate in a full mesh on the same size network. The smaller spoke routers do not need to build out all possible spoke-to-spoke links; these routers need to build only the ones they are currently using.

Next Hop Server Selection

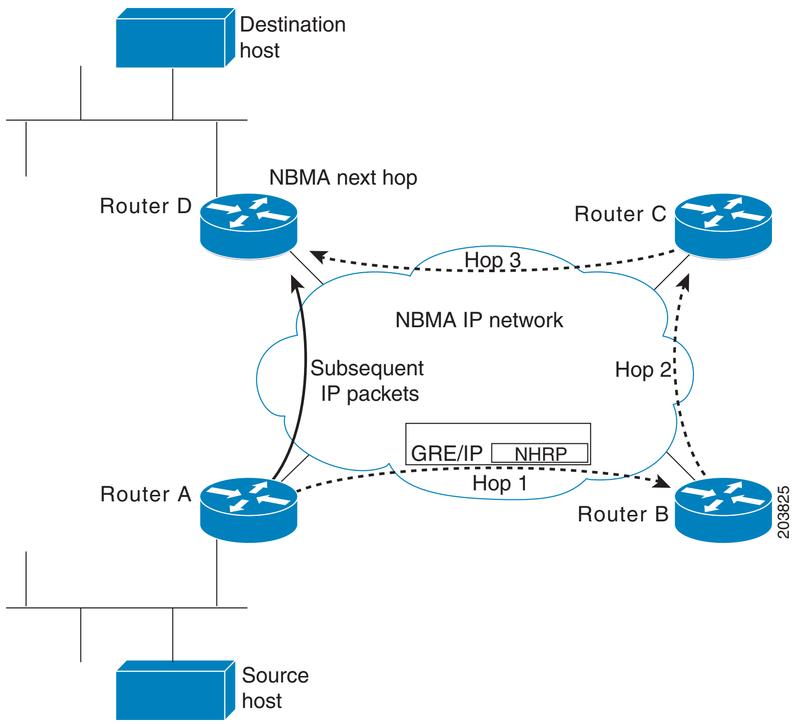
NHRP resolution requests traverse one or more hops (hubs) within the base hub-and-spoke NBMA subnetwork before reaching the station that is expected to generate a response. Each station (including the source station) chooses a neighboring NHS to which it forwards the request. The NHS selection procedure typically involves performing a routing decision based upon the network layer destination address of the NHRP request. The NHRP resolution request eventually arrives at a station that generates an NHRP resolution reply. This responding station either serves the destination, or is the destination itself. The responding station generates a reply using the source address from within the NHRP packet to determine where the reply should be sent.

The Cisco implementation of NHRP also supports and extends the IEEE RFC 2332, *NBMA Next Hop Resolution Protocol (NHRP)*.

The Cisco implementation of NHRP supports IP Version 4 at the network layer and at the link layer with multipoint GRE, Ethernet, Switched Multimegabit Data Service (SMDS), Frame Relay, and ATM. Although NHRP is available on Ethernet, NHRP need not be implemented over Ethernet media because Ethernet is capable of broadcasting and the standard Ethernet IP ARP protocol is sufficient.

Figure 1 illustrates four routers connected to an NBMA network. Within the network are IP routers necessary for the routers to communicate with each other by tunneling the IP data packets in GRE IP tunnel packets. The infrastructure layer routers support logical IP tunnel circuit connections represented by hops 1, 2, and 3 of the figure. When router A attempts to forward an IP packet from the source host to the destination host, NHRP is triggered. On behalf of the source host, router A sends an NHRP resolution request packet encapsulated in a GRE IP packet, which takes three hops across the network to reach router D, connected to the destination host. After router A receives a positive NHRP resolution reply, router A determines that router D is the NBMA IP next hop, and router A sends subsequent data IP packets for the destination to router D in one GRE IP tunnel hop.

Figure 1 Next Hop Resolution Protocol



With NHRP, once the NBMA next hop is determined, the source either starts sending data packets to the destination (in a connectionless NBMA network such as GRE IP or SMDS) or establishes a virtual virtual circuit (VC) connection to the destination. This connection is configured with the desired bandwidth and quality of service (QoS) characteristics for a connection-oriented NBMA network such as Frame Relay, ATM, or with DMVPN where an IPsec encyrption peering must be established.

Other address resolution methods can be used while NHRP is deployed. IP hosts that rely upon the Logical IP Subnet (LIS) model might require ARP servers and services over the NBMA network, and deployed hosts might not implement NHRP, but might continue to support ARP variations. NHRP is designed to eliminate the suboptimal routing that results from the LIS model, and can be deployed with existing ARP services without interfering with them.

NHRP Registration

NHRP registrations are sent from NHCs to their configured NHSs every one-third of the NHRP holdtime (**ip nhrp holdtime** *value*), unless the **ip nhrp registration timeout** *value* command is configured, in which case registrations are sent out according to the configured timeout value. If an NHRP registration reply is not received for an NHRP registration request, the NHRP registration request is retransmitted at timeouts of 1, 2, 4, 8, 16, 32, and 64 seconds, then the sequence starts over again at 1.

The NHS is declared down if an NHRP registration reply is not received after 3 retransmission (7 seconds), and an NHRP resolution packets will no longer be sent to or by way of that NHS. NHRP registrations will continue to be sent in the intervals 0, 1, 2, 4, 8, 16, 32, 64 probing the NHS until an NHRP registration reply is received. As soon as an NHRP registration reply is received the NHS is immediately declared up, the NHRP registration requests revert to being sent every one-third of NHRP holdtime or the value configured in the **ip nhrp registration timeout** command, and the NHS can again be sent NHRP resolution requests. The **show ip nhrp nhs** {**detail**} command can be used to check the state of the NHRP NHSs.

NHRP Used with a DMVPN

NHRP is used to facilitate building a VPN. In this context, a VPN consists of a virtual Layer 3 network that is built on top of an actual Layer 3 network. The topology you use over the VPN is largely independent of the underlying network, and the protocols you run over it are completely independent of it. The VPN network (DMVPN) is based on GRE IP logical tunnels that can be protected by adding in IPsec to encrypt the GRE IP tunnels.

Connected to the NBMA network are one or more stations that implement NHRP, and are known as NHSs and NHCs. All routers running Cisco IOS Release 10.3 or later releases can implement NHRP and, thus, can act as NHSs or NHCs. To get the base functionality of DMVPN (GRE IP+IPsec), which uses NHRP, you must run Cisco IOS Release 12.3(9), 12.3(8)T, or a later release.

Note For the latest extensions and enhancements to NHRP, you must use Cisco IOS Release 12.4 or Cisco IOS Release 12.4T.

Dynamic Spoke-to-Spoke Tunnels

In addition to NHRP registration of NHCs with NHSs, NHRP provides the capability for NHCs (spokes) to find a shortcut path over the infrastructure of the network (IP network, SMDS) or build a shortcut switched virtual circuit (SVC) over a switched infrastructure network (Frame Relay and ATM) directly to another NHC (spoke) bypassing hops through the NHSs (hubs). This capability allows the building of very large NHRP NBMA networks. In this way, the bandwidth and CPU limitations of the hub do not limit the over all bandwidth of the NHRP NBMA network. This capability effectively creates a full-mesh-capable network without having to discover all possible connections beforehand. This is called a dynamic-mesh network, where there is a base hub-and-spoke network of NHCs and NHSs for transporting NHRP and dynamic routing protocol information (and data traffic) and dynamic direct spoke-to-spoke links that are built when there is data traffic to use the link and torn down when the data traffic stops.

The mesh network allows individual spoke routers to directly connect to anywhere in the NBMA network, even though they are capable of connecting only to a limited number at the same time. This allows each spoke in the network to participate in the whole network up to its capabilities without limiting another spoke from participating up to its capability. If a full-mesh network were to be built, then all spokes would have to be sized to handle all possible tunnels at the same time.

For example, in a network of 1000 nodes, a full mesh spoke would have to be large and powerful because it must always support 999 tunnels (one to every other node). In a dynamic-mesh network, a spoke needs to support only a limited number of tunnels to its NHSs (hubs) plus any currently active tunnels to other spokes. Also, if a spoke cannot build more spoke-to-spoke tunnels, then it will send its data traffic by way of the spoke-hub-spoke path. In this way, connectivity is always preserved, even when the preferred single hop path is not available.

Developmental Phases of DMVPN and NHRP

The developmental phases described in this section are actually DMVPN phases combining mGRE plus NHRP and IPsec. These phases are important because they provide the functionality needed to support dynamic spoke-to-spoke tunnels.

- **Phase 1** is the hub-and-spoke capability only. This phase will not be discussed here.
- **Phase 2** adds spoke-to-spoke capability.
- **Phase 3** changes spoke-to-spoke capability in order to scale to larger NBMA networks.

Note Phase 1 does not support spoke-to-spoke tunnels.

NHRP gathers the information that it needs to build spoke-to-spoke tunnels by using NHRP resolution request and reply packets that are sent via the spoke-hub-spoke path through the NBMA network. NHRP also has to be triggered (or know when) to collect this information for building the spoke-to-spoke tunnels, because it brings up the spoke-to-spoke tunnel only when there is data traffic to use it. The two ways that NHRP does this are described the following sections.

Phase 2

In phase 2, NHRP brings up the NHC-to-NHS tunnel and a dynamic routing protocol is used to distribute routing information about all of the networks that are available behind the hub and all of the other spokes. Included in this information is the IP next hop of the destination spoke that is supporting a particular destination network.

When a data packet is to be forwarded it will get the outbound interface and the IP next hop from the matching routing table network entry. If the NHRP interface is the outbound interface then it looks for an NHRP mapping entry for that IP next hop. If there is no matching of NHRP mapping entry, then NHRP is triggered to send an NHRP resolution request to get the mapping information (IP next-hop address to physical layer address). The NHRP registration reply packet contains this mapping information and when this information is received the spoke will have sufficient information to correctly encapsulate the data packet to go directly to the remote spoke, taking one hop across the infrastructure network. One of the downsides to this technique is that each spoke must have all of the individual routes in its routing table for all possible destination networks behind the hub and other spokes. Keeping this routing information distributed and up to date can put a significant load on the routing protocol running over the VPN network.

Phase 3

NHRP brings up the NHC and NHS tunnel and a dynamic routing protocol is used to distribute routing information about all of the networks that are available behind all of the spokes to the hub. The hub then resends this routing information out to the spokes, but in this case the hub can summarize the routing information. It sets the IP next hop for all the network destinations to be the NHS (hub) itself. This can significantly reduce the amount of information that the routing protocol needs to distribute from the hub to the spokes, thus reducing the load on the routing protocol running on the hub.

When a data packet is to be forwarded, it again will get the outbound interface and the IP next hop from the matching routing table network entry. If the NHRP interface is the outbound interface then it looks for an NHRP mapping entry for that IP next hop. In this case the IP next hop will be the hub for which it already has an NHRP mapping entry (it already has a tunnel with the hub (NHS)), so the spoke will send only the data packet to the hub.

The hub will receive the data packet and it will check its routing table. Because this data packet is destined for a network behind another spoke it will be forwarded back out the NHRP interface to the next hop toward that spoke. At this point the hub detects that the packet arrived and was sent back out the NHRP interface. This means that the data packet is taking at least two hops within the NHRP network and therefore this path via the hub is not the optimal one-hop path. The hub therefore sends an NHRP redirect message to the spoke. In the redirect message is information to the spoke about the data packet IP destination that triggered the NHRP redirect message.

When the spoke receives the NHRP redirect it will create and send an NHRP resolution request for the data IP destination from the NHRP redirect message. The NHRP resolution request will be forwarded through the path to the remote spoke that services the network for that IP destination.

The remote spoke will generate an NHRP resolution reply with its own NBMA address and the whole subnet (from its routing table) that matches the data IP destination from the NHRP resolution request packet. The remote spoke will then send the NHRP resolution reply directly back to the local spoke. At this point there is now sufficient information for data traffic to be sent over the direct spoke-to-spoke path that was just built.

Note The method for Phase 2 was implemented in Cisco IOS Release 12.4(6)T and uses the NHRP commands **ip nhrp redirect** and **ip nhrp shortcut**.

The IP routing table and the routes learned by way of the hub are important when building spoke-to-spoke tunnels. Therefore the availability of the NHSs (hubs) is critical for the functioning of an NHRP-based network. When there is only one hub and that hub goes down, the spoke removes the routes that it learned from the hub from its routing table, because it lost the hub as its routing neighbor. However, the spoke does not delete any of the spoke-to-spoke tunnels (NHRP mappings) that are now up. Even though the spoke-to-spoke tunnel is still there the spoke will not be able to use the tunnel because its routing table no longer has a route to the destination network. The spoke has a path (spoke-to-spoke tunnel), but does not know to use it (no routing table entry).

In addition, when the routing entries are removed there is no trigger into NHRP for NHRP to remove NHRP mapping entries. Eventually NHRP will time out the current dynamic NHRP mapping entries that it had when the hub went down because they are not being used. Only at that time does NHRP remove the mapping entry.

In **Phase 2**, if there still happened to be a route in the routing table (could be a static route) with the correct IP next hop, then the spoke could still use the spoke-to-spoke tunnel even when the hub is down. NHRP will not be able to refresh the mapping entry because the NHRP resolution request or response would need to go through the hub.

In **Phase 3** you would need a route that only points out the tunnel interface. It would not need have to have the correct IP next hop (NHRP ignores the IP next-hop in Phase 3). Also NHRP will be able to refresh the NHRP mapping entry, because the NHRP resolution request or response will go over the direct spoke-to-spoke tunnel.

If you have two (or more) NHS hubs within a single NBMA network (single mGRE, Frame Relay, or ATM interface), then when the first (primary) hub goes down, the spoke router will still remove the routes from the routing table that it learned from this hub, but it will also be learning the same routes (higher metric) from the second (backup) hub, so it will immediately install these routes. Therefore the spoke-to-spoke traffic would continue going over the spoke-to-spoke tunnel, and be unaffected by the primary hub outage.

Spoke Refresh Mechanism

Spoke-to-spoke tunnels are designed to be dynamic, in that they are created only when there is data traffic to use the tunnel and they are removed when there is no longer any data traffic using the tunnel. This section describes the mechanism to refresh the spoke-to-spoke tunnel when it is still being used (no packet loss) and to detect and remove the spoke-to-spoke tunnel when it is no longer being used.

Process Switching

Each time a data packet is switched using an NHRP mapping entry the "used" flag is set on the mapping entry. Then when the NHRP background process runs (every 60 seconds) the following happens:

- If the expire time is >120 seconds and the "used" flag is set, then the "used" flag is cleared.
- If the expire time is <= 120 seconds and the "used" flag is set, then the entry is refreshed.
- If the expire time is <= 120 seconds and the "used" flag is not set, then nothing is done.

CEF Switching

NHRP has no knowledge about when a packet is Cisco Express Forwarding (CEF) switched through the spoke-to-spoke tunnel.

When the NHRP background process runs the following happens:

- If the expire time is > 120 seconds then nothing is done.
- If the expire time is <= 120 seconds, then the corresponding CEF adjacency is marked "stale". If the CEF adjacency is then used to switch a packet, CEF will mark the adjacency "fresh" and trigger NHRP to refresh the mapping entry.

In both the process and CEF switching cases, refreshed means that another NHRP resolution request is sent and response is needed to keep the entry from expiring. If the expiration time goes to 0 then the NHRP mapping entry is deleted. Also, if this entry is the last mapping entry with this NBMA address and if the router is CEF switching, then the CEF adjacency will be cleared and marked incomplete.

If the IPsec **tunnel protection ipsec profile** *name* command is used on an NHRP mGRE interface, then the following also occurs:

1. The corresponding crypto socket entry will be deleted.
2. The corresponding crypto map entry will be deleted.
3. The corresponding IPsec security associations (SAs) and Internet Security Association and Key Management Protocol (ISAKMP) SAs will be deleted.
4. Just prior to removing the ISAKMP SA, Phase 2 and Phase 1 delete notify messages will be sent to the ISAKMP peer.
5. The ISAKMP peer will delete the corresponding IPsec SAs and ISAKMP SAs.
6. Via the crypto socket the ISAKMP peer's NHRP mapping entry will have its expire time set to 5 seconds, unless it is a static NHRP mapping entry.
7. When the NHRP mapping entry expires and if it is the last mapping entry with this NBMA address, then the ISAKMP peer also does items 1 through 5.

How to Configure NHRP

To implement basic NHRP functionality the first two tasks are required. After NHRP is operational, and depending on your network setup, you can use the other optional tasks to further configure or modify the operation of NHRP.

Note In the following tasks, DMVPN (GRE IP with IPSEC) is referred to and used for all examples because DMVPN is the primary solution where NHRP is used.

This section contains the following procedures:

Configuring a GRE Tunnel for Multipoint Operation

You can enable a GRE tunnel to operate in multipoint fashion. A tunnel network of multipoint tunnel interfaces can be thought of as an NBMA network. When multiple GRE tunnels are configured on the same router they must either have unique tunnel ID keys or unique tunnel source addresses. NHRP is required on mGRE tunnel interfaces, because it provides the VPN-layer-IP to NBMA-layer- IP address mappings for forwarding IP data packets over the mGRE tunnel.

Note Prior to Cisco IOS Release 12.3(11)T, all mGRE interfaces required the configuration of a tunnel ID key. After Cisco IOS Release 12.3(11)T this is optional, but if multiple GRE (mGRE) interfaces are configured on the same router without a tunnel ID key, then the mGRE interfaces be configured with unique tunnel source addresses.

The tunnel ID key is carried in each GRE packet, it is not carried in any NHRP messages. We do not recommend relying on this key for security purposes.

Perform this task to configure a GRE tunnel for multipoint (NBMA) operation.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **tunnel mode gre multipoint**
5. **tunnel key** *key-number*
6. **ip nhrp network-id** *number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`enable`** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**`configure terminal`** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**`interface`** *`type number`* **Example:** `Router(config)# interface tunnel 100`	Configures an interface and enters interface configuration mode.
Step 4	**`tunnel mode gre multipoint`** **Example:** `Router(config-if)# tunnel mode gre multipoint`	Enables a GRE tunnel to be used in multipoint NBMA mode.
Step 5	**`tunnel key`** *`key-number`* **Example:** `Router(config-if)# tunnel key 3`	(Optional) Sets the tunnel ID key • See the "NHRP on a Multipoint Tunnel: Example" section on page 33 for an example of NHRP configured on a multipoint tunnel.
Step 6	**`ip nhrp network-id`** *`number`* **Example:** `Router(config-if)# ip nhrp network-id 1`	Enables NHRP on the interface.

Enabling NHRP on an Interface

The NHRP network ID is used to define the NHRP domain for an NHRP interface and differentiate between multiple NHRP domains or networks, when two or more NHRP domains (GRE tunnel interfaces) are available on the same NHRP node (router). The NHRP network ID is used to help keep two NHRP networks (clouds) separate from each other when both are configured on the same router.

The NHRP network ID is a local only parameter. It is significant only to the local router and it is not transmitted in NHRP packets to other NHRP nodes. For this reason the actual value of the NHRP network ID configured on a router need not match the same NHRP network ID on another router where both of these routers are in the same NHRP domain. As NHRP packets arrive on a GRE interface, they are assigned to the local NHRP domain in the NHRP network ID that is configured on that interface.

Note This method of assigning a network ID is similar to the Open Shortest Path First (OSPF) concept of process ID in the **router ospf** *id* command . If more than one OSPF process is configured, then the OSPF neighbors and any routing data that they provide is assigned to the OSPF process (domain) by which interfaces map to the *network* arguments under the different **router ospf** *id* configuration blocks.

We recommend that the same NHRP network ID be used on the GRE interfaces on all router that are in the same NHRP network. It is then easier to track which GRE interfaces are members of which the NHRP network.

NHRP domains (network IDs) can be unique on each GRE tunnel interface on a router. This is required when running DMVPN Phase 1 or Phase 2 or when using a tunnel key on the GRE interfaces. This places each GRE interface into a different NHRP domain, which is equivalent to each being in a unique DMVPN network.

NHRP domains can span across GRE tunnel interfaces on a route. This option is available when running DMVPN Phase 3 and not using a tunnel key on the GRE tunnel interfaces. In this case the effect of using the same NHRP network ID on the GRE tunnel interfaces is to “glue” the two GRE interfaces into a single NHRP network (DMVPN network).

Perform this task to enable NHRP for an interface on a router. In general, all NHRP stations within a logical NBMA network should be configured with the same network identifier.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address network-mask*
5. **ip nhrp network-id** *number*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface tunnel 100`	Configures an interface and enters interface configuration mode.
Step 4	**ip address** *ip-address network-mask* ***Example:*** `Router(config-if)# ip address 10.0.0.1 255.255.255.0`	Enables IP and gives the interface an IP address.
Step 5	**ip nhrp network-id** *number* **Example:** `Router(config-if)# ip nhrp network-id 1`	Enables NHRP on the interface.
Step 6	**end** **Example:** `Router(config)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring a Static IP-to-NBMA Address Mapping on a Station

To participate in NHRP, a station connected to an NBMA network must be configured with the IP and NBMA addresses of its NHSs. The format of the NBMA address depends on the medium you are using. For example, GRE uses a network service access point (NSAP) address, Ethernet uses a MAC address, and SMDS uses an E.164 address.

These NHSs may also be the default or peer routers of the station, so their addresses can be obtained from the network layer forwarding table of the station.

If the station is attached to several link layer networks (including logical NBMA networks), the station should also be configured to receive routing information from its NHSs and peer routers so that it can determine which IP networks are reachable through which link layer networks.

Perform this task to configure static IP-to-NBMA address mapping on a station (host or router). To enable IP multicast and broadcast packets to be sent to the statically configured station, use the **ip nhrp map multicast** *nbma-address* command. This step is required on multipoint GRE tunnels and not required on point-point RE tunnels.

Note The IGP routing protocol uses IP multicast or broadcast, so this step, though optional, is often required.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** type *number*
4. **ip nhrp map** *ip-address nbma-address*
5. **ip nhrp map multicast** *nbma-address*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface tunnel 100`	Configures an interface and enters interface configuration mode.
Step 4	**ip nhrp map** *ip-address nbma-address* **Example:** `Router(config-if)# ip nhrp map 10.0.0.2 172.16.1.2`	Configures static IP-to-NBMA address mapping on the station.
Step 5	**ip nhrp map multicast** *nbma-address* **Example:** `Router(config-if)# ip nhrp map multicast 172.16.12`	(Optional) Adds an NBMA address to receive multicast or broadcast packets sent out the interface. **Note** This command is not required on point-to-point GRE (p=pGre) tunnels.

Statically Configuring a Next Hop Server

A NHS normally uses the network layer forwarding table to determine where to forward NHRP packets and to find the egress point from an NBMA network. A NHS may also be statically configured with a set of IP address prefixes that correspond to the IP addresses of the stations it serves, and their logical NBMA network identifiers.

Perform this task to statically configure a Next Hop Server.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip nhrp nhs** *nhs-address* [*net-address* [*netmask*]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`configure terminal` **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	`interface` *`type number`* **Example:** `Router(config)# interface tunnel 100`	Configures an interface and enters interface configuration mode.
Step 4	`ip nhrp nhs` *`nhs-address`* `[`*`net-address`* `[`*`netmask`*`]]` **Example:** `Router(config-if)# ip nhrp nhs 10.0.0.2`	Statically configures a Next Hop Server. • To configure multiple networks that the Next Hop Server serves, repeat the **ip nhrp nhs** command with the same Next Hop Server address, but different IP network addresses. • To configure additional Next Hop Servers, repeat the **ip nhrp nhs** command.

Changing the Length of Time NBMA Addresses Are Advertised as Valid

You can change the length of time that NBMA addresses are advertised as valid in positive NHRP responses. In this context, *advertised* means how long the Cisco IOS software tells other routers to keep the address mappings it is providing in NHRP responses. The default length of time is 7200 seconds (2 hours). Perform this task to change the length of time.

This controls how long a spoke-to-spoke shortcut path will stay up after it is no longer used or how often the spoke-to-spoke short-cut path mapping entry will be refreshed if it is still being used. We recommend that a value from 300 to 600 seconds be used.

The **ip nhrp holdtime** command controls how often the NHRP NHC will send NHRP registration requests to its configured NHRP NHSs. The default is to send NHRP Registrations every one third the NHRP holdtime value (default = 2400 seconds (40 minutes)). The optional **ip nhrp registration timeout** *value* command can be used to set the interval for sending NHRP registration requests independently from the NHRP holdtime.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip nhrp holdtime** *seconds*
5. **ip nhrp registration timeout** *seconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`configure terminal` **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	`interface` *type number* **Example:** `Router(config)# interface tunnel 100`	Configures an interface and enters interface configuration mode.
Step 4	`ip nhrp holdtime` *seconds* **Example:** `Router(config-if)# ip nhrp holdtime 600`	Changes the number of seconds that NHRP NBMA addresses are advertised as valid in positive NHRP responses. • In this example, NHRP NBMA addresses are advertised as valid in positive NHRP responses for 10 minutes.
Step 5	`ip nhrp registration timeout` *seconds* **Example:** `Router(config-if)# ip nhrp registration timeout 100`	(Optional)Changes the interval that NHRP NHCs send NHRP registration requests to configured NHRP NHSs. • In this example, NHRP registration requests are now sent every 100 seconds (default value is one third NHRP holdtime value).

Specifying the NHRP Authentication String

Configuring an authentication string ensures that only routers configured with the same string can communicate using NHRP. Therefore, if the authentication scheme is to be used, the same string must be configured in all devices configured for NHRP on a fabric. Perform this task to specify the authentication string for NHRP on an interface.

Note We recommend using an NHRP authentication string, especially to help keep multiple NHRP domains separate from each other. The NHRP authentication string is not encrypted, so it cannot be used as a true authentication for an NHRP node trying to enter the NHRP network (cloud).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip nhrp authentication** *string*
5. **exit**
6. **show ip nhrp** [**dynamic** | **static**] [*type number*]
7. **show ip nhrp traffic**
8. **show ip nhrp nhs** [**detail**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface tunnel 100`	Configures an interface and enters interface configuration mode.
Step 4	**ip nhrp authentication** *string* **Example:** `Router(config-if)# ip nhrp authentication specialxx`	Specifies an authentication string. • All routers configured with NHRP within one logical NBMA network must share the same authentication string.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits interface configureation mode and returns to privileged EXEC mode.
Step 6	**show ip nhrp** [**dynamic** | **static**] [*type number*] **Example:** `Router# show ip nhrp`	Displays the IP NHRP cache, can be limited to dynamic or static cache entries for a specific interface.
Step 7	**show ip nhrp traffic** **Example:** `Router# show ip nhrp traffic`	Displays NHRP traffic statistics.
Step 8	**show ip nhrp nhs** [**detail**] **Example:** `Router# show ip nhrp nhs detail`	Displays NHRP holdtime details.

Configuring NHRP Server-Only Mode

You can configure an interface so that it cannot initiate NHRP resolution requests to establish NHRP shortcut SVCs but can respond only to NHRP resolution requests. Configure NHRP server-only mode on routers you do not want placing NHRP resolution requests.

If an interface is placed in NHRP server-only mode, you have the option to specify the**ip nhrp server-only** [**non-caching**] command keyword. In this case, NHRP does not store mapping information in the NHRP cach, such as NHRP responses that go through the router. To save memory and block building of NHRP shortcuts, the non-caching option is generally used on a router located between two other NHRP routers (NHRP hubs).

Perform this task to configure NHRP server-only mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip nhrp server-only** [**non-caching**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`enable`** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**`configure terminal`** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**`interface`** *`type number`* **Example:** `Router(config)# interface tunnel 100`	Configures an interface and enters interface configuration mode.
Step 4	**`ip nhrp server-only`** `[`**`non-caching`**`]` **Example:** `Router(config-if)# ip nhrp server-only non-caching`	Configures NHRP server-only mode.

