



IP Application Services Configuration Guide, Cisco IOS Release 15M&T

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

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

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

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: [www.cisco.com go trademarks](http://www.cisco.com/go/trademarks). Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2019 Cisco Systems, Inc. All rights reserved.



CONTENTS

CHAPTER 1

Configuring Enhanced Object Tracking 1

| | |
|-----------------------------------------------------|----|
| Finding Feature Information | 1 |
| Restrictions for Enhanced Object Tracking | 2 |
| Information About Enhanced Object Tracking | 2 |
| Feature Design of Enhanced Object Tracking | 2 |
| Interface State Tracking | 2 |
| Scaled Route Metrics | 3 |
| IP SLA Operation Tracking | 4 |
| Enhanced Object Tracking and Embedded Event Manager | 4 |
| EOT Support for Carrier Delay | 5 |
| Enhanced Object Tracking for Mobile IP Applications | 5 |
| Benefits of Enhanced Object Tracking | 5 |
| How to Configure Enhanced Object Tracking | 6 |
| Tracking the Line-Protocol State of an Interface | 6 |
| Tracking the IP-Routing State of an Interface | 7 |
| Tracking IP-Route Reachability | 9 |
| Tracking the Threshold of IP-Route Metrics | 11 |
| Tracking the State of an IP SLAs Operation | 13 |
| Tracking the Reachability of an IP SLAs IP Host | 14 |
| Configuring a Tracked List and Boolean Expression | 15 |
| Configuring a Tracked List and Threshold Weight | 17 |
| Configuring a Tracked List and Threshold Percentage | 18 |
| Configuring Track List Defaults | 20 |
| Configuring Tracking for Mobile IP Applications | 21 |
| Configuration Examples for Enhanced Object Tracking | 22 |
| Example: Interface Line Protocol | 22 |

| | |
|--------------------------------------------------|----|
| Example: Interface IP Routing | 23 |
| Example: IP-Route Reachability | 23 |
| Example: IP-Route Threshold Metric | 24 |
| Example: IP SLAs IP Host Tracking | 24 |
| Example: Boolean Expression for a Tracked List | 25 |
| Example: Threshold Weight for a Tracked List | 26 |
| Example: Threshold Percentage for a Tracked List | 26 |
| Example: Mobile IP Application Tracking | 27 |
| Additional References | 27 |
| Feature Information for Enhanced Object Tracking | 28 |
| Glossary | 30 |

CHAPTER 2
Configuring IP Services 33

| | |
|------------------------------------------------------------|----|
| Finding Feature Information | 33 |
| Information About IP Services | 33 |
| IP Source Routing | 33 |
| ICMP Overview | 34 |
| ICMP Unreachable Error Messages | 34 |
| ICMP Mask Reply Messages | 35 |
| ICMP Redirect Messages | 35 |
| Denial of Service Attack | 35 |
| Path MTU Discovery | 36 |
| Cisco IP Accounting | 37 |
| Show and Clear Commands for IOS Sockets | 37 |
| How to Configure IP Services | 38 |
| Protecting Your Network from DOS Attacks | 38 |
| Configuring ICMP Unreachable Rate Limiting User Feedback | 39 |
| Setting the MTU Packet Size | 41 |
| Configuring IP Accounting | 42 |
| Monitoring and Maintaining the IP Network | 43 |
| Configuration Examples for IP Services | 49 |
| Example: Protecting Your Network from DOS Attacks | 49 |
| Example: Configuring ICMP Unreachable Destination Counters | 49 |
| Example: Setting the MTU Packet Size | 50 |

| | |
|---------------------------------------|----|
| Example: Configuring IP Accounting | 50 |
| Additional References For IP Services | 50 |
| Feature Information for IP Services | 51 |

CHAPTER 3
Configuring IPv4 Broadcast Packet Handling 53

| | |
|------------------------------------------------------------------------------------------------------------------|----|
| Finding Feature Information | 53 |
| Information About IPv4 Broadcast Packet Handling | 53 |
| IP Unicast Address | 53 |
| IP Broadcast Address | 54 |
| IP Directed Broadcast Address | 54 |
| IP Directed Broadcasts | 55 |
| IP Multicast Addresses | 55 |
| Early IP Implementations | 56 |
| DHCP and IPv4 Broadcast Packets | 56 |
| UDP Broadcast Packet Forwarding | 56 |
| UDP Broadcast Packet Flooding | 57 |
| IP Broadcast Flooding Acceleration | 57 |
| Default UDP Port Numbers | 58 |
| Default IP Broadcast Address | 58 |
| UDP Broadcast Packet Case Study | 58 |
| UDP Broadcast Packet Forwarding | 59 |
| UDP Broadcast Packet Flooding | 61 |
| How to Configure IP Broadcast Packet Handling | 64 |
| Enabling IP Directed Broadcasts Without an Access List | 64 |
| Enabling IP Directed Broadcasts with an Access List | 65 |
| Enabling Forwarding of UDP Broadcast Packets to a Specific Host | 66 |
| Enabling Forwarding of UDP Broadcast Packets to a Range of Hosts | 67 |
| Changing the Default IP Broadcast Address for All Interfaces to 0.0.0.0 on Routers Without Nonvolatile Memory | 69 |
| Changing the Default IP Broadcast Address for All Interfaces to 0.0.0.0 on Routers with Nonvolatile Memory | 70 |
| Changing the IP Broadcast Address to Any IP Address on One or More Interfaces in a Router | 71 |
| Configuring UDP Broadcast Packet Flooding | 72 |
| Configuration Examples for IP Broadcast Packet Handling | 74 |

| | |
|--------------------------------------------------------------|----|
| Example: Enabling IP Directed Broadcasts with an Access List | 74 |
| Example: Configuring UDP Broadcast Packet Flooding | 74 |
| Additional References for WCCP—Configurable Router ID | 75 |
| Feature Information for IP Broadcast Packet Handling | 76 |

CHAPTER 4**Configuring TCP 77**

| | |
|-----------------------------------------------------------------|----|
| Finding Feature Information | 77 |
| Prerequisites for TCP | 77 |
| Restrictions for TCP | 78 |
| Information About TCP | 78 |
| TCP Services | 78 |
| TCP Connection Establishment | 78 |
| TCP Connection Attempt Time | 79 |
| TCP Selective Acknowledgment | 79 |
| TCP Time Stamp | 79 |
| TCP Maximum Read Size | 80 |
| TCP Path MTU Discovery | 80 |
| TCP Window Scaling | 80 |
| TCP Sliding Window | 81 |
| TCP Outgoing Queue Size | 81 |
| TCP Congestion Avoidance | 81 |
| TCP Explicit Congestion Notification | 82 |
| TCP MSS Adjustment | 82 |
| TCP Applications Flags Enhancement | 82 |
| TCP Show Extension | 83 |
| TCP MIB for RFC 4022 Support | 83 |
| TCP Keepalive Timer | 83 |
| How to Configure TCP | 83 |
| Configuring TCP Performance Parameters | 83 |
| Configuring the MSS Value and MTU for Transient TCP SYN Packets | 85 |
| Configuring the MSS Value for IPv6 Traffic | 86 |
| Verifying TCP Performance Parameters | 87 |
| Configuring Keepalive Parameters | 91 |
| Configuration Examples for TCP | 92 |

| | |
|------------------------------------------------------------|----|
| Example: Verifying the Configuration of TCP ECN | 92 |
| Example: Configuring the TCP MSS Adjustment | 94 |
| Example: Configuring the TCP Application Flags Enhancement | 95 |
| Example: Displaying Addresses in IP Format | 95 |
| Example: Configuring Keepalive Parameters | 96 |
| Additional References | 96 |
| Feature Information for TCP | 97 |

CHAPTER 5**Configuring UDP Forwarding Support for IP Redundancy Virtual Router Groups 101**

| | |
|-------------------------------------------------------------------------------------------|-----|
| Finding Feature Information | 101 |
| Prerequisites for UDP Forwarding Support for IP Redundancy Virtual Router Groups | 102 |
| Information About UDP Forwarding Support for IP Redundancy Virtual Router Groups | 102 |
| Benefits of the UDP Forwarding Support for Virtual Router Groups Feature | 102 |
| How to Configure UDP Forwarding Support for IP Redundancy Virtual Router Groups | 102 |
| Configuring UDP Forwarding Support for IP Redundancy Virtual Router Groups | 102 |
| Configuration Examples for UDP Forwarding Support for IP Redundancy Virtual Router Groups | 104 |
| Example: Configuring UDP Forwarding Support for IP Redundancy Virtual Router Groups | 104 |
| Additional References | 104 |
| Feature Information for UDP Forwarding Support for IP Redundancy Virtual Router Groups | 105 |

CHAPTER 6**Configuring WCCP 107**

| | |
|---------------------------------------------|-----|
| Finding Feature Information | 107 |
| Prerequisites for WCCP | 108 |
| Restrictions for WCCP | 108 |
| Information About WCCP | 109 |
| WCCP Overview | 109 |
| Layer 2 Forwarding Redirection and Return | 110 |
| WCCP Mask Assignment | 110 |
| Hardware Acceleration | 111 |
| WCCPv1 Configuration | 111 |
| WCCPv2 Configuration | 112 |
| WCCPv2 Support for Services Other Than HTTP | 114 |
| WCCPv2 Support for Multiple Routers | 114 |
| WCCPv2 MD5 Security | 114 |

| | |
|--------------------------------------------------------------|-----|
| WCCPv2 Web Cache Packet Return | 114 |
| WCCPv2 Load Distribution | 115 |
| WCCP VRF Support | 115 |
| WCCP VRF Tunnel Interfaces | 115 |
| WCCP Bypass Packets | 118 |
| WCCP Closed Services and Open Services | 118 |
| WCCP Outbound ACL Check | 118 |
| WCCP Service Groups | 119 |
| WCCP—Check All Services | 120 |
| WCCP Interoperability with NAT | 120 |
| WCCP Troubleshooting Tips | 120 |
| How to Configure WCCP | 121 |
| Configuring WCCP | 121 |
| Configuring Closed Services | 123 |
| Registering a Router to a Multicast Address | 124 |
| Using Access Lists for a WCCP Service Group | 125 |
| Enabling the WCCP Outbound ACL Check | 127 |
| Enabling WCCP Interoperability with NAT | 128 |
| Verifying and Monitoring WCCP Configuration Settings | 130 |
| Configuration Examples for WCCP | 131 |
| Example: Changing the Version of WCCP on a Router | 131 |
| Example: Configuring a General WCCPv2 Session | 132 |
| Example: Setting a Password for a Router and Content Engines | 132 |
| Example: Configuring a Web Cache Service | 132 |
| Example: Running a Reverse Proxy Service | 133 |
| Example: Registering a Router to a Multicast Address | 133 |
| Example: Using Access Lists | 133 |
| Example: WCCP Outbound ACL Check Configuration | 134 |
| Example: Verifying WCCP Settings | 134 |
| Example: Enabling WCCP Interoperability with NAT | 136 |
| Additional References | 136 |
| Feature Information for WCCP | 138 |

| | |
|-----------------------------------------------------------------|-----|
| Finding Feature Information | 141 |
| Restrictions for Object Tracking: IPv6 Route Tracking | 141 |
| Information About Object Tracking: IPv6 Route Tracking | 142 |
| Enhanced Object Tracking and IPv6 Route Tracking | 142 |
| How to Configure Object Tracking: IPv6 Route Tracking | 142 |
| Tracking the IPv6-Routing State of an Interface | 142 |
| Tracking the Threshold of IPv6-Route Metrics | 143 |
| Tracking IPv6-Route Reachability | 145 |
| Configuration Examples for Object Tracking: IPv6 Route Tracking | 146 |
| Example: Tracking the IPv6-Routing State of an Interface | 146 |
| Example: Tracking the Threshold of IPv6-Route Metrics | 147 |
| Example: Tracking IPv6-Route Reachability | 147 |
| Additional References for Object Tracking: IPv6 Route Tracking | 147 |
| Feature Information for Object Tracking: IPv6 Route Tracking | 148 |

CHAPTER 8
IPv6 Static Route Support for Object Tracking 149

| | |
|--------------------------------------------------------------------------|-----|
| Finding Feature Information | 149 |
| Information About IPv6 Static Route Support for Object Tracking | 149 |
| IPv6 Static Route Support for Object Tracking Overview | 149 |
| Routing Table Insertion | 150 |
| Routing Table Insertion Criteria | 150 |
| How to Configure IPv6 Static Route Support for Object Tracking | 150 |
| Configuring the IPv6 Static Routing Support for Object Tracking | 150 |
| Configuration Examples for IPv6 Static Route Support for Object Tracking | 152 |
| Example: IPv6 Static Route Object Tracking | 152 |
| Additional References for IPv6 Static Route Support for Object Tracking | 152 |
| Feature Information for IPv6 Static Route Support for Object Tracking | 153 |