Controlling the Triggering of NHRP

There are two ways to control when NHRP is triggered on any platform. These methods are described in the following sections:

- Triggering NHRP on a per-Destination Basis, page 19
- Triggering NHRP on a Packet Count Basis, page 20

Triggering NHRP on a per-Destination Basis

You can specify an IP access list that is used to decide which IP packets can trigger the sending of NHRP resolution requests. By default, all non-NHRP packets trigger NHRP resolution requests. To limit which IP packets trigger NHRP resolution requests, define an access list and then apply it to the interface.

Note NHRP resolution requests are used to build direct paths between two NHRP nodes. Even though certain traffic is excluded from triggering the building of this path, if the path is already built then this "excluded" traffic will use the direct path.

Perform the following task to trigger NHRP on a per-destination basis.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*]

   or

   access-list *access-list-number* {**deny** | **permit**} *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**tos** *tos*] [**established**] [**log**]
4. **interface** *type number*
5. **ip nhrp interest** *access-list-number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*] or **access-list** *access-list-number* {**deny** | **permit**} *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**tos** *tos*] [**established**] [**log**] **Example:** `Router(config)# access-list 101 permit ip any any` or `Router(config)# access-list 101 deny ip any 10.3.0.0 0.0.255.255`	Defines a standard or extended IP access list.
Step 4	**interface** *type number* **Example:** `Router(config)# interface tunnel 100`	Configures an interface and enters interface configuration mode.
Step 5	**ip nhrp interest** *access-list-number* **Example:** `Router(config-if)# ip nhrp interest 101`	Specifies an IP access list that controls NHRP requests. • In this example, only the packets that pass extended access list 101 are subject to the default SVC triggering and teardown rates.

Triggering NHRP on a Packet Count Basis

By default, when the software attempts to send a data packet to a destination for which it has determined that NHRP can be used, it sends an NHRP request for that destination. Perform this task to configure the system to wait until a specified number of data packets have been sent to a particular destination before NHRP is attempted.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip nhrp use** *usage-count*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface tunnel 100`	Configures an interface and enters interface configuration mode.
Step 4	**ip nhrp use** *usage-count* **Example:** `Router(config-if)# ip nhrp use 5`	Specifies how many data packets are sent to a destination before NHRP is attempted. • In this example, if in the first minute five packets are sent to the first destination and five packets are sent to a second destination, then a single NHRP request is generated for the second destination. • If in the second minute the same traffic is generated and no NHRP responses have been received, then the system resends its request for the second destination.

Triggering NHRP Based on Traffic Thresholds

NHRP can run on Cisco Express Forwarding platforms when NHRP runs with Border Gateway Protocol (BGP). You can configure NHRP to initiate SVCs once a configured traffic rate is reached. Similarly, SVCs can be torn down when traffic falls to another configured rate.

Prior to Cisco IOS Release 12.0, a single packet could trigger an SVC. Now you can configure the traffic rate that must be reached before NHRP sets up or tears down an SVC. Because SVCs are created only for burst traffic, you can conserve resources.

To configure the NHRP triggering and teardown of SVCs based on traffic rate, perform the following tasks. The first task is required; the second and third tasks are optional.

- Changing the Rate for Triggering SVCs, page 23 (required)
- Changing the Sampling Time Period and Sampling Rate, page 23 (optional)
- Applying the Triggering and Teardown Rates to Specific Destinations, page 24 (optional)

Prerequisites

Before you configure the feature whereby NHRP initiation is based on traffic rate, the following conditions must exist in the router:

- GRE must be configured.
- CEF switching or distributed CEF (dCEF) switching must be enabled.

- BGP must be configured on all routers in the network where these enhancements are running.

If your network has CEF switching or dCEF switching and you want NHRP to work (whether with default values or changed values), configure the **ip cef accounting non-recursive** command .

Restrictions

Cisco IOS releases prior to Release 12.0 implemented NHRP draft version 4. Cisco IOS Release 12.0 and later releases implement NHRP draft version 11. These versions are not compatible. Therefore, all routers running NHRP in a network must run the same version of NHRP in order to communicate with each other. All routers must run Cisco IOS Release 12.0 and later releases, or all routers must run a release prior to Release 12.0, but not a combination of the two.

When NHRP runs with BGP, there is way to control the triggering of NHRP packets. This method consists of SVCs being initiated based on the input traffic rate to a given BGP next hop.

When BGP discovers a BGP next hop and enters this BGP route into the routing table, an NHRP request is sent to the BGP next hop. When an NHRP reply is received, a subsequent route is put in the NHRP cache that directly corresponds to the BGP next hop.

A new NHRP request is sent to the same BGP next hop to repopulate the NHRP cache. When an NHRP cache entry is generated, a subsequent map statement to the same BGP next hop is also created.

Aggregate traffic to each BGP next hop is measured and monitored. Once the aggregate traffic has met or exceeded the configured trigger rate, NHRP creates an SVC and sends traffic directly to that destination router. The router tears down the SVC to the specified destinations when the aggregate traffic rate falls to or below the configured teardown rate.

By default, NHRP will set up an SVC for a destination when aggregate traffic for that destination is more than 1 kbps over a running average of 30 seconds. Similarly, NHRP will tear down the SVC when the traffic for that destination drops to 0 kbps over a running average of 30 seconds. There are several ways to change the rate at which SVC setup or teardown occurs. You can change the number of kbps thresholds, or the load interval, or both.

Perform this task to change the number of kilobits per second at which NHRP sets up or tears down the SVC to this destination.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip nhrp trigger-svc** *trigger-threshold teardown-threshold*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface tunnel 100`	Configures an interface and enters interface configuration mode.
Step 4	**ip nhrp trigger-svc** *trigger-threshold teardown-threshold* **Example:** `Router(config-if)# ip nhrp trigger-svc 100 5`	Changes the rate at which NHRP sets up or tears down SVCs. • In this example, the triggering and teardown thresholds are set to 100 kbps and 5 kbps, respectively.

Changing the Sampling Time Period and Sampling Rate

You can change the length of time over which the average trigger rate or teardown rate is calculated. By default, the period is 30 seconds; the range is from 30 to 300 seconds in 30-second increments. This period is for calculations of aggregate traffic rate internal to Cisco IOS software only, and it represents a worst-case time period for taking action. In some cases, the software will act sooner, depending on the ramp-up and fall-off rate of the traffic.

If your Cisco hardware has a Virtual Interface Processor, version 2 adapter, you must perform the following task to change the sampling time. By default, the port adapter sends the traffic statistics to the Route Processor every 10 seconds. If you are using NHRP in dCEF switching mode, you must change this update rate to 5 seconds.

Perform this task to change the sampling time period and the sampling rate.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip cef traffic-statistics** [**load-interval** *seconds*]
4. **ip cef traffic-statistics** [**update-rate** *seconds*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`enable`** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**`configure terminal`** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**`ip cef traffic-statistics`** `[`**`load-interval`** *`seconds`*`]` **Example:** `Router(config)# ip cef traffic-statistics load-interval 120`	Changes the length of time in a sampling period during which trigger and teardown thresholds are averaged. • In this example, the triggering and teardown thresholds are calculated based on an average over 120 seconds.
Step 4	**`ip cef traffic-statistics`** `[`**`update-rate`** *`seconds`*`]` **Example:** `Router(config)# ip cef traffic-statistics update-rate 5`	Specifies the frequency that the port adapter sends the accounting statistics to the RP. • When using NHRP in distributed CEF switching mode, this value must be set to 5 seconds. The default value is 10 seconds.

Applying the Triggering and Teardown Rates to Specific Destinations

Perform this task to impose the triggering and teardown rates on certain destinations. By default, all destinations are measured and monitored for NHRP triggering.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*]

   or

   access-list *access-list-number* {**deny** | **permit**} *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**tos** *tos*] [**log**]
4. **interface** *type number*

1. **ip nhrp interest** *access-list*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`enable`** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**`configure terminal`** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**`access-list`** *`access-list-number`* `{`**`deny`** `|` **`permit`**`}` *`source`* `[`*`source-wildcard`*`]` or **`access-list`** *`access-list-number`* `{`**`deny`** `|` **`permit`**`}` *`protocol source source-wildcard destination destination-wildcard`* `[`**`precedence`** *`precedence`*`]` `[`**`tos`** *`tos`*`]` `[`**`log`**`]` **Example:** `Router(config)# access-list 101 permit ip any any` or `Router(config)# access-list 101 deny ip any 10.3.0.0 0.0.255.255`	Defines a standard or extended IP access list. • In the example an extended access list is defined.
Step 4	**`interface`** *`type number`* **Example:** `Router(config)# interface tunnel 100`	Configures an interface and enters interface configuration mode.
Step 5	**`ip nhrp interest`** *`access-list-number`* **Example:** `Router(config-if)# ip nhrp interest 101`	Specifies an IP access list that controls NHRP requests. • In this example, only the packets that pass extended access list 101 are subject to the default SVC triggering and teardown rates.

Controlling the NHRP Packet Rate

There is the maximum value for the number of NHRP messages that the local NHRP process can handle within a set period of time. This limit protects the router against things like a runaway NHRP process sending NHRP requests or an application (worm) that is doing an IP address scan that is triggering many spoke-to-spoke tunnels.

The larger the *Max-send-interval* the more NHRP packets the system can process and send. These messages do not use much memory and the CPU usage is not be very large per message, however excessive messages causing excessive CPU usage can degrade system performance.

To set a reasonable *Max-send-interval* consider the following information:

- Number of spoke routers being handled by this hub and how often they send NHRP registration requests. To support this load you would need:

    Number of spokes/registration timeout * *Max-send-interval*

For example:

500 spokes with 100 second Registration timeout

Max-send-interval = 500/100*10 = 50

- The maximum number of spoke-to-spoke tunnels that are expected to be up at any one time across the NBMA network:

  spoke-to-spoke tunnels/NHRP holdtime * *Max-send-interval*

  This would cover spoke-to-spoke tunnel creation and the refreshing of spoke-to-spoke tunnels that are used for longer periods of time.

  Then add these together and multiply this by 1.5 or 2.0 to give a buffer.

- The *max-send-interval* can be used to keep the long-term average number of NHRP messages allowed to be sent constant, but allow greater peaks.

By default, the maximum rate at which the software sends NHRP packets is five packets per 10 seconds. The software maintains a per-interface quota of NHRP packets (whether generated locally or forwarded) that can be sent.

Perform this task to change the maximum rate at which NRHP packets will be handled.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip nhrp max-send** *pkt-count* **every** *interval*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`enable`** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**`configure terminal`** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**`interface`** *`type number`* **Example:** `Router(config)# interface tunnel 100`	Configures an interface and enters interface configuration mode.
Step 4	**`ip nhrp max-send`** *`pkt-count`* **`every`** *`interval`* **Example:** `Router(config-if)# ip nhrp max-send 10 every 10`	In this example, 10 NHRP packets can be sent from the interface every 10 seconds (twice the default rate).

Suppressing Forward and Reverse Record Options

To dynamically detect link layer filtering in NBMA networks (for example, SMDS address screens), and to provide loop detection and diagnostic capabilities, NHRP incorporates a Route Record in request and reply packets. The Route Record options contain the network (and link layer) addresses of all intermediate Next Hop Servers between the source and destination (in the forward direction) and between the destination and source (in the reverse direction).

By default, Forward Record options and Reverse Record options are included in NHRP request and reply packets. Perform the following task to suppress forward and reverse record options.

Note Forward and Reverse Record information is required for the proper operation of NHRP, especially in a DMVPN network. Therefore you must not configure suppression of this information.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no ip nhrp record**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface tunnel 100`	Configures an interface and enters interface configuration mode.
Step 4	**no ip nhrp record** **Example:** `Router(config-if)# no ip nhrp record`	Suppresses Forward and Reverse Record options.

Specifying the NHRP Responder IP Address

An NHRP requester that wants to know which Next Hop Server generates an NHRP reply packet can include the responder address option in its NHRP request packet. The Next Hop Server that generates the NHRP reply packet then complies by inserting its own IP address in the NHRP reply. The Next Hop Server uses the primary IP address of the specified interface.

Perform this task to specify which interface the Next Hop Server uses for the NHRP responder IP address.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip nhrp responder** *type number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface serial 0`	Configures a serial interface and enters interface configuration mode.
Step 4	**ip nhrp responder** *type number* **Example:** `Router(config-if)# ip nhrp responder serial 0`	Specifies which interface the Next Hop Server uses for the NHRP responder IP address. • In this example, any NHRP requests for the Responder Address will cause this router acting as a next-hop server to supply the primary IP address of serial interface 0 in the NHRP reply packet. • If an NHRP reply packet being forwarded by a Next Hop Server contains the IP address of that server, the Next Hop Server generates an error indication of type "NHRP Loop Detected" and discards the reply.

Clearing the NHRP Cache

The NHRP cache can contain entries of statically configured NHRP mappings and dynamic entries caused by the Cisco IOS software learning addresses from NHRP packets. To clear statically configured entries, use the **no ip nhrp map** command in interface configuration mode.

Perform the following task to clear the NHRP cache.

SUMMARY STEPS

1. **enable**
2. **clear ip nhrp** [*ip-address*] [*ip-mask*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear ip nhrp** [*ip-address*] [*ip-mask*] **Example:** `Router# clear ip nhrp`	Clears the IP NHRP cache of dynamic entries. • This command does not clear any static (configured) IP to NBMA address mappings from the NHRP cache.

Configuration Examples for NHRP

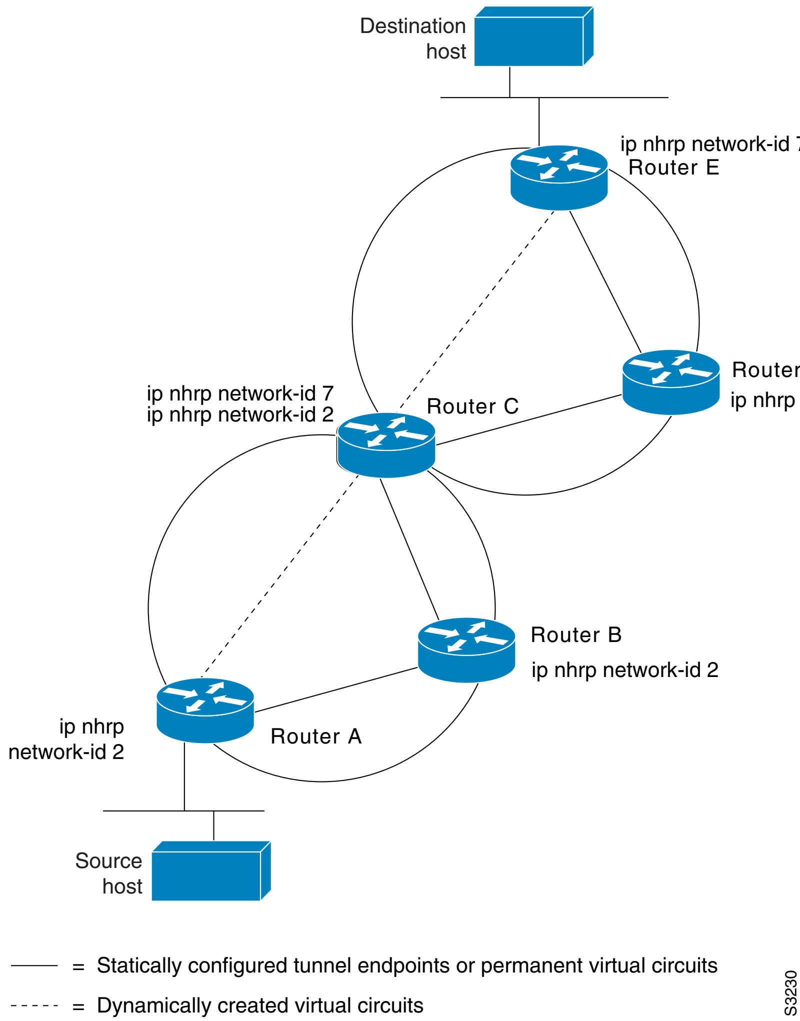
This section provides the following configuration examples:

- Physical Network Designs for Logical NBMA: Examples, page 30
- Applying NHRP Rates to Specific Destinations: Example, page 32
- NHRP on a Multipoint Tunnel: Example, page 33
- Show NHRP: Examples, page 34

Physical Network Designs for Logical NBMA: Examples

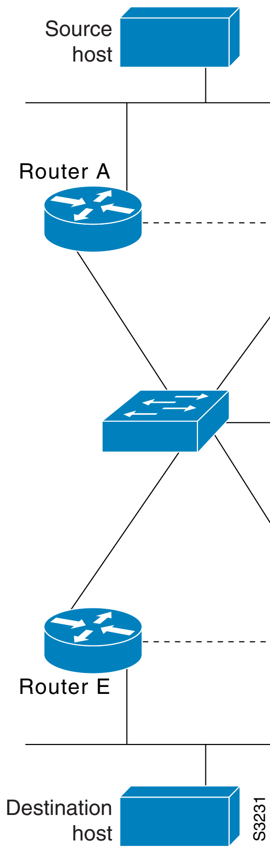
A logical NBMA network is considered the group of interfaces and hosts participating in NHRP and having the same network identifier. Figure 2 illustrates two logical NBMA networks (shown as circles) configured over a single physical NBMA network. Router A can communicate with routers B and C because they share the same network identifier (2). Router C can also communicate with routers D and E because they share network identifier 7. After address resolution is complete, router A can send IP packets to router C in one hop, and router C can send them to router E in one hop, as shown by the dotted lines.

Figure 2 Two Logical NBMA Networks over One Physical NBMA Network



The physical configuration of the five routers in Figure 2 might actually be that shown in Figure 3. The source host is connected to router A and the destination host is connected to router E. The same switch serves all five routers, making one physical NBMA network.

Figure 3 Physical Configuration of a Sample NBMA Network



Refer again to Figure 2. Initially, before NHRP has resolved any NBMA addresses, IP packets from the source host to the destination host travel through all five routers connected to the switch before reaching the destination. When router A first forwards the IP packet toward the destination host, router A also generates an NHRP request for the IP address of the destination host. The request is forwarded to router C, whereupon a reply is generated. Router C replies because it is the egress router between the two logical NBMA networks.

Similarly, router C generates an NHRP request of its own, to which router E replies. In this example, subsequent IP traffic between the source and the destination still requires two hops to traverse the NBMA network, because the IP traffic must be forwarded between the two logical NBMA networks. Only one hop would be required if the NBMA network were not logically divided.

Applying NHRP Rates to Specific Destinations: Example

In the following example, only the packets that pass extended access list 101 are subject to the default SVC triggering and teardown rates:

```
interface tunnel 100
 ip nhrp interest 101
!
access-list 101 permit ip any any
access-list 101 deny ip any 10.3.0.0 0.0.255.255
```

NHRP on a Multipoint Tunnel: Example

With multipoint tunnels, a single tunnel interface may be connected to multiple neighboring routers. Unlike point-to-point tunnels, a tunnel destination need not be configured. In fact, if configured, the tunnel destination must correspond to an IP multicast address. Broadcast or multicast packets to be sent over the tunnel interface can then be sent by sending the GRE packet to the multicast address configured as the tunnel destination.

Multipoint tunnels require that you configure a tunnel key. Otherwise, unexpected GRE traffic could easily be received by the tunnel interface. For simplicity, we recommend that the tunnel key correspond to the NHRP network identifier.

In the following example, routers A, B, C, and D all share an Ethernet segment. Minimal connectivity over the multipoint tunnel network is configured, thus creating a network that can be treated as a partially meshed NBMA network. Due to the static NHRP map entries, router A knows how to reach router B, router B knows how to reach router C, router C knows how to reach router D, and router D knows how to reach Router A.

When router A initially attempts to send an IP packet to router D, the packet is forwarded through routers B and C. The routers use NHRP to quickly learn the NBMA addresses of each other (in this case, IP addresses assigned to the underlying Ethernet network). The partially meshed tunnel network readily becomes fully meshed, at which point any of the routers can directly communicate over the tunnel network without their IP traffic requiring an intermediate hop.

The significant portions of the configurations for routers A, B, C, and D follow:

Router A Configuration

```
interface tunnel 0
 no ip redirects
 ip address 11.0.0.1 255.0.0.0
 ip nhrp map 11.0.0.2 10.0.0.2
 ip nhrp network-id 1
 ip nhrp nhs 11.0.0.2
 tunnel source ethernet 0
 tunnel mode gre multipoint
 tunnel key 1

interface ethernet 0
 ip address 10.0.0.1 255.0.0.0
```

Router B Configuration

```
interface tunnel 0
 no ip redirects
 ip address 11.0.0.2 255.0.0.0
 ip nhrp map 11.0.0.3 10.0.0.3
 ip nhrp network-id 1
 ip nhrp nhs 11.0.0.3
 tunnel source ethernet 0
 tunnel mode gre multipoint
 tunnel key 1

interface ethernet 0
 ip address 10.0.0.2 255.0.0.0
```

Router C Configuration

```
interface tunnel 0
 no ip redirects
 ip address 11.0.0.3 255.0.0.0
 ip nhrp map 11.0.0.4 10.0.0.4
```

```
 ip nhrp network-id 1
 ip nhrp nhs 11.0.0.4
 tunnel source ethernet 0
 tunnel mode gre multipoint
 tunnel key 1

interface ethernet 0
 ip address 10.0.0.3 255.0.0.0
```

Router D Configuration

```
interface tunnel 0
 no ip redirects
 ip address 11.0.0.4 255.0.0.0
 ip nhrp map 11.0.0.1 10.0.0.1
 ip nhrp network-id 1
 ip nhrp nhs 11.0.0.1
 tunnel source ethernet 0
 tunnel mode gre multipoint
 tunnel key 1

interface ethernet 0
 ip address 10.0.0.4 255.0.0.0
```

Show NHRP: Examples

The following is sample output from the **show ip nhrp** command:

```
Router# show ip nhrp


10.0.0.2 255.255.255.255, tunnel 100 created 0:00:43 expire 1:59:16

 Type: dynamic Flags: authoritative

 NBMA address: 10.1111.1111.1111.1111.1111.1111.1111.1111.1111.11

10.0.0.1 255.255.255.255, Tunnel0 created 0:10:03 expire 1:49:56

 Type: static Flags: authoritative

 NBMA address: 10.1.1.2
```

The fields in the sample display are as follows:

- The IP address and its network mask in the IP-to-NBMA address cache. The mask is always 255.255.255.255 because Cisco does not support aggregation of NBMA information through NHRP.
- The interface type and number and how long ago it was created (hours:minutes:seconds).
- The time in which the positive and negative authoritative NBMA address will expire (hours:minutes:seconds). This value is based on the **ip nhrp holdtime** command.
- Type of interface:
  - dynamic—NBMA address was obtained from the NHRP Request packet.
  - static—NBMA address was statically configured.
- Flags:

- authoritative—Indicates that the NHRP information was obtained from the Next Hop Server or router that maintains the NBMA-to-IP address mapping for a particular destination.
- implicit—Indicates that the information was learned from the source mapping information of an NHRP resolution request received by the local router, or from an NHRP resolution packet being forwarded through the local router.
- negative—For negative caching; indicates that the requested NBMA mapping could not be obtained.
- unique—Indicates that this NHRP mapping entry must be unique; it cannot be overwritten with a mapping entry that has the same IP address but a different NBMA address.
- registered—Indicates the NHRP mapping entry was created by an NHRP registration request.
- used—Indicates the NHRP mapping was used to forward data packets within the last 60 seconds.
- router—Indicates an NHRP mapping entry that is from a remote router that is providing access to a network or host behind the remote router.
- local—Indicates an NHRP mapping entry for networks local to this router for which this router has answered an NHRP resolution request.
- (no socket)—Indicates an NHRP mapping entry for which IPsec socket (for encryption) has not been triggered. These mapping entries are not used to forward data packets.
- nat—Indicates an NHRP mapping entry for which IPsec socket (for encryption) has not been triggered. These mapping entries are not used to forward data packets.
- NBMA address—Nonbroadcast multiaccess address. The address format is appropriate for the type of network being used (for example, GRE, Ethernet, SMDS, or multipoint tunnel

The following is sample output from the **show ip nhrp traffic** command which displays NHRP traffic statistics:

```
Router# show ip nhrp traffic

Tunnel0

  request packets sent: 2

  request packets received: 4

  reply packets sent: 4

  reply packets received: 2

  register packets sent: 0

  register packets received: 0

  error packets sent: 0

  error packets received: 0
```

The fields shown in the sample display are as follows:

- Tunnel0—Interface type and number.
- request packets sent—Number of NHRP request packets originated from this station.
- request packets received—Number of NHRP request packets received by this station.
- reply packets sent—Number of NHRP reply packets originated from this station.
- reply packets received—Number of NHRP reply packets received by this station.
- register packets sent—Number of NHRP register packets originated from this station. Routers and access servers do not send register packets, so this value is 0.
- register packets received—Number of NHRP register packets received by this station. Routers or access servers do not send register packets, so this value is 0.
- error packets sent—Number of NHRP error packets originated by this station.
- error packets received—Number of NHRP error packets received by this station.

The following example shows output for a specific tunnel, tunnel7:

```
Router# show ip nhrp traffic interface tunnel7

Tunnel7: Max-send limit:100Pkts/10Sec, Usage:0%

Sent: Total 79

18 Resolution Request 10 Resolution Reply 42 Registration Request

0 Registration Reply 3 Purge Request 6 Purge Reply

0 Error Indication 0 Traffic Indication

Rcvd: Total 69

10 Resolution Request 15 Resolution Reply 0 Registration Request

36 Registration Reply 6 Purge Request 2 Purge Reply

0 Error Indication 0 Traffic Indication
```

NHRP holdtime = 600, NHRP registration timeout not set. NHRP registrations will be sent every 200 seconds so the time to detect that an NHS is down would range from 7 to 207 seconds with an average of 107 seconds.

```
Router# show ip nhrp nhs detail

Legend:
E=Expecting replies
R=Responding

Tunnel0:
10.0.0.1 E req-sent 14793 req-failed 1 repl-recv 14751 (00:25:07 ago)
10.0.0.2 req-sent 26 req-failed 9 repl-recv 0
Legend:
E=Expecting replies
R=Responding

Tunnel1:
10.0.1.1 RE req-sent 14765 req-failed 1 repl-recv 14763 (00:01:07 ago)

Pending Registration Requests:
```

```
Registration Request: Reqid 29507, Ret 64 NHS 10.0.0.1
Registration Request: Reqid 29511, Ret 64 NHS 10.0.0.2
```

10.0.0.1 is new (expecting replies) and is down.
10.0.0.2 is old (not expecting replies) and is assumed up.
10.0.1.1 is new (expecting replies) and is up.

Additional References

The following sections provide references related to the configuring NHRP.

Related Documents

Related Topic	Document Title
The DMVPN feature allows users to better scale large and small IP Security (IPsec) Virtual Private Networks (VPNs) by combining generic routing encapsulation (GRE) tunnels, IPsec encryption, and Next Hop Resolution Protocol (NHRP).	*Dynamic Multipoint VPN (DMVPN)*
Routers in a Dynamic Multipoint VPN (DMVPN) network can use the Next Hop Resolution Protocol (NHRP) to discover the addresses of other routers and networks behind those routers that are connected to a DMVPN nonbroadcast multiaccess (NBMA) network. NHRP provides an ARP-like solution that alleviates NBMA network problems, such as hub failure, decreased reliability, and complex configurations.	*Shortcut Switching Enhancements for NHRP in DMVPN Networks*
NRHP commands	*Cisco IOS IP Addressing Services Command Reference*

RFCs

RFC	Title
RFC 2332	NBMA Next Hop Resolution Protocol (NHRP)

Technical Assistance

Description	Link
The Cisco Technical Support & Documentation website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, tools, and technical documentation. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport

Feature Information for Configuring NHRP

Table 1 lists the release history for this feature.

Table 1 lists the features in this module and provides links to specific configuration information.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for Configuring NHRP

Feature Name	Releases	Feature Information
Next Hop Resolution Protocol	Cisco IOS XE Release 2.1	This feature was integrated on the Cisco ASR 1000 Series Routers.

NAT

Configuring Network Address Translation Features Roadmap

This roadmap lists the features documented in the Network Address Translation modules and maps the features to the modules in which they appear.

Roadmap History

This module was first published on May 2, 2005, and last updated on May 2, 2005.

Features and Release Support

Table 1 lists Network Address Translation feature support for the following Cisco IOS software release trains:

- Cisco IOS Releases 12.2T, 12.3, and 12.3T

Only features that were introduced or modified in Cisco IOS Release 12.2 (1) or a later release appear in the table. *Not all features may be supported in your Cisco IOS software release*.

Cisco IOS software images are specific to a Cisco IOS software release, a feature set, and a platform. Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Table 1 Supported Network Address Translation Features

Release	Feature Name	Feature Description	Where Documented
Cisco IOS Releases 12.2T, 12.3, and 12.3T			
12.2(2)T	NAT Support for H.323 v2 RAS	Cisco IOS NAT supports all H.225 and H.245 message types, including those sent in the RAS protocol.	Using Application Level Gateways with NAT
12.2(4)T 12.2(4)T2	NAT—Static Mapping Support with HSRP for High Availability	Static mapping support for HSRP allows the option of having only the HSRP active router respond to an incoming ARP for a router configured with a NAT address.	Configuring NAT for High Availability
12.2(4)T 12.2(4)T2	NAT - Translation of External IP addresses only	Using the NAT translation of external IP address only feature, NAT can be configured to ignore all embedded IP addresses for any application and traffic type.	Configuring NAT for IP Address Conservation
12.2.(4)T	NAT-Ability to Use Route Maps with Static Translation	The dynamic translation command can specify a route map to be processed instead of an access-list. A route map allows you to match any combination of access-list, next-hop IP address, and output interface to determine which pool to use. The ability to use route maps with static translations enables NAT multihoming capability with static address translations.	Configuring NAT for IP Address Conservation
12.2(8)T	NAT Support for SIP feature	NAT Support for SIP adds the ability to deploy Cisco IOS NAT between VoIP solutions based on SIP.	Using Application Level Gateways with NAT
12.2(8)T	NAT Support for SIP feature	NAT Support for SIP adds the ability to deploy Cisco IOS NAT between VoIP solutions based on SIP.	Using Application Level Gateways with NAT
12.2(13)T	Support for IPSec ESP Through NAT	IPSec ESP Through NAT provides the ability to support multiple concurrent IP Security (IPSec) Encapsulating Security Payload (ESP) tunnels or connections through a Cisco IOS Network Address Translation (NAT) device configured in Overload or Port Address Translation (PAT) mode.	Using Application Level Gateways with NAT
12.2(13)T	Network Address Translation (NAT) Integration with MPLS VPNs	This feature allows multiple Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs) to be configured on a single device to work together.	Integrating NAT with MPLS VPNs
12.2(13)T	NAT Stateful Failover of Network Address Translation	The NAT Stateful Failover of Network Address Translation feature represents Phase 1 of the stateful failover capability. It introduces support for two or more network address translators to function as a translation group.	Configuring NAT for High Availability
12.2(15)T	The NAT Support for IPSec ESP— Phase II feature	The NAT Support for IPSec ESP— Phase II feature provides support for Internet Key Exchange (IKE) and ESP without encapsulation in tunnel mode through a Cisco IOS router configured with NAPT.	Using Application Level Gateways with NAT

Table 1 Supported Network Address Translation Features (continued)

Release	Feature Name	Feature Description	Where Documented
12.3(4)T	Rate Limiting NAT Translation	The Rate Limiting NAT Translation feature provides the ability to limit the maximum number of concurrent network address translation (NAT) operations on a router. In addition to giving users more control over how NAT addresses are used, the Rate Limiting NAT Translation feature can be used to limit the effects of viruses, worms, and denial-of-service attacks.	Configuring NAT for IP Address Conservation
12.3(7)T	NAT-Static IP Support	The NAT - Static IP Support feature provides support for users with static IP addresses, enabling those users to establish an IP session in a Public Wireless LAN environment.	Configuring NAT for IP Address Conservation
12.3(7)T	NAT RTSP Support Using NBAR	The Real Time Streaming Protocol (RTSP) is a client-server multimedia presentation control protocol that supports multimedia application delivery. Some of the applications that use RTSP include Windows Media Services (WMS) by Microsoft, QuickTime by Apple Computer, and RealSystem G2 by RealNetworks.	Configuring NAT for IP Address Conservation
12.3(7)T	NAT Stateful Failover for Asymmetric Outside-to-Inside ALG Support	The NAT Stateful Failover for Asymmetric Outside-to-Inside and Application Layer Gateway (ALG) Support feature improves the ability to handle asymmetric paths by allowing multiple routing paths from outside-to-inside, and per-packet load balancing. This feature also provides seamless failover translated IP sessions with traffic that includes embedded IP addressing such as Voice over IP, FTP, and Domain Name System (DNS) applications.	Configuring NAT for High Availability
12.3(11)T	NAT H.245 Tunneling Support	The NAT H.245 Tunneling Support feature allows H.245 tunneling in H.323 Application Level Gateways (ALGs)	Using Application Level Gateways with NAT
12.3(13)T	NAT Default Inside Server	The NAT Default Inside Server feature provides for the need to forward packets from the outside to a specified inside local address.	Configuring NAT for IP Address Conservation
12.3(14)T	NAT Virtual Interface (NVI)	The NAT Virtual Interface (NVI) feature removes the requirement to configure an interface as either Network Address Translation (NAT) inside or NAT outside. An interface can be configured to use NAT or not use NAT.	Configuring NAT for IP Address Conservation
12.3(14)T	NAT Routemaps Outside-to-Inside Support	The NAT Routemaps Outside-to-Inside Support feature enables the deployment of a NAT routemap configuration that will allow IP sessions to be initiated from the outside to the inside.	Configuring NAT for IP Address Conservation

This module first published May 2, 2005. Last updated May 2, 2005

Configuring NAT for IP Address Conservation

NAT enables private IP internetworks that use nonregistered IP addresses to connect to the Internet. NAT operates on a router, usually connecting two networks together, and translates the private (not globally unique) address in the internal network into legal addresses before packets are forwarded onto another network. NAT can be configured to advertise only one address for the entire network to the outside world. This ability provides additional security, effectively hiding the entire internal network behind that one address.

NAT is also used at the Enterprise edge to allow internal users access to the Internet and to allow Internet access to internal devices such as mail servers.

Module History

This module was first published on May 2, 2005, and was last updated on February 27, 2006.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all features. To find information about feature support and configuration, use the "Feature Information for Configuring NAT for IP Address Conservation" section on page 49.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Configuring NAT for IP Address Conservation

Access Lists

All access lists required for use with the tasks in this module should be configured prior to beginning the configuration task. For information about how to configure an access list, refer to the *IP Access List Sequence Numbering* document at the following URL:

http://www.cisco.com/en/US/docs/ios/security/configuration/guide/sec_ip_entry_numbrng.html

Note If you specify an access list to use with a NAT command, NAT does not support the commonly used **permit ip any any** command in the access list.

Defining the NAT Requirements, Objectives, and Interfaces

Before configuring NAT in your network, it is important to understand on which interfaces NAT will be configured and for what purposes. You can use the questions below to determine how you will use NAT and how NAT will need to be configured.

1. Define NAT inside and outside interfaces by answering the following questions:
   - Do users exist off multiple interfaces?
   - Are there multiple interfaces going to the Internet?
2. Define what is trying to be accomplished with NAT by answering the following questions:
   - Should NAT allow internal users to access the Internet?
   - Should NAT allow the Internet to access internal devices such as a mail server?
   - Should NAT redirect TCP traffic to another TCP port or address?
   - Will NAT be used during a network transition?
   - Should NAT allow overlapping networks to communicate?
   - Should NAT allow networks with different address schemes to communicate?
   - Should NAT allow the use of an application level gateway?

Restrictions for Configuring NAT for IP Address Conservation

- NAT is not practical if large numbers of hosts in the stub domain communicate outside of the domain.
- Some applications use embedded IP addresses in such a way that it is impractical for a NAT device to translate them. These applications may not work transparently or at all through a NAT device.
- NAT also hides the identity of hosts, which may be an advantage or a disadvantage depending on the desired result.
- A router configured with NAT must not advertise the local networks to the outside. However, routing information that NAT receives from the outside can be advertised in the stub domain as usual.
- If you specify an access list to use with a NAT command, NAT does not support the commonly used **permit ip any any** command in the access list.

Information About Configuring NAT for IP Address Conservation

To configure NAT for IP address conservation, you should understand the following concepts:

Benefits of Configuring NAT for IP Address Conservation

NAT allows organizations to resolve the problem of IP address depletion when they have existing networks and need to access the Internet. Sites that do not yet possess NIC-registered IP addresses must acquire them, and if more than 254 clients are present or planned, the scarcity of Class B addresses becomes a serious issue. Cisco IOS NAT addresses these issued by mapping thousands of hidden internal addresses to a range of easy-to-get Class C addresses.

Sites that already have registered IP addresses for clients on an internal network may want to hide those addresses from the Internet so that hackers cannot directly attack the clients. With client addresses hidden, a degree of security is established. Cisco IOS NAT gives LAN administrators complete freedom to expand Class A addressing, which is drawn from the reserve pool of the Internet Assigned Numbers Authority (RFC 1597). This expansion occurs within the organization without concern for addressing changes at the LAN/Internet interface.

Cisco IOS can selectively or dynamically perform NAT. This flexibility allows the network administrator to use a mix of RFC 1597 and RFC 1918 addresses or registered addresses. NAT is designed for use on a variety of routers for IP address simplification and conservation. In addition, Cisco IOS NAT allows the selection of which internal hosts are available for NAT.

A significant advantage of NAT is that it can be configured without requiring changes to hosts or routers other than those few routers on which NAT will be configured.

Purpose of NAT

Two key problems facing the Internet are depletion of IP address space and scaling in routing. NAT is a feature that allows the IP network of an organization to appear from the outside to use different IP address space than what it is actually using. Thus, NAT allows an organization with nonglobally routable addresses to connect to the Internet by translating those addresses into globally routable address space. NAT also allows a more graceful renumbering strategy for organizations that are changing service providers or voluntarily renumbering into classless interdomain routing (CIDR) blocks. NAT is described in RFC 1631.

Beginning with Cisco IOS Release 12.1(5)T, NAT supports all H.225 and H.245 message types, including FastConnect and Alerting as part of the H.323 version 2 specification. Any product that makes use of these message types will be able to pass through a Cisco IOS NAT configuration without any static configuration. Full support for NetMeeting Directory (Internet Locator Service) is also provided through Cisco IOS NAT.

How NAT Works

A router configured with NAT will have at least one interface to the inside network and one to the outside network. In a typical environment, NAT is configured at the exit router between a stub domain and backbone. When a packet is leaving the domain, NAT translates the locally significant source address into a globally unique address. When a packet is entering the domain, NAT translates the globally unique destination address into a local address. If more than one exit point exists, each NAT must have the same translation table. If the software cannot allocate an address because it has run out of addresses, it drops the packet and sends an ICMP host unreachable packet.

Uses of NAT

NAT can be used for the following applications:

- When you want to connect to the Internet, but not all your hosts have globally unique IP addresses. NAT enables private IP internetworks that use nonregistered IP addresses to connect to the Internet. NAT is configured on the router at the border of a stub domain (referred to as the *inside network*) and a public network such as the Internet (referred to as the *outside network*). NAT translates the internal local addresses to globally unique IP addresses before sending packets to the outside network.
- When you must change your internal addresses. Instead of changing them, which can be a considerable amount of work, you can translate them by using NAT.
- When you want to do basic load sharing of TCP traffic. You can map a single global IP address to many local IP addresses by using the TCP load distribution feature.

As a solution to the connectivity problem, NAT is practical only when relatively few hosts in a stub domain communicate outside of the domain at the same time. When this is the case, only a small subset of the IP addresses in the domain must be translated into globally unique IP addresses when outside communication is necessary, and these addresses can be reused when no longer in use.

NAT Inside and Outside Addresses

With reference to NAT, the term *inside* refers to those networks that are owned by an organization and that must be translated. Inside this domain, hosts will have addresses in the one address space, while on the outside, they will appear to have addresses in another address space when NAT is configured. The first address space is referred to as the *local* address space and the second is referred to as the *global* address space.

Similarly, *outside* refers to those networks to which the stub network connects, and which are generally not under the control of the organization. Hosts in outside networks can be subject to translation also, and can thus have local and global addresses.

NAT uses the following definitions:

- **Inside local address**—The IP address that is assigned to a host on the inside network. The address is probably not a legitimate IP address assigned by the Network Information Center (NIC) or service provider.
- **Inside global address**—A legitimate IP address (assigned by the NIC or service provider) that represents one or more inside local IP addresses to the outside world.
- **Outside local address**—The IP address of an outside host as it appears to the inside network. Not necessarily a legitimate address, it was allocated from address space routable on the inside.

- **Outside global address**—The IP address assigned to a host on the outside network by the owner of the host. The address was allocated from a globally routable address or network space.

Types of NAT

NAT operates on a router—generally connecting only two networks together—and translates your private (inside local) addresses within the internal network, into public (inside global) addresses before any packets are forwarded to another network. This functionality give you the option to configure NAT so that it will advertise only a single address for your entire network to the outside world. Doing this effectively hides the internal network from the world, giving you some additional security.

NAT types include:

- Static Address Translation—Static NAT—allows one-to-one mapping between local and global addresses.
- Dynamic Address Translation—Dynamic NAT—maps unregistered IP addresses to registered IP addresses of out of a pool of registered IP addresses.
- Overloading—a form of dynamic NAT that maps multiple unregistered IP addresses to a single registered IP address (many to one) using different ports. This method is also known as Port Address Translation (PAT). By using PAT (NAT Overload), thousands of users can be connected to the Internet using only one real global IP address.

How to Configure NAT for IP Address Conservation

The tasks described in this section configure NAT for IP address conservation. No single task in this section is required; however, at least one of the tasks must be performed. More than one of the tasks may be needed. This section contains the following procedures:

Configuring the Inside Source Addresses

Inside source address can be configured for static or dynamic translation. Perform one of the following tasks depending on your requirements:

- Configuring Static Translation of Inside Source Addresses, page 7
- Configuring Dynamic Translation of Inside Source Addresses, page 9

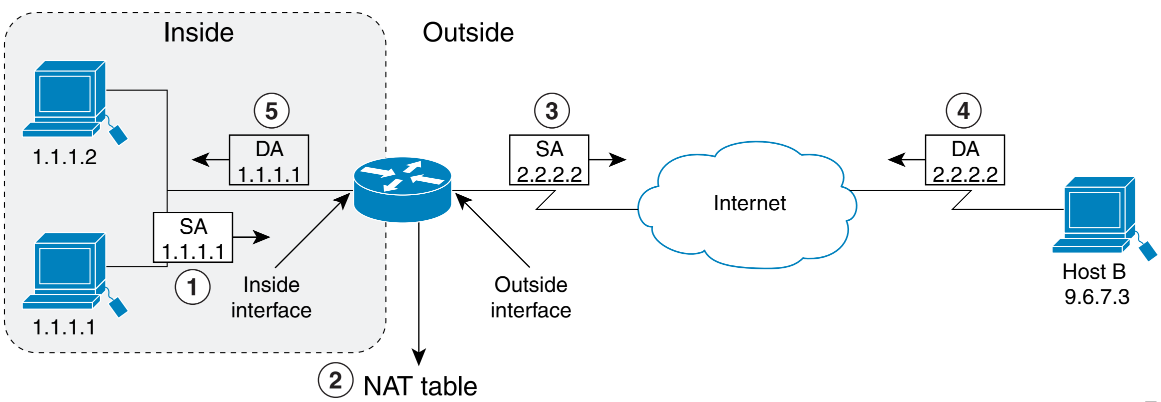
Inside Source Address Translation

You can translate your own IP addresses into globally unique IP addresses when communicating outside of your network. You can configure static or dynamic inside source translation as follows:

- *Static translation* establishes a one-to-one mapping between your inside local address and an inside global address. Static translation is useful when a host on the inside must be accessible by a fixed address from the outside.
- *Dynamic translation* establishes a mapping between an inside local address and a pool of global addresses.

Figure 1 illustrates a router that is translating a source address inside a network to a source address outside the network.

Figure 1 NAT Inside Source Translation



Protocol	Inside Local IP Address	Inside Global IP Address	Outside Global IP Address	Outside Local IP Address
-- -- -- --	1.1.1.2 1.1.1.1	2.2.2.3 2.2.2.2	-- -- -- -- -- -- -- --	-- -- -- -- -- -- -- --

The following process describes inside source address translation, as shown in Figure 1:

1. The user at host 1.1.1.1 opens a connection to host B.
2. The first packet that the router receives from host 1.1.1.1 causes the router to check its NAT table:
   - If a static translation entry was configured, the router goes to Step 3.
   - If no translation entry exists, the router determines that source address (SA) 1.1.1.1 must be translated dynamically, selects a legal, global address from the dynamic address pool, and creates a translation entry. This type of entry is called a *simple entry.*
3. The router replaces the inside local source address of host 1.1.1.1 with the global address of the translation entry and forwards the packet.

4. Host B receives the packet and responds to host 1.1.1.1 by using the inside global IP destination—Address (DA) 2.2.2.2.
5. When the router receives the packet with the inside global IP address, it performs a NAT table lookup by using the inside global address as a key. It then translates the address to the inside local address of host 1.1.1.1 and forwards the packet to host 1.1.1.1.

Host 1.1.1.1 receives the packet and continues the conversation. The router performs Steps 2 through 5 for each packet.

Configuring Static Translation of Inside Source Addresses

Configure static translation of inside source addresses when you want to allow one-to-one mapping between your inside local address and an inside global address. Static translation is useful when a host on the inside must be accessible by a fixed address from the outside.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat inside source static** *local-ip global-ip*
4. **interface** *type number*
5. **ip address** *ip-address mask* **secondary**
6. **ip nat inside**
7. **exit**
8. **interface** *type number*
9. **ip address** *ip-address mask*
10. **ip nat outside**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat inside source static** *local-ip global-ip* **Example:** `Router(config)# ip nat inside source static 10.10.10.1 172.16.131.1`	Establishes static translation between an inside local address and inside global address.
Step 4	**interface** *type number* **Example:** `Router(config)# interface ethernet 1`	Specifies an interface and enters interface configuration mode.
Step 5	**ip address** *ip-address mask* [**secondary**] **Example:** `Router(config-if)# ip address 10.114.11.39 255.255.255.0`	Sets a primary IP address for an interface.
Step 6	**ip nat inside** **Example:** `Router(config-if)# ip nat inside`	Marks the interface as connected to the inside.
Step 7	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to configuration mode.
Step 8	**interface** *type number* **Example:** `Router(config)# interface ethernet 0`	Specifies a different interface and returns interface configuration mode.
Step 9	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 172.31.232.182 255.255.255.240`	Sets a primary IP address for an interface.
Step 10	**ip nat outside** **Example:** `Router(config-if)# ip nat outside`	Marks the interface as connected to the outside.

Configuring Dynamic Translation of Inside Source Addresses

Dynamic translation establishes a mapping between an inside local address and a pool of global addresses. Dynamic translation is useful when multiple users on a private network need to access the Internet. The dynamically configured pool IP address may be used as needed and are released for use by other users when access to the Internet is no longer required.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*}
4. **access-list** *access-list-number* **permit** *source* [*source-wildcard*]
5. **ip nat inside source list** *access-list-number* **pool** *name*
6. **interface** *type number*
7. **ip address** *ip-address mask*
8. **ip nat inside**
9. **exit**
10. **interface** *type number*
11. **ip address** *ip-address mask*
12. **ip nat outside**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*} **Example:** `Router(config)# ip nat pool net-208 171.69.233.208 171.69.233.223 prefix-length 28`	Defines a pool of global addresses to be allocated as needed.
Step 4	**access-list** *access-list-number* **permit** *source* [*source-wildcard*] **Example:** `Router(config)# access-list 1 permit 192.5.34.0 0.0.0.255`	Defines a standard access list permitting those addresses that are to be translated.

	Command or Action	Purpose
Step 5	**ip nat inside source list** *access-list-number* **pool** *name* **Example:** `Router(config)# ip nat inside source list 1 pool net-208`	Establishes dynamic source translation, specifying the access list defined in the prior step.
Step 6	**interface** *type number* **Example:** `Router(config)# interface ethernet 1`	Specifies an interface and enters interface configuration mode.
Step 7	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 10.114.11.39 255.255.255.0`	Sets a primary IP address for the interface.
Step 8	**ip nat inside** **Example:** `Router(config-if)# ip nat inside`	Marks the interface as connected to the inside.
Step 9	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to configuration mode.
Step 10	**interface** *type number* **Example:** `Router(config-if)# interface ethernet 0`	Specifies a different interface and returns to interface configuration mode.
Step 11	**ip address** *ip-address mask* **Example:** `Router(config)# ip address 172.69.232.182 255.255.255.240`	Sets a primary IP address for the interface.
Step 12	**ip nat outside** **Example:** `Router(config-if)# ip nat outside`	Marks the interface as connected to the outside.

Allowing Internal Users Access to the Internet Using NAT

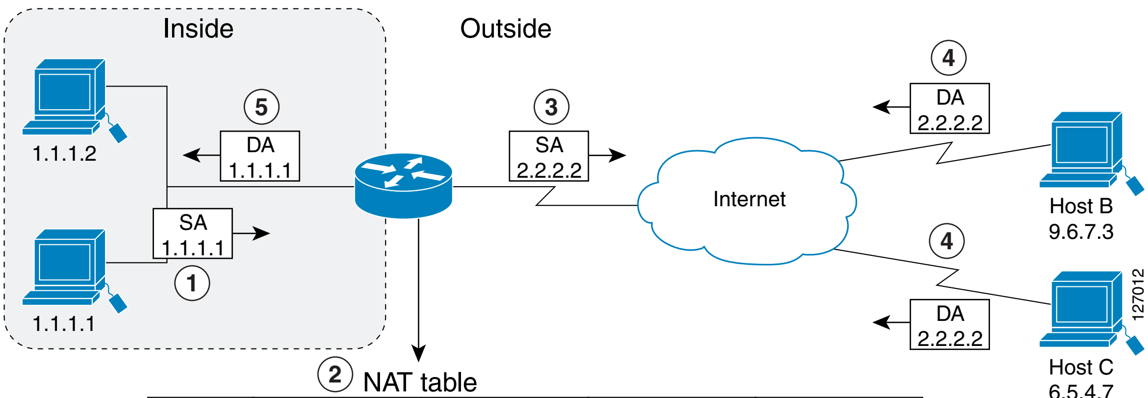
Perform this task to allow your internal users access to the internet and conserve addresses in the inside global address pool using overloading of global addresses.

Inside Global Addresses Overloading

You can conserve addresses in the inside global address pool by allowing the router to use one global address for many local addresses. When this overloading is configured, the router maintains enough information from higher-level protocols (for example, TCP or UDP port numbers) to translate the global address back to the correct local address. When multiple local addresses map to one global address, the TCP or UDP port numbers of each inside host distinguish between the local addresses.

Figure 2 illustrates NAT operation when one inside global address represents multiple inside local addresses. The TCP port numbers act as differentiators.

Figure 2 NAT Overloading Inside Global Addresses



Protocol	Inside Local IP address:port	Inside Global IP address:port	Outside Global IP address:port	Outside Local IP address
TCP	1.1.1.2:1723	2.2.2.2:1723	6.5.4.7:23	6.5.4.7:23
TCP	1.1.1.1:1024	2.2.2.2:1024	9.6.7.3:23	9.6.7.3:23

The router performs the following process in overloading inside global addresses, as shown in Figure 2. Both host B and host C believe they are communicating with a single host at address 2.2.2.2. They are actually communicating with different hosts; the port number is the differentiator. In fact, many inside hosts could share the inside global IP address by using many port numbers.

1. The user at host 1.1.1.1 opens a connection to host B.
2. The first packet that the router receives from host 1.1.1.1 causes the router to check its NAT table:
   - If no translation entry exists, the router determines that address 1.1.1.1 must be translated, and sets up a translation of inside local address 1.1.1.1 to a legal global address.
   - If overloading is enabled, and another translation is active, the router reuses the global address from that translation and saves enough information to be able to translate back. This type of entry is called an *extended entry.*
3. The router replaces the inside local source address 1.1.1.1 with the selected global address and forwards the packet.
4. Host B receives the packet and responds to host 1.1.1.1 by using the inside global IP address 2.2.2.2.

5. When the router receives the packet with the inside global IP address, it performs a NAT table lookup, using the protocol, the inside global address and port, and the outside address and port as a key; translates the address to inside local address 1.1.1.1; and forwards the packet to host 1.1.1.1.

Host 1.1.1.1 receives the packet and continues the conversation. The router performs Steps 2 through 5 for each packet.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat pool** *name start-ip end-ip* {**netmask** *netmask*| **prefix-length** *prefix-length*}
4. **access-list** *access-list-number* **permit** *source* [*source-wildcard]*
5. **ip nat inside source list** *access-list-number* **pool** *name* **overload**
6. **interface** *type number*
7. **ip address** *ip-address mask*
8. **ip nat inside**
9. **exit**
10. **interface** *type number*
11. **ip address** *ip-address mask*
12. **ip nat outside**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`configure terminal` **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*} **Example:** `Router(config)# ip nat pool net-208 171.69.233.208 171.69.233.233 netmask 255.255.255.240`	Defines a pool of global addresses to be allocated as needed.
Step 4	**access-list** *access-list-number* **permit** *source* [*source-wildcard*] **Example:** `Router(config)# access-list 1 permit 192.5.34.0 0.0.0.255`	Defines a standard access list permitting those addresses that are to be translated. • The access list must permit only those addresses that are to be translated. (Remember that there is an implicit "deny all" at the end of each access list.) An access list that is too permissive can lead to unpredictable results.

	Command or Action	Purpose
Step 5	**ip nat inside source list** *access-list-number* **pool** *name* **overload** **Example:** `Router(config)# ip nat inside source list 1 pool net-208 overload`	Establishes dynamic source translation with overloading, specifying the access list defined in the prior step.
Step 6	**interface** *type number* **Example:** `Router(config)# interface ethernet 1`	Specifies an interface and enters interface configuration mode.
Step 7	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 10.114.11.39 255.255.255.0`	Sets a primary IP address for the interface.
Step 8	**ip nat inside** **Example:** `Router(config-if)# ip nat inside`	Marks the interface as connected to the inside.
Step 9	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to configuration mode.
Step 10	**interface** *type number* **Example:** `Router(config)# interface ethernet 0`	Specifies a different interface and returns to interface configuration mode.
Step 11	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 172.69.232.182 255.255.255.240`	Sets a primary IP address for the interface.
Step 12	**ip nat outside** **Example:** `Router(config-if)# ip nat outside`	Marks the interface as connected to the outside.

Configuring Address Translation Timeouts

The tasks in this section are presented together because they address similar objectives, but you must select the one that is applicable to the specific configuration of NAT.