CHAPTER 9
Stream Control Transmission Protocol 155

| | |
|-----------------------------|-----|
| Finding Feature Information | 155 |
| Prerequisites for SCTP | 156 |
| Information About SCTP | 156 |
| Benefits of SCTP | 156 |
| SCTP Release 2 | 156 |

| | |
|---------------------------------------------------------------|-----|
| SCTP Show Clear CLI Enhancements | 156 |
| SCTP Stream Reset | 157 |
| SCTP Authentication | 157 |
| SCTP Add-IP | 158 |
| How to Configure SCTP | 158 |
| Configuring SCTP Authentication Parameters | 159 |
| Displaying Information About SCTP Associations and Parameters | 159 |
| Troubleshooting SCTP Associations and Parameters | 163 |
| Configuration Examples for SCTP | 174 |
| Example: Defining SCTP Authentication Parameters | 174 |
| Additional References | 175 |
| Feature Information for SCTP | 176 |

CHAPTER 10**WCCP—Configurable Router ID 179**

| | |
|--------------------------------------------------------|-----|
| Finding Feature Information | 179 |
| Restrictions for WCCP—Configurable Router ID | 179 |
| Information About WCCP—Configurable Router ID | 180 |
| WCCP—Configurable Router ID Overview | 180 |
| How to Configure WCCP—Configurable Router ID | 180 |
| Configuring a Preferred WCCP Router ID | 180 |
| Configuration Examples for WCCP—Configurable Router ID | 181 |
| Example: Configuring a Preferred WCCP Router ID | 181 |
| Additional References for WCCP—Configurable Router ID | 181 |
| Feature Information for WCCP—Configurable Router ID | 182 |

CHAPTER 11**WCCP—Fast Timers 185**

| | |
|--------------------------------------------------|-----|
| Finding Feature Information | 185 |
| Information About WCCP—Fast Timers | 185 |
| WCCP—Fast Timers Overview | 185 |
| How to Configure WCCP—Fast Timers | 186 |
| Displaying WCCP—Fast Timers Information | 186 |
| Configuration Examples for WCCP—Fast Timers | 187 |
| Example: Displaying WCCP-Fast Timers Information | 187 |
| Additional References For IP Services | 187 |

Feature Information for WCCP—Fast Timers 188

CHAPTER 12

WCCPv2—IPv6 Support 189

- Finding Feature Information 189
- Prerequisites for WCCPv2—IPv6 Support 189
- Restrictions for WCCPv2—IPv6 Support 190
- Information About WCCPv2—IPv6 Support 190
 - WCCP Overview 190
 - Layer 2 Forwarding Redirection and Return 191
 - WCCP Mask Assignment 191
 - WCCP Hash Assignment 192
 - WCCPv2 Configuration 192
 - WCCPv2 Support for Services Other Than HTTP 193
 - WCCPv2 Support for Multiple Routers 193
 - WCCPv2 MD5 Security 193
 - WCCPv2 Web Cache Packet Return 194
 - WCCPv2 Load Distribution 194
 - WCCP VRF Support 194
 - IPv6 WCCP Tunnel Interface 195
 - WCCP Bypass Packets 197
 - WCCP Closed Services and Open Services 197
 - WCCP Outbound ACL Check 198
 - WCCP Service Groups 198
 - WCCP—Check All Services 199
 - WCCP Interoperability with NAT 200
 - WCCP—Configurable Router ID Overview 200
 - WCCP Troubleshooting Tips 200
- How to Configure WCCPv2—IPv6 Support 201
 - Configuring a General WCCPv2—IPv6 Session 201
 - Configuring Services for WCCPv2—IPv6 202
 - Registering a Router to a Multicast Address for WCCPv2— IPv6 204
 - Using Access Lists for WCCPv2—IPv6 Service Group 205
 - Enabling the WCCP—IPv6 Outbound ACL Check 207
 - Enabling WCCPv2—IPv6 Interoperability with NAT 208

| | |
|--------------------------------------------------------------------------|-----|
| Verifying and Monitoring WCCPv2—IPv6 Configuration Settings | 210 |
| Configuration Examples for WCCPv2—IPv6 Support | 211 |
| Example: Configuring a General WCCPv2—IPv6 Session | 211 |
| Example: WCCPv2—IPv6—Setting a Password for a Router and Content Engines | 212 |
| Example: WCCPv2—IPv6—Configuring a Web Cache Service | 212 |
| Example: WCCPv2—IPv6—Running a Reverse Proxy Service | 212 |
| Example: WCCPv2—IPv6—Registering a Router to a Multicast Address | 212 |
| Example: WCCPv2—IPv6—Using Access Lists for a WCCPv2 IPv6 Service Group | 213 |
| Example: WCCPv2—IPv6—Configuring Outbound ACL Check | 213 |
| Example: WCCPv2—IPv6—Enabling WCCP Interoperability with NAT | 214 |
| Example: WCCPv2—IPv6—Verifying WCCP Settings | 214 |
| Additional References | 216 |
| Feature Information for WCCPv2—IPv6 Support | 216 |



CHAPTER

1

Configuring Enhanced Object Tracking

Before the introduction of the Enhanced Object Tracking feature, the Hot Standby Router Protocol (HSRP) had a simple tracking mechanism that allowed you to track the interface line-protocol state only. If the line-protocol state of the interface went down, the HSRP priority of the router was reduced, allowing another HSRP router with a higher priority to become active.

The Enhanced Object Tracking feature separates the tracking mechanism from HSRP and creates a separate standalone tracking process that can be used by other processes and HSRP. This feature allows tracking of other objects in addition to the interface line-protocol state.

A client process such as HSRP, Virtual Router Redundancy Protocol (VRRP), or Gateway Load Balancing Protocol (GLBP), can register its interest in tracking objects and then be notified when the tracked object changes state.

- [Finding Feature Information, on page 1](#)
- [Restrictions for Enhanced Object Tracking, on page 2](#)
- [Information About Enhanced Object Tracking, on page 2](#)
- [How to Configure Enhanced Object Tracking, on page 6](#)
- [Configuration Examples for Enhanced Object Tracking, on page 22](#)
- [Additional References, on page 27](#)
- [Feature Information for Enhanced Object Tracking, on page 28](#)
- [Glossary, on page 30](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

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

Restrictions for Enhanced Object Tracking

Enhanced Object Tracking is not stateful switchover (SSO)-aware and cannot be used with Hot Standby Routing Protocol (HSRP), Virtual Router Redundancy Protocol (VRRP), or Gateway Load Balancing Protocol (GLBP) in SSO mode.

Information About Enhanced Object Tracking

Feature Design of Enhanced Object Tracking

The Enhanced Object Tracking feature provides complete separation between the objects to be tracked and the action to be taken by a client when a tracked object changes. Thus, several clients such as HSRP, VRRP, or GLPB can register their interest with the tracking process, track the same object, and each take different action when the object changes.

Each tracked object is identified by a unique number that is specified on the tracking CLI. Client processes use this number to track a specific object.

The tracking process periodically polls the tracked objects and notes any change of value. The changes in the tracked object are communicated to interested client processes, either immediately or after a specified delay. The object values are reported as either up or down.

You can configure a combination of tracked objects in a list and a flexible method for combining objects using Boolean logic. This functionality includes the following capabilities:

- **Threshold**—The tracked list can be configured to use a weight or percentage threshold to measure the state of the list. Each object in a tracked list can be assigned a threshold weight. The state of the tracked list is determined by whether the threshold has been met.
- **Boolean "and" function**—When a tracked list has been assigned a Boolean "and" function, each object defined within a subset must be in an up state so that the tracked object can become up.
- **Boolean "or" function**—When the tracked list has been assigned a Boolean "or" function, at least one object defined within a subset must be in an up state so that the tracked object can become up.

With CSCtg75700, a maximum of 1000 objects can be tracked. Although 1000 tracked objects can be configured, each tracked object uses CPU resources. The amount of available CPU resources on a router depends on variables such as traffic load and how other protocols are configured and run. The ability to use 1000 tracked objects depends on the available CPU. Testing should be conducted on site to ensure that the service works under the specific site traffic conditions.

Interface State Tracking

An IP-routing object is considered up when the following criteria exist:

- IP routing is enabled and active on the interface.
- The interface line-protocol state is up.

- The interface IP address is known. The IP address is configured or received through Dynamic Host Configuration Protocol (DHCP) or IP Control Protocol (IPCP) negotiation.

Interface IP routing will go down when one of the following criteria exists:

- IP routing is disabled globally.
- The interface line-protocol state is down.
- The interface IP address is unknown. The IP address is not configured or received through DHCP or IPCP negotiation.

Tracking the IP-routing state of an interface using the **track interface ip routing** command can be more useful in some situations than just tracking the line-protocol state using the **track interface line-protocol** command, especially on interfaces for which IP addresses are negotiated. For example, on a serial interface that uses the PPP, the line protocol could be up (link control protocol [LCP] negotiated successfully), but IP could be down (IPCP negotiation failed).

The **track interface ip routing** command supports the tracking of an interface with an IP address acquired through any of the following methods:

- Conventional IP address configuration
- PPP/IPCP
- DHCP
- Unnumbered interface

You can configure Enhanced Object Tracking to consider the carrier-delay timer when tracking the IP-routing state of an interface by using the **carrier-delay** command in tracking configuration mode.

Scaled Route Metrics

The **track ip route** command enables tracking of a route in the routing table. If a route exists in the table, the metric value is converted into a number. To provide a common interface to tracking clients, normalize route metric values to the range from 0 to 255, where 0 is connected and 255 is inaccessible. Scaled metrics can be tracked by setting thresholds. Up and down state notification occurs when the thresholds are crossed. The resulting value is compared against threshold values to determine the tracking state as follows:

- State is up if the scaled metric for that route is less than or equal to the up threshold.
- State is down if the scaled metric for that route is greater than or equal to the down threshold.

Tracking uses a per-protocol configurable resolution value to convert the real metric to the scaled metric. The table below shows the default values used for the conversion. You can use the **track resolution** command to change the metric resolution default values.

Table 1: Metric Conversion

| Route Type ¹ | Metric Resolution |
|----------------------------------------------------|-------------------|
| Static | 10 |
| Enhanced Interior Gateway Routing Protocol (EIGRP) | 2560 |

| Route Type ¹ | Metric Resolution |
|----------------------------------------------------|-------------------|
| Open Shortest Path First (OSPF) | 1 |
| Intermediate System-to-Intermediate System (IS-IS) | 10 |

¹ RIP is scaled directly to the range from 0 to 255 because its maximum metric is less than 255.

For example, a change in 10 in an IS-IS metric results in a change of 1 in the scaled metric. The default resolutions are designed so that approximately one 2-Mbps link in the path will give a scaled metric of 255.

Scaling the very large metric ranges of EIGRP and IS-IS to a 0 to 255 range is a compromise. The default resolutions will cause the scaled metric to exceed the maximum limit with a 2-Mb/s link. However, this scaling allows a distinction between a route consisting of three Fast-Ethernet links and a route consisting of four Fast-Ethernet links.

IP SLA Operation Tracking

Object tracking of IP Service Level Agreements (SLAs) operations allows tracking clients to track the output from IP SLAs objects and use the provided information to trigger an action.

Cisco IOS IP SLAs is a network performance measurement and diagnostics tool that uses active monitoring. Active monitoring is the generation of traffic in a reliable and predictable manner to measure network performance. software uses IP SLAs to collect real-time metrics such as response time, network resource availability, application performance, jitter (interpacket delay variance), connect time, throughput, and packet loss.

These metrics can be used for troubleshooting, for proactive analysis before problems occur, and for designing network topologies.

Every IP SLAs operation maintains an operation return-code value. This return code is interpreted by the tracking process. The return code can return OK, OverThreshold, and several other return codes. Different operations can have different return-code values, so only values common to all operation types are used.

Two aspects of an IP SLAs operation can be tracked: state and reachability. The difference between these aspects is the acceptance of the OverThreshold return code. The table below shows the state and reachability aspects of IP SLAs operations that can be tracked.

Table 2: Comparison of State and Reachability Operations

| Tracking | Return Code | Track State |
|--------------|--------------------------|-------------|
| State | OK | Up |
| | (all other return codes) | Down |
| Reachability | OK or OverThreshold | Up |
| | (all other return codes) | Down |

Enhanced Object Tracking and Embedded Event Manager

Enhanced Object Tracking (EOT) is now integrated with Embedded Event Manager (EEM) to allow EEM to report on status change of a tracked object and to allow EOT to track EEM objects. A new type of tracking

object--a stub object--is created. The stub object can be modified by an external process through a defined Application Programming Interface (API). See the Embedded Event Manager Overview document in the *Network Management Configuration Guide* for more information on how EOT works with EEM.