Perform one of the following tasks:

- Changing the Translation Timeout Default, page 14
- Changing the Default Timeouts When Overloading Is Configured, page 14

Changing the Translation Timeout Default

By default, dynamic address translations time out after some period of non-use. You can change the default values on timeouts, if necessary. When overloading is not configured, simple translation entries time out after 24 hours.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat translation timeout** *seconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`configure terminal` **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	`ip nat translation timeout` *seconds* **Example:** `Router(config)# ip nat translation timeout 500`	Changes the timeout value for dynamic address translations that do not use overloading.

Changing the Default Timeouts When Overloading Is Configured

If you have configured overloading, you have more control over translation entry timeout, because each entry contains more context about the traffic using it. To change timeouts on extended entries, use the following commands as needed.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat translation udp-timeout** *seconds*
4. **ip nat translation dns-timeout** *seconds*
5. **ip nat translation tcp-timeout** *seconds*
6. **ip nat translation finrst-timeout** *seconds*
7. **ip nat translation icmp-timeout** *seconds*
8. **ip nat translation syn-timeout** *seconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat translation udp-timeout** *seconds* **Example:** `Router(config)# ip nat translation udp-timeout 300`	(Optional) Changes the UDP timeout value from 5 minutes.
Step 4	**ip nat translation dns-timeout** *seconds* **Example:** Router(config)# **ip nat translation dns-timeout** *45*	(Optional) Changes the DNS timeout value from 1 minute.
Step 5	**ip nat translation tcp-timeout** *seconds* **Example:** Router(config)# i**p nat translation tcp-timeout** *2500*	(Optional) Changes the TCP timeout value from 24 hours.
Step 6	**ip nat translation finrst-timeout** *seconds* **Example:** Router(config)# **ip nat translation finrst-timeout** *45*	(Optional) Changes the Finish and Reset timeout value from 1 minute.
Step 7	**ip nat translation icmp-timeout** *seconds* **Example:** Router(config)# ip **nat translation icmp-timeout** *45*	(Optional) Changes the ICMP timeout value from 24 hours.
Step 8	**ip nat translation syn-timeout** *seconds* **Example:** `Router(config)# ip nat translation syn-timeout 45`	(Optional) Changes the Synchronous (SYN) timeout value from 1 minute.

Allowing Overlapping Networks to Communicate Using NAT

The tasks in this section are group together because they perform the same action but are executed differently depending on the type of translation that is implemented: static or dynamic.

Perform the task that applies to the translation type that is implemented.

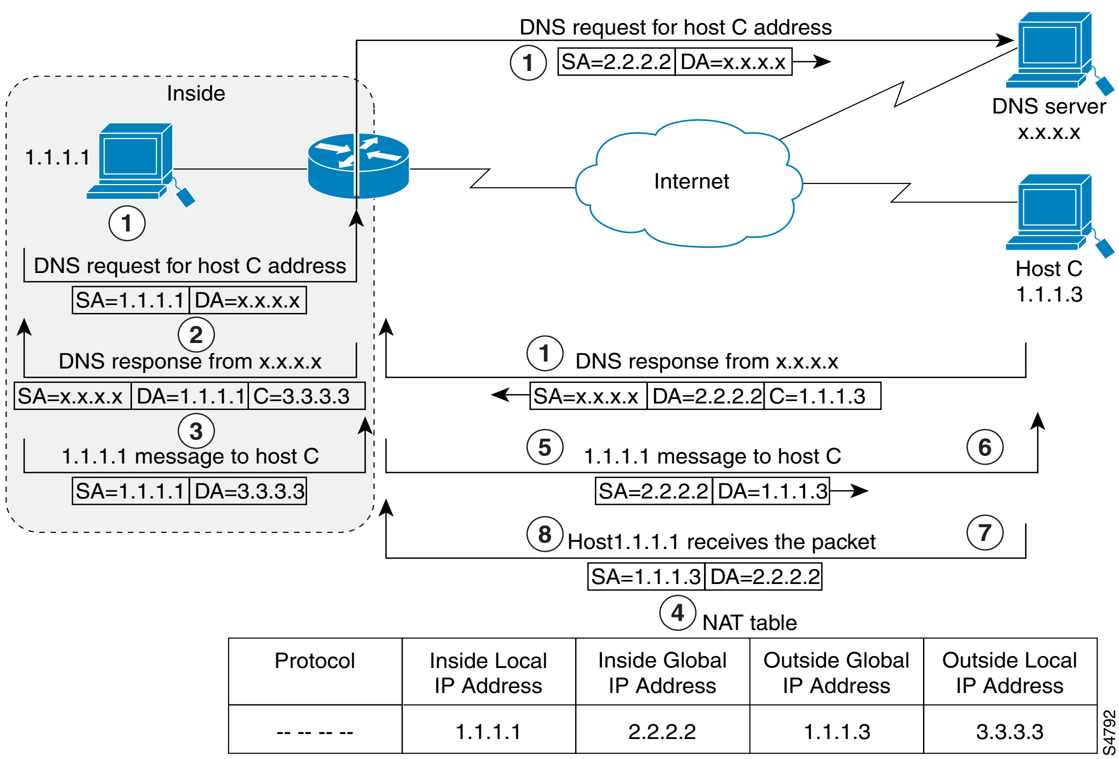
- Configuring Static Translation of Overlapping Networks, page 17
- Configuring Dynamic Translation of Overlapping Networks, page 19

Address Translation of Overlapping Networks

NAT is used to translate your IP addresses, which could occur because your IP addresses are not legal, officially assigned IP addresses. Perhaps you chose IP addresses that officially belong to another network. The case of an address used both illegally and legally is called *index overlapping*. You can use NAT to translate inside addresses that overlap with outside addresses.

Figure 3 shows how NAT translates overlapping networks.

Figure 3 NAT Translating Overlapping Addresses



The router performs the following process when translating overlapping addresses:

1. The user at host 1.1.1.1 opens a connection to host C by name, requesting a name-to-address lookup from a DNS server.
2. The router intercepts the DNS reply and translates the returned address if there is an overlap (that is, the resulting legal address resides illegally in the inside network). To translate the return address, the router creates a simple translation entry mapping the overlapping address 1.1.1.3 to an address from a separately configured, outside local address pool.

The router examines every DNS reply from everywhere, ensuring that the IP address is not in the stub network. If it is, the router translates the address.

3. Host 1.1.1.1 opens a connection to 3.3.3.3.
4. The router sets up translations mapping inside local and global addresses to each other, and outside global and local addresses to each other.
5. The router replaces the SA with the inside global address and replaces the DA with the outside global address.
6. Host C receives the packet and continues the conversation.
7. The router does a lookup, replaces the DA with the inside local address, and replaces the SA with the outside local address.
8. Host 1.1.1.1 receives the packet and the conversation continues, using this translation process.

Configuring Static Translation of Overlapping Networks

Configure static translation of overlapping networks if your IP addresses in the stub network are legitimate IP addresses belonging to another network and you want to communicate with those hosts or routers using static translation.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat inside source static** *local-ip global-ip*
4. **interface** *type number*
5. **ip address** *ip-address mask*
6. **ip nat inside**
7. **exit**
8. **interface** *type number*
9. **ip address** *ip-address mask*
10. **ip nat outside**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`enable`** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**`configure terminal`** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ip nat inside source static** *local-ip global-ip* **Example:** `Router(config)# ip nat inside source static 192.168.121.33 2.2.2.1`	Establishes static translation between an inside local address and inside global address.
Step 4	**interface** *type number* **Example:** `Router(config)# interface ethernet 1`	Specifies an interface and enters interface configuration mode.
Step 5	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 10.114.11.39 255.255.255.0`	Sets a primary IP address for the interface.
Step 6	**ip nat inside** **Example:** `Router(config-if)# ip nat inside`	Marks the interface as connected to the inside.
Step 7	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to configuration mode.
Step 8	**interface** *type number* **Example:** `Router(config)# interface ethernet 0`	Specifies a different interface and returns to interface configuration mode.
Step 9	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 172.69.232.182 255.255.255.240`	Sets a primary IP address for the interface.
Step 10	**ip nat outside** **Example:** `Router(config-if)# ip nat outside`	Marks the interface as connected to the outside.

What to Do Next

When you have completed all required configuration, go to the “Monitoring and Maintaining NAT” module.

Configuring Dynamic Translation of Overlapping Networks

Configure dynamic translation of overlapping networks if your IP addresses in the stub network are legitimate IP addresses belonging to another network and you want to communicate with those hosts or routers using dynamic translation.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*}
4. **access-list** *access-list-number* **permit** *source* [*source-wildcard*]
5. **ip nat outside source list** *access-list-number* **pool** *name*
6. **interface** *type number*
7. **ip address** *ip-address mask*
8. **ip nat inside**
9. **exit**
10. **interface** *type number*
11. **ip address** *ip-address mask*
12. **ip nat outside**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`configure terminal` **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*} **Example:** `Router(config)# ip nat pool net-10 10.0.1.0 10.0.1.255 prefix-length 24`	Defines a pool of global addresses to be allocated as needed.

	Command or Action	Purpose
Step 4	**access-list** *access-list-number* **permit** *source* [*source-wildcard*] **Example:** `Router(config)# access-list 1 permit 9.114.11.0 0.0.0.255`	Defines a standard access list permitting those addresses that are to be translated. • The access list must permit only those addresses that are to be translated. (Remember that there is an implicit "deny all" at the end of each access list.) An access list that is too permissive can lead to unpredictable results.
Step 5	**ip nat outside source list** *access-list-number* **pool** *name* **Example:** `Router(config)# ip nat outside source list 1 pool net-10`	Establishes dynamic outside source translation, specifying the access list defined in the prior step.
Step 6	**interface** *type number* **Example:** `Router(config)# interface ethernet 1`	Specifies an interface and enters interface configuration mode.
Step 7	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 10.114.11.39 255.255.255.0`	Sets a primary IP address for the interface.
Step 8	**ip nat inside** **Example:** `Router(config-if)# ip nat inside`	Marks the interface as connected to the inside.
Step 9	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to configuration mode.
Step 10	**interface** *type number* **Example:** `Router(config)# interface ethernet 0`	Specifies a different interface and returns to interface configuration mode.
Step 11	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 172.69.232.182 255.255.255.240`	Sets a primary IP address for the interface.
Step 12	**ip nat outside** **Example:** `Router(config-if)# ip nat outside`	Marks the interface as connected to the outside.

Configuring the NAT Virtual Interface

The NAT Virtual Interface (NVI) feature removes the requirement to configure an interface as either Network Address Translation (NAT) inside or NAT outside. An interface can be configured to use NAT or not use NAT.

This section contains the following procedures:

- Restrictions for NAT Virtual Interface, page 22
- Enabling a Static NAT Virtual Interface, page 23

Before you configure the NAT Virtual Interface feature, you should understand the following concepts:

- NAT Virtual Interface Design, page 21
- Benefits of NAT Virtual Interface, page 21

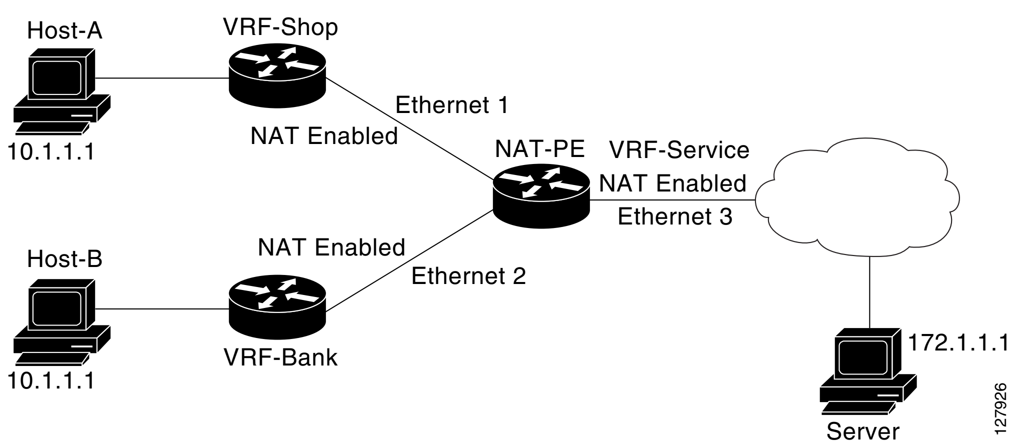
NAT Virtual Interface Design

The NAT Virtual Interface feature allows all NAT traffic flows on the virtual interface, eliminating the need to specify inside and outside domains. When a domain is specified, the translation rules are applied either before or after route decisions depending on the traffic flow from inside to outside or outside to inside. The translation rules are applied only after the route decision for an NVI.

When a NAT pool is shared for translating packets from multiple networks connected to a NAT router, an NVI is created and a static route is configured that forwards all packets addressed to the NAT pool to the NVI. The standard interfaces connected to various networks will be configured to identify that the traffic originating and receiving on the interfaces needs to be translated.

Figure 4 shows a typical NAT virtual interface configuration.

Figure 4 NAT Virtual Interface Typical Configuration



Benefits of NAT Virtual Interface

- A NAT table is maintained per interface for better performance and scalability.
- Domain specific NAT configurations can be eliminated.

Restrictions for NAT Virtual Interface

Routemaps are not supported.

Enabling a Dynamic NAT Virtual Interface

Perform this task to enable a dynamic NAT virtual interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip nat enable**
5. **exit**
6. **ip nat pool** *name start-ip end-ip* **netmask** *netmask* **add-route**
7. **ip nat source list** *access-list-number* **pool** *name* **vrf** *name*
8. **ip nat source list** *access-list-number* **pool** *name* **vrf** *name* **overload**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface FastEthernet 1`	Configures an interface type and enters interface configuration mode.
Step 4	**ip nat enable** **Example:** `Router(config-if)# ip nat enable`	Configures an interface connecting VPNs and the Internet for NAT.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Returns to global configuration mode.
Step 6	**ip nat pool** *name start***-***ip end***-***ip* **netmask** *netmask* **add-route** **Example:** `Router(config)# ip nat pool pool1 200.1.1.1 200.1.1.20 netmask 255.255.255.0 add-route`	Configures a NAT pool and associated mappings.
Step 7	**ip nat source list** *access-list-number* **pool** *number* **vrf** *name* **Example:** `Router(config)# ip nat source list 1 pool 1 vrf shop`	Configures a NAT virtual interface without inside or outside specification for the specified customer.
Step 8	**ip nat source list** *access-list-number* **pool** *number* **vrf** *name* **overload** **Example:** `Router(config)# ip nat source list 1 pool 1 vrf bank overload`	Configures a NAT virtual interface without inside or outside specification for the specified customer.

Enabling a Static NAT Virtual Interface

Perform this task to enable a static NAT virtual interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip nat enable**
5. **exit**
6. **ip nat source static** *local-ip global-ip* **vrf** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface FastEthernet 1`	Configures an interface type and enters interface configuration mode.
Step 4	**ip nat enable** **Example:** `Router(config-if)# ip nat enable`	Configures an interface connecting VPNs and the Internet for NAT.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Returns to global configuration mode.
Step 6	**ip nat source static** *local-ip global-ip* **vrf** *name* **Example:** `Router(config)# ip nat source static 192.168.123.1 192.168.125.10 vrf bank`	Configures a static NVI.

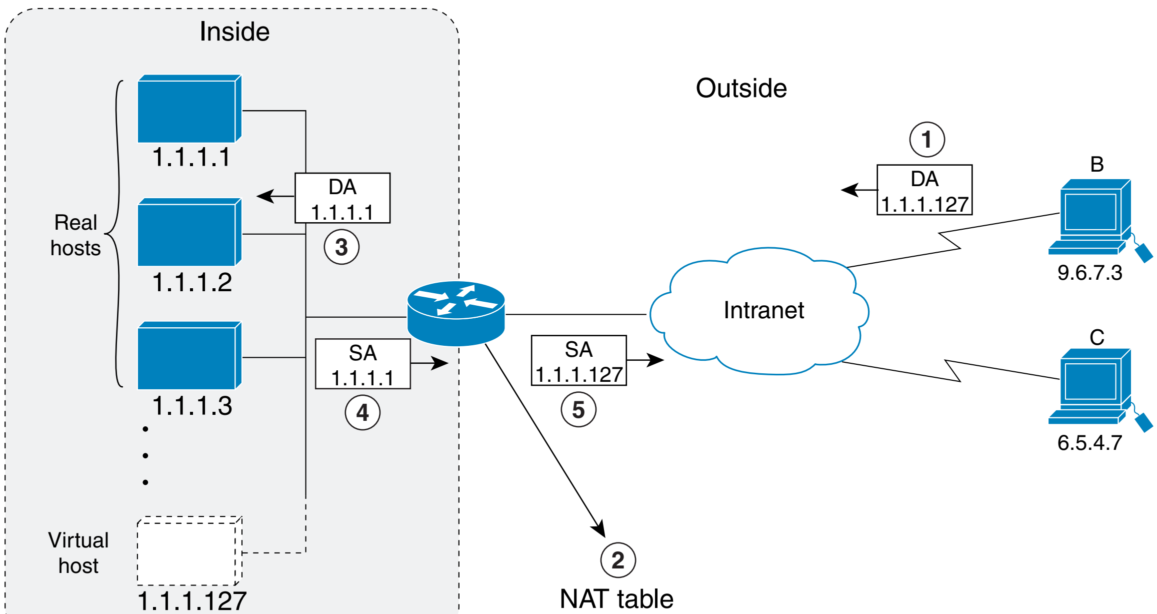
Avoiding Server Overload Using TCP Load Balancing

Perform this task to configure server TCP load balancing by way of destination address rotary translation. These commands allow you to map one virtual host to many real hosts. Each new TCP session opened with the virtual host will be translated into a session with a different real host.

TCP Load Distribution for NAT

Another use of NAT is unrelated to Internet addresses. Your organization may have multiple hosts that must communicate with a heavily used host. Using NAT, you can establish a virtual host on the inside network that coordinates load sharing among real hosts. DAs that match an access list are replaced with addresses from a rotary pool. Allocation is done on a round-robin basis, and only when a new connection is opened from the outside to the inside. Non-TCP traffic is passed untranslated (unless other translations are in effect). Figure 5 illustrates this feature.

Figure 5 NAT TCP Load Distribution



Protocol	Inside Local IP address:port	Inside Global IP address:port	Outside Global IP address:port	Outside Local IP address
TCP TCP TCP	1.1.1.1:23 1.1.1.2:23 1.1.1.3:23	1.1.1.127:23 1.1.1.127:23 1.1.1.127:23	9.6.7.5:3058 6.5.4.7:4371 9.6.7.3:3062	9.6.7.5:3058 6.5.4.7:4371 9.6.7.3:3062

The router performs the following process when translating rotary addresses:

1. The user on host B (9.6.7.3) opens a connection to the virtual host at 1.1.1.127.
2. The router receives the connection request and creates a new translation, allocating the next real host (1.1.1.1) for the inside local IP address.
3. The router replaces the destination address with the selected real host address and forwards the packet.
4. Host 1.1.1.1 receives the packet and responds.
5. The router receives the packet, performs a NAT table lookup using the inside local address and port number, and the outside address and port number as the key. The router then translates the source address to the address of the virtual host and forwards the packet.

The next connection request will cause the router to allocate 1.1.1.2 for the inside local address.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*} **type rotary**
4. **access-list** *access-list-number* **permit** *source* [*source-wildcard*]
5. **ip nat inside destination-list** *access-list-number* **pool** name
6. **interface** *type number*
7. **ip address** *ip-address mask*
8. **ip nat inside**
9. **exit**
10. **interface** *type number*
11. **ip address** *ip-address mask*
12. **ip nat outside**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*} **type rotary** **Example:** `Router(config)# ip nat pool real-hosts 192.168.15.2 192.168.15.15 prefix-length 28 type rotary`	Defines a pool of addresses containing the addresses of the real hosts.
Step 4	**access-list** *access-list-number* **permit** *source* [*source-wildcard*] **Example:** `Router(config)# access-list 1 permit 9.114.11.0 0.0.0.255`	Defines an access list permitting the address of the virtual host.
Step 5	**ip nat inside destination-list** *access-list-number* **pool** *name* **Example:** `Router(config)# ip nat inside destination-list 2 pool real-hosts`	Establishes dynamic inside destination translation, specifying the access list defined in the prior step.

	Command or Action	Purpose
Step 6	**interface** *type number* **Example:** `Router(config)# interface ethernet 0`	Specifies an interface and enters interface configuration mode.
Step 7	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 192.168.15.17 255.255.255.240`	Sets a primary IP address for the interface.
Step 8	**ip nat inside** **Example:** `Router(config-if)# ip nat inside`	Marks the interface as connected to the inside.
Step 9	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to configuration mode.
Step 10	**interface** *type number* **Example:** `Router(config)# interface serial 0`	Specifies a different interface and returns to interface configuration mode.
Step 11	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 192.168.15.129 255.255.255.240`	Sets a primary IP address for the interface.
Step 12	**ip nat outside** **Example:** `Router(config-if)# ip nat outside`	Marks the interface as connected to the outside.

Using Route Maps for Address Translation Decisions

For NAT, a route map to be processed instead of an access list. A route map allows you to match any combination of access-list, next-hop IP address, and output interface to determine which pool to use. The ability to use route maps with static translations enables NAT multihoming capability with static address translations. Multihomed internal networks now can host common services such as the Internet and Domain Name System (DNS), which are accessed from different outside networks.

Benefits of Using Route Maps For Address Translation

- The ability to configure route map statements provides the option of using IP Security (IPSec) with NAT.
- Translation decisions can be made based on the destination IP address when static translation entries are used.

Prerequisites

All route maps required for use with this task should be configured prior to beginning the configuration task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat inside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* [**overload**] | **static** *local-ip global-ip* **route-map** *map-name*}
4. **exit**
5. **show ip nat translations** [**verbose**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat inside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* [**overload**] | **static** *local-ip global-ip* **route-map** *map-name*} **Example:** `Router(config)# ip nat inside source static 11.1.1.2 192.68.1.21 route-map isp2`	Enables route mapping with static NAT configured on the NAT inside interface.
Step 4	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.
Step 5	**show ip nat translations** [**verbose**] **Example:** `Router# show ip nat translations`	(Optional) Displays active NAT.

Enabling NAT Routemaps Outside-to-Inside Support

The NAT Routemaps Outside-to-Inside Support feature enables the deployment of a NAT routemap configuration that will allow IP sessions to be initiated from the outside to the inside. Perform this task to enable NAT Routemaps Outside-to-Inside Support.

Routemaps Outside-to-Inside Support Design

An initial session from inside-to-outside is required to trigger a NAT. New translation sessions can then be initiated from outside-to-inside to the inside host that triggered the initial translation.

When routemaps are used to allocate global addresses, the global address can allow return traffic, and the return traffic is allowed only if the return traffic matches the defined routemap in the reverse direction. Current functionality remains unchanged by not creating additional entries to allow the return traffic for a routemap-based dynamic entry unless the **reversible** keyword is used with the **ip nat inside source** command.

Restrictions

- Only IP hosts that are part of the routemap configuration will allow outside sessions.
- Outside-to-Inside support is not available with Port Address Translation (PAT).
- Outside sessions must use an access list.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat pool** *name start-ip end-ip* **netmask** *netmask*
4. **ip nat pool** *name start-ip end-ip* **netmask** *netmask*
5. **ip nat inside source rout-map** *name* **pool** *name* **[reversible]**
6. **ip nat inside source rout-map** *name* **pool** *name* **[reversible]**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat pool** *name start-ip end-ip* **netmask** *netmask* **Example:** `Router# ip nat pool POOL-A 30.1.10.1 30.1.10.126 netmask 255.255.255.128`	Defines a pool of network addresses for NAT.
Step 4	**ip nat pool** *name start-ip end-ip* **netmask** *netmask* **Example:** `Router# ip nat pool POOL-B 30.1.20.1 30.1.20.126 netmask 255.255.255.128`	Defines a pool of network addresses for NAT.
Step 5	**ip nat inside source route-map** *name* **pool** *name* **reversible** **Example:** `Router# ip nat inside source route-map MAP-A pool POOL-A reversible`	Enables outside-to-inside initiated sessions to use routemaps for destination-based NAT.
Step 6	**ip nat inside source route-map** *name* **pool** *name* **reversible** **Example:** `Router# ip nat inside source route-map MAP-B pool POOL-B reversible`	Enables outside-to-inside initiated sessions to use routemaps for destination-based NAT.

Configuring NAT of External IP Addresses Only

When configuring NAT of external IP addresses only, NAT can be configured to ignore all embedded IP addresses for any application and traffic type. Traffic between a host and the outside world flows through the internal network. A router configured for NAT translates the packet to an address that is able to be routed inside the internal network. If the intended destination is the outside world, the packet gets translated back to an external address and sent out.

Benefits of Configuring NAT of External IP Addresses Only

- Supports public and private network architecture with no specific route updates.

- Gives the end client a usable IP address at the starting point. This address will be the address used for IP Security connections and traffic.
- Allows the use of network architecture that requires only the header translation.
- Allows an Enterprise to use the Internet as its enterprise backbone network.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat inside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* [**overload**] | **static network** *local-ip global-ip* **no-payload**}
4. **ip nat inside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* [**overload**] | **static** {**tcp** | **upd**} *local-port global-port* **no-payload**}
5. **ip nat inside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* [**overload**] | **static** [**network**] *local-network-mask global-network-mask* **no-payload**}
6. **ip nat outside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* [**overload**] | **static** *local-ip global-ip* **no-payload**}
7. **ip nat outside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* [**overload**] | **static** {**tcp** | **upd**} *local-port global-port* **no-payload**}
8. **ip nat outside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* [**overload**] | **static** [**network**] *local-network-mask global-network-mask* **no-payload**}
9. **exit**
10. **show ip nat translations** [**verbose**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat inside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* [**overload**] | **static** [**network**] *local-ip global-ip* **no-payload**} **Example:** `Router(config)# ip nat inside source static network 4.1.1.0 192.168.251.0/24 no-payload`	Disables the network packet translation on the inside host router.
Step 4	**ip nat inside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* [**overload**] | **static** {**tcp** | **upd**} *local-port global-port* **no-payload**} **Example:** `Router(config)# ip nat inside source static tcp 10.1.1.1 2000 192.1.1.1 2000 no-payload`	Disables port packet translation on the inside host router.
Step 5	**ip nat inside source** {**list** {*access-list-number* | *access-list-name}* **pool** *pool-name* [**overload**] |**static** [**network**] *local-network-mask global-network-mask* **no-payload**} **Example:** `Router(config)# p nat inside source static 10.1.1.1 192.1.1.1 no-payload`	Disables the packet translation on the inside host router.
Step 6	**ip nat outside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* [**overload**] | **static** *local-ip global-ip* **no-payload**} **Example:** `Router(config)# ip nat outside source static 10.1.1.1 192.1.1.1 no-payload`	Disables packet translation on the outside host router.
Step 7	**ip nat outside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* [**overload**] | **static** {**tcp** | **upd**} *local-port global-port* **no-payload**} **Example:** `Router(config)# ip nat outside source static tcp 10.1.1.1 20000 192.1.1.1 20000 no-payload`	Disables port packet translation on the outside host router.

	Command or Action	Purpose
Step 8	**ip nat outside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* [**overload**] | **static** [**network**] *local-network-mask global-network-mask* **no-payload**} **Example:** `Router(config)# ip nat outside source static network 4.1.1.0 192.168.251.0/24 no-payload`	Disables network packet translation on the outside host router.
Step 9	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.
Step 10	**show ip nat translations** [**verbose**] **Example:** `Router# show ip nat translations`	Displays active NAT.

Configuring NAT for a Default Inside Server

The NAT Default Inside Server feature provides for the need to forward packets from the outside to a specified inside local address. Traffic is redirected that does not match any existing dynamic translations or static port translations, and the packets are not dropped. For online games, outside traffic comes on different User Datagram Ports (UDP).

Dynamic mapping and interface overload can be configured for the PC traffic and also for the gaming device. If a packet is destined for the 806 interface from the outside and there is not a match in the NAT table for the fully extended entry or a match for the static port entry, it will be forwarded to the gaming device using a simple static entry created as a result of the new command line interface (CLI).

Restrictions

- This feature is used for configuring gaming devices with a different IP address than the PC. To avoid unwanted traffic or attacks, access lists should be used.
- For traffic going from the PC to the outside world, it is better that a route map be used so that extended entries are created.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat inside source static** *local-ip* **interface** *type number*
4. ***ip nat inside source static tcp*** *local-ip local-port* ***interface*** *global-port*
5. **exit**
6. **show ip nat translations [verbose]**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat inside source static** *local-ip* **interface** *type number* **Example:** `Router(config)# ip nat inside source static 1.1.1.1 interface Ethernet1/1`	Enables static NAT on the interface.
Step 4	**ip nat inside source static tcp** *local-ip local-port* **interface** *global-port* **Example:** `Router(config)# ip nat inside source static tcp 1.1.1.1 23 interface 23`	(Optional) Enables the use of telnet to the router from the outside.
Step 5	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.
Step 6	**show ip nat translations** [**verbose**] **Example:** `Router# show ip nat translations`	(Optional) Displays active NAT.

Configuring NAT RTSP Support Using NBAR

The Real Time Streaming Protocol (RTSP) is a client-server multimedia presentation control protocol that supports multimedia application delivery. Some of the applications that use RTSP include Windows Media Services (WMS) by Microsoft, QuickTime by Apple Computer, and RealSystem G2 by RealNetworks.

When the RTSP protocol passes through a NAT router, the embedded address and port must be translated in order for the connection to be successful. NAT uses Network Based Application Recognition (NBAR) architecture to parse the payload and translate the embedded information in the RTSP payload.

RTSP is enabled by default. Use the following commands to re-enable RTSP on a NAT router if this configuration has been disabled.

SUMMARY STEPS

- **enable**
- **configure terminal**
- **ip nat service rtsp port** *port-number*
- **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat service rtsp port** *port-number* **Example:** `Router(config)# ip nat service rtsp port 554`	Enables RTSP packets by NAT.
Step 4	**end** **Example:** `Router(config)# end`	Saves the configuration and exits global configuration mode.

Configuring Support for Users with Static IP Addresses

Configuring support for users with static IP addresses enables those users to establish an IP session in a Public Wireless LAN environment.

The NAT Static IP Support feature extends the capabilities of Public Wireless LAN providers to support users configured with a static IP address. By configuring a router to support users with a static IP address, Public Wireless LAN providers extend their services to a greater number of potential users, which can lead to greater user satisfaction and additional revenue.

Users with static IP addresses can use services of the public wireless LAN provider without changing their IP address. NAT entries are created for static IP clients and a routable address is provided.

This section contains the following procedures:

Configuring Static IP Support, page 36

Verifying Static IP Support, page 38

Public Wireless LAN

A Public Wireless LAN provides users of mobile computing devices with wireless connections to a public network, such as the Internet.

RADIUS

Remote Authentication Dial-In User Service (RADIUS) is a distributed client/server system that secures networks against unauthorized access. Communication between a network access server (NAS) and a RADIUS server is based on the User Datagram Protocol (UDP). Generally, the RADIUS protocol is considered a connectionless service. Issues related to server availability, retransmission, and timeouts are handled by the RADIUS-enabled devices rather than the transmission protocol.

RADIUS is a client/server protocol. The RADIUS client is typically a NAS, and the RADIUS server is usually a daemon process running on a UNIX or Windows NT machine. The client passes user information to designated RADIUS servers and acts on the response that is returned. RADIUS servers receive user connection requests, authenticate the user, and then return the configuration information necessary for the client to deliver service to the user. A RADIUS server can act as a proxy client to other RADIUS servers or other kinds of authentication servers.

Prerequisites

Before configuring support for users with static IP addresses for NAT, you must first enable NAT on your router and configure a RADIUS server host. For additional information on NAT and RADIUS configuration, see the "Related Documents" section on page 48.

Configuring Static IP Support

Perform this task to configure the NAT Static IP Support feature.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip nat inside**
5. **exit**
6. **ip nat allow-static-host**
7. **ip nat pool** *name start-ip end-ip* **netmask** *netmask* **accounting** *list-name*
8. **ip nat inside source list** *access-list-number* **pool** *name*
9. **access-list** *access-list-number* **deny ip** *source*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface ethernet 1`	Specifies the interface to be configured, and enters interface configuration mode.
Step 4	`ip nat inside` **Example:** `Router(config-if)# ip nat inside`	Marks the interface as connected to the inside.
Step 5	`exit` **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 6	**ip nat allow-static-host** **Example:** `Router(config)# ip nat allow-static-host`	Enables static IP address support. • Dynamic Address Resolution Protocol (ARP) learning will be disabled on this interface, and NAT will control the creation and deletion of ARP entries for the static-IP host.
Step 7	ip nat pool *name start-ip end-ip* netmask *netmask* accounting *list-name* **Example:** `Router(config)# ip nat pool xyz 171.1.1.1 171.1.1.10 netmask 255.255.255.0 accounting WLAN-ACCT`	Specifies an existing RADUIS profile name to be used for authentication of the static IP host.
Step 8	**ip nat inside source list** *access-list-number* **pool** *name* **Example:** `Router(config)# ip nat inside source list 1 pool net-208`	Specifies the access list and pool to be used for static IP support. • The specified access list must permit all traffic.
Step 9	**access-list** *access-list-number* **deny ip** *source* **Example:** `Router(config)# access-list 1 deny ip 192.168.196.51`	Removes the router's own traffic from NAT. • The *source* argument is the IP address of the router that supports the NAT Static IP Support feature.

Verifying Static IP Support

To verify the NAT Static IP Support feature, use the following command.

SUMMARY STEPS

1. **show ip nat translations verbose**

DETAILED STEPS

Step 1 **show ip nat translations verbose**

Use this command to verify that NAT is configured to support static IP addresses, for example:

```
Router# show ip nat translations verbose

--- 171.1.1.11         10.1.1.1           ---               ---
create 00:05:59, use 00:03:39, left 23:56:20, Map-Id(In): 1, flags: none wlan-flags:
Secure ARP added, Accounting Start sent Mac-Address:0010.7bc2.9ff6 Input-IDB:Ethernet1/2,
use_count: 0, entry-id:7, lc_entries: 0
```

Configuring Support for ARP Ping in a Public Wireless LAN

When the static IP client's NAT entry times out, the NAT entry and the secure ARP entry associations are deleted for the client. Reauthentication with the Service Selection Gateway (SSG) is needed for the client to reestablish WLAN services. The ARP Ping feature enables the NAT entry and the secure ARP entry to not be deleted when the static IP client exists in the network where the IP address is unchanged after authentication.

An ARP ping is necessary to determine static IP client existence and to restart the NAT entry timer.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat pool** *name start-ip end-ip* **prefix-length** *prefix-length* **[accounting]** *method-list-name* **[arp-ping]**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`configure terminal` **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ip nat pool** *name start-ip end-ip* **prefix-length [accounting]** *method-list-name* **[arp-ping]** **Example:** `Router(config)# ip nat pool net-208 171.69.233.208 171.69.233.223 prefix-length 28 accounting radius1 arp-ping`	Defines a pool of IP addresses for NAT.
Step 4	ip nat translation arp-ping-timeout [*timeout-value*] **Example:** `Router(config)# ip nat translation arp-ping-timeout 600`	Changes the amount of time after each network address translation.

Limiting the Number of Concurrent NAT Operations

Limiting the number of concurrent NAT operations using the Rate Limiting NAT Translation feature provides users more control over how NAT addresses are used. The Rate Limiting NAT Translation feature can be used to limit the effects of viruses, worms, and denial-of-service attacks.

Benefits of Limiting the Number of concurrent NAT Operations

Since NAT is a CPU-intensive process, router performance can be adversely affected by denial-of-service attacks, viruses, and worms that target NAT. The Rate Limiting NAT Translation feature allows you to limit the maximum number of concurrent NAT requests on a router.

Denial-of-Service Attacks

A denial-of-service (DoS) attack typically involves the misuse of standard protocols or connection processes with the intent to overload and disable a target, such as a router or web server. DoS attacks can come from a malicious user or from a computer infected with a virus or worm. When the attack comes from many different sources at once, such as when a virus or worm has infected many computers, it is known as a distributed denial-of-service (DDoS) attack. Such DDoS attacks can spread rapidly and involve thousands of systems.

Viruses and Worms That Target NAT

Viruses and worms are malicious programs designed to attack computer and networking equipment. While viruses are typically embedded in discrete applications and only run when executed, worms self-propagate and can quickly spread on their own. Although a specific virus or worm may not expressly target NAT, it might use NAT resources to propagate itself. The Rate Limiting NAT Translation feature can be used to limit the impact of viruses and worms that originate from specific hosts, access control lists, and VPN routing and forwarding (VRF) instances.

Prerequisites

- Classify current NAT usage and determine the sources of requests for NAT. If a specific host, access control list, or VRF instance is generating an unexpectedly high number of NAT requests, it may be the source of a malicious virus or worm attack.
- Once you have identified the source of excess NAT requests, you can set a NAT rate limit that contains a specific host, access control list, or VRF instance, or you can set a general limit for the maximum number of NAT requests allowed regardless of their source.

SUMMARY STEPS

1. **enable**
2. **show ip nat translations**
3. **configure terminal**
4. **ip nat translation max-entries** {*number* | **all-vrf** *number* | **host** *ip-address number* | **list** *listname number* | **vrf** *name number*}
5. **end**
6. **show ip nat statistics**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`show ip nat translations` **Example:** `Router# show ip nat translations`	(Optional) Displays active NAT. • If a specific host, access control list, or VRF instance is generating an unexpectedly high number of NAT requests, it may be the source of a malicious virus or worm attack.
Step 3	`configure terminal` **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 4	**ip nat translation max-entries** {*number* | **all-vrf** *number* | **host** *ip-address number* | **list** *listname number* | **vrf** *name number*} **Example:** `Router(config)# ip nat translation max-entries 300`	Configures the maximum number of NAT entries allowed from the specified source. • The maximum number of allowed NAT entries is 2147483647, although a typical range for a NAT rate limit is 100 to 300 entries. • When configuring a NAT rate limit for all VRF instances, each VRF instance is limited to the maximum number of NAT entries that you specify. • When configuring a NAT rate limit for a specific VRF instance, you can specify a maximum number of NAT entries for the named VRF instance that is greater than or less than that allowed for all VRF instances.
Step 5	`end` **Example:** `Router(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.
Step 6	`show ip nat statistics` **Example:** `Router# show ip nat statistics`	(Optional) Displays current NAT usage information, including NAT rate limit settings. • After setting a NAT rate limit, use the **show ip nat statistics** command to verify current NAT rate limit settings.

Configuration Examples for Configuring NAT for IP Address Conservation

This section provides the following configuration examples:

Configuring Static Translation of Inside Source Addresses: Examples

The following example translates between inside hosts addressed from the 9.114.11.0 network to the globally unique 171.69.233.208/28 network. Further packets from outside hosts addressed from the 9.114.11.0 network (the true 9.114.11.0 network) are translated to appear to be from the 10.0.1.0/24 network.

```
ip nat pool net-208 171.69.233.208 171.69.233.223 prefix-length 28
ip nat pool net-10 10.0.1.0 10.0.1.255 prefix-length 24
ip nat inside source list 1 pool net-208
ip nat outside source list 1 pool net-10
!
interface ethernet 0
 ip address 171.69.232.182 255.255.255.240
 ip nat outside
!
interface ethernet 1
 ip address 9.114.11.39 255.255.255.0
 ip nat inside
!
access-list 1 permit 9.114.11.0 0.0.0.255
```

The following example shows NAT configured on the Provider Edge (PE) router with a static route to the shared service for the gold and silver Virtual Private Networks (VPNs). NAT is configured as inside source static one-to-one translations.

```
ip nat pool outside 4.4.4.1 4.4.4.254 netmask 255.255.255.0
ip nat outside source list 1 pool mypool
access-list 1 permit 168.58.18.0 0.0.0.255
ip nat inside source static 192.168.121.33 2.2.2.1 vrf gold
ip nat inside source static 192.169.121.33.2.2.2.2 vrf silver
```

Configuring Dynamic Translation of Inside Source Addresses: Example

The following example translates between inside hosts addressed from either the 192.168.1.0 or 192.168.2.0 network to the globally unique 171.69.233.208/28 network:

```
ip nat pool net-208 171.69.233.208 171.69.233.223 prefix-length 28
ip nat inside source list 1 pool net-208
!
interface ethernet 0
 ip address 171.69.232.182 255.255.255.240
 ip nat outside
!
interface ethernet 1
 ip address 192.168.1.94 255.255.255.0
 ip nat inside
!
access-list 1 permit 192.168.1.0 0.0.0.255
access-list 1 permit 192.168.2.0 0.0.0.255
```

The following example translates only traffic local to the provider edge device running NAT (NAT-PE):

```
ip nat inside source list 1 interface e 0 vrf shop overload
ip nat inside source list 1 interface e 0 vrf bank overload
!
ip route vrf shop 0.0.0.0 0.0.0.0 192.1.1.1
ip route vrf bank 0.0.0.0 0.0.0.0 192.1.1.1
!
access-list 1 permit 10.1.1.1.0 0.0.0.255
!
```

```
ip nat inside source list 1 interface e 1 vrf shop overload
ip nat inside source list 1 interface e 1 vrf bank overload
!
ip route vrf shop 0.0.0.0 0.0.0.0 172.1.1.1 global
ip route vrf bank 0.0.0.0 0.0.0.0 172.1.1.1 global
access-list 1 permit 10.1.1.0 0.0.0.255
```

Overloading Inside Global Addresses: Example

The following example creates a pool of addresses named net-208. The pool contains addresses from 171.69.233.208 to 171.69.233.233. Access list 1 allows packets having the SA from 192.168.1.0 to 192.168.1.255. If no translation exists, packets matching access list 1 are translated to an address from the pool. The router allows multiple local addresses (192.168.1.0 to 192.168.1.255) to use the same global address. The router retains port numbers to differentiate the connections.

```
ip nat pool net-208 171.69.233.208 171.69.233.233 netmask 255.255.255.240
ip nat inside source list 1 pool net-208 overload
!
interface serial0
 ip address 171.69.232.182 255.255.255.240
 ip nat outside
!
interface ethernet0
 ip address 192.168.1.94 255.255.255.0
 ip nat inside
!
access-list 1 permit 192.168.1.0 0.0.0.255
```

Translating Overlapping Address: Example

In the following example, the addresses in the local network are being used legitimately by someone else on the Internet. An extra translation is required to access that external network. Pool net-10 is a pool of outside local IP addresses. The **ip nat outside source list 1 pool net-10** statement translates the addresses of hosts from the outside overlapping network to addresses in that pool.

```
ip nat pool net-208 171.69.233.208 171.69.233.223 prefix-length 28
ip nat pool net-10 10.0.1.0 10.0.1.255 prefix-length 24
ip nat inside source list 1 pool net-208
ip nat outside source list 1 pool net-10
!
interface serial 0
 ip address 171.69.232.192 255.255.255.240
 ip nat outside
!
interface ethernet0
 ip address 192.168.1.94 255.255.255.0
 ip nat inside
!
access-list 1 permit 192.168.1.0 0.0.0.255
```

Enabling NAT Virtual Interface: Example

The following example shows how to configure NAT virtual interfaces without the use of inside or outside source addresses:

```
interface Ethernet0/0
 ip vrf forwarding bank
```

```
 ip address 192.168.122.1 255.255.255.0
 ip nat enable
!
interface Ethernet1/0
 ip vrf forwarding park
 ip address 192.168.122.1 255.255.255.0
 ip nat enable
!
interface Serial2/0
 ip vrf forwarding services
 ip address 192.168.123.2 255.255.255.0
ip nat enable
!
ip nat pool NAT 192.168.25.20 192.168.25.30 netmask 255.255.255.0 add-route
ip nat source list 1 pool NAT vrf bank overload
ip nat source list 1 pool NAT vrf park overload
ip nat source static 192.168.123.1 192.168.125.10 vrf services
!
access-list 1 permit 192.168.122.20
access-list 1 permit 192.168.122.0 0.0.0.255
!
```

Avoiding Server Overload Using Load Balancing: Example

In the following example, the goal is to define a virtual address, connections to which are distributed among a set of real hosts. The pool defines the addresses of the real hosts. The access list defines the virtual address. If a translation does not already exist, TCP packets from serial interface 0 (the outside interface) whose destination matches the access list are translated to an address from the pool.

```
ip nat pool real-hosts 192.168.15.2 192.168.15.15 prefix-length 28 type rotary
ip nat inside destination list 2 pool real-hosts
!
interface serial 0
 ip address 192.168.15.129 255.255.255.240
 ip nat outside
!
interface ethernet 0
 ip address 192.168.15.17 255.255.255.240
 ip nat inside
!
access-list 2 permit 192.168.15.1
```

Enabling NAT Route Mapping: Example

The following example shows the use of route mapping with static NATs:

```
interface Ethernet3
ip address 172.68.1.100 255.255.255.0
ip nat outside
media-type 10BaseT
!
interface Ethernet4
ip address 192.68.1.100 255.255.255.0
ip nat outside
media-type 10BaseT
!
interface Ethernet5
ip address 11.1.1.100 255.255.255.0
```

```
ip nat inside
media-type 10BaseT
!
router rip
network 172.68.0.0
network 192.68.1.0
!
ip nat inside source static 11.1.1.2 192.68.1.21 route-map isp2
ip nat inside source static 11.1.1.2 172.68.1.21 route-map isp1
ip nat inside source static 11.1.1.1 192.68.1.11 route-map isp2
ip nat inside source static 11.1.1.1 172.68.1.11 route-map isp1
!
access-list 101 permit ip 11.1.1.0 0.0.0.255 172.0.0.0 0.255.255.255.
access-list 102 permit ip 11.1.1.0 0.0.0.255 192.0.0.0 0.255.255.255
!
route-map isp2 permit 10
match ip address 102
set ip next-hop 192.68.1.1
!
route-map isp1 permit 10
match ip address 101
set ip next-hop 172.68.1.1
```

Enabling NAT Routemaps Outside-to-Inside Support: Example

The following example shows how to configure routemap A and routemap B to allow outside-to-inside translation for a destination-based NAT.

```
ip nat pool POOL-A 30.1.10.1 30.1.10.126 netmask 255.255.255.128
ip nat pool POOL-B 30.1.20.1 30.1.20.126 netmask 255.255.255.128
ip nat inside source route-map MAP-A pool POOL-A reversible
ip nat inside source route-map MAP-B pool POOL-B reversible
!
ip access-list extended ACL-A
 permit ip any 30.1.10.128 0.0.0.127
ip access-list extended ACL-B
 permit ip any 30.1.20.128 0.0.0.127
!
route-map MAP-A permit 10
 match ip address ACL-A
!
route-map MAP-B permit 10
 match ip address ACL-B
```

Configuring NAT Translation of External IP Addresses Only: Example

The following example shows how to translate the packet to an address that is able to be routed inside the internal network:

```
interface ethernet 3
ip address 20.1.1.1 255.255.255.0
ip nat outside
no ip mroute-cache
media-type 10BaseT
!
interface Ethernet4
ip address 192.168.15.1 255.255.255.0
ip nat inside
no ip mroute-cache
media-type 10BaseT
```

```
!
router rip
network 20.0.0.0
Network 192.168.15.0
!
ip nat outside source static network 4.1.1.0 192.168.251.0/24 no-payload
!
ip route 2.1.1.0 255.255.255.0 Ethernet4
ip route 4.1.1.0 255.255.255.0 Ethernet3
```

Configuration Examples for NAT Static IP Support

This section provides the following configuration examples:

- Configuring NAT Static IP Support: Example, page 46
- Creating a RADIUS Profile for NAT Static IP Support: Example, page 46

Configuring NAT Static IP Support: Example

The following example shows how to enable static IP address support for the router at 192.168.196.51:

```
interface ethernet 1
 ip nat inside
ip nat allow-static-host
ip nat pool xyz 171.1.1.1 171.1.1.10 netmask 255.255.255.0 accounting WLAN-ACCT
ip nat inside source list 1 pool net-208
access-list 1 deny ip 192.168.196.51
```

Creating a RADIUS Profile for NAT Static IP Support: Example

The following example shows how to create a RADIUS profile for use with the NAT Static IP Support feature:

```
aaa new-model
!
aaa group server radius WLAN-RADIUS
 server 168.58.88.1 auth-port 1645 acct-port 1645
 server 168.58.88.1 auth-port 1645 acct-port 1646
!
aaa accounting network WLAN-ACCT start-stop group WLAN-RADIUS
aaa session-id common
ip radius source-interface Ethernet3/0
radius-server host 168.58.88.1 auth-port 1645 acct-port 1646
radius-server key cisco
```

Configuration Examples for Rate Limiting NAT Translation

This section provides the following configuration examples:

- Setting a Global NAT Rate Limit: Example, page 47
- Setting NAT Rate Limits for a Specific VRF Instance: Example, page 47
- Setting NAT Rate Limits for All VRF Instances: Example, page 47
- Setting NAT Rate Limits for Access Control Lists: Example, page 47

-

Setting a Global NAT Rate Limit: Example

The following example shows how to limit the maximum number of allowed NAT entries to 300:

```
ip nat translation max-entries 300
```

Setting NAT Rate Limits for a Specific VRF Instance: Example

The following example shows how to limit the VRF instance named "vrf1" to 150 NAT entries:

```
ip nat translation max-entries vrf vrf1 150
```

Setting NAT Rate Limits for All VRF Instances: Example

The following example shows how to limit each VRF instance to 200 NAT entries:

```
ip nat translation max-entries all-vrf 200
```

The following example shows how to limit the VRF instance named "vrf2" to 225 NAT entries, but limit all other VRF instances to 100 NAT entries each:

```
ip nat translation max-entries all-vrf 100
ip nat translation max-entries vrf vrf2 225
```

Setting NAT Rate Limits for Access Control Lists: Example

The following example shows how to limit the access control list named "vrf3" to 100 NAT entries:

```
ip nat translation max-entries list vrf3 100
```

Setting NAT Rate Limits for an IP Address: Example

The following example shows how to limit the host at IP address 127.0.0.1 to 300 NAT entries:

```
ip nat translation max-entries host 127.0.0.1 300
```

Where to Go Next

- To configure NAT for use with application level gateways, see the "Using Application Level Gateways with NAT" module.
- To verify, monitor, and maintain NAT, see the "Monitoring and Maintaining NAT" module.
- To integrate NAT with MPLS VPNs, see the "Integrating NAT with MPLS VPNs" module
- To configure NAT for high availability, see the "Configuring NAT for High Availability" module.

Additional References

The following sections provide references related to Configuring NAT for IP Address Conservation.

Related Documents

Related Topic	Document Title
Using NAT with MPLS VPNs	"Integrating NAT with MPLS VPNs" module
Using HSRP and SNAT for high availability	"Configuring NAT for High Availability" module
NAT maintenance	"Monitoring and Maintaining NAT" module
NAT commands: complete command syntax, command mode command history, defaults, usage guidelines, and examples	*Cisco IOS IP Addressing Services Command Reference*

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 1597	Internet Assigned Numbers Authority
RFC 1631	The IP Network Address Translation (NAT)
RFC 1918	Address Allocation for Private Internets
RFC 2663	IP Network Address Translation (NAT) Terminology and Considerations
RFC 3022	Traditional IP Network Address Translation (Traditional NAT)

Technical Assistance

Description	Link
The Cisco Technical Support & Documentation website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport

Feature Information for Configuring NAT for IP Address Conservation

Table 1 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.2(4)T, 12.2(4)2T, 12.3(13)T or later appear in the table.

Not all commands may be available in your Cisco IOS software release. For details on when support for specific commands was introduced, see the command reference documents.

If you are looking for information on a feature in this technology that is not documented here, see the "Configuring Network Address Translation Features Roadmap."

Cisco IOS software images are specific to a Cisco IOS software release, a feature set, and a platform. Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for Configuring NAT for IP Address Conservation

Feature Name	Releases	Feature Configuration Information
NAT Ability to Use Route Maps with Static Translation	12.2.(4)T	This feature provides a dynamic translation command that can specify a route map to be processed instead of an access-list. A route map allows you to match any combination of access-list, next-hop IP address, and output interface to determine which pool to use. The ability to use route maps with static translations enables NAT multihoming capability with static address translations. The following section provides information about this feature: • "Using Route Maps for Address Translation Decisions" section on page 27
NAT Default Inside Server	12.3(13)T	The NAT Default Inside Server feature provides for the need to forward packets from the outside to a specified inside local address. The following section provides information about this feature: • "Configuring NAT for a Default Inside Server" section on page 33
NAT Routemaps Outside-to-Inside Support	12.3(14)T	The NAT Routemaps Outside-to-Inside Support feature enables the deployment of a NAT routemap configuration that will allow IP sessions to be initiated from the outside to the inside. The following sections provide information about this feature: • "Enabling NAT Routemaps Outside-to-Inside Support" section on page 28 • "Enabling NAT Routemaps Outside-to-Inside Support: Example" section on page 45
NAT RTSP Support Using NBAR	12.3(7)T	The Real Time Streaming Protocol (RTSP) is a client-server multimedia presentation control protocol that supports multimedia application delivery. Some of the applications that use RTSP include Windows Media Services (WMS) by Microsoft, QuickTime by Apple Computer, and RealSystem G2 by RealNetworks. The following section provides information about this feature: • "Configuring NAT RTSP Support Using NBAR" section on page 34

Table 1 Feature Information for Configuring NAT for IP Address Conservation

Feature Name	Releases	Feature Configuration Information
NAT Static IP Support	12.3(7)T Cisco IOS XE Release 2.1	The NAT Static IP Support feature provides support for users with static IP addresses, enabling those users to establish an IP session in a Public Wireless LAN environment. The following sections provide information about this feature: • "Configuring Support for Users with Static IP Addresses" section on page 35 • "Configuration Examples for NAT Static IP Support" section on page 46
NAT Translation of External IP addresses only	12.2(4)T 12.2(4)T2 Cisco IOS XE Release 2.1	Using the NAT of external IP address only feature, NAT can be configured to ignore all embedded IP addresses for any application and traffic type. The following sections provide information about this feature: • "Configuring NAT of External IP Addresses Only" section on page 30 • "Configuring NAT of External IP Addresses Only" section on page 30
NAT Virtual Interface (NVI)	12.3(14)T	The NAT Virtual Interface (NVI) feature removes the requirement to configure an interface as either Network Address Translation (NAT) inside or NAT outside. An interface can be configured to use NAT or not use NAT. The following sections provide information about this feature: • Configuring the NAT Virtual Interface, page 21 • "Enabling NAT Virtual Interface: Example" section on page 43

Table 1 Feature Information for Configuring NAT for IP Address Conservation

Feature Name	Releases	Feature Configuration Information
Rate Limiting NAT Translation feature	12.3(4)T Cisco IOS XE Release 2.1	The Rate Limiting NAT Translation feature provides the ability to limit the maximum number of concurrent Network Address Translation (NAT) operations on a router. In addition to giving users more control over how NAT addresses are used, the Rate Limiting NAT Translation feature can be used to limit the effects of viruses, worms, and denial-of-service attacks. The following sections provide information about this feature: • "Limiting the Number of Concurrent NAT Operations" section on page 39 • "Configuration Examples for Rate Limiting NAT Translation" section on page 46
Configuring Support for ARP Ping in a Public Wireless LAN	12.4(6)T	The ARP Ping feature enables the NAT entry and the secure ARP entry to not be deleted when the static IP client exists in the network where the IP address is unchanged after authentication. The following section provides information about this feature: • "Configuring Support for ARP Ping in a Public Wireless LAN" section on page 38

Using Application Level Gateways with NAT

Network Address Translation (NAT) performs translation service on any Transmission Control Protocol/User Datagram Protocol (TCP/UDP) traffic that does not carry source and/or destination IP addresses in the application data stream. These protocols include HTTP, Trivial File Transfer Protocol (TFTP), telnet, archie, finger, Network TimeProtocol (NTP), Network File System (NFS), remote login (rlogin), remote shell protocol (rsh), and remote copy protocol (rcp). Specific protocols that do imbed IP address information within the payload require support of an Application Level Gateway (ALG).