EOT Support for Carrier Delay

The EOT Support for Carrier Delay feature enables Enhanced Object Tracking (EOT) to consider the carrier-delay timer when tracking the status of an interface.

If a link fails, by default there is a two-second timer that must expire before an interface and the associated routes are declared as being down. If a link goes down and comes back up before the carrier delay timer expires, the down state is effectively filtered, and the rest of the software on the switch is not aware that a link-down event occurred. You can configure the carrier-delay seconds command in interface configuration mode to extend the timer up to 60 seconds.

When EOT is configured on an interface, the tracking may detect the interface is down before a configured carrier-delay timer has expired. This is because EOT looks at the interface state and does not consider the carrier delay timer. Use the **carrier-delay** command in tracking configuration mode to enable tracking to consider the carrier-delay timer configured on an interface.

Enhanced Object Tracking for Mobile IP Applications

The Enhanced Object Tracking Support for Mobile IP feature enables EOT to monitor the presence of Home Agent, Packet Data Serving Node (PDSN), or Gateway GPRS Support Node (GGSN) traffic on a router for mobile wireless applications.

When a redundant pair of Home Agents running HSRP between them loses connectivity, both HSRP nodes become active. Once the connectivity is restored between the two nodes, a graceful way is needed to restore proper HSRP states without losing Home Agent bindings. During the time of no connectivity, one of the nodes will continue to process Home Agent, GGSN, or PDSN traffic while the other will not. The node that continues to process traffic needs to remain active once connectivity is restored. To ensure that the active node remains in the active state, the priority of the HSRP group member that does not process Home Agent traffic is reduced. Reducing the priority of the node that is not processing Home Agent traffic ensures that this node will become the standby after connectivity is restored. When connectivity is restored, the normal Home Agent state synchronization will get all bindings back into the inactive node and, depending on the preempt configuration, it may switch over again. This state synchronization ensures that no Mobile IP, GGSN, or PDSN bindings are lost.

For more information on configuring Mobile IP services, see the following Cisco IOS configuration guides:

- *Cisco IOS Mobile Wireless Home Agent Configuration Guide*
- *Cisco IOS Mobile Wireless Gateway GPRS Support Node Configuration Guide*
- *Cisco IOS Mobile Wireless Packet Data Serving Node Configuration Guide*
- *Cisco IOS IP Mobility Configuration Guide*

Benefits of Enhanced Object Tracking

- Increases the availability and speed of recovery of a network.
- Decreases the number of network outages and their duration.

- Enables client processes such as VRRP and GLBP to track objects individually or as a list of objects. Prior to the introduction of this functionality, the tracking process was embedded within HSRP.

How to Configure Enhanced Object Tracking

Tracking the Line-Protocol State of an Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **track timer interface** {seconds | msec milliseconds}
4. **track object-number interface type number line-protocol**
5. **carrier-delay**
6. **delay** {up seconds [down [seconds] | [up seconds] down seconds]}
7. **end**
8. **show track object-number**

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|--------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Step 1 | enable Example: Device> enable | Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted. |
| Step 2 | configure terminal Example: Device# configure terminal | Enters global configuration mode. |
| Step 3 | track timer interface {seconds msec milliseconds} Example: Device(config)# track timer interface 5 | (Optional) Specifies the interval in which the tracking process polls the tracked object. <ul style="list-style-type: none"> • The default interval that the tracking process polls interface objects is 1 second. <p>Note All polling frequencies can be configured down to 500 milliseconds, overriding the minimum 1-second interval configured using the msec keyword and <i>milliseconds</i> argument.</p> |
| Step 4 | track object-number interface type number line-protocol Example: | Tracks the line-protocol state of an interface and enters tracking configuration mode. |

| | Command or Action | Purpose |
|---------------|--------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| | Device(config)# track 3 interface GigabitEthernet 0/0 line-protocol | |
| Step 5 | carrier-delay Example: Device(config-track)# carrier-delay | (Optional) Enables EOT to consider the carrier-delay timer when tracking the status of an interface. |
| Step 6 | delay {up seconds [down [seconds] [up seconds] down seconds]} Example: Device(config-track)# delay up 30 | (Optional) Specifies a period of time (in seconds) to delay communicating state changes of a tracked object. |
| Step 7 | end Example: Device(config-track)# end | Exits to privileged EXEC mode. |
| Step 8 | show track object-number Example: Device# show track 3 | (Optional) Displays tracking information. <ul style="list-style-type: none"> • Use this command to verify the configuration. |

Example

The following example shows the state of the line protocol on an interface when it is tracked:

```
Device# show track 3

Track 3
  Interface GigabitEthernet 0/0 line-protocol
  Line protocol is Up
    1 change, last change 00:00:05
  Tracked by:
    HSRP GigabitEthernet 0/3 1
```

Tracking the IP-Routing State of an Interface

SUMMARY STEPS

1. enable
2. configure terminal
3. track timer interface {seconds | msec milliseconds}
4. track object-number interface type number ip routing
5. carrier-delay

6. `delay {up seconds [down seconds] | [up seconds] down seconds}`
7. `end`
8. `show track object-number`

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Step 1 | enable Example: <pre>Device> enable</pre> | Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted. |
| Step 2 | configure terminal Example: <pre>Device# configure terminal</pre> | Enters global configuration mode. |
| Step 3 | track timer interface {seconds msec milliseconds} Example: <pre>Device(config)# track timer interface 5</pre> | (Optional) Specifies the interval in which the tracking process polls the tracked object. <ul style="list-style-type: none"> • The default interval that the tracking process polls interface objects is 1 second. <p>Note All polling frequencies can be configured down to 500 milliseconds, overriding the minimum 1-second interval configured using the msec keyword and <i>milliseconds</i> argument.</p> |
| Step 4 | track object-number interface type number ip routing Example: <pre>Device(config)# track 1 interface Gigabitethernet 0/0 ip routing</pre> | Tracks the IP-routing state of an interface and enters tracking configuration mode. <ul style="list-style-type: none"> • IP-route tracking tracks an IP route in the routing table and the ability of an interface to route IP packets. |
| Step 5 | carrier-delay Example: <pre>Device(config-track)# carrier-delay</pre> | (Optional) Enables EOT to consider the carrier-delay timer when tracking the status of an interface. |
| Step 6 | delay {up seconds [down seconds] [up seconds] down seconds} Example: <pre>Device(config-track)# delay up 30</pre> | (Optional) Specifies a period of time (in seconds) to delay communicating state changes of a tracked object. |
| Step 7 | end Example: <pre>Device(config-track)# end</pre> | Returns to privileged EXEC mode. |

| | Command or Action | Purpose |
|---------------|-----------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| Step 8 | show track <i>object-number</i> Example: Device# show track 1 | Displays tracking information. <ul style="list-style-type: none"> • Use this command to verify the configuration. |

Example

The following example shows the state of IP routing on an interface when it is tracked:

```
Device# show track 1

Track 1
  Interface GigabitEthernet 0/1 ip routing
  IP routing is Up
    1 change, last change 00:01:08
  Tracked by:
    HSRP GigabitEthernet 0/3 1
```

Tracking IP-Route Reachability

Perform this task to track the reachability of an IP route. A tracked object is considered up when a routing table entry exists for the route and the route is accessible.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **track timer ip route** {*seconds* | **msec** *milliseconds*}
4. **track *object-number* ip route *ip-address/prefix-length* reachability**
5. **delay** {**up** *seconds* [**down** *seconds*] | [**up** *seconds*] **down** *seconds*}
6. **ip vrf *vrf-name***
7. **end**
8. **show track *object-number***

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|----------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------|
| Step 1 | enable Example: Device> enable | Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted. |
| Step 2 | configure terminal Example: Device# configure terminal | Enters global configuration mode. |

| | Command or Action | Purpose |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Step 3 | track timer ip route <i>{seconds msec milliseconds}</i> Example: <pre>Device(config)# track timer ip route 20</pre> | (Optional) Specifies the interval in which the tracking process polls the tracked object. <ul style="list-style-type: none"> The default interval that the tracking process polls IP-route objects is 15 seconds. Note All polling frequencies can be configured down to 500 milliseconds, overriding the minimum 1-second interval configured using the msec keyword and <i>milliseconds</i> argument. |
| Step 4 | track object-number ip route ip-address/prefix-length reachability Example: <pre>Device(config)# track 4 ip route 10.16.0.0/16 reachability</pre> | Tracks the reachability of an IP route and enters tracking configuration mode. |
| Step 5 | delay {up seconds [down seconds] [up seconds] down seconds} Example: <pre>Device(config-track)# delay up 30</pre> | (Optional) Specifies a period of time (in seconds) to delay communicating state changes of a tracked object. |
| Step 6 | ip vrf vrf-name Example: <pre>Device(config-track)# ip vrf VRF2</pre> | (Optional) Configures a VPN routing and forwarding (VRF) table. |
| Step 7 | end Example: <pre>Device(config-track)# end</pre> | Returns to privileged EXEC mode. |
| Step 8 | show track object-number Example: <pre>Device# show track 4</pre> | (Optional) Displays tracking information. <ul style="list-style-type: none"> Use this command to verify the configuration. |

Example

The following example shows the state of the reachability of an IP route when it is tracked:

```
Device# show track 4

Track 4
  IP route 10.16.0.0 255.255.0.0 reachability
  Reachability is Up (RIP)
```

```

1 change, last change 00:02:04
First-hop interface is Ethernet0/1
Tracked by:
  HSRP Ethernet0/3 1

```

Tracking the Threshold of IP-Route Metrics

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **track timer ip route** {seconds | msec milliseconds}
4. **track resolution ip route** {eigrp | isis | ospf | static} resolution-value
5. **track object-number ip route** ip-address/prefix-length metric threshold
6. **delay** {up seconds [down seconds] | [up seconds] down seconds}
7. **ip vrf** vrf-name
8. **threshold metric** {up number [down number] | down number [up number]}
9. **end**
10. **show track object-number**

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|-------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Step 1 | enable Example: Device> enable | Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted. |
| Step 2 | configure terminal Example: Device# configure terminal | Enters global configuration mode. |
| Step 3 | track timer ip route {seconds msec milliseconds} Example: Device(config)# track timer ip route 20 | (Optional) Specifies the interval in which the tracking process polls the tracked object. <ul style="list-style-type: none"> • The default interval that the tracking process polls IP-route objects is 15 seconds. Note All polling frequencies can be configured down to 500 milliseconds, overriding the minimum 1-second interval configured using the msec keyword and <i>milliseconds</i> argument. |
| Step 4 | track resolution ip route {eigrp isis ospf static} resolution-value Example: | (Optional) Specifies resolution parameters for a tracked object. <ul style="list-style-type: none"> • Use this command to change the default metric resolution values. |

| | Command or Action | Purpose |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | Device(config)# track resolution ip route eigrp 300 | |
| Step 5 | <p>track <i>object-number</i> ip route <i>ip-address/prefix-length</i> metric threshold</p> <p>Example:</p> <pre>Device(config)# track 6 ip route 10.16.0.0/16 metric threshold</pre> | <p>Tracks the scaled metric value of an IP route to determine if it is above or below a threshold and enters tracking configuration mode.</p> <ul style="list-style-type: none"> • The default down value is 255, which equates to an inaccessible route. • The default up value is 254. |
| Step 6 | <p>delay {up <i>seconds</i> [<i>down seconds</i>] [up <i>seconds</i>] down <i>seconds</i>}</p> <p>Example:</p> <pre>Device(config-track)# delay up 30</pre> | (Optional) Specifies a period of time (in seconds) to delay communicating state changes of a tracked object. |
| Step 7 | <p>ip vrf <i>vrf-name</i></p> <p>Example:</p> <pre>Device(config-track)# ip vrf VRF1</pre> | (Optional) Configures a VRF table. |
| Step 8 | <p>threshold metric {up <i>number</i> [<i>down number</i>] down <i>number</i> [<i>up number</i>] }</p> <p>Example:</p> <pre>Device(config-track)# threshold metric up 254 down 255</pre> | (Optional) Sets a metric threshold other than the default value. |
| Step 9 | <p>end</p> <p>Example:</p> <pre>Device(config-track)# end</pre> | Exits to privileged EXEC mode. |
| Step 10 | <p>show track <i>object-number</i></p> <p>Example:</p> <pre>Device# show track 6</pre> | <p>(Optional) Displays tracking information.</p> <ul style="list-style-type: none"> • Use this command to verify the configuration. |

Example

The following example shows the metric threshold of an IP route when it is tracked:

```
Device# show track 6

Track 6
  IP route 10.16.0.0 255.255.0.0 metric threshold
```



```

Metric threshold is Up (RIP/6/102)
  1 change, last change 00:00:08
Metric threshold down 255 up 254
First-hop interface is Ethernet0/1
Tracked by:
  HSRP Ethernet0/3 1