The support for IPSec ESP Through NAT feature provides the ability to support multiple concurrent IP Security (IPSec) Encapsulating Security Payload (ESP) tunnels or connections through a Cisco IOS NAT device configured in Overload or Port Address Translation (PAT) mode.

Module History

This module was first published on May 2, 2005, and last updated on May 2, 2008.

Finding Feature Information in This Module

To find information about feature support and configuration, use the "Feature Information for Using Application Level Gateways with NAT" section on page 12.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Using Application Level Gateways with NAT

- Before performing the tasks in this module, you should be familiar with the concepts described in the Configuring NAT for IP Address Conservation module.
- All access lists required for use with the tasks in this module should be configured prior to beginning the configuration task. For information about how to configure an access list, see the "*IP Access List Sequence Numbering*" document at the following URL:

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios122s/122snwft/release/122s14/fsaclseq.htm
- Before performing the tasks in this module, you should verify that Session Initiation Protocol (SIP) and H.323 have not been disabled. SIP and H.323 are enabled by default.

Information About Configuring Application Level Gateways with NAT

To configure ALGs with NAT, you should understand the following concept:

- Application Level Gateway, page 2

Application Level Gateway

An application level gateway is an application that translates IP address information inside the payload of an applications packet.

How to Configure Application Level Gateways with NAT

This section contains the following procedures:

- Configuring IPSec Through NAT, page 2
- Deploying NAT Between an IP Phone and Cisco CallManager, page 9

Configuring IPSec Through NAT

This section contains the following tasks related to configuring IPSec through NAT:

- Configuring IPSec ESP Through NAT, page 5 (required)
- Enabling Preserve Port, page 7 (optional)
- Disabling SPI Matching on the NAT Device or Changing the Default Port, page 7 (required)
- Enabling SPI Matching on the Endpoints, page 8 (required)

Benefits of Configuring NAT IPSec

- NAT support for SIP adds the ability to deploy Cisco IOS NAT between VoIP solutions based on SIP.
- Customers can control their IP address scheme and include complete support for H.323 v2 gatekeeper designs.
- NAT enables customers to deploy private IP addresses within their network and perform translation to public IP addresses when connecting to the Internet or interconnecting with another corporate network.
- Normally ESP entries in the translation table are delayed from being transmitted until a reply is received from the destination. With predictable security parameter indexes (SPIs) and SPI matching, the delay can be eliminated since the SPI entries are matched. Some third-party concentrators require both the source and incoming ports to use port 500. Use of the **preserve-port** keyword with the **ip nat service** command preserves the ports rather than changing one, which is required with regular NAT.

IP Security

IP Security (IPSec) is a set of extensions to the IP protocol family in a framework of open standards for ensuring secure private communications over the Internet. Based on standards developed by the Internet Engineering Task Force (IETF), IPSec ensures confidentiality, integrity, and authenticity of data communications across the public network and provides cryptographic security services.

Secure tunnels between two peers, such as two routers, are provided and decisions are made as to which packets are considered sensitive and should be sent through these secure tunnels, and which parameters should be used to protect these sensitive packets by specifying characteristics of these tunnels. When the IPSec peer receives a sensitive packet, it sets up the appropriate secure tunnel and sends the packet through the tunnel to the remote peer.

IPSec using ESP can pass through a router running NAT without any specific support from it as long as Network Address Port Translation (NAPT) or address overloading are not configured.

There are a number of factors to consider when attempting an IPSec Virtual Private Network (VPN) connection that traverses a NAPT device that represents multiple private internal IP addresses as a single public external IP address. Such factors include the capabilities of the VPN server and client, the capabilities of the NAPT device, and whether more than one simultaneous connection is attempted across the NAPT device.

There are two possible methods for configuring IPSec on a router with NAPT:

- Encapsulate IPSec in a Layer 4 protocol such as TCP or UDP. In this case, IPSec is *sneaking* through NAT. The NAT device is unaware of the encapsulation.
- Add IPSec specific support to NAPT. IPSec works with NAT in this case as opposed to *sneaking* through NAT. The NAT Support for IPSec ESP— Phase II feature provides support for Internet Key Exchange (IKE) and ESP without encapsulation in tunnel mode through a Cisco IOS router configured with NAPT.

The recommended protocols to use when conducting IPSec sessions that traverse a NAPT device are TCP and UDP but not all VPN servers or clients support TCP or UDP.

SPI Matching

Security Parameter Index (SPI) matching is used to establish VPN connections between multiple pairs of destinations. NAT entries will immediately be placed in the translation table for endpoints matching the configured access list. SPI matching is available only for endpoints that choose SPIs according to the predictive algorithm implemented in Cisco IOS Release 12.2(15)T.

Voice and Multimedia over IP Networks

SIP is a protocol developed by the Internet Engineering Task Force (IETF) Multiparty Multimedia Session Control (MMUSIC) Working Group. The Cisco SIP functionality equips Cisco routers to signal the setup of voice and multimedia calls over IP networks. SIP provides an alternative to H.323 within the Voice over IP (VoIP) internetworking software.

Session Description Protocol (SDP) is a protocol that describes multimedia sessions. SDP may be used in SIP message bodies to describe multimedia sessions used for creating and controlling multimedia sessions with two or more participants.

The NAT Support for SIP feature allows SIP embedded messages passing through a router configured with NAT to be translated and encoded back to the packet. An ALG is used with NAT to translate the SIP or SDP messages.

NAT Support of H.323 v2 RAS

Cisco IOS NAT supports all H.225 and H.245 message types, including those sent in the Registration, Admission, and Status (RAS) protocol. RAS provides a number of messages that are used by software clients and Voice over IP (VoIP) devices to register their location, request assistance in call setup, and control bandwidth. The RAS messages are directed toward an H.323 gatekeeper.

Some RAS messages include IP addressing information in the payload, typically meant to register a user with the gatekeeper or learn about another user already registered. If these messages are not known to NAT, they cannot be translated to an IP address that will be visible to the public.

Previously, NAT did not support H.323 v2 RAS messages. With this enhancement, embedded IP addresses can be inspected for potential address translation.

NAT Support for H.323 v3 and v4 in v2 Compatibility Mode

H.323 is an ITU-T specification for transmitting audio, video, and data across a packet network. Four versions of the H.323 protocols are currently in use: v1, v2, v3, and v4. The NAT Support for H.323 v3 and v4 in v2 Compatibility Mode feature enables Cisco NAT routers to support messages coded in H.323 v3 and v4 when those messages contain fields compatible with H.323 v2. This feature does not add support for H.323 capabilities introduced in v3 and v4, such as new message types or new fields that require address translation.

NAT H.245 Tunneling Support

NAT H.245 tunneling allows H.245 tunneling in H.323 ALGs. NAT H.245 tunneling provides a mechanism for supporting H.245 tunnel message which are needed to create a media channel setup.

In order for an H.323 call to take place, an H.225 connection on TCP port 1720 needs to be opened. When the H.225 connection is opened, the H.245 session is initiated and established. This connection can take place on a separate channel from the H.225 or it can be done using H.245 tunneling on the same H.225 channel whereby the H.245 messages are embedded in the H.225 messages and sent on the previously established H.225 channel.

If the H.245 tunneled message is not understood, the media address or port is going to be left untranslated by the Cisco IOS NAT resulting in failure in media traffic. H.245 FastConnect procedures will not help because FastConnect is terminated as soon as an H.245 tunneled message is sent.

Restrictions

- NAT will translate only embedded IP version 4 addresses.

Configuring IPSec ESP Through NAT

IPSec ESP Through NAT provides the ability to support multiple concurrent IPSec ESP tunnels or connections through a Cisco IOS NAT device configured in Overload or PAT mode.

Perform this task to configure IPSec ESP through NAT.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat** [**inside** | **outside**] **source static** *local-ip global-ip*
4. **exit**
5. **show ip nat translations**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat** [**inside** | **outside**] **source static** *local-ip global-ip* **Example:** `Router(config)# ip nat inside source static 10.10.10.10 172.16.30.30`	Enables static NAT.

	Command or Action	Purpose
Step 4	**exit** **Example:** `Router(config)# exit`	Returns to privileged EXEC mode.
Step 5	**show ip nat translations** **Example:** `Router# show ip nat translations`	(Optional) Displays active NATs.

Enabling Preserve Port

This task is used for IPSec traffic using port 500 for the source and incoming ports. Perform this task to enable port 500 to be preserved for both source and incoming ports.

Restrictions

This task is required by certain VPN concentrators but will cause problems with other concentrators. Cisco VPN devices generally do not use this feature.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat service list** *access-list-number* **ike preserve-port**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`enable`** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**`configure terminal`** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**`ip nat service list`** *`access-list-number`* **`ike preserve-port`** **Example:** `Router(config)# ip nat service list 10 ike preserve-port`	Specifies a port other than the default port.

Disabling SPI Matching on the NAT Device or Changing the Default Port

Security parameter index (SPI) matching is used to establish VPN connections between multiple pairs of destinations. NAT entries are immediately placed in the translation table for endpoints matching the configured access list. SPI Matching is available only for endpoints that choose SPIs according to the predictive algorithm implemented in Cisco IOS Release 12.2(15)T.

The generation of SPIs that are predictable and symmetric is enabled. SPI Matching should be used in conjunction with NAT devices when multiple ESP connections across a NAT device are desired.

SPI Matching is enabled by default for listening on port 2000. This task may be used to either change the default port or disable SPI matching.

Prerequisites

Cisco IOS software must be running on both the source router and the remote gateway enabling parallel processing.

Restrictions

SPI matching must be configured on the NAT device and both endpoint devices.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat service list** *access-list-number* **esp spi-match**
4. no ip nat service list *access-list-number* esp spi-match

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat service list** *access-list-number* **esp spi-match** **Example:** `Router(config)# ip nat service list 10 esp spi-match`	Specifies a port other than the default port. • This example shows how to enter ESP traffic matching list 10 into the NAT table, making the assumption that both devices are Cisco devices and are configured to provide matchable SPIs.
Step 4	**no ip nat service list** *access-list-number* **esp spi-match** **Example:** `Router(config)# no ip nat service list 10 esp spi-match`	Disables SPI matching.

Enabling SPI Matching on the Endpoints

Perform this task to enable SPI matching on both endpoints.

Prerequisites

Cisco IOS software must be running on both the source router and the remote gateway enabling parallel processing.

Restrictions

SPI matching must be configure on the NAT device and both endpoint devices.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec spi-matching**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto ipsec spi-matching** **Example:** `Router(config)# crypto ipsec spi-matching`	Enables SPI matching on both endpoints.

Deploying NAT Between an IP Phone and Cisco CallManager

This section describes deploying Cisco's Skinny Client Control Protocol (SCCP) for a Cisco IP phone to Cisco CallManager (CCM) communication. The task in this section deploys NAT between an IP phone and CCM.

NAT Support of Skinny Client Control Protocol

Cisco IP phones use the SCCP to connect with and register to CCM.

To be able to deploy Cisco IOS NAT between the IP phone and CCM in a scalable environment, NAT needs to be able to detect the SCCP and understand the information passed within the messages. Messages flow back and forth that include IP address and port information used to identify other IP phone users with which a call can be placed.

The SCCP client to CCM communication typically flows from inside to outside. DNS should be used to resolve the CCM IP address connection when the CCM is on the inside (behind the NAT device), or static NAT should be configured to reach the CCM in the inside.

When an IP phone attempts to connect to the CCM and it matches the configured NAT rules, NAT will translate the original source IP address and replace it with one from the configured pool. This new address will be reflected in the CCM and be visible to other IP phone users.

NAT Support of SCCP Fragmentation

Skinny control messages are exchanged over TCP. If either the IP phone or CCM has been configured to have TCP maximum segment size (MSS) lower than the skinny control message payload, the skinny control message will be segmented across multiple TCP segments. Prior to this feature skinny control message exchanges would fail in a TCP segmentation scenario because NAT skinny ALG was not able to reassemble the skinny control messages. The NAT SCCP Fragmentation Support feature adds support for TCP segments for NAT skinny ALG. A fragmented payload that requires an IP or port translation will no longer be dropped.

Skinny control messages can also be IP fragmented but they are supported using Virtual Fragmentation Reassembly (VFR).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat service skinny tcp port** *number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat service skinny tcp port** *number* **Example:** `Router(config)# ip nat service skinny tcp port 20002`	Configures the skinny protocol on the specified TCP port.

Configuration Examples for Using Application Level Gateways with NAT

This section provides the following configuration examples:

- Configuring IPSec ESP Through NAT: Example, page 11
- Enabling the Preserve Port: Example, page 11
- Enabling SPI Matching: Example, page 11
- Configuring SPI Matching on the Endpoint Routers: Example, page 11
- Deploying NAT Between an IP Phone and Cisco CallManager: Example, page 11

Configuring IPSec ESP Through NAT: Example

The following example shows NAT configured on the Provider Edge (PE) router with a static route to the shared service for the gold and silver Virtual Private Networks (VPNs). NAT is configured as inside source static 1- to-1 translations.

```
ip nat pool outside 4.4.4.1 4.4.4.254 netmask 255.255.255.0
ip nat outside source list 1 pool mypool
access-list 1 permit 168.58.18.0 0.0.0.255
ip nat inside source static 192.168.121.33 2.2.2.1 vrf gold
ip nat inside source static 192.169.121.33.2.2.2.2 vrf silver
```

Enabling the Preserve Port: Example

The following example shows how to configure TCP port 500 of the third-party concentrator:

```
ip nat service list 10 ike preserve-port
```

Enabling SPI Matching: Example

The following example shows how to enable SPI matching:

```
ip nat service list 10 esp spi-match
```

Configuring SPI Matching on the Endpoint Routers: Example

The following example show how to enable SPI matching on the endpoint routers:

```
crypto ipsec spi-matching
```

Deploying NAT Between an IP Phone and Cisco CallManager: Example

The following example shows how to configure the 20002 port of the CallManager:

```
ip nat service skinny tcp port 20002
```

Where to Go Next

- To learn about Network Address Translation and configure NAT for IP address conservation, see the “Configuring NAT for IP Address Conservation” module.
- To verify monitor, and maintain NAT, see the “Monitoring and Maintaining NAT” module.
- To integrate NAT with MPLS VPNs, see the “Integrating NAT with MPLS VPNs” module.
- To configure NAT for high availability, see the “Configuring NAT for High Availability” module.

Additional References

The following sections provide references related to using application level gateways with NAT.

Related Documents

Related Topic	Document Title
NAT commands: complete command syntax, command mode, defaults, usage guidelines, and examples	*Cisco IOS IP Addressing Services Command Reference*

Standards

Standards	Title
None	

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport

Feature Information for Using Application Level Gateways with NAT

Table 1 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.2(1) or later appear in the table.

Not all commands may be available in your Cisco IOS software release. For details on when support for specific commands was introduced, see the command reference documents.

For information on a feature in this technology that is not documented here, see the “Configuring Network Address Translation Features Roadmap.”

Cisco IOS software images are specific to a Cisco IOS software release, a feature set, and a platform. Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for Using Application Level Gateways with NAT

Feature Name	Releases	Feature Configuration Information
The NAT Support for IPSec ESP— Phase II feature	12.2(15)T Cisco IOS XE Release 2.1	The NAT Support for IPSec ESP— Phase II feature provides support for Internet Key Exchange (IKE) and ESP without encapsulation in tunnel mode through a Cisco IOS router configured with NAPT. The following sections provide information about this feature: • “Configuring IPSec Through NAT” section on page 2 • “Configuring IPSec ESP Through NAT: Example” section on page 11
NAT Support for SIP feature	12.2(8)T Cisco IOS XE Release 2.1	NAT Support for SIP adds the ability to deploy Cisco IOS NAT between VoIP solutions based on SIP. The following section provides information about this feature: • “Configuring IPSec Through NAT” section on page 2
NAT Support for H.323 v2 RAS feature	12.2(2)T	Cisco IOS NAT supports all H.225 and H.245 message types, including those sent in the RAS protocol. The following section provides information about this feature: • “NAT Support of H.323 v2 RAS” section on page 4
Support for IPSec ESP Through NAT	12.2(13)T Cisco IOS XE Release 2.1	IPSec ESP Through NAT provides the ability to support multiple concurrent IP Security (IPSec) Encapsulating Security Payload (ESP) tunnels or connections through a Cisco IOS Network Address Translation (NAT) device configured in Overload or Port Address Translation (PAT) mode. The following section provides information about this feature: • “Configuring IPSec ESP Through NAT” section on page 5

Table 1 Feature Information for Using Application Level Gateways with NAT

Feature Name	Releases	Feature Configuration Information
NAT Support for H.323 v3 and v4 in v2 Compatibility Mode	12.3(2)T Cisco IOS XE Release 2.1	The NAT Support for H.323 v3 and v4 in v2 Compatibility Mode feature enables Cisco NAT routers to support messages coded in H.323 v3 and v4 when those messages contain fields compatible with H.323 v2. This feature does not add support for H.323 capabilities introduced in v3 and v4, such as new message types or new fields that require address translation. The following section provides information about this feature: • "NAT Support for H.323 v3 and v4 in v2 Compatibility Mode" section on page 4
NAT H.245 Tunneling Support	12.3(11)T	The NAT H.245 Tunneling Support feature allows H.245 tunneling in H.323 Application Level Gateways (ALGs). The following section provides information about this feature: • "NAT H.245 Tunneling Support" section on page 4
NAT SCCP Fragmentation Support	12.4(6)T	The NAT SCCP Fragmentation Support feature adds support for TCP segments for NAT skinny ALG. A fragmented payload that requires an IP or port translation will no longer be dropped. The following section provides information about this feature: • "NAT Support of SCCP Fragmentation" section on page 10

Configuring NAT for High Availability

This module contains procedures for configuring Network Address Translation (NAT) to support the increasing need for highly resilient IP networks. This network resiliency is required where application connectivity needs to continue unaffected by failures to links and routers at the NAT border.

Module History

This module was first published on May 2, 2005, and last updated on May 2, 2005.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all features. To find information about feature support and configuration, use the "Feature Information for Configuring NAT for High Availability" section on page 20.

Contents

Prerequisites for Configuring NAT for High Availability

- Before performing the tasks in this module, you should be familiar with the concepts described in the "Configuring NAT for IP Address Conservation" module.
- All access lists required for use with the tasks in this module should be configured prior to beginning the configuration tasks. For information about how to configure an access list, see the "*IP Access List Sequence Numbering*" document at the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122s/122snwft/release/122s14/fsaclseq.htm

Note If you specify an access list to use with a NAT command, NAT does not support the commonly used **permit ip any any** command in the access list.

Restrictions for Configuring NAT for High Availability

The Address Resolution Protocol (ARP) queries are always replied to by the Hot Standby Routing Protocol (HSRP) active router. If the active HSRP router fails upstream devices will point to the new HSRP active router and will not have an ARP entry pointing to the original active router, which may no longer be available.

Information About Configuring NAT for High Availability

To configure NAT for High availability, you should understand the following concepts:

- Stateful NAT, page 2
- NAT Stateful Failover for Asymmetric Outside-to-Inside ALG Support, page 2
- Interaction with HSRP, page 2
- Translation Group, page 3
- Address Resolution with ARP, page 3

Stateful NAT

Stateful NAT (SNAT) enables continuous service for dynamically mapped NAT sessions. Sessions that are statically defined receive the benefit of redundancy without the need for SNAT. In the absence of SNAT, sessions that use dynamic NAT mappings would be severed in the event of a critical failure and would have to be reestablished.

SNAT can be used with protocols that do not need payload translation.

NAT Stateful Failover for Asymmetric Outside-to-Inside ALG Support

NAT stateful failover for asymmetric outside-to-inside and Application Layer Gateway (ALG) support improves the ability to handle asymmetric paths by allowing multiple routing paths from outside-to-inside, and per-packet load balancing. This feature also provides seamless failover translated IP sessions with traffic that includes embedded IP addressing such as Voice over IP, FTP, and Domain Name System (DNS) applications.

Interaction with HSRP

SNAT can be configured to operate with the Hot Standby Routing Protocol (HSRP) to provide redundancy. Active and Standby state changes are managed by HSRP.

SNAT applies a more global context to the task of forwarding a particular datagram. Consideration is given to understanding the application state along with forwarding. Devices can take action to avoid potential failures that will have less impact on the flow and to the application that is transmitting data. Multiple NAT routers that share stateful context can work cooperatively and thereby increase service availability.

Translation Group

Two or more network address translators function as a translation group. One member of the group handles traffic requiring translation of IP address information. It also informs the backup translator of active flows as they occur. The backup translator can then use information from the active translator to prepare duplicate translation table entries, and in the event that the active translator is hindered by a critical failure, the traffic can rapidly be switched to the backup. The traffic flow continues since the same network address translations are used, and the state of those translations has been previously defined.

Address Resolution with ARP

A device in IP can have both a local address (which uniquely identifies the device on its local segment or LAN) and a network address (which identifies the network to which the device belongs). The local address is more properly known as a data link address because it is contained in the data link layer (Layer 2 of the OSI model) part of the packet header and is read by data-link devices (bridges and all device interfaces, for example). The local address is referred to as the MAC address, because the MAC sub-layer within the data link layer processes addresses for the layer.

To communicate with a device on Ethernet, for example, the Cisco IOS software first must determine the 48-bit MAC or local data-link address of that device. The process of determining the local data-link address from an IP address is called address resolution. The process of determining the IP address from a local data-link address is called reverse address resolution.

The software uses three forms of address resolution: Address Resolution Protocol (ARP), proxy ARP, and Probe (similar to ARP). The software also uses the Reverse Address Resolution Protocol (RARP). ARP, proxy ARP, and RARP are defined in RFCs 826, 1027, and 903, respectively. Probe is a protocol developed by the Hewlett-Packard Company (HP) for use on IEEE-802.3 networks.

ARP is used to associate IP addresses with media or MAC addresses. Taking an IP address as input, ARP determines the associated media address. Once a media or MAC address is determined, the IP address or media address association is stored in an ARP cache for rapid retrieval. Then the IP datagram is encapsulated in a link-layer frame and sent over the network. Encapsulation of IP datagrams and ARP requests and replies on IEEE 802 networks other than Ethernet is specified by the Subnetwork Access Protocol (SNAP).

How to Configure NAT for High Availability

This module describes three methods for configuring NAT for high availability:

- Configuring the Stateful Failover of NAT, page 4 (optional)
- Configuring NAT Stateful Failover for Asymmetric Outside-to-Inside and ALG Support, page 8 (optional)
- Configuring NAT Static Mapping Support for HSRP, page 14 (optional)

Configuring the Stateful Failover of NAT

The NAT Stateful Failover of Network Address Translation feature represents Phase 1 of the stateful failover capability. It introduces support for two or more network address translators to function as a translation group. A backup router running NAT provides translation services in the event the active translator fails. Protocols that do not need payload translations, such as HTTP and telnet, are supported by stateful NAT (SNAT).

This section contains the following procedures:

- Configuring SNAT with HSRP, page 4 (optional)
- Configuring SNAT on the Primary (Active) Router, page 6 (optional)
- Configuring SNAT on the Backup (Standby) Router, page 7 (optional)

Restrictions for Configuring Stateful Failover of NAT

The following applications and protocols are not supported in Phase I:

- Application Level Gateway (ALG)
- FTP
- NetMeeting Directory (ILS)
- RAS
- SIP
- Skinny
- TFTP
- Asymmetrical routing

Configuring SNAT with HSRP

Perform this task to configure Stateful NAT using HSRP to provide router backup facilities.

Note This task must be performed on both the **active** and the **standby** routers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **standby** [*group-name]* **ip** [*ip-address* [**secondary**]]
5. **exit**
6. **ip nat stateful id** *id-number {***redundancy** *name* **mapping-id** *map-number}*
7. **ip nat pool** *name start-ip end-ip* **prefix-length** *prefix-length*
8. **ip nat inside source** {**route-map** *name* **pool** *pool-name* **mapping-id** *map-number*} [**overload**]
9. **exit**

10. **show ip snat distributed verbose**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`enable`** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. • Enter your password if prompted.
Step 2	**`configure terminal`** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**`interface`** *`type number`* **Example:** `Router(config)# interface ethernet 1/1`	Enters interface configuration mode.
Step 4	**`standby [`***`group-name]`* **`ip`** `[`*`ip-address`* `[`**`secondary`**`]]` **Example:** `Router(config-if)# standby SNATHSRP ip 10.1.1.1 secondary`	Enables the HSRP protocol.
Step 5	**`exit`** **Example:** `Router(config-if)# exit`	Returns to global configuration mode.
Step 6	**`ip nat stateful id`** *`id-number`* `{`**`redundancy`** *`name`* **`mapping-id`** *`map-number}`* **Example:** `Router(config)# ip nat stateful id 1 redundancy snathsrp mapping-id 10`	Specifies SNAT on routers configured for HSRP.
Step 7	**`ip nat pool`** *`name start-ip end-ip`* **`prefix-length`** *`prefix-length`* **Example:** `Router(config)# ip nat pool snatpool1 10.1.1.1 10.1.1.9 prefix-length 24`	Defines a pool of IP addresses.
Step 8	**`ip nat inside source`** `{`**`route-map`** *`name`* **`pool`** *`pool-name`* **`mapping-id`** *`map-number`*`}` `[`**`overload`**`]` **Example:** `Router(config)# ip nat inside source route-map rm-101 pool snatpool1 mapping-id 10 overload`	Enables stateful NAT for the HSRP translation group.

	Command or Action	Purpose
Step 9	**exit** **Example:** `Router> exit`	Returns to privileged EXEC mode.
Step 10	**show ip snat distributed verbose** **Example:** `Router# show ip snat distributed verbose`	(Optional) Displays active stateful NAT translations.

Configuring SNAT on the Primary (Active) Router

Perform this task to manually configure your primary SNAT router. When you have completed this task, perform the steps in the "Configuring SNAT on the Backup (Standby) Router" section on page 7.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat stateful id** *id-number* **primary** *ip-address* **peer** *ip-address* **mapping-id** *map-number*
4. **ip nat pool** *name start-ip end-ip* {**prefix-length** *prefix-length*}
5. **ip nat inside source route-map** *name* **pool** *pool-name* **mapping-id** *map-number* [**overload**]
6. **exit**
7. **show ip snat distributed verbose**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure** terminal **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat stateful id** *id-number* **primary** *ip-address* **peer** *ip-address* **mapping-id** *map-number* **Example:** `Router(config)# ip nat stateful id 1 primary 10.10.10.10 peer 10.22.22.22 mapping-id 10`	Specifies stateful NAT on the primary router.

	Command or Action	Purpose
Step 4	**ip nat pool** *name start-ip end-ip* **prefix-length** *prefix-length* **Example:** `Router(config)# ip nat pool SNATPOOL1 10.1.1.1 10.1.1.9 prefix-length 24`	Defines a pool of IP addresses.
Step 5	**ip nat inside source route-map** *name* **pool** *pool-name* **mapping-id** *map-number* [**overload**] **Example:** `Router(config)# ip nat inside source route-map rm-101 pool snatpool1 mapping-id 10 overload`	Enables stateful NAT for the HSRP translation group.
Step 6	**exit** **Example:** `Router> exit`	Returns to privileged EXEC mode.
Step 7	**show ip snat distributed verbose** **Example:** `Router# show ip snat distributed verbose`	(Optional) Displays active stateful NAT translations.

Configuring SNAT on the Backup (Standby) Router

Perform this task to manually configure your backup (standby) SNAT router.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat stateful id** *id-number* **back-up** *ip-address* **peer** *ip-address* **mapping-id** *map-number*
4. **ip nat pool** *name start-ip end-ip* **prefix-length** *prefix-length*
5. **ip nat inside source route-map** *name* **pool** *pool-name* **mapping-id** *map-number* [**overload**]
6. **exit**
7. **show ip snat distributed verbose**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat stateful id** *id-number* **backup** *ip-address* **peer** *ip-address* **mapping-id** *map-number* **Example:** `Router(config)# ip nat stateful id 1 backup 10.2.2.2 peer 10.10.10.10 mapping-id 10`	Specifies stateful NAT on the backup router.
Step 4	**ip nat pool** *name start-ip end-ip* **prefix-length** *prefix-length* **Example:** `Router(config)# ip nat pool SNATPOOL1 10.1.1.1 10.1.1.9 prefix-length 24`	Defines a pool of IP addresses.
Step 5	**ip nat inside source route-map** *name* **pool** *pool-name* **mapping-id** *map-number* [**overload**] **Example:** `Router(config)# ip nat inside source route-map rm-101 pool snatpool1 mapping-id 10 overload`	Enables stateful NAT for the HSRP translation group.
Step 6	**exit** **Example:** `Router> exit`	Returns to privileged EXEC mode.
Step 7	**show ip snat distributed verbose** **Example:** `Router# show ip snat distributed verbose`	(Optional) Displays active stateful NAT translations.

Configuring NAT Stateful Failover for Asymmetric Outside-to-Inside and ALG Support

Stateful NAT Phase I required all sessions to pass through the primary NAT router that controlled the NAT translation entries unless the primary NAT router was unavailable. This requirement assured integrity of the translation information by guarding against the possibility of some packets relevant to NAT session control from traversing the backup without the primary being aware of it. Without synchronized IP sessions NAT eventually times out the IP session entries and the result is IP session states that are out of sequence.

This section contains the following procedures:

- Configuring SNAT with HSRP, page 11 (required)
- Configuring SNAT Primary/Backup, page 12 (required)

Prerequisites for Configuring the NAT Stateful Failover for Asymmetric Outside-to-Inside and ALG Support Feature

Each router must have the same Network Address Translation (NAT) configurations.

Benefits of Configuring Stateful Failover for Asymmetric Outside-to-Inside Support

The stateful failover asymmetric outside-to-inside enhancement provides the following benefits:

- Ability to support multiple routing paths from outside-to-inside
- Ability to handle per-packet load balancing of asymmetric routing from outside-to-inside

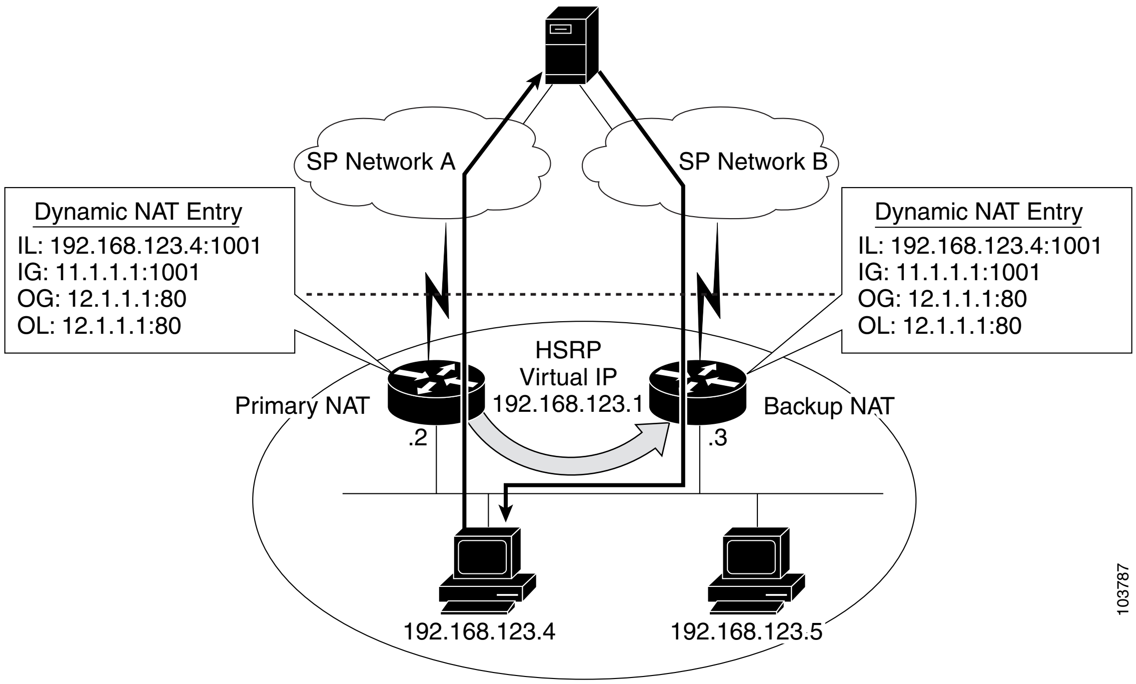
How Stateful Failover for Asymmetric Outside-to-Inside Support Works

Stateful failover for asymmetric outside-to-inside support enables two NAT routers to participate in a primary/backup design. One of the routers is elected as the primary NAT router and a second router acts as the backup router. As traffic is actively translated by the primary NAT router it updates the backup NAT router with the NAT translation state from NAT translation table entries. If the primary NAT router fails or is out of service, the backup NAT router will automatically take over. When the primary comes back into service it will take over and request an update from the backup NAT router. Return traffic is handled by either the primary or the backup NAT translator and NAT translation integrity is preserved.

When the backup NAT router receives asymmetric IP traffic and performs NAT of the packets, it will update the primary NAT router to ensure both the primary and backup NAT translation tables remain synchronized.

Figure 1 on page 10 shows a typical configuration that uses the NAT Stateful Failover for Asymmetric Outside-to-Inside and ALG Support feature.

Figure 1 Stateful NAT Asymmetric Outside-to-Inside Support



How Stateful Failover for ALGs Works

The stateful failover embedded addressing enhancement allows the secondary or backup NAT router to properly handle NAT and delivery of IP traffic. NAT inspects all IP traffic entering interfaces that have been configured with the NAT feature. The inspection consists of matching the incoming traffic against a set of translations rules and performs an address translation if a match occurs. The following are examples:

- Matching a source address range
- Matching a specific destination address range
- Matching a list of applications known to NAT that might require a specific source port for control plane negotiation, or embedded source IP addresses within the application protocol

Some of the applications and protocols that embed source port or IP address information include:

- H.323 Registration, Admission, and Status (RAS) Protocol
- DNS queries
- NetMeeting Internet Locator Server (ILS)
- Internet Control Message Protocol (ICMP)
- Simple Mail Transfer Protocol (SMTP)
- Point-to-Point Tunneling Protocol (PPTP)
- Network File System (NFS)
- Cisco Selsius Skinny Client Protocol (SCCP)

A complete list of current ALG protocols supported by Cisco IOS NAT can be found at

http://www.cisco.com/en/US/tech/tk648/tk361/tech_brief09186a00801af2b9.html

Configuring SNAT with HSRP

To configure your Hot Standby Router Protocol (HSRP) router with Stateful Network Address Translation (SNAT), use the following commands:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **standby** [*group-name]* **ip** [*ip-address* [**secondary**]]
5. **exit**
6. **ip nat stateful id** *ip-address* **redundancy** *group-name* **mapping-id** *map-id*
7. **ip nat pool** *name start-ip end-ip* **prefix-length** *prefix-length*
8. **ip nat inside source route-map** *name* **pool** *pool-name* **mapping-id** *map-id* [**overload**]
9. **ip nat inside destination list** *number* **pool** *name* **mapping-id** *map-id*
10. **ip nat outside source static** *global-ip local-ip* **extendable mapping-id** *map-id*
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface ethernet 1/1`	Enters interface configuration mode.
Step 4	**standby [***group-name]* **ip** [*ip-address* [**secondary**]] **Example:** `Router(config-if)# standby SNATHSRP ip 11.1.1.1 secondary`	Enables the HSRP protocol.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Returns to global configuration mode.

	Command or Action	Purpose
Step 6	**ip nat stateful id** *ip-address* **redundancy** *group-name* **mapping-id** *map-id* **Example:** `Router(config)# ip nat stateful id 1 redundancy snathsrp mapping-id 10`	Specifies SNAT on routers configured for HSRP.
Step 7	**ip nat pool** *name start-ip end-ip* **prefix-length** *prefix-length* **Example:** `Router(config)# ip nat pool snatpool1 11.1.1.1 11.1.1.9 prefix-length 24`	Defines a pool of IP addresses.
Step 8	**ip nat inside source static route-map** *name* **pool** *pool-name* **mapping-id** *map-id* [**overload**] **Example:** `Router(config)# ip nat inside source static route-map rm-101 pool snatpool2 mapping-id 10 overload`	Enables stateful NAT for the HSRP translation group.
Step 9	**ip nat inside destination list** *number* **pool** *name* **mapping-id** *map-id* **Example:** `Router(config)# ip nat inside destination list 1 pool snatpool2 mapping-id 10`	Enables the local SNAT router to distribute a particular set of locally created entries to a peer SNAT router.
Step 10	**ip nat outside source static** *global-ip local-ip* **extendable mapping-id** *map-id* **Example:** `Router(config)# ip nat outside source static 1.1.1.1 2.2.2.2 extendable mapping-id 10`	Enables stateful NAT for the HSRP translation group.
Step 11	**end** **Example:** `Router(config)# end`	Exits global configuration mode. • Use the **end** command to save your configuration and leave configuration mode.

Configuring SNAT Primary/Backup

Use the following commands to enable the NAT Stateful Failover for Asymmetric Outside-to-Inside and ALG Support feature:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat stateful id** *id-number* **primary** *ip-address* **peer** *ip-address* **mapping-id** *map-number*
4. **ip nat pool** *name start-ip end-ip* **prefix-length** *prefix-length*

5. **ip nat inside source static route-map** *name* **pool** *pool-name* **mapping-id** *map-id* [**overload**]
6. **ip nat inside destination list** *number* **pool** *name* **mapping-id** *map-id*
7. **ip nat outside source static** *global-ip local-ip* **extendable mapping-id** *map-id*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat stateful id** *id-number* **primary** *ip-address* **peer** *ip-address* **mapping-id** *map-id* **Example:** `Router(config)# ip nat stateful id 1 primary 1.1.1.1 peer 2.2.2.2 mapping-id 10`	Specifies stateful NAT on the primary router.
Step 4	**ip nat pool** *name start-ip end-ip* **prefix-length** *prefix-length* **Example:** `Router(config)# parser config cache interface`	Defines a pool of IP addresses.
Step 5	**ip nat inside source static route-map** *name* **pool** *pool-name* **mapping-id** *map-id* **[overload]** **Example:** `Router(config)# ip nat inside source static route-map rm-101 pool snatpool2 mapping-id 10 overload`	Enables stateful NAT of the inside source address to distribute a particular set of locally created entries to a peer SNAT router.
Step 6	**ip nat inside destination list** *number* **pool** *name* **mapping-id** *map-id* **Example:** `Router(config)# ip nat inside destination list 1 pool snatpool2 mapping-id 10 overload`	Defines the inside destination address that enables the local SNAT router to distribute locally created entries to a peer SNAT router.

	Command or Action	Purpose
Step 7	**ip nat outside source Static** *global-ip local-ip* **extendable mapping-id** *map-id* **Example:** `Router(config)# ip nat outside source static 1.1.1.1 2.2.2.2 extendable mapping-id 10`	Enables stateful NAT of the outside source address to distribute a particular set of locally created entries to a peer SNAT router.
Step 8	**end** **Example:** `Router(config)# end`	Exits global configuration mode. • Use the **end** command to save your configuration and leave configuration mode.

Configuring NAT Static Mapping Support for HSRP

When an Address Resolution Protocol (ARP) query is triggered for an address that is configured with NAT static mapping and owned by the router, NAT responds with the burned in MAC (BIA MAC) address on the interface to which the ARP is pointing. Two routers are acting as HSRP active and standby. Their NAT inside interfaces must be enabled and configured to belong to a group.

Both of the following tasks are required and must be performed on both the active and standby routers to configure NAT static mapping support for HSRP:

- Enabling HSRP on the NAT Interface, page 14 (required)
- Enabling Static NAT in an HSRP Environment, page 16 (required)

Restrictions for Configuring Static Mapping Support for HSRP

- Configuring static mapping support for HSRP provides NAT support in the presence of HSRP using static mapping configuration only.
- Static NAT mappings must be mirrored on two or more HSRP routers, because NAT state will not be exchanged between the routers running NAT in an HSRP group.
- Behavior will be unpredictable if both HSRP routers have the same static NAT and are not configured with the **hsrp** keyword linking them to the same HSRP group.

Benefits of Configuring Static Mapping Support for HSRP

- Using static mapping support for HSRP, failover is ensured without having to time out and repopulate upstream ARP caches in a high-availability environment, where HSRP router pairs have identical NAT configuration for redundancy.
- Static mapping support for HSRP allows the option of having only the HSRP active router respond to an incoming ARP for a router configured with a NAT address.

Enabling HSRP on the NAT Interface

Perform this task to enable HSRP on the NAT interface of both the active and standby routers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask*
5. **no ip redirects**
6. **ip nat** {**inside** | **outside**}
7. **standby** [*group-number*] **ip** [*ip-address* [**secondary**]]
8. **standby name** [*group-name*]
9. *end*
10. **show standby**
11. **show ip nat translations** [**verbose**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`enable`** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. • Enter your password if prompted.
Step 2	**`configure terminal`** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**`interface`** *`type number`* **Example:** `Router(config)# interface ethernet 1/1`	Enters interface configuration mode.
Step 4	**`ip address`** *`ip-address mask`* **Example:** `Router(config-if)# ip address 192.168.1.27 255.255.255.0`	Sets the primary IP address on the interface.
Step 5	**`no ip redirects`** **Example:** `Router(config-if)# no ip redirects`	Disables the sending of redirect messages
Step 6	**`ip nat`** `{`**`inside`** `|` **`outside`**`}` `Router(config)# ip nat inside`	Marks the interface as connected to the inside or outside.
Step 7	**`standby`** `[`*`group-number`*`]` **`ip`** `[`*`ip-address`* `[`**`secondary`**`]]` **Example:** `Router(config-if)# standby 10 ip 192.168.5.30`	Enables the HSRP protocol.

	Command or Action	Purpose
Step 8	**standby** [*group-number*] **name** [*group-name*] **Example:** `Router(config-if)# standby 10 name HSRP1`	Sets the HSRP group name.
Step 9	**end** **Example:** `Router(config-if)# exit`	Returns to privileged EXEC mode.
Step 10	**show standby** **Example:** `Router# show standby`	(Optional) Displays HSRP information
Step 11	**show ip nat translations** [**verbose**] **Example:** `Router# show ip nat translations verbose`	(Optional) Displays active NAT translations.

What to Do Next

Go to the next section and enable static NAT in the HSRP environment.

Enabling Static NAT in an HSRP Environment

To enable static mapping support with HRSP for high availability, perform this task on both the active and standby routers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat inside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name*} [**overload**] | **static** *local-ip global- ip* **redundancy** *group-name*}
4. **ip nat outside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name*} [**overload**] | **static** *local-ip global- ip* **redundancy** *group-name*}
5. exit
6. **show ip nat translations** [**verbose**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat inside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name*} [**overload**] | **static** *local-ip global-ip* **redundancy** *group-name*} `Router(config)# ip nat inside source static 192.168.5.33 10.10.10.5 redundancy HSRP1`	Enables the router to respond to ARP queries using BIA MAC, if HSRP is configured on the NAT inside interface.
Step 4	**ip nat outside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name*} [**overload**] | **static** *local-ip global-ip* **redundancy** *group-name*} `Router(config)# ip nat outside source static 192.168.5.33 10.10.10.5 redundancy HSRP1`	Enables the router to respond to ARP queries using BIA MAC, if HSRP is configured on the NAT outside interface.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Returns to privileged EXEC mode.
Step 6	**show ip nat translations** [**verbose**] **Example:** `Router# show ip nat translations verbose`	(Optional) Displays active NAT translations.

Configuration Example for NAT for High Availability

This section provides the following configuration examples:

Configuring Stateful NAT: Examples

The following examples show configuring stateful NAT with HSRP and configuring stateful NAT primary and backup routers.

SNAT with HSRP Example

```
ip nat Stateful id 1
```

```
redundancy SNATHSRP
mapping-id 10
ip nat pool SNATPOOL1 10.1.1.1 10.1.1.9 prefix-length 24
ip nat inside source route-map rm-101 pool SNATPOOL1 mapping-id 10 overload
ip classless
ip route 10.1.1.0 255.255.255.0 Null0
no ip http server
ip pim bidir-enable
```

Configuring SNAT Primary/Backup Example

```
ip nat Stateful id 1
primary 10.88.194.17
peer 10.88.194.18
mapping-id 10
!
ip nat Stateful id 2
backup 10.88.194.18
peer 10.88.194.17
mapping-id 10
```

Configuration Examples for NAT Stateful Failover for Asymmetric Outside-to-Inside and ALG Support

This section contains the following examples:

- Configuring SNAT with HSRP, page 11
- Enabling HSRP on the NAT Interface, page 14

Configuring SNAT with HSRP: Example

The following example shows how to configure SNAT with HSRP.

```
ip nat Stateful id 1
redundancy SNATHSRP
mapping-id 10
ip nat pool SNATPOOL1 11.1.1.1 11.1.1.9 prefix-length 24
ip nat inside source route-map rm-101 pool SNATPOOL1 mapping-id 10 overload
ip classless
ip route 11.1.1.0 255.255.255.0 Null0
no ip http server
ip pim bidir-enable
```

Configuring SNAT Primary/Backup: Example

The following example shows how to configure SNAT on the primary/backup router.

```
ip nat Stateful id 1
primary 10.88.194.17
peer 10.88.194.18
mapping-id 10
!
ip nat Stateful id 2
backup 10.88.194.17
peer 10.88.194.17
mapping-id 10
```

Configuring Static NAT in an HSRP Environment: Examples

The following example shows support for NAT with a static configuration in an HSRP environment. Two routers are acting as HSRP active and standby, and the NAT inside interfaces are HSRP enabled and configured to belong to the group HSRP1.

Active Router Configuration

```
interface BVI10
 ip address 192.168.5.54 255.255.255.255.0
 no ip redirects
 ip nat inside
 standby 10 priority 105 preempt
 standby 10 name HSRP1
 standby 10 ip 192.168.5.30
 standby 10 track Ethernet2/1
!
!
 ip default-gateway 10.0.18.126
 ip nat inside source static 192.168.5.33 10.10.10.5 redundancy HSRP1
 ip classless
 ip route 10.10.10.0 255.255.255.0 Ethernet2/1
 ip route 172.22.33.0 255.255.255.0 Ethernet2/1
 no ip http server
```

Standby Router Configuration

```
interface BVI10
 ip address 192.168.5.56 255.255.255.255.0
 no ip redirects
 ip nat inside
 standby 10 priority 100 preempt
 standby 10 name HSRP1
 standby 10 ip 192.168.5.30
 standby 10 track Ethernet3/1
!
 ip default-gateway 10.0.18.126
 ip nat inside source static 192.168.5.33 3.3.3.5 redundancy HSRP1
 ip classless
 ip route 10.0.32.231 255.255.255 Ethernet3/1
 ip route 10.10.10.0 255.255.255.0 Ethernet3/1
 no ip http server
```

Additional References

The following sections provide references related toNAT for high availability.

Related Documents

Related Topic	Document Title
NAT configuration tasks	“Configuring NAT for IP Address Conservation” module
Using NAT with MPLS VPNs	“Integrating NAT with MPLS VPNs” module

Related Topic	Document Title
NAT maintenance	"Monitoring and Maintaining NAT" module
NAT commands: complete command syntax, command mode, command history, usage guidelines, and examples	*Cisco IOS IP Addressing Services Command Reference*

Standards

Standards	Title
None	

MIBs

MIBs	MIBs Link
• None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 903	*Reverse Address Resolution Protocol*
RFC 826	*Ethernet Address Resolution Protocol: Or converting network protocol addresses to 48.bit Ethernet address for transmission on Ethernet hardware*
RFC 1027	*Using ARP to implement transparent subnet gateways*

Feature Information for Configuring NAT for High Availability

Table 1 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Releases 12.2(4) or later appear in the table.

Not all commands may be available in your Cisco IOS software release. For details on when support for a specific command was introduced, see the command reference documentation.

For information on a feature in this technology that is not documented here, see the "Configuring Network Address Translation Features Roadmap."

Cisco IOS software images are specific to a Cisco IOS software release, a feature set, and a platform. Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for Configuring NAT for High Availability

Feature Name	Releases	Feature Configuration Information
NAT—Static Mapping Support with HSRP for High Availability	12.2(4)T 12.2(4)T2	Static mapping support for HSRP allows the option of having only the HSRP active router respond to an incoming ARP for a router configured with a NAT address. The following sections provide information about this feature: • "Configuring NAT Stateful Failover for Asymmetric Outside-to-Inside and ALG Support" section on page 8 • "Configuring Static NAT in an HSRP Environment: Examples" section on page 19
NAT Stateful Failover of Network Address Translation	12.2(13)T	The NAT Stateful Failover of Network Address Translation feature represents Phase 1 of the stateful failover capability. It introduces support for two or more network address translators to function as a translation group. The following sections provide information about this feature: • "Configuring the Stateful Failover of NAT" section on page 4 • "Configuring Stateful NAT: Examples" section on page 17
NAT Stateful Failover for Asymmetric Outside-to-Inside ALG Support	12.3(7)T	The NAT Stateful Failover for Asymmetric Outside-to-Inside and Application Layer Gateway (ALG) Support feature improves the ability to handle asymmetric paths by allowing multiple routing paths from outside-to-inside, and per-packet load balancing. This feature also provides seamless failover translated IP sessions with traffic that includes embedded IP addressing such as Voice over IP, FTP, and Domain Name System (DNS) applications. The following sections provide information about this feature: • "Configuring NAT Stateful Failover for Asymmetric Outside-to-Inside and ALG Support" section on page 8 • "Configuration Examples for NAT Stateful Failover for Asymmetric Outside-to-Inside and ALG Support" section on page 18

Technical Assistance

The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport

Integrating NAT with MPLS VPNs

Network Address Translation (NAT) Integration with MPLS VPNs feature allows multiple Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs) to be configured on a single device to work together. NAT can differentiate which MPLS VPN it receives IP traffic from even if the MPLS VPNs are all using the same IP addressing scheme. This enhancement enables multiple MPLS VPN customers to share services while ensuring that each MPLS VPN is completely separate from the other.

Module History

This module was first published on May 2, 2005, and last updated on May 2, 2005.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all features. To find information about feature support and configuration, use the "Feature Information for Integrating NAT with MPLS VPNs" section on page 12.

Contents

Prerequisites for Integrating NAT with MPLS VPNs

- Before performing the tasks in this module, you should be familiar with the concepts described in the "Configuring NAT for IP Address Conservation" module.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

- All access lists required for use with the tasks in this module should be configured prior to beginning the configuration task. For information about how to configure an access list, see the "IP *Access List Sequence Numbering*" document at the following URL:

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios122s/122snwft/release/122s14/fsaclseq.htm

Note If you specify an access list to use with a NAT command, NAT does not support the commonly used **permit ip any any** command in the access list.

Restrictions for Integrating NAT with MPLS VPNs

Inside VPN to VPN with NAT is not supported.

Information About Integrating NAT with MPLS VPNs

To integrate NAT with MPLS VPNs, you should understand the following concepts:

- Benefits of NAT Integration with MPLS VPNs, page 2
- Implementation Options for Integrating Nat with MPLS VPNs, page 2
- Scenarios for Implementing NAT on the PE Router, page 2

Benefits of NAT Integration with MPLS VPNs

MPLS service providers would like to provide value-added services such as Internet connectivity, domain name servers (DNS), and voice over IP (VoIP) service to their customers. The providers require that their customers; IP addresses be different when reaching the services. Because MPLS VPN allows customers to use overlapped IP addresses in their networks, NAT must be implemented to make the services possible.

Implementation Options for Integrating Nat with MPLS VPNs

There are two approaches to implementing NAT in the MPLS VPN network. NAT can be implemented on the customer edge (CE) router, which is already supported by NAT, or it can be implemented on a provider edge (PE) router. The NAT Integration with MPLS VPNs feature enables the implementation of NAT on a PE router in an MPLS cloud.

Scenarios for Implementing NAT on the PE Router

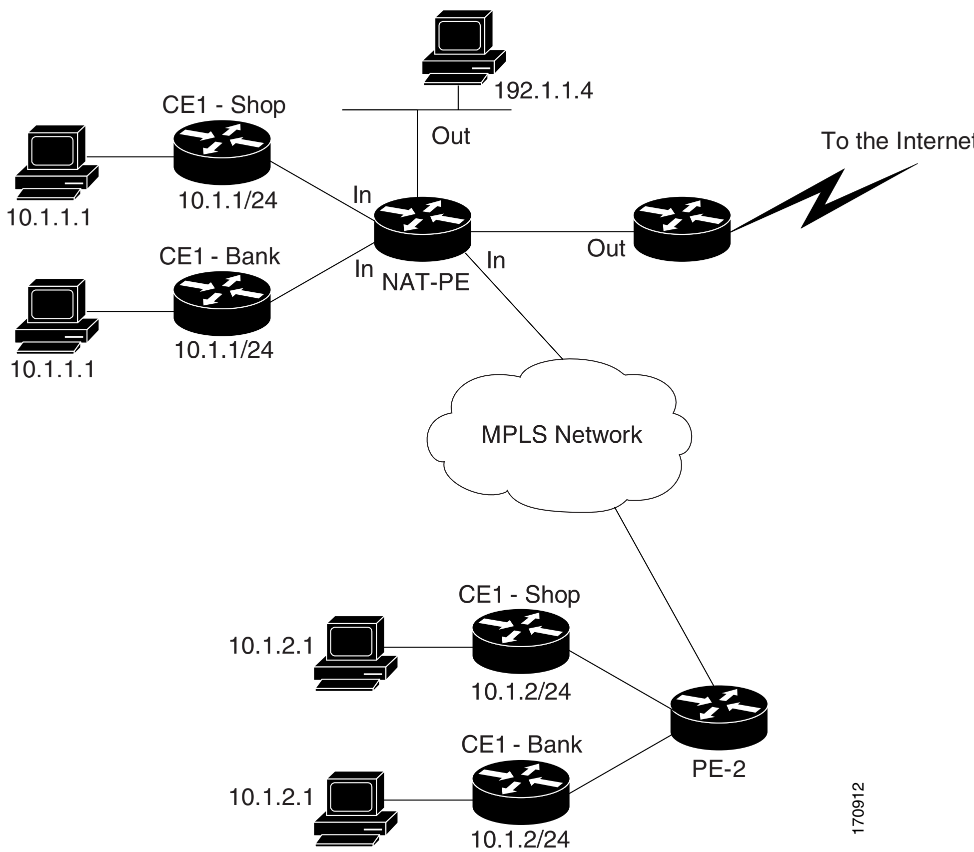
NAT could be implemented on the PE router in the following scenarios:

- Service point—Shared access can be from a generic interface or from a VPN interface.
- NAT point—NAT can be configured on the PE router that is directly connected to the shared access gateway, or on the PE router that is not directly connected to the shared access gateway.