```

Tracking the State of an IP SLAs Operation

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **track *object-number* ip sla *operation-number* state**
4. **delay {*up seconds* [*down seconds* | [*up seconds*] *down seconds*}**
5. **end**
6. **show track *object-number***

DETAILED STEPS

| | Command or Action | Purpose |
|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Step 1 | enable Example: Device> enable | Enables privileged EXEC mode. • Enter your password if prompted. |
| Step 2 | configure terminal Example: Device# configure terminal | Enters global configuration mode. |
| Step 3 | track <i>object-number</i> ip sla <i>operation-number</i> state Example: Device(config)# track 2 ip sla 4 state | Tracks the state of an IP SLAs object and enters tracking configuration mode. With CSCsf08092, the track rtr command was replaced by the track ip sla command. |
| Step 4 | delay {<i>up seconds</i> [<i>down seconds</i> [<i>up seconds</i>] <i>down seconds</i>} Example: Device(config-track)# delay up 60 down 30 | (Optional) Specifies a period of time (in seconds) to delay communicating state changes of a tracked object. |
| Step 5 | end Example: Device(config-track)# end | Exits to privileged EXEC mode. |
| Step 6 | show track <i>object-number</i> | (Optional) Displays tracking information. |

| Command or Action | Purpose |
|---------------------------------------------|-------------------------------------------------------------------------------------------------|
| Example: Device# show track 2 | <ul style="list-style-type: none"> Use this command to verify the configuration. |

Example

The following example shows the state of the IP SLAs tracking:

```
Device# show track 2

Track 2
  IP SLA 1 state
  State is Down
    1 change, last change 00:00:47
  Latest operation return code: over threshold
  Latest RTT (milliseconds) 4
  Tracked by:
    HSRP Ethernet0/1 3
```

Tracking the Reachability of an IP SLAs IP Host

SUMMARY STEPS

- enable
- configure terminal
- track *object-number* ip sla *operation-number* reachability
- delay {up *seconds* [down *seconds*] | [up *seconds*] down*seconds*}
- end
- show track *object-number*

DETAILED STEPS

| Command or Action | Purpose |
|-------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Step 1 enable Example: Device> enable | Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted. |
| Step 2 configure terminal Example: Device# configure terminal | Enters global configuration mode. |
| Step 3 track <i>object-number</i> ip sla <i>operation-number</i> reachability Example: | Tracks the reachability of an IP SLAs IP host and enters tracking configuration mode. <p>Note With CSCsf08092, the track rtr command was replaced by the track ip sla command.</p> |

| | Command or Action | Purpose |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| | Device(config)# track 2 ip sla 4 reachability | |
| Step 4 | delay {up seconds [down seconds] [up seconds] downseconds} Example: Device(config-track)# delay up 30 down 10 | (Optional) Specifies a period of time (in seconds) to delay communicating state changes of a tracked object. |
| Step 5 | end Example: Device(config-track)# end | Exits to privileged EXEC mode. |
| Step 6 | show track object-number Example: Device# show track 3 | (Optional) Displays tracking information. <ul style="list-style-type: none"> • Use this command to verify the configuration. |

Example

The following example shows whether the route is reachable:

```
Device# show track 3

Track 3
  IP SLA 1 reachability
  Reachability is Up
    1 change, last change 00:00:47
  Latest operation return code: over threshold
  Latest RTT (millisecs) 4
  Tracked by:
    HSRP Ethernet0/1 3
```

Configuring a Tracked List and Boolean Expression

Perform this task to configure a tracked list of objects and a Boolean expression to determine the state of the list. A tracked list contains one or more objects. The Boolean expression enables two types of calculations by using either “and” or “or” operators. For example, when you configure tracking for two interfaces using the “and” operator up means that *both* interfaces are up, and down means that either interface is down.

You may configure a tracked list state to be measured using a weight or percentage threshold. See the [Configuring a Tracked List and Threshold Weight](#) section and the [Configuring a Tracked List and Threshold Percentage](#) section.

Before you begin

An object must exist before it can be added to a tracked list.



Note The “not” operator is specified for one or more objects and negates the state of the object.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **track** *track-number* **list boolean** {**and** | **or**}
4. **object** *object-number* [**not**]
5. **delay** {**up** *seconds* [**down** *seconds*] | [**up** *seconds*] **down** *seconds*}
6. **end**

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Step 1 | enable Example: Device> enable | Enables privileged EXEC mode. • Enter your password if prompted. |
| Step 2 | configure terminal Example: Device# configure terminal | Enters global configuration mode. |
| Step 3 | track <i>track-number</i> list boolean { and or } | Configures a tracked list object and enters tracking configuration mode. |
| | Example: Device(config)# track 100 list boolean and | |
| Step 4 | object <i>object-number</i> [not] Example: Device(config-track)# object 3 not | Specifies the object to be tracked. • The <i>object-number</i> argument has a valid range from 1 to 500. There is no default. The optional not keyword negates the state of the object. Note The example means that when object 3 is up, the tracked list detects object 3 as down. |
| Step 5 | delay { up <i>seconds</i> [down <i>seconds</i>] [up <i>seconds</i>] down <i>seconds</i> } | (Optional) Specifies a tracking delay in seconds between up and down states. |
| | Example: Device(config-track)# delay up 3 | |
| Step 6 | end Example: | Returns to privileged EXEC mode. |

| | Command or Action | Purpose |
|--|---------------------------|---------|
| | Device(config-track)# end | |

Configuring a Tracked List and Threshold Weight

Perform this task to configure a list of tracked objects, to specify that weight be used as the threshold, and to configure a weight for each of the objects in the list of tracked objects. A tracked list contains one or more objects. Enhanced object tracking uses a threshold weight to determine the state of each object by comparing the total weight of all objects that are up against a threshold weight for each object.

You can also configure a tracked list state to be measured using a Boolean calculation or threshold percentage. See the [Configuring a Tracked List and Boolean Expression](#) section and the [Configuring a Tracked List and Threshold Percentage](#) section.

Before you begin

An object must exist before it can be added to a tracked list.



Note You cannot use the Boolean “not” operator in a weight or percentage threshold list.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **track** *track-number* **list threshold weight**
4. **object** *object-number* [**weight** *weight-number*]
5. **threshold weight** {**up** *number* **down** *number* | **up** *number* | **down** *number*}
6. **delay** {**up** *seconds* [**down** *seconds*] | [**up** *seconds*] **down** *seconds*}
7. **end**

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|----------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------|
| Step 1 | enable Example: Device> enable | Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted. |
| Step 2 | configure terminal Example: Device# configure terminal | Enters global configuration mode. |
| Step 3 | track <i>track-number</i> list threshold weight Example: | Configures a tracked list object and enters tracking configuration mode. The keywords are as follows: |

| | Command or Action | Purpose |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | Device(config)# track 100 list threshold weight | <ul style="list-style-type: none"> • threshold —Specifies that the state of the tracked list is based on a threshold. • weight —Specifies that the threshold is based on a specified weight. |
| Step 4 | object <i>object-number</i> [weight <i>weight-number</i>] Example: Device(config-track)# object 3 weight 30 | Specifies the object to be tracked. The <i>object-number</i> argument has a valid range from 1 to 500. There is no default. The optional weight keyword specifies a threshold weight for each object. |
| Step 5 | threshold weight { up <i>number</i> down <i>number</i> up <i>number</i> down <i>number</i> } Example: Device(config-track)# threshold weight up 30 | Specifies the threshold weight. <ul style="list-style-type: none"> • up number —Valid range is from 1 to 255. • down number—Range depends upon what you select for the up keyword. For example, if you configure 25 for up, you will see a range from 0 to 24 for down. |
| Step 6 | delay { up <i>seconds</i> [down <i>seconds</i>] [up <i>seconds</i>] down <i>seconds</i> } Example: Device(config-track)# delay up 3 | (Optional) Specifies a tracking delay in seconds between up and down states. |
| Step 7 | end Example: Device(config-track)# end | Returns to privileged EXEC mode. |

Configuring a Tracked List and Threshold Percentage

Perform this task to configure a tracked list of objects, to specify that a percentage will be used as the threshold, and to specify a percentage for each object in the list. A tracked list contains one or more objects. Enhanced object tracking uses the threshold percentage to determine the state of the list by comparing the assigned percentage of each object to the list.

You may also configure a tracked list state to be measured using a Boolean calculation or threshold weight. See the [Configuring a Tracked List and Boolean Expression](#) section and the [Configuring a Tracked List and Threshold Weight](#) section.



Note You cannot use the Boolean “not” operator in a weight or percentage threshold list.

Before you begin

An object must exist before it can be added to a tracked list.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **track** *track-number* **list threshold percentage**
4. **object** *object-number*
5. **threshold percentage** {**up** *number* [**down** *number*] | **down** *number* [**up** *number*]}
6. **delay** {**up** *seconds* [**down** *seconds*] | [**up** *seconds*] **down** *seconds*}
7. **end**

DETAILED STEPS

| | Command or Action | Purpose |
|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Step 1 | enable Example: Device> enable | Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted. |
| Step 2 | configure terminal Example: Device# configure terminal | Enters global configuration mode. |
| Step 3 | track <i>track-number</i> list threshold percentage Example: Device(config)# track 100 list threshold percentage | Configures a tracked list object and enters tracking configuration mode. The keywords are as follows: <ul style="list-style-type: none"> • threshold —Specifies that the state of the tracked list is based on a threshold. • percentage —Specifies that the threshold is based on a percentage. |
| Step 4 | object <i>object-number</i> Example: Device(config-track)# object 3 | Specifies the object to be tracked. <ul style="list-style-type: none"> • The <i>object-number</i> argument has a valid range from 1 to 500. There is no default. |
| Step 5 | threshold percentage { up <i>number</i> [down <i>number</i>] down <i>number</i> [up <i>number</i>]} Example: Device(config-track)# threshold percentage up 30 | Specifies the threshold percentage. <ul style="list-style-type: none"> • up <i>number</i>—Valid range is from 1 to 100. • down <i>number</i> —Range depends upon what you have selected for the up keyword. For example, if you specify 25 as up, a range from 26 to 100 is displayed for the down keyword. |
| Step 6 | delay { up <i>seconds</i> [down <i>seconds</i>] [up <i>seconds</i>] down <i>seconds</i> } Example: | (Optional) Specifies a tracking delay in seconds between up and down states. |

| | Command or Action | Purpose |
|---------------|------------------------------------------------------------|----------------------------------|
| | Device(config-track)# delay up 3 | |
| Step 7 | end Example: Device(config-track)# end | Returns to privileged EXEC mode. |

Configuring Track List Defaults

Perform this task to configure a default delay value for a tracked list, a default object, and default threshold parameters for a tracked list.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **track** *track-number*
4. **default** {**delay** | **object** *object-number* | **threshold percentage**}
5. **end**

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Step 1 | enable Example: Device> enable | Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted. |
| Step 2 | configure terminal Example: Device# configure terminal | Enters global configuration mode. |
| Step 3 | track <i>track-number</i> Example: Device(config)# track 3 | Enters tracking configuration mode. |
| Step 4 | default { delay object <i>object-number</i> threshold percentage } Example: Device(config-track)# default delay | Specifies a default delay value for a tracked list, a default object, and default threshold parameters for a tracked list. <ul style="list-style-type: none"> • delay —Reverts to the default delay. • object <i>object-number</i>—Specifies a default object for the track list. The valid range is from 1 to 1000. |

| | Command or Action | Purpose |
|---------------|-----------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------|
| | | <ul style="list-style-type: none"> • threshold percentage—Specifies a default threshold percentage. |
| Step 5 | end Example: <pre>Device(config-track)# end</pre> | Returns to privileged EXEC mode. |

Configuring Tracking for Mobile IP Applications

Perform this task to configure a tracked list of Mobile IP application objects.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **track *track-number* application home-agent**
4. **exit**
5. **track *track-number* application pdsn**
6. **exit**
7. **track *track-number* application ggsn**
8. **end**

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------|
| Step 1 | enable Example: <pre>Device> enable</pre> | Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted. |
| Step 2 | configure terminal Example: <pre>Device# configure terminal</pre> | Enters global configuration mode. |
| Step 3 | track <i>track-number</i> application home-agent Example: <pre>Device(config)# track 100 application home-agent</pre> | (Optional) Tracks the presence of Home Agent traffic on a router and enters tracking configuration mode. |
| Step 4 | exit Example: <pre>Device(config-track)# exit</pre> | Returns to global configuration mode. |

| | Command or Action | Purpose |
|---------------|---------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------|
| Step 5 | track <i>track-number</i> application pdsn Example: Device(config)# track 100 application pdsn | (Optional) Tracks the presence of Packet Data Serving Node (PDSN) traffic on a router tracking configuration mode. |
| Step 6 | exit Example: Device(config-track)# exit | Returns to global configuration mode. |
| Step 7 | track <i>track-number</i> application ggsn Example: Device(config)# track 100 application ggsn | (Optional) Tracks the presence of Gateway GPRS Support Node (GGSN) traffic on a router tracking configuration mode. |
| Step 8 | end Example: Device(config)# end | Returns to privileged EXEC mode. |

Configuration Examples for Enhanced Object Tracking

Example: Interface Line Protocol

In the following example, the tracking process is configured to track the line-protocol state of GigabitEthernet interface 1/0/0. HSRP on GigabitEthernet interface 0/0/0 then registers with the tracking process to be informed of any changes to the line-protocol state of GigabitEthernet interface 1/0/0. If the line protocol on GigabitEthernet interface 1/0/0 goes down, the priority of the HSRP group is reduced by 10.

Router A Configuration

```
Device(config)# track 100 interface GigabitEthernet1/0/0 line-protocol
!
Device(config)# interface GigabitEthernet0/0/0
Device(config-if)# ip address 10.1.0.21 255.255.0.0
Device(config-if)# standby 1 preempt
Device(config-if)# standby 1 ip 10.1.0.1
Device(config-if)# standby 1 priority 110
Device(config-if)# standby 1 track 100 decrement 10
```

Router B Configuration

```
Device(config)# track 100 interface GigabitEthernet1/0/0 line-protocol
!
Device(config)# interface GigabitEthernet0/0/0
Device(config-if)# ip address 10.1.0.22 255.255.0.0
```

```
Device(config-if)# standby 1 preempt
Device(config-if)# standby 1 ip 10.1.0.1
Device(config-if)# standby 1 priority 105
Device(config-if)# standby 1 track 100 decrement 10
```

Example: Interface IP Routing

In the following example, the tracking process is configured to track the IP-routing capability of GigabitEthernet interface 1/0/0. HSRP on GigabitEthernet interface 0/0/0 then registers with the tracking process to be informed of any changes to the IP-routing state of GigabitEthernet interface 1/0/0. If the IP-routing state on GigabitEthernet interface 1/0/0 goes down, the priority of the HSRP group is reduced by 10.

If both serial interfaces are operational, Router A will be the HSRP active router because it has the higher priority. However, if IP on GigabitEthernet interface 1/0/0 in Router A fails, the HSRP group priority will be reduced and Router B will take over as the active router, thus maintaining a default virtual gateway service to hosts on the 10.1.0.0 subnet.

See the figure below for a sample topology.

Figure 1: Topology for IP-Routing Support



Router A Configuration

```
Device(config)# track 100 interface GigabitEthernet1/0/0 ip routing
!
Device(config)# interface GigabitEthernet0/0/0
Device(config-if)# ip address 10.1.0.21 255.255.0.0
Device(config-if)# standby 1 preempt
Device(config-if)# standby 1 ip 10.1.0.1
Device(config-if)# standby 1 priority 110
Device(config-if)# standby 1 track 100 decrement 10
```

Router B Configuration

```
Device(config)# track 100 interface GigabitEthernet1/0/0 ip routing
!
Device(config)# interface GigabitEthernet0/0/0
Device(config-if)# ip address 10.1.0.22 255.255.0.0
Device(config-if)# standby 1 preempt
Device(config-if)# standby 1 ip 10.1.0.1
Device(config-if)# standby 1 priority 105
Device(config-if)# standby 1 track 100 decrement 10
```

Example: IP-Route Reachability

In the following example, the tracking process is configured to track the reachability of IP route 10.2.2.0/24:

Router A Configuration

```
Device(config)# track 100 ip route 10.2.2.0/24 reachability
!
Device(config)# interface GigabitEthernet0/0/0
Device(config-if)# ip address 10.1.1.21 255.255.255.0
Device(config-if)# standby 1 preempt
Device(config-if)# standby 1 ip 10.1.1.1
Device(config-if)# standby 1 priority 110
Device(config-if)# standby 1 track 100 decrement 10
```

Router B Configuration

```
Device(config)# track 100 ip route 10.2.2.0/24 reachability
!
Device(config)# interface GigabitEthernet0/0/0
Device(config-if)# ip address 10.1.1.22 255.255.255.0
Device(config-if)# standby 1 preempt
Device(config-if)# standby 1 ip 10.1.1.1
Device(config-if)# standby 1 priority 105
Device(config-if)# standby 1 track 100 decrement 10
```

Example: IP-Route Threshold Metric

In the following example, the tracking process is configured to track the threshold metric of IP route 10.2.2.0/24:

Router A Configuration

```
Device(config)# track 100 ip route 10.2.2.0/24 metric threshold
!
Device(config)# interface GigabitEthernet0/0/0
Device(config-if)# ip address 10.1.1.21 255.255.255.0
Device(config-if)# standby 1 preempt
Device(config-if)# standby 1 ip 10.1.1.1
Device(config-if)# standby 1 priority 110
Device(config-if)# standby 1 track 100 decrement 10
```

Router B Configuration

```
Device(config)# track 100 ip route 10.2.2.0/24 metric threshold
!
Device(config)# interface GigabitEthernet0/0/0
Device(config-if)# ip address 10.1.1.22 255.255.255.0
Device(config-if)# standby 1 preempt
Device(config-if)# standby 1 ip 10.1.1.1
Device(config-if)# standby 1 priority 105
Device(config-if)# standby 1 track 100 decrement 10
```

Example: IP SLAs IP Host Tracking

The following example shows how to configure IP host tracking for IP SLAs operation 1 prior to CSCsf08092:

```
Device(config)# ip sla 1
Device(config-ip-sla)# icmp-echo 10.51.12.4
```

```

Device(config-ip-sla-echo) # timeout 1000
Device(config-ip-sla-echo) # threshold 2
Device(config-ip-sla-echo) # frequency 3
Device(config-ip-sla-echo) # request-data-size 1400
Device(config-ip-sla-echo) # exit
Device(config) # ip sla schedule 1 start-time now life forever
Device(config-ip-sla) # track 2 rtr 1 state
Device(config-ip-sla) # exit
Device(config) # track 3 rtr 1 reachability
Device(config-track) # exit
Device(config) # interface ethernet0/1
Device(config-if) # ip address 10.21.0.4 255.255.0.0
Device(config-if) # no shutdown
Device(config-if) # standby 3 ip 10.21.0.10
Device(config-if) # standby 3 priority 120
Device(config-if) # standby 3 preempt
Device(config-if) # standby 3 track 2 decrement 10
Device(config-if) # standby 3 track 3 decrement 10

```

The following example shows how to configure IP host tracking for IP SLAs operation 1 prior to CSCsf08092:

```

Device(config) # ip sla 1
Device(config-ip-sla) # icmp-echo 10.51.12.4
Device(config-ip-sla-echo) # threshold 2
Device(config-ip-sla-echo) # timeout 1000
Device(config-ip-sla-echo) # frequency 3
Device(config-ip-sla-echo) # request-data-size 1400
Device(config-ip-sla-echo) # exit
Device(config) # ip sla schedule 1 start-time now life forever
Device(config) # track 2 ip sla 1 state
Device(config-track) # exit
Device(config) # track 3 ip sla 1 reachability
Device(config-track) # exit
Device(config) # interface ethernet0/1
Device(config-if) # ip address 10.21.0.4 255.255.0.0
Device(config-if) # no shutdown
Device(config-if) # standby 3 ip 10.21.0.10
Device(config-if) # standby 3 priority 120
Device(config-if) # standby 3 preempt
Device(config-if) # standby 3 track 2 decrement 10
Device(config-if) # standby 3 track 3 decrement 10

```

Example: Boolean Expression for a Tracked List

In the following example, a track list object is configured to track two GigabitEthernet interfaces when both interfaces are up and when either interface is down:

```

Device(config) # track 1 interface GigabitEthernet2/0/0 line-protocol
Device(config) # track 2 interface GigabitEthernet2/1/0 line-protocol
Device(config-track) # exit
Device(config) # track 100 list boolean and
Device(config-track) # object 1
Device(config-track) # object 2

```

In the following example, a track list object is configured to track two GigabitEthernet interfaces when either interface is up and when both interfaces are down:

```

Device(config) # track 1 interface GigabitEthernet2/0/0 line-protocol
Device(config) # track 2 interface GigabitEthernet2/1/0 line-protocol

```

Example: Threshold Weight for a Tracked List

```
Device(config-track)# exit
Device(config)# track 101 list boolean or
Device(config-track)# object 1
Device(config-track)# object 2
```

The following configuration example shows that tracked list 4 has two objects and one object state is negated (if the list is up, the list detects that object 2 is down):

```
Device(config)# track 4 list boolean and
Device(config-track)# object 1
Device(config-track)# object 2 not
```

Example: Threshold Weight for a Tracked List

In the following example, three GigabitEthernet interfaces in tracked list 100 are configured with a threshold weight of 20 each. The down threshold is configured to 0 and the up threshold is configured to 40:

```
Device(config)# track 1 interface GigabitEthernet2/0/0 line-protocol
Device(config)# track 2 interface GigabitEthernet2/1/0 line-protocol
Device(config)# track 3 interface GigabitEthernet2/2/0 line-protocol
Device(config-track)# exit
Device(config)# track 100 list threshold weight
Device(config-track)# object 1 weight 20
Device(config-track)# object 2 weight 20
Device(config-track)# object 3 weight 20
Device(config-track)# threshold weight up 40 down 0
```

In the example above the track-list object goes down only when all three serial interfaces go down, and comes up again only when at least two interfaces are up (because $20 + 20 \geq 40$). The advantage of this configuration is that it prevents the track-list object from coming up if two interfaces are down and the third interface is flapping.

The following configuration example shows that if object 1 and object 2 are down, then track list 4 is up, because object 3 satisfies the up threshold value of up 30. But, if object 3 is down, both objects 1 and 2 need to be up in order to satisfy the threshold weight.

```
Device(config)# track 4 list threshold weight
Device(config-track)# object 1 weight 15
Device(config-track)# object 2 weight 20
Device(config-track)# object 3 weight 30
Device(config-track)# threshold weight up 30 down 10
```

This configuration may be useful to you if you have two small bandwidth connections (represented by object 1 and 2) and one large bandwidth connection (represented by object 3). Also the down 10 value means that once the tracked object is up, it will not go down until the threshold value is lower or equal to 10, which in this example means that all connections are down.

Example: Threshold Percentage for a Tracked List

In the following example, four GigabitEthernet interfaces in track list 100 are configured for an up threshold percentage of 75. The track list is up when 75 percent of the interfaces are up and down when fewer than 75 percent of the interfaces are up.

```
Device(config)# track 1 interface GigabitEthernet2/0/0 line-protocol
Device(config)# track 2 interface GigabitEthernet2/1/0 line-protocol
```

```

Device(config)# track 3 interface GigabitEthernet2/2/0 line-protocol
Device(config)# track 4 interface GigabitEthernet2/3/0 line-protocol
Device(config-track)# exit
Device(config)# track 100 list threshold percentage
Device(config-track)# object 1
Device(config-track)# object 2
Device(config-track)# object 3
Device(config-track)# object 4
Device(config-track)# threshold percentage up 75

```

Example: Mobile IP Application Tracking

The following example shows how to configure EOT to track Mobile IP, GGSN, and PDSN traffic on a router:

```

Device(config)# track 1 application home-agent
Device(config-track)# exit
Device(config)# track 2 application ggsn
Device(config-track)# exit
Device(config)# track 3 application pdsn

```

Additional References

Related Documents

| Related Topic | Document Title |
|---------------------------------------|--------------------------------------------------------------|
| Cisco IOS commands | Cisco IOS Master Commands List, All Releases |
| Embedded Event Manager | <i>Embedded Event Manager Overview</i> |
| HSRP concepts and configuration tasks | <i>Configuring HSRP</i> |
| GLBP concepts and configuration tasks | <i>Configuring GLBP</i> |
| IP SLAs commands | <i>Cisco IOS IP SLAs Command Reference</i> |
| VRRP concepts and configuration tasks | <i>Configuring VRRP</i> |
| GLBP, HSRP, and VRRP commands | <i>Cisco IOS IP Application Services Command Reference</i> |

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 software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs |

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 and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password. | http://www.cisco.com/cisco/web/support/index.html |