- NAT interface—The shared access gateway interface most often is configured as the outside interface of NAT. The inside interface of NAT can be either the PE-CE interface of a VPN, the interface to the MPLS backbone, or both. The shared access gateway interface can also be configured as the inside interface.
- Routing type—Common service can be Internet connectivity or a common server. For Internet connectivity, a default route should be propagated to all the VPN customers that use the service. For common server access, a static or dynamically learned route should be propagated to the VPN customers.
- NAT configuration—NAT can have different configurations: static, dynamic, pool/interface overloading, and route-map.

Figure 1 shows a typical NAT integration with MPLS VPNs. The PE router connected to the internet and centralized mail service is employed to do the address translation.

Figure 1 Typical NAT Integration with MPLS VPNs



How to Integrate NAT with MPLS VPNs

Perform one or more of the following tasks depending on the type of translation you wish to configure for your network:

- Configuring Inside Dynamic NAT with MPLS VPNs, page 4 (optional)
- Configuring Inside Static NAT with MPLS VPNs, page 6 (optional)

- Configuring Outside Dynamic NAT with MPLS VPNs, page 7 (optional)
- Configuring Outside Static NAT with MPLS VPNs, page 8 (optional)

Configuring Inside Dynamic NAT with MPLS VPNs

Perform this task to configure your NAT PE router for dynamic translations to integrate with MPLS VPNs.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat pool** *name start-ip end-ip* **netmask** *netmask*
4. **ip nat** [**inside** | **outside**] **source** [**list** {*access-list-number* | *access-list-name*} | **route-map** *name*] [**interface** *type number* | **pool** *pool-name*] **vrf** *vrf-name* [**overload**]
5. Repeat Step 4 for all VPNs being configured.
6. **ip route vrf** *vrf-name prefix mask interface-type interface-number next-hop-address*
7. Repeat Step 6 for all VPNs being configured.
8. **exit**
9. **show ip nat translations vrf** *vrf-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`enable`** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. • Enter your password if prompted.
Step 2	**`configure terminal`** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**`ip nat pool`** *`name start-ip end-ip`* **`netmask`** *`netmask`* **Example:** `Router(config)# ip nat pool inside 2.2.2.10 2.2.2.10 netmask 255.255.255.0`	Defines a pool of IP addresses for NAT.

	Command or Action	Purpose
Step 4	**ip nat** [**inside** | **outside**] **source** [**list** {*access-list-number* | *access-list-name*} | **route-map** *name*] [**interface** *type number* | **pool** *pool-name*] **vrf** *vrf-name* [**overload**] **Example:** `Router(config)# ip nat inside source list 1 pool mypool vrf shop overload`	Allows NAT to be configured on a particular VPN.
Step 5	Repeat Step 4 for each VPN being configured	Allows NAT to be configured on a particular VPN.
Step 6	**ip route vrf** *vrf-name prefix mask interface-type interface-number next-hop-address* **Example:** Router(config)# `ip route vrf shop 0.0.0.0 0.0.0.0 ethernet 0 168.58.88.2`	Allows NAT to be configured on a particular VPN.
Step 7	Repeat Step 6 for each VPN being configured.	Allows NAT to be configured on a particular VPN.
Step 8	**exit** **Example:** `Router> exit`	Returns to privileged EXEC mode.
Step 9	**show ip nat translations vrf** *vrf-name* **Example:** Router# show ip nat translations vrf shop	(Optional) Displays the settings used by virtual routing/forwarding (VRF) table translations.

Configuring Inside Static NAT with MPLS VPNs

Perform this task to configure your NAT PE router for static translations to integrate with MPLS VPNs.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat inside source** {**static** {**esp** *local-ip* **interface** *type number* | *local-ip global-ip*}} [**extendable** | **mapping-id** *map-id* | **no-alias** | **no-payload** | **redundancy** *group-name* | **route-map** | **vrf** *name*]
4. Repeat Step 3 for each VPN being configured.
5. **ip route vrf** *vrf-name* **prefix** *prefix mask next-hop-address* **global**
6. Repeat Step 5 for each VPN being configured.
7. **exit**
8. **show ip nat translations vrf** *vrf-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`enable`** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. • Enter your password if prompted.
Step 2	**`configure terminal`** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**`ip nat inside source`** {**`static`** {**`esp`** *`local-ip`* **`interface`** *`type number`* | *`local-ip global-ip`*}} [**`extendable`** | **`mapping-id`** *`map-id`* | **`no-alias`** | **`no-payload`** | **`redundancy`** *`group-name`* | **`route-map`** | **`vrf`** *`name`*] **Example:** Router(config)# `ip nat inside source static 192.168.121.113 2.2.2.1 vrf shop`	Enables inside static translation on the VRF.
Step 4	Repeat Step 3 for each VPN being configured.	Enables inside static translation on the VRF.
Step 5	**`ip route vrf`** *`vrf-name`* **`prefix`** *`prefix mask next-hop-address`* **`global`** **Example:** Router(config)# `ip route vrf shop 0.0.0.0 0.0.0.0 168.58.88.2 global`	Allows the route to be shared by several customers.
Step 6	Repeat Step 5 for each VPN being configured.	Allows the route to be shared by several customers.

	Command or Action	Purpose
Step 7	**exit** **Example:** `Router> exit`	Returns to privileged EXEC mode.
Step 8	**show ip nat translations vrf** *vrf-name* **Example:** Router# show ip nat translations vrf shop	(Optional) Displays the settings used by VRF translations.

Configuring Outside Dynamic NAT with MPLS VPNs

Perform this step to configure your NAT PE router for dynamic outside translations to integrate with MPLS VPNs.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat pool outside** *global-ip local-ip* **netmask** ***netmask***
4. **ip nat inside source static** *local-ip global-ip* **vrf** ***vrf-name***
5. Repeat Step 4 for each VRF being configured.
6. **ip nat outside source static** *global-ip local-ip* **vrf** *vrf-name*
7. **exit**
8. **show ip nat translations vrf** *vrf-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat pool outside** *global-ip local-ip* **netmask** *netmask* **Example:** Router(config)# `ip nat pool outside 4.4.4.1 4.4.4.254 netmask 255.255.255.00`	Allows the configured VRF to be associated with the NAT translation rule.

	Command or Action	Purpose
Step 4	**ip nat inside source static** *local-ip global-ip* **vrf** *vrf-name* **Example:** Router(config)# `ip nat inside source static 192.168.121.113 2.2.2.1 vrf shop`	Allows the route to be shared by several customers.
Step 5	Repeat Step 4 for each VRF being configured.	Allows the route to be shared by several customers.
Step 6	**ip nat outside source static** *global-ip local-ip* **vrf** *vrf-name* **Example:** Router(config)# `ip nat outside source static 168.58.88.2 4.4.4.1 vrf shop`	Enables NAT translation of the outside source address.
Step 7	**exit** **Example:** `Router> exit`	Returns to privileged EXEC mode.
Step 8	**show ip nat translations vrf** *vrf-name* **Example:** Router# show ip nat translations vrf shop	(Optional) Displays the settings used by VRF translations.

Configuring Outside Static NAT with MPLS VPNs

Perform this task to configure your NAT PE router for static outside translations to integrate with MPLS VPNs.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat pool inside** *global-ip local-ip* **netmask** *netmask*
4. Repeat Step 3 for each pool being configured.
5. **ip nat inside source list** *access-list-number* **pool** *pool-name* **vrf** *vrf-name*
6. Repeat Step 5 for each pool being configured.
7. **ip nat outside source static** *global-ip local-ip* **vrf** *vrf-name*
8. Repeat Step 7 for all VPNs being configured.
9. **exit**
10. **show ip nat translations vrf** *vrf-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure** {**terminal** | **memory** | **network**} **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat pool inside** *global-ip local-ip* **netmask** *netmask* **Example:** Router(config)# `ip nat pool inside1 2.2.1.1 2.2.1.254 netmask 255.255.255.0`	Allows the configured VRF to be associated with the NAT translation rule.
Step 4	Repeat Step 3 for each pool being configured.	Allows the configured VRF to be associated with the NAT translation rule.
Step 5	**ip nat inside source list** *access-list-number* **pool** *pool-name* **vrf** *vrf-name* **Example:** Router(config)# `ip nat inside source list 1 pool inside2 vrf shop`	Allows the route to be shared by several customers.
Step 6	Repeat Step 5 for each pool being configured.	Defines the access list.
Step 7	**ip nat outside source static** *global-ip local-ip* vrf *vrf-name* **Example:** Router(config)# `ip nat outside source static 168.58.88.2 4.4.4.1 vrf shop`	Allows the route to be shared by several customers.
Step 8	Repeat Step 7 for all VPNs being configured.	Allows the route to be shared by several customers.
Step 9	**exit** **Example:** `Router> exit`	Returns to privileged EXEC mode.
Step 10	**show ip nat translations vrf** *vrf-name* **Example:** Router# show ip nat translations vrf shop	(Optional) Displays the settings used by VRF translations.

Configuration Examples for Integrating NAT with MPLS VPNs

This section provides the following configuration examples:

- Configuring Inside Dynamic NAT with MPLS VPNs: Example, page 10
- Configuring Outside Dynamic NAT with MPLS VPNs: Example, page 11
- Configuring Inside Static NAT with MPLS VPNs: Example, page 10
- Configuring Outside Static NAT with MPLS VPNs: Example, page 11

Configuring Inside Dynamic NAT with MPLS VPNs: Example

The following example shows configuring inside Dynamic NAT with MPLS VPNs.

```
!
ip nat pool inside 2.2.2.10 2.2.2.10 netmask 255.255.255.0
ip nat inside source list 1 pool inside vrf bank overload
ip nat inside source list 1 pool inside vrf park overload
ip nat inside source list 1 pool inside vrf shop overload
!
ip route vrf shop 0.0.0.0 0.0.0.0 Ethernet1/3 168.58.88.2
ip route vrf bank 0.0.0.0 0.0.0.0 Ethernet1/3 168.58.88.2
ip route vrf park 0.0.0.0 0.0.0.0 Ethernet1/3 168.58.88.2
!
access-list 1 permit 192.168.0.0 0.0.255.255
```

Configuring Inside Static NAT with MPLS VPNs: Example

The following example shows configuring inside static NAT with MPLS VPNs.

```
!
ip nat inside source static 192.168.121.113 2.2.2.1 vrf shop
ip nat inside source static 192.168.122.49 2.2.2.2 vrf shop
ip nat inside source static 192.168.121.113 2.2.2.3 vrf bank
ip nat inside source static 192.168.22.49 2.2.2.4 vrf bank
ip nat inside source static 192.168.121.113 2.2.2.5 vrf park
ip nat inside source static 192.168.22.49 2.2.2.6 vrf park
ip nat inside source static 192.168.11.1 2.2.2.11 vrf shop
ip nat inside source static 192.168.11.3 2.2.2.12 vrf shop
ip nat inside source static 140.48.5.20 2.2.2.13 vrf shop
!
ip route 2.2.2.1 255.255.255.255 Ethernet1/0 192.168.121.113
ip route 2.2.2.2 255.255.255.255 Ethernet1/0 192.168.121.113
ip route 2.2.2.3 255.255.255.255 Serial2/1.1 192.168.121.113
ip route 2.2.2.4 255.255.255.255 Serial2/1.1 192.168.121.113
ip route 2.2.2.5 255.255.255.255 FastEthernet0/0 192.168.121.113
ip route 2.2.2.6 255.255.255.255 FastEthernet0/0 192.168.121.113
ip route 2.2.2.11 255.255.255.255 Ethernet1/0 192.168.121.113
ip route 2.2.2.12 255.255.255.255 Ethernet1/0 192.168.121.113
ip route 2.2.2.13 255.255.255.255 Ethernet1/0 192.168.121.113
```

Configuring Outside Dynamic NAT with MPLS VPNs: Example

The following example shows configuring outside dynamic NAT with MPLS VPNs.

```
!
ip nat pool outside 4.4.4.1 4.4.4.254 netmask 255.255.255.0
ip nat inside source static 192.168.121.113 2.2.2.1 vrf shop
ip nat inside source static 192.168.122.49 2.2.2.2 vrf shop
ip nat inside source static 192.168.121.113 2.2.2.3 vrf bank
ip nat inside source static 192.168.22.49 2.2.2.4 vrf bank
ip nat inside source static 192.168.121.113 2.2.2.5 vrf park
ip nat inside source static 192.168.22.49 2.2.2.6 vrf park
ip nat outside source list 1 pool outside
!
```

Configuring Outside Static NAT with MPLS VPNs: Example

The following example shows configuring outside static NAT with MPLS VPNs.

```
!
ip default-gateway 10.1.15.1
ip nat pool inside1 2.2.1.1 2.2.1.254 netmask 255.255.255.0
ip nat pool inside2 2.2.2.1 2.2.2.254 netmask 255.255.255.0
ip nat pool inside3 2.2.3.1 2.2.3.254 netmask 255.255.255.0
ip nat inside source list 1 pool inside2 vrf bank
ip nat inside source list 1 pool inside3 vrf park
ip nat inside source list 1 pool inside1 vrf shop
ip nat outside source static 168.58.88.2 4.4.4.1 vrf bank
ip nat outside source static 18.68.58.1 4.4.4.2 vrf park
ip nat outside source static 168.58.88.1 4.4.4.3 vrf shop
ip classless
ip route 192.170.10.0 255.255.255.0 Ethernet1/0 192.168.121.113
ip route 192.170.11.0 255.255.255.0 Serial2/1.1 192.168.121.113
ip route 192.170.12.0 255.255.255.0 FastEthernet0/0 192.168.121.113
ip route vrf shop 0.0.0.0 0.0.0.0 168.58.88.2 global
ip route vrf bank 0.0.0.0 0.0.0.0 168.58.88.2 global
ip route vrf park 0.0.0.0 0.0.0.0 168.58.88.2 global
no ip http server
!
access-list 1 permit 192.168.0.0 0.0.255.255
```

Where to Go Next

- To learn about Network Address Translation and configure NAT for IP address conservation, see the “Configuring NAT for IP Address Conservation” module.
- To verify, monitor, and maintain NAT, see the “Monitoring and Maintaining NAT” module.
- To use NAT with application level gateways, see the “Using Application Level Gateways with NAT” module.
- To configure NAT for high availability, see the “Configuring NAT for High Availability” module.

Additional References

The following sections provide references related to NAT.

Related Documents

Related Topic	Document Title
NAT commands: complete command syntax, command mode, command history, defaults, usage guidelines and examples	*Cisco IOS IP Addressing Services Command Reference*
NAT high availability	"Configuring NAT for High Availability" module
Application Level Gateways	"Using Application Level Gateways with NAT"
Maintain and monitor NAT	"Monitoring and Maintaining NAT" module
IP Address Conservation	"Configuring NAT for IP Address Conservation" module

Standards

Standards	Title
None	

MIBs

MIBs	MIBs Link
• None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs[1]	Title
RFC 2547	BGP/MPLS VPNs

1. Not all supported RFCs are listed.

Feature Information for Integrating NAT with MPLS VPNs

Table 1 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.1(13) T or later appear in the table.

Not all commands may be available in your Cisco IOS software release. For details on when support for a specific command was introduced, see the command reference documentation.

For information on a feature in this technology that is not documented here, see the "Configuring Network Address Translation Features Roadmap."

Cisco IOS software images are specific to a Cisco IOS software release, a feature set, and a platform. Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for Integrating NAT with MPLS VPNs

Feature Name	Releases	Feature Configuration Information
Network Address Translation (NAT) Integration with MPLS VPNs feature	12.1(13)T	This feature allows multiple Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs) to be configured on a single device to work together. The following sections provide information about this feature: • "Information About Integrating NAT with MPLS VPNs" section on page 2 • "How to Integrate NAT with MPLS VPNs" section on page 3

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport

Monitoring and Maintaining NAT

This module describes how to:

- Monitor Network Address Translation (NAT) using translation information and statistics displays.
- Maintain NAT by clearing NAT translations before the timeout has expired.
- Enable logging of NAT translation by way of syslog to log and track system error messages, exceptions, and other information.

Module History

This module was first published on May 2, 2005, and last updated on May 2, 2005.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all features. To find information about feature support and configuration, use the "Feature Information for Monitoring and Maintaining NAT" section on page 9.

Contents

Prerequisites for Monitoring and Maintaining NAT

Before performing the tasks in the module, you should be familiar with the concepts described in the "Configuring NAT for IP Address Conservation" module and have NAT configured.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Information About Monitoring and Maintaining NAT

Before performing the tasks in this module, you should understand the following concepts:

- NAT Display Contents, page 2
- Syslog Usage, page 3

NAT Display Contents

There are two basic types of IP NAT translation information: translation entries and statistics.

Translation Entries

Translation entry information includes the following:

- The protocol of the port identifying the address.
- The legitimate IP address that represents one or more inside local IP addresses to the outside world.
- The IP address assigned to a host on the inside network; probably not a legitimate address assigned by the NIC or service provider.
- The IP address of an outside host as it appears to the inside network; probably not a legitimate address assigned by the NIC or service provider.
- The IP address assigned to a host on the outside network by its owner.
- The time since the entry was created (in hours:minutes:seconds).
- The time since the entry was last used (in hours:minutes:seconds).
- Flags indicating the type of translation. Possible flags are:
  - extended—Extended translation
  - static—Static translation
  - destination—Rotary translation
  - outside—Outside translation
  - timing out—Translation will no longer be used, due to a TCP finish (FIN) or reset (RST) flag.

Statistical Information

Statistical information includes the following:

- The total number of translations active in the system. This number is incremented each time a translation is created and is decremented each time a translation is cleared or times out.
- A list of interfaces marked as outside with the **ip nat outside** command.
- A list of interfaces marked as inside with the **ip nat inside** command.
- The number of times the software does a translations table lookup and finds an entry.
- The number of times the software does a translations table lookup, fails to find an entry, and must try to create one.
- A cumulative count of translations that have expired since the router was booted.
- Information about dynamic mappings.

- Information about an inside source translation.
- The access list number being used for the translation.
- The name of the pool.
- The number of translations using this pool.
- The IP network mask being used in the pool.
- The starting IP address in the pool range.
- The ending IP address in the pool range.
- The type of pool. Possible types are generic or rotary.
- The number of addresses in the pool available for translation.
- The number of addresses being used.
- The number of failed allocations from the pool.

Syslog Usage

Syslog Analysis lets you centrally log and track system error messages, exceptions, and other information (such as device configuration changes). You can use the logged error message data to analyze router and network performance. You can customize Syslog Analysis to produce the information and message reports important to your operation.

For more information see the *Resource Manager Essentials and Syslog Analysis: How-To* document*:*

http://www.cisco.com/warp/public/477/RME/rme_syslog.html

How to Monitor and Maintain NAT

This section contains the following procedures:

- Displaying NAT Translation Information, page 3 (optional)
- Clearing NAT Entries Before the Timeout, page 5 (optional)
- Enabling Syslog for Logging NAT Translations, page 6 (optional)

Displaying NAT Translation Information

Perform this task to display: translation data and statistical information.

SUMMARY STEPS

1. **enable**
2. **show ip nat translations** [**verbose**]
3. **show ip nat statistics**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip nat translations** [**verbose**] **Example:** `Router> show ip nat translations`	(Optional) Displays active NAT translations.
Step 3	**show ip nat statistics** **Example:** `Router> show ip nat statistics`	(Optional) Displays active NAT translation statistics.

Displaying NAT Translation Information: Examples

This section contains the following examples:

- Displaying NAT Translations, page 4
- Displaying NAT Statistics, page 5

Displaying NAT Translations

The following is sample output from the **show ip nat translations** command. Without overloading, two inside hosts are exchanging packets with some number of outside hosts.

```
Router# show ip nat translations

Pro Inside global      Inside local       Outside local      Outside global

--- 171.69.233.209     192.168.1.95       ---                ---

--- 171.69.233.210     192.168.1.89       ---                --
```

With overloading, a translation for a Domain Name Server (DNS) transaction is still active, and translations for two Telnet sessions (from two different hosts) are also active. Note that two different inside hosts appear on the outside with a single IP address.

```
Router# show ip nat translations

Pro Inside global        Inside local       Outside local      Outside global

udp 171.69.233.209:1220  192.168.1.95:1220  171.69.2.132:53    171.69.2.132:53

tcp 171.69.233.209:11012 192.168.1.89:11012 171.69.1.220:23    171.69.1.220:23

tcp 171.69.233.209:1067  192.168.1.95:1067  171.69.1.161:23    171.69.1.161:23
```

The following is sample output that includes the **verbose** keyword:

```
Router# show ip nat translations verbose

Pro Inside global        Inside local       Outside local      Outside global

udp 171.69.233.209:1220  192.168.1.95:1220  171.69.2.132:53    171.69.2.132:53

        create 00:00:02, use 00:00:00, flags: extended

tcp 171.69.233.209:11012 192.168.1.89:11012 171.69.1.220:23    171.69.1.220:23

        create 00:01:13, use 00:00:50, flags: extended

tcp 171.69.233.209:1067  192.168.1.95:1067  171.69.1.161:23    171.69.1.161:23

        create 00:00:02, use 00:00:00, flags: extended
```

Displaying NAT Statistics

The following is sample output from the **show ip nat statistics** command:

```
Router# show ip nat statistics
Total translations: 2 (0 static, 2 dynamic; 0 extended)
Outside interfaces: Serial0
Inside interfaces: Ethernet1
Hits: 135  Misses: 5
Expired translations: 2
Dynamic mappings:
-- Inside Source
access-list 1 pool net-208 refcount 2
 pool net-208: netmask 255.255.255.240
        start 171.69.233.208 end 171.69.233.221
        type generic, total addresses 14, allocated 2 (14%), misses 0
```

Clearing NAT Entries Before the Timeout

By default, dynamic address translations will time out from the NAT translation table at some point. Perform this task to clear the entries before the timeout.

SUMMARY STEPS

1. **enable**
2. **clear ip nat translation inside** *global-ip local-ip* [**outside** *local-ip global-ip*]
3. **clear ip nat translation outside** *global-ip local-ip*
4. **clear ip nat translation** ***protocol*** **inside** ***global-ip global-port local-ip local-port*** **[outside** ***local-ip local-port-global-ip global-port***]
5. **clear ip nat translation** {* | [**forced**] | [**inside** *global-ip local-ip*] [**outside** *local-ip global-ip*]}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear ip nat translation inside** *global-ip* *local-ip* [**outside** *local-ip global-ip*] **Example:** `Router# clear ip nat translation udp inside 171.69.233.209 1220 192.168.1.95 1220 171.69.2.132 53 171.69.2.132 53`	(Optional) Clears a simple dynamic translation entry containing an inside translation, or both inside and outside translation.
Step 3	**clear ip nat translation outside** *global-ip* *local-i***p** **Example:** `Router# clear ip nat translation outside 171.69.233.209 1220 192.168.1.95`	(Optional) Clears a simple dynamic translation entry containing an outside translation.
Step 4	**clear ip nat translation** ***protocol*** **inside** ***global-ip global-port local-ip local-p[ort*** [**outside** ***local-ip local-port-global-ip global-port***] **Example:** `clear ip nat translation udp inside 171.69.233.209 1220 192.168.1.95 1220 171.69.2.132 53 171.69.2.132 53`	(Optional) Clears a UDP translation entry.
Step 5	**clear ip nat translation** {* | [**forced**] | [**inside** *global-ip local-ip*] [**outside** *local-ip global-ip*]} **Example:** `Router# clear ip nat translation *`	(Optional) Clears all dynamic translations.

Enabling Syslog for Logging NAT Translations

The logging of NAT translations can be enabled and disabled by way of the **syslog** command.

Syslog Analysis lets you centrally log and track system error messages, exceptions, and other information (such as NAT translations). You can use the logged error message data to analyze router and network performance. You can customize Syslog Analysis to produce the information and message reports important to your operation.

Prerequisites

Prior to performing this task, you must specify the necessary **syslog** commands such as making sure that logging is enabled, configuring the server's IP address, and establishing the level of messages to be trapped.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat log translations syslog**
4. **no logging console** (optional)

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`configure terminal` **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	`ip nat log translations syslog` **Example:** `Router(config)# ip nat log translations syslog`	Enables the syslog for logging NAT translations.
Step 4	`no logging console` **Example:** `Router(config)# no logging console`	(Optional) Disables the log display to the console. • Logging to the console is enable by default.

Examples for Monitoring and Maintaining NAT

- Clearing UDP NAT Translations: Example, page 8
- Enabling Syslog: Example, page 8

Clearing UDP NAT Translations: Example

The following example shows the NAT entries before and after the User Datagram Protocol (UDP) entry is cleared:

```
Router# show ip nat translation

Pro Inside global      Inside local       Outside local      Outside global
udp 171.69.233.209:1220 192.168.1.95:1220 171.69.2.132:53    171.69.2.132:53
tcp 171.69.233.209:11012 192.168.1.89:11012 171.69.1.220:23  171.69.1.220:23
tcp 171.69.233.209:1067 192.168.1.95:1067 171.69.1.161:23    171.69.1.161:23

Router# clear ip nat translation udp inside 171.69.233.209 1220 192.168.1.95 1220
171.69.2.132 53 171.69.2.132 53

Router# show ip nat translation

Pro Inside global      Inside local       Outside local      Outside global
tcp 171.69.233.209:11012 192.168.1.89:11012 171.69.1.220:23  171.69.1.220:23
tcp 171.69.233.209:1067 192.168.1.95:1067 171.69.1.161:23    171.69.1.161:23
```

Enabling Syslog: Example

The following example shows enabling NAT entries into syslog.

```
Router(config)# logging on
Router(config)# logging 1.1.1.1
Router(config)# logging trap informational
Router(Config)# ip nat log translations syslog
```

The format of NAT information logged (for example, for ICMP Ping via NAT Overload configurations) will be as follows:

```
Apr 25 11:51:29 [10.0.19.182.204.28] 1: 00:01:13: NAT:Created icmp
135.135.5.2:7 171 12.106.151.30:7171 54.45.54.45:7171
54.45.54.45:7171
Apr 25 11:52:31 [10.0.19.182.204.28] 8: 00:02:15: NAT:Deleted icmp
135.135.5.2:7 172 12.106.151.30:7172 54.45.54.45:7172
54.45.54.45:7172
```

Where to Go Next

- To configure NAT for use with application level gateways, see the “Using Application Level Gateways with NAT” module.
- To integrate NAT with MPLS VPNs, see the “Integrating NAT with MPLS VPNs” module.
- To configure NAT for high availability, see the “Configuring NAT for High Availability” module.

Additional References

The following sections provide references related to Monitoring and Maintaining NAT.

Related Documents

Related Topic	Document Title
NAT commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*"IP Addressing Commands*" chapter in the *Cisco IOS IP Command Reference, Volume 1 of 3: Addressing and Services,* Release 12.3.

Standards

Standards	Title
None	

MIBs

MIBs	MIBs Link
• None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport

Feature Information for Monitoring and Maintaining NAT

Table 1 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in IOS Release 12.2(1) or later appear in the table.

Not all commands may be available in your Cisco IOS software release. For details on when support for a specific command was introduced, see the command reference documentation.

For information on a feature in this technology that is not documented here, see the "Configuring Network Address Translation Features Roadmap."

Cisco IOS software images are specific to a Cisco IOS software release, a feature set, and a platform. Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for Monitoring and Maintaining NAT

Feature Name	Releases	Feature Configuration Information
This table is intentionally left blank because no features were introduced or modified in Cisco IOS Release 12.2(1) or later. This table will be updated when feature information is added to this module.	—	—

This module first published May 2, 2005. Last updated May 2, 2005