Feature Information for Enhanced Object Tracking

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

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

Table 3: Feature Information for Enhanced Object Tracking

| Feature Name | Releases | Feature Configuration Information |
|------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Enhanced Tracking Support | Cisco IOS XE 3.1.0SG 12.2(15)T 12.2(25)S 12.2(28)SB 12.2(33)SRA 12.2(33)SXH Cisco IOS XE 3.9S | The Enhanced Tracking Support feature separates the tracking mechanism from HSRP and creates a separate standalone tracking process that can be used by other Cisco IOS processes as well as HSRP. This feature allows tracking of other objects in addition to the interface line-protocol state. The following commands were introduced or modified by this feature: debug track, delay tracking, ip vrf, show track, standby track, threshold metric, track interface, track ip route, track timer. |
| FHRP--Enhanced Object Tracking Integration with Embedded Event Manager | 12.2(33)SRB 12.2(33)SXI 12.4(2)T | EOT is now integrated with EEM to allow EEM to report on a status change of a tracked object and to allow EOT to track EEM objects. The following commands were introduced or modified by this feature: action track read, action track set, default-state, event resource, event rf, event track, show track, track stub. |
| FHRP--Enhanced Object Tracking of IP SLAs Operations | Cisco IOS XE 3.1.0SG 12.2(25)S 12.2(27)SBC 12.2(33)SRA 12.2(33)SXH 12.3(4)T 15.0(1)S | This feature enables First Hop Redundancy Protocols (FHRPs) and other Enhanced Object Tracking (EOT) clients to track the output from IP SLAs objects and use the provided information to trigger an action. The following command was introduced by this feature: track rtr. |
| FHRP--Enhanced Object Tracking Support for Mobile IP | 12.4(11)T | The FHRP--Enhanced Object Tracking Support for Mobile IP feature provides new tracking objects needed by mobile wireless applications to track the presence of Home Agent, GGSN, or PDSN traffic on a router. The following command was introduced by this feature: track application. |
| FHRP--EOT Deprecation of rtr Keyword | 12.2(33)SRE 12.2(33)SXI1 12.4(20)T | This feature replaces the track rtr command with the track ip sla command. The following command was introduced by this feature: track ip sla. |
| FHRP--Object Tracking List | Cisco IOS XE 3.1.0SG 12.2(30)S 12.2(31)SB2 12.2(33)SRA 12.2(33)SXH 12.3(8)T 15.0(1)S | This feature enhances the tracking capabilities to enable the configuration of a combination of tracked objects in a list, and a flexible method of combining objects using Boolean logic. The following commands were introduced or modified by this feature: show track, threshold percentage, threshold weight, track list, track resolution. |

| Feature Name | Releases | Feature Configuration Information |
|-------------------------------|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| EOT Support for Carrier Delay | 12.4(9)T | The EOT Support for Carrier Delay feature enables Enhanced Object Tracking (EOT) to consider the carrier-delay timer when tracking the status of an interface. The following commands were introduced or modified by this feature: carrier-delay (tracking) , show track . |

Glossary

DHCP—Dynamic Host Configuration Protocol. DHCP is a protocol that delivers IP addresses and configuration information to network clients.

GGSN—Gateway GPRS Support Node. A wireless gateway that allows mobile cell phone users to access the public data network (PDN) or specified private IP networks. The GGSN function is implemented on the Cisco routers.

GLBP—Gateway Load Balancing Protocol. Provides automatic router backup for IP hosts that are configured with a single default gateway on an IEEE 802.3 LAN. Multiple first-hop routers on the LAN combine to offer a single virtual first-hop IP router while sharing the IP packet forwarding load. Other routers on the LAN may act as redundant (GLBP) routers that will become active if any of the existing forwarding routers fail.

GPRS—General Packet Radio Service. A 2.5G mobile communications technology that enables mobile wireless service providers to offer their mobile subscribers with packet-based data services over GSM networks.

GSM network—Global System for Mobile Communications network. A digital cellular technology that is used worldwide, predominantly in Europe and Asia. GSM is the world's leading standard in digital wireless communications.

Home Agent—A Home Agent is a router on the home network of the Mobile Node (MN) that maintains an association between the home IP address of the MN and its care-of address, which is the current location of the MN on a foreign or visited network. The HA redirects packets by tunneling them to the MN while it is away from the home network.

HSRP—Hot Standby Router Protocol. Provides high network availability and transparent network topology changes. HSRP creates a Hot Standby router group with a lead router that services all packets sent to the Hot Standby address. The lead router is monitored by other routers in the group, and if it fails, one of these standby routers inherits the lead position and the Hot Standby group address.

IPCP—IP Control Protocol. The protocol used to establish and configure IP over PPP.

LCP—Link Control Protocol. The protocol used to establish, configure, and test data-link connections for use by PPP.

PDSN—Packet Data Serving Node. The Cisco PDSN is a standards-compliant, wireless gateway that enables packet data services in a Code Division Multiplex Access (CDMA) environment. Acting as an access gateway, the Cisco PDSN provides simple IP and Mobile IP access, foreign-agent support, and packet transport for Virtual Private Networks (VPN).

PPP—Point-to-Point Protocol. Provides router-to-router and host-to-network connections over synchronous and asynchronous circuits. PPP is most commonly used for dial-up Internet access. Its features include address notification, authentication via CHAP or PAP, support for multiple protocols, and link monitoring.

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 provider edge router.

VRRP—Virtual Router Redundancy Protocol. Eliminates the single point of failure inherent in the static default routed environment. VRRP specifies an election protocol that dynamically assigns responsibility for a virtual router to one of the VRRP routers on a LAN. The VRRP router that controls the IP addresses associated with a virtual router is called the master, and forwards packets sent to these IP addresses. The election process provides dynamic failover in the forwarding responsibility should the master become unavailable. Any of the virtual router IP addresses on a LAN can then be used as the default first-hop router by end hosts.



CHAPTER 2

Configuring IP Services

This module describes how to configure optional IP services. For a complete description of the IP services commands in this chapter, refer to the *Cisco IOS IP Application Services Command Reference*. To locate documentation of other commands that appear in this module, use the master command list, or search online.

- [Finding Feature Information, on page 33](#)
- [Information About IP Services, on page 33](#)
- [How to Configure IP Services, on page 38](#)
- [Configuration Examples for IP Services, on page 49](#)
- [Additional References For IP Services, on page 50](#)
- [Feature Information for IP Services, on page 51](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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

Information About IP Services

IP Source Routing

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

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

routing is rarely used for legitimate purposes in networks. Some older IP implementations do not process source-routed packets properly, and it may be possible to crash devices running these implementations by sending them datagrams with source routing options. Disable IP source routing whenever possible. Disabling IP source routing will cause a Cisco router to never forward an IP packet that carries a source routing option.

ICMP Overview

Originally created for the TCP/IP suite in RFC 792, the Internet Control Message Protocol (ICMP) was designed to report a small set of error conditions. ICMP can also report a wide variety of error conditions and provide feedback and testing capabilities. Each message uses a common format and is sent and received by using the same protocol rules.

ICMP enables IP to perform addressing, datagram packaging, and routing by allowing encapsulated messages to be sent and received between IP devices. These messages are encapsulated in IP datagrams just like any other IP message. When the message is generated, the original IP header is encapsulated in the ICMP message and these two pieces are encapsulated within a new IP header to be returned as an error report to the sending device.

ICMP messages are sent in several situations: when a datagram cannot reach its destination, when the gateway does not have the buffering capacity to forward a datagram, and when the gateway can direct the host to send traffic on a shorter route. To avoid the infinite regress of messages about messages, no ICMP messages are sent about ICMP messages.

ICMP does not make IP reliable or ensure the delivery of datagrams or the return of a control message. Some datagrams may be dropped without any report of their loss. The higher-level protocols that use IP must implement their own reliability procedures if reliable communication is required.

ICMP Unreachable Error Messages

Type 3 error messages are sent when a message cannot be delivered completely to the application at a destination host. Six codes contained in the ICMP header describe the unreachable condition as follows:

- 0—Network unreachable
- 1—Host unreachable
- 2—Protocol unreachable
- 3—Port unreachable
- 4—Fragmentation needed and the “don’t fragment” (DF) bit is set
- 5—Source route failed

software can suppress the generation of ICMP unreachable destination error messages, which is called rate-limiting. The default is no unreachable messages more often than once every half second. Separate intervals can be configured for code 4 and all other unreachable destination error messages. However, there is no method of displaying how many ICMP messages have not been sent.

The ICMP Unreachable Destination Counters feature provides a method to count and display the unsent Type 3 messages. This feature also provides console logging with error messages when there are periods of excessive rate limiting that would indicate a Denial of Service (DoS) attack against the router.

If the software receives a nonbroadcast packet destined for itself that uses an unknown protocol, it sends an ICMP protocol unreachable message back to the source. Similarly, if the software receives a packet that it is

unable to deliver to the final destination because it knows of no route to the destination address, it sends an ICMP host unreachable message to the source. This functionality is enabled by default.

Disable ICMP host unreachable messages whenever possible. ICMP supports IP traffic by relaying information about paths, routes, and network conditions. These messages can be used by an attacker to gain network mapping information.

Because the null interface is a packet sink, packets forwarded there will always be discarded and, unless disabled, will generate host unreachable messages. In that case, if the null interface is being used to block a Denial-of-Service attack, these messages flood the local network with these messages. Disabling these messages prevents this situation. In addition, because all blocked packets are forwarded to the null interface, an attacker receiving host unreachable messages could use those messages to determine Access Control List (ACL) configuration. If the “null 0” interface is configured on your router, disable ICMP host unreachable messages for discarded packets or packets routed to the null interface.

ICMP Mask Reply Messages

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

These messages can be used by an attacker to gain network mapping information.

ICMP Redirect Messages

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

In a properly functioning IP network, a router will send redirects only to hosts on its own local subnets, no end node will ever send a redirect, and no redirect will ever be traversed more than one network hop. However, an attacker may violate these rules; some attacks are based on this. Disabling ICMP redirects will cause no operational impact to the network, and it eliminates this possible method of attack.

Denial of Service Attack

Denial of service has become a growing concern, especially when considering the associated costs of such an attack. DoS attacks can decrease the performance of networked devices, disconnect the devices from the network, and cause system crashes. When network services are unavailable, enterprises and service providers suffer the loss of productivity and sales.

The objective of a DoS attack is to deprive a user or organization access to services or resources. If a Website is compromised by a DoS attack, millions of users could be denied access to the site. DoS attacks do not typically result in intrusion or the illegal theft of information. Instead of providing access to unauthorized users, DoS attacks can cause much aggravation and cost to the target customer by preventing authorized

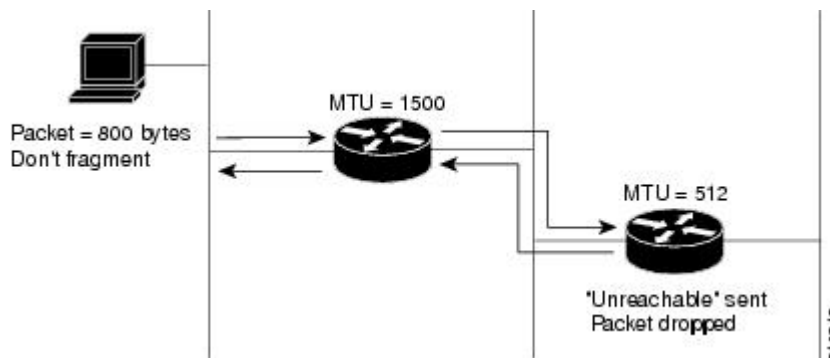
access. Distributed DoS (DDoS) attacks amplify DoS attacks in that a multitude of compromised systems coordinate to flood targets with attack packets, thereby causing denial of service for users of the targeted systems.

A DoS attack occurs when a stream of ICMP echo requests (pings) are broadcast to a destination subnet. The source addresses of these requests are falsified to be the source address of the target. For each request sent by the attacker, many hosts on the subnet will respond flooding the target and wasting bandwidth. The most common DoS attack is called a “smurf” attack, named after an executable program and is in the category of network-level attacks against hosts. DoS attacks can be easily detected when error-message logging of the ICMP Unreachable Destination Counters feature is enabled.

Path MTU Discovery

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

Figure 2: IP Path MTU Discovery



IP Path MTU Discovery is useful when a link in a network goes down, forcing the use of another, different MTU-sized link (and different routers). As shown in the figure above, suppose a router is sending IP packets over a network where the MTU in the first router is set to 1500 bytes, but the second router is set to 512 bytes. If the “don’t fragment” (DF) bit of the datagram is set, the datagram would be dropped because the 512-byte router is unable to forward it. All packets larger than 512 bytes are dropped in this case. The second router returns an ICMP destination unreachable message to the source of the datagram with its Code field indicating “Fragmentation needed and DF set.” (1) To support IP Path MTU Discovery, it would also include the MTU of the next hop network link in the low-order bits of an unused header field.

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



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

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

Cisco IP Accounting

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

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

Show and Clear Commands for IOS Sockets

The Show and Clear Commands for IOS Sockets feature introduces the **show udp**, **show sockets**, and **clear sockets** commands. These new commands are useful for monitoring and managing the Cisco IOS Socket library.

In Cisco IOS software, sockets are a per process entity. This means that the maximum number of sockets is per process and all sockets are managed on a per process basis. For example, each Cisco IOS process could have a socket with file descriptor number 1. This is unlike UNIX or other operating systems that have per system file descriptor allocations.

The **show** and **clear** commands operate on a per process basis to be consistent with the current functionality. Thus, any action taken by the commands will be applicable only to a particular process at a time as selected by the process ID entered on the CLI.

Many applications have a need for **show** and **clear** commands, which primarily aid in debugging. The following scenarios provide examples of when these commands might be useful:

- The application H.323 is using sockets for voice calls. According to the current number of calls, there is still space for more sockets. However, no more sockets can be opened. You can now use the **show sockets** command to find out if the socket space is indeed exhausted or if there are unused sockets available.

- An application is waiting for a particular socket event to happen. A UDP segment was seen, but the application never became active. You can use the **show udp** command to display the list of events being monitored to determine if a UDP socket event is being monitored or if the socket library failed to activate the application.
- An application wants to forcibly close all the sockets for a particular process. You can use the **clear sockets** command to close both the sockets and the underlying TCP or UDP connection or Stream Control Transmission Protocol (SCTP) association.

How to Configure IP Services

Protecting Your Network from DOS Attacks

ICMP supports IP traffic by relaying information about paths, routes, and network conditions. ICMP messages can be used by an attacker to gain network mapping information. IP source routing allows the source IP host to specify a route through the IP network and is rarely used for legitimate purposes in networks. Some older IP implementations do not process source-routed packets properly, and it may be possible to crash devices running these implementations by sending them datagrams with source routing options.

Whenever possible, ICMP messages and IP source routing should be disabled.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no ip source-route**
4. **interface** *type/number/slot*
5. **no ip unreachable**
6. **no ip redirects**
7. **no ip mask-reply**

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|----------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------|
| Step 1 | enable Example: Device> enable | Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted. |
| Step 2 | configure terminal Example: Device# configure terminal | Enters global configuration mode. |
| Step 3 | no ip source-route Example: | Disables IP source routing. |

| | Command or Action | Purpose |
|---------------|----------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | Device(config)# no ip source-route | |
| Step 4 | interface <i>type/number/slot</i> Example: Device(config)# interface GigabitEthernet 0/0/0 | Specifies the interface to configure and enters interface configuration mode. |
| Step 5 | no ip unreachable Example: Device(config-if)# no ip unreachable | Disables the sending of ICMP protocol unreachable and host unreachable messages. This command is enabled by default. Note Disabling the unreachable messages also disables IP Path MTU Discovery because path discovery works by having the software send unreachable messages. |
| Step 6 | no ip redirects Example: Device(config-if)# no ip redirects | Disables the sending of ICMP redirect messages to learn routes. This command is enabled by default. |
| Step 7 | no ip mask-reply Example: Device(config-if)# no ip mask-reply | Disables the sending of ICMP mask reply messages. |

Configuring ICMP Unreachable Rate Limiting User Feedback

Perform this task to clear all of the unreachable destination packet statistics and to specify an interval number for unreachable destination messages. This task also configures a packet counter (threshold) and interval to trigger a logging message to a console. This task is beneficial to begin a new log after the thresholds have been set.

SUMMARY STEPS

1. **enable**
2. **clear ip icmp rate-limit** [*interface-type interface-number*]
3. **configure terminal**
4. **ip icmp rate-limit unreachable** [df] [ms] [log [*packets*] [*interval-ms*]]
5. **exit**
6. **show ip icmp rate-limit** [*interface-type interface-number*]

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|-------------------|-------------------------------|
| Step 1 | enable | Enables privileged EXEC mode. |

| | Command or Action | Purpose |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | Example: <pre>Router> enable</pre> | <ul style="list-style-type: none"> Enter your password if prompted. |
| Step 2 | clear ip icmp rate-limit [<i>interface-type interface-number</i>] Example: <pre>Router# clear ip icmp rate-limit ethernet 2/3</pre> | Clears all current ICMP unreachable statistics for all configured interfaces. The optional <i>interface-type</i> and <i>interface-number</i> arguments clear the statistics for only one interface. |
| Step 3 | configure terminal Example: <pre>Router# configure terminal</pre> | Enters global configuration mode. |
| Step 4 | ip icmp rate-limit unreachable [df] [<i>ms</i>] [log [<i>packets</i>] [<i>interval-ms</i>]] Example: <pre>Router(config)# ip icmp rate-limit unreachable df log 1100 12000</pre> | <p>Specifies the rate limitation of ICMP unreachable destination messages and the error message log threshold for generating a message. The default is no unreachable messages are sent more often than once every half second.</p> <p>The arguments and keywords are as follows:</p> <ul style="list-style-type: none"> df --(Optional) When “don’t fragment” (DF) bit is set in the ICMP header, a datagram cannot be fragmented. If the df keyword is not specified, all other types of destination unreachable messages are sent. ms --(Optional) Interval at which unreachable messages are generated. The valid range is from 1 to 4294967295. log --(Optional) List of error messages. The arguments are as follows: <ul style="list-style-type: none"> <i>packets</i>--(Optional) Number of packets that determine a threshold for generating a log. The default is 1000. <i>interval-ms</i>--(Optional) Time limit for an interval for which a logging message is triggered. The default is 60000, which is 1 minute. <p>Note Counting begins as soon as this command is configured.</p> |
| Step 5 | exit Example: <pre>Router# exit</pre> | Exits to privileged EXEC mode. |
| Step 6 | show ip icmp rate-limit [<i>interface-type interface-number</i>] Example: | (Optional) Displays all current ICMP unreachable statistics for all configured interfaces. The optional <i>interface-type</i> |

| | Command or Action | Purpose |
|--|----------------------------------------------|--------------------------------------------------------------------------------------|
| | Router# show ip icmp rate-limit ethernet 2/3 | and <i>interface-number</i> arguments display the statistics for only one interface. |

Example

The following output using the **show ip icmp rate-limit** command displays the unreachable destinations by interface:

```
Router# show ip icmp rate-limit
Interval (millisecond)    DF bit unreachable    All other unreachable
                        500                    500
Interface                # DF bit unreachable  # All other unreachable
-----                -
Ethernet0/0              0                      0
Ethernet0/2              0                      0
Serial3/0/3              0                      19
The greatest number of unreachable is on serial interface 3/0/3.
```

Setting the MTU Packet Size

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

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

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

Perform this task to set the MTU packet size for a specified interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type/number/slot*
4. **ip mtu** *bytes*
5. **end**

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|------------------------------------------------------------|-------------------------------------------------------------------------|
| Step 1 | enable Example: Device> enable | Enables privileged EXEC mode. • Enter your password if prompted. |

| | Command or Action | Purpose |
|---------------|--------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------|
| Step 2 | configure terminal Example: Device# configure terminal | Enters global configuration mode. |
| Step 3 | interface <i>type/number/slot</i> Example: Device(config)# interface GigabitEthernet 0/0/0 | Specifies the interface to configure and enters interface configuration mode. |
| Step 4 | ip mtu <i>bytes</i> Example: Device(config-if)# ip mtu 300 | Sets the IP MTU packet size for an interface. |
| Step 5 | end Example: Device(config-if)# end | Returns to privileged EXEC mode. |

Configuring IP Accounting

To configure IP accounting, perform this task for each interface.

SUMMARY STEPS

1. enable
2. configure terminal
3. ip accounting-threshold *threshold*
4. ip accounting-list *ip-address wildcard*
5. ip accounting-transits *count*
6. interface *type number*
7. ip accounting [access-violations] [output-packets]
8. ip accounting mac-address {input | output}

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|--------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------|
| Step 1 | enable Example: Router> enable | Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted. |
| Step 2 | configure terminal Example: | Enters global configuration mode. |

| | Command or Action | Purpose |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | Router# configure terminal | |
| Step 3 | ip accounting-threshold <i>threshold</i> Example: Router(config)# ip accounting-threshold 500 | (Optional) Sets the maximum number of accounting entries to be created. |
| Step 4 | ip accounting-list <i>ip-address wildcard</i> Example: Router(config)# ip accounting-list 192.31.0.0 0.0.255.255 | (Optional) Filters accounting information for hosts. |
| Step 5 | ip accounting-transits <i>count</i> Example: Router(config)# ip accounting-transits 100 | (Optional) Controls the number of transit records that will be stored in the IP accounting database. |
| Step 6 | interface <i>type number</i> Example: Router(config)# interface GigabitEthernet 1/0/0 | Specifies the interface and enters interface configuration mode. |
| Step 7 | ip accounting [access-violations] [output-packets] Example: Router(config-if)# ip accounting access-violations | Configures basic IP accounting. <ul style="list-style-type: none"> • Use the optional access-violations keyword to enable IP accounting with the ability to identify IP traffic that fails IP access lists. • Use the optional output-packets keyword to enable IP accounting based on the IP packets output on the interface. |
| Step 8 | ip accounting mac-address { input output } Example: Router(config-if)# ip accounting mac-address output | (Optional) Configures IP accounting based on the MAC address of received (input) or transmitted (output) packets. |

Monitoring and Maintaining the IP Network

IP Accounting collects the number of bytes and packets processed by the network element on the source or destination IP address, or on the basis of the **ip precedence** command. The information collected can be used to identify users for network usage billing, monitoring, and troubleshooting.

SUMMARY STEPS

1. **clear ip traffic**
2. **clear ip accounting** [**checkpoint**]

3. `clear sockets process-id`
4. `show ip accounting [checkpoint] [output-packets | access-violations]`
5. `show interface type number mac`
6. `show interface [type number] precedence`
7. `show ip redirects`
8. `show sockets process-id [detail] [events]`
9. `show udp [detail]`
10. `show ip traffic`

DETAILED STEPS

Step 1 `clear ip traffic`

To clear all IP traffic statistical counters on all interfaces, use the following command:

Example:

```
Router# clear ip traffic
```

Step 2 `clear ip accounting [checkpoint]`

You can remove all contents of a particular cache, table, or database. Clearing a cache, table, or database can become necessary when the contents of the particular structure have become or are suspected to be invalid. To clear the active IP accounting database when IP accounting is enabled, use the following command:

Example:

```
Router# clear ip accounting
```

To clear the checkpointed IP accounting database when IP accounting is enabled, use the following command:

Example:

```
Router# clear ip accounting checkpoint
```

Step 3 `clear sockets process-id`

To close all IP sockets and clear the underlying transport connections and data structures for the specified process, use the following command:

Example:

```
Router# clear sockets 35
```

```
All sockets (TCP, UDP and SCTP) for this process will be cleared.
Do you want to proceed? [yes/no]: y
Cleared sockets for PID 35
```

Step 4 `show ip accounting [checkpoint] [output-packets | access-violations]`

To display access list violations, use the **show ip accounting** command. To use this command, you must first enable IP accounting on a per-interface basis.

Use the **checkpoint** keyword to display the checkpointed database. Use the **output-packets** keyword to indicate that information pertaining to packets that passed access control and were routed should be displayed. Use the **access-violations** keyword to display the number of the access list failed by the last packet for the source and destination pair. The number of packets reveals how aggressive the attack is upon a specific destination. If you do not specify the **access-violations** keyword, the command defaults to displaying the number of packets that have passed access lists and were routed.

If neither the **output-packets** nor **access-violations** keyword is specified, **output-packets** is the default.

The following is sample output from the **show ip accounting** command:

Example:

```
Router# show ip accounting
```

| Source | Destination | Packets | Bytes |
|---------------|---------------|---------|---------|
| 172.16.19.40 | 192.168.67.20 | 7 | 306 |
| 172.16.13.55 | 192.168.67.20 | 67 | 2749 |
| 172.16.2.50 | 192.168.33.51 | 17 | 1111 |
| 172.16.2.50 | 172.31.2.1 | 5 | 319 |
| 172.16.2.50 | 172.31.1.2 | 463 | 30991 |
| 172.16.19.40 | 172.16.2.1 | 4 | 262 |
| 172.16.19.40 | 172.16.1.2 | 28 | 2552 |
| 172.16.20.2 | 172.16.6.100 | 39 | 2184 |
| 172.16.13.55 | 172.16.1.2 | 35 | 3020 |
| 172.16.19.40 | 192.168.33.51 | 1986 | 95091 |
| 172.16.2.50 | 192.168.67.20 | 233 | 14908 |
| 172.16.13.28 | 192.168.67.53 | 390 | 24817 |
| 172.16.13.55 | 192.168.33.51 | 214669 | 9806659 |
| 172.16.13.111 | 172.16.6.23 | 27739 | 1126607 |
| 172.16.13.44 | 192.168.33.51 | 35412 | 1523980 |
| 192.168.7.21 | 172.163.1.2 | 11 | 824 |
| 172.16.13.28 | 192.168.33.2 | 21 | 1762 |
| 172.16.2.166 | 192.168.7.130 | 797 | 141054 |
| 172.16.3.11 | 192.168.67.53 | 4 | 246 |
| 192.168.7.21 | 192.168.33.51 | 15696 | 695635 |
| 192.168.7.24 | 192.168.67.20 | 21 | 916 |
| 172.16.13.111 | 172.16.10.1 | 16 | 1137 |

accounting threshold exceeded for 7 packets and 433 bytes

The following is sample output from the **show ip accounting access-violations** command. The output pertains to packets that failed access lists and were not routed:

Example:

```
Router# show ip accounting access-violations
```

| Source | Destination | Packets | Bytes | ACL |
|--------------|---------------|---------|-------|-----|
| 172.16.19.40 | 192.168.67.20 | 7 | 306 | 77 |
| 172.16.13.55 | 192.168.67.20 | 67 | 2749 | 185 |
| 172.16.2.50 | 192.168.33.51 | 17 | 1111 | 140 |
| 172.16.2.50 | 172.16.2.1 | 5 | 319 | 140 |
| 172.16.19.40 | 172.16.2.1 | 4 | 262 | 77 |

Accounting data age is 41

Step 5 **show interface type number mac**

To display information for interfaces configured for MAC accounting, use the **show interface mac** command. The following is sample output from the **show interface mac** command:

Example:

```
Router# show interface ethernet 0/1 mac
```

```
Ethernet0/1
Input (511 free)
0007.f618.4449(228): 4 packets, 456 bytes, last: 2684ms ago
Total: 4 packets, 456 bytes
Output (511 free)
0007.f618.4449(228): 4 packets, 456 bytes, last: 2692ms ago
Total: 4 packets, 456 bytes
```

Step 6 **show interface** [*type number*] **precedence**

To display information for interfaces configured for precedence accounting, use the **show interface precedence** command.

The following is sample output from the **show interface precedence** command. In this example, the total packet and byte counts are calculated for the interface that receives (input) or sends (output) IP packets and sorts the results based on IP precedence.

Example:

```
Router# show interface ethernet 0/1 precedence

Ethernet0/1
Input
Precedence 0:  4 packets, 456 bytes
Output
Precedence 0:  4 packets, 456 bytes
```

Step 7 **show ip redirects**

To display the address of the default router and the address of hosts for which an ICMP redirect message has been received, use the **show ip redirects** command.

Example:

```
Router# show ip redirects

Default gateway is 172.16.80.29

Host          Gateway          Last Use      Total Uses  Interface
172.16.1.111  172.16.80.240   0:00         9   Ethernet0
172.16.1.4    172.16.80.240   0:00         4   Ethernet0
```

Step 8 **show sockets** *process-id* [**detail**] [**events**]

To display the number of sockets currently open and their distribution with respect to the transport protocol process specified by the *process-id* argument, use the **show sockets** command. The following sample output from the **show sockets** command displays the total number of open sockets for the specified process:

Example:

```
Router# show sockets 35

Total open sockets - TCP:7, UDP:0, SCTP:0
```

The following sample output shows information about the same open processes with the **detail** keyword specified:

Example:

```
Router# show sockets 35 detail

      FD LPort FPort Proto Type      TransID
      0 5000  0      TCP  STREAM  0x6654DEBC
State: SS_ISBOUND
Options: SO_ACCEPTCONN

      1 5001  0      TCP  STREAM  0x6654E494
State: SS_ISBOUND
Options: SO_ACCEPTCONN

      2 5002  0      TCP  STREAM  0x656710B0
State: SS_ISBOUND
Options: SO_ACCEPTCONN
```

```

    3 5003 0    TCP    STREAM 0x65671688
State: SS_ISBOUND
Options: SO_ACCEPTCONN

    4 5004 0    TCP    STREAM 0x65671C60
State: SS_ISBOUND
Options: SO_ACCEPTCONN

    5 5005 0    TCP    STREAM 0x65672238
State: SS_ISBOUND
Options: SO_ACCEPTCONN

    6 5006 0    TCP    STREAM 0x64C7840C
State: SS_ISBOUND
Options: SO_ACCEPTCONN

Total open sockets - TCP:7, UDP:0, SCTP:0

```

The following example displays IP socket event information:

Example:

```

Router# show sockets 35 events

Events watched for this process: READ
FD Watched Present Select Present

0 --- --- R-- R--

```

Step 9

show udp [detail]

To display IP socket information about UDP processes, use the **show udp** command. The following example shows how to display detailed information about UDP sockets:

Example:

```

Router# show udp detail

Proto Remote Port Local Port In Out Stat TTY OutputIF
17 10.0.0.0 0 10.0.21.70 67 0 0 2211 0
Queues: output 0
input 0 (drops 0, max 50, highwater 0)
Proto Remote Port Local Port In Out Stat TTY OutputIF
17 10.0.0.0 0 10.0.21.70 2517 0 0 11 0
Queues: output 0
input 0 (drops 0, max 50, highwater 0)
Proto Remote Port Local Port In Out Stat TTY OutputIF
17 10.0.0.0 0 10.0.21.70 5000 0 0 211 0
Queues: output 0
input 0 (drops 0, max 50, highwater 0)
Proto Remote Port Local Port In Out Stat TTY OutputIF
17 10.0.0.0 0 10.0.21.70 5001 0 0 211 0
Queues: output 0
input 0 (drops 0, max 50, highwater 0)
Proto Remote Port Local Port In Out Stat TTY OutputIF
17 10.0.0.0 0 10.0.21.70 5002 0 0 211 0
Queues: output 0
input 0 (drops 0, max 50, highwater 0)
Proto Remote Port Local Port In Out Stat TTY OutputIF
17 10.0.0.0 0 10.0.21.70 5003 0 0 211 0
Queues: output 0
input 0 (drops 0, max 50, highwater 0)
Proto Remote Port Local Port In Out Stat TTY OutputIF
17 10.0.0.0 0 10.0.21.70 5004 0 0 211 0

```

```
Queues: output 0
        input 0 (drops 0, max 50, highwater 0)
```

Step 10 show ip traffic

To display IP protocol statistics, use the **show ip traffic** command. The following example shows that the IP traffic statistics have been cleared by the **clear ip traffic** command:

Example:

```
Router# clear ip traffic
```

```
Router# show ip traffic
```

IP statistics:

```
Rcvd: 0 total, 0 local destination
      0 format errors, 0 checksum errors, 0 bad hop count
      0 unknown protocol, 0 not a gateway
      0 security failures, 0 bad options, 0 with options
Opts: 0 end, 0 nop, 0 basic security, 0 loose source route
      0 timestamp, 0 extended security, 0 record route
      0 stream ID, 0 strict source route, 0 alert, 0 cipso
      0 other
Frgs: 0 reassembled, 0 timeouts, 0 couldn't reassemble
      0 fragmented, 0 couldn't fragment
Bcast: 0 received, 0 sent
Mcast: 0 received, 0 sent
Sent: 0 generated, 0 forwarded
Drop: 0 encapsulation failed, 0 unresolved, 0 no adjacency
      0 no route, 0 unicast RPF, 0 forced drop
```

ICMP statistics:

```
Rcvd: 0 format errors, 0 checksum errors, 0 redirects, 0 unreachable
      0 echo, 0 echo reply, 0 mask requests, 0 mask replies, 0 quench
      0 parameter, 0 timestamp, 0 info request, 0 other
      0 irdp solicitations, 0 irdp advertisements
Sent: 0 redirects, 0 unreachable, 0 echo, 0 echo reply
      0 mask requests, 0 mask replies, 0 quench, 0 timestamp
      0 info reply, 0 time exceeded, 0 parameter problem
      0 irdp solicitations, 0 irdp advertisements
```

UDP statistics:

```
Rcvd: 0 total, 0 checksum errors, 0 no port
Sent: 0 total, 0 forwarded broadcasts
```

TCP statistics:

```
Rcvd: 0 total, 0 checksum errors, 0 no port
Sent: 0 total
```

Probe statistics:

```
Rcvd: 0 address requests, 0 address replies
      0 proxy name requests, 0 where-is requests, 0 other
Sent: 0 address requests, 0 address replies (0 proxy)
      0 proxy name replies, 0 where-is replies
```

EGP statistics:

```
Rcvd: 0 total, 0 format errors, 0 checksum errors, 0 no listener
Sent: 0 total
```

IGRP statistics:

```
Rcvd: 0 total, 0 checksum errors
Sent: 0 total
```

OSPF statistics:

```
Rcvd: 0 total, 0 checksum errors
```

```
    0 hello, 0 database desc, 0 link state req
    0 link state updates, 0 link state acks

Sent: 0 total

IP-IGRP2 statistics:
Rcvd: 0 total
Sent: 0 total

PIMv2 statistics: Sent/Received
Total: 0/0, 0 checksum errors, 0 format errors
Registers: 0/0, Register Stops: 0/0, Hellos: 0/0
Join/Prunes: 0/0, Asserts: 0/0, grafts: 0/0
Bootstraps: 0/0, Candidate_RP_Advertisements: 0/0

IGMP statistics: Sent/Received
Total: 0/0, Format errors: 0/0, Checksum errors: 0/0
Host Queries: 0/0, Host Reports: 0/0, Host Leaves: 0/0
DVMRP: 0/0, PIM: 0/0
```

Configuration Examples for IP Services

Example: Protecting Your Network from DOS Attacks

The following example shows how to change some of the ICMP defaults for Gigabit Ethernet interface 0/0/0 to prevent ICMP from relaying information about paths, routes, and network conditions, which can be used by an attacker to gain network mapping information.

Disabling the unreachable messages will have a secondary effect: it will also disable IP Path MTU Discovery, because path discovery works by having the software send Unreachable messages. If you have a network segment with a small number of devices and an absolutely reliable traffic pattern—which could easily happen on a segment with a small number of rarely used user devices—you would be disabling options that your device would be unlikely to use anyway.

```
Device(config)# no ip source-route
Device(config)# interface GigabitEthernet 0/0/0
Device(config-if)# no ip unreachables
Device(config-if)# no ip redirects
Device(config-if)# no ip mask-reply
```

Example: Configuring ICMP Unreachable Destination Counters

The following example shows how to clear all of the unreachable destination packet statistics and to specify an interval number for unreachable destination messages. This example also shows how to configure a packet counter threshold and interval to trigger a logging message to a console.

```
Router# clear ip icmp rate-limit ethernet 0/0
Router# configure terminal
Router(config)# ip icmp rate-limit unreachable df log 1100 12000
```

Example: Setting the MTU Packet Size

The following example shows how to change the default MTU packet size for Gigabit Ethernet interface 0/0/0:

```
Device(config)# interface GigabitEthernet 0/0/0
Device(config-if)# ip mtu 300
```

Example: Configuring IP Accounting

The following example shows how to enable IP accounting based on the source and destination MAC address and based on IP precedence for received and transmitted packets:

```
Router# configure terminal
Router(config)# interface ethernet 0/5
Router(config-if)# ip accounting mac-address input
Router(config-if)# ip accounting mac-address output
Router(config-if)# ip accounting precedence input
Router(config-if)# ip accounting precedence output
```

The following example shows how to enable IP accounting with the ability to identify IP traffic that fails IP access lists and with the number of transit records that will be stored in the IP accounting database limited to 100:

```
Router# configure terminal
Router(config)# ip accounting-transits 100
Router(config)# interface ethernet 0/5
Router(config-if)# ip accounting output-packets
Router(config-if)# ip accounting access-violations
```

Additional References For IP Services

Related Documents

| Related Topic | Document Title |
|----------------------------------|---------------------------------------------------------------------|
| Cisco IOS commands | Cisco IOS Master Commands List, All Releases |
| IP application services commands | Cisco IOS IP Application Services Command Reference |

Standards and RFCs

| Standard | Title |
|----------|------------------------------------------------|
| RFC 1256 | ICMP Router Discovery Messages |

Technical Assistance

| Description | Link |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|
| The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password. | http://www.cisco.com/cisco/web/support/index.html |

Feature Information for IP Services

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

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

Table 4: Feature Information for IP Services

| Feature Name | Releases | Feature Information |
|----------------------------------------------|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Clear IP Traffic CLI | 12.4(2)T 12.2(31)SB2 | The Clear IP Traffic CLI feature introduced the clear ip traffic command to clear all IP traffic statistics on a router instead of reloading the router. For added safety, the user will see a confirmation prompt when entering this command. In Cisco IOS Release 12.4(2)T, this feature was introduced. The following command was introduced by this feature: clear ip traffic . |
| ICMP Unreachable Rate Limiting User Feedback | 12.4(2)T 12.2(31)SB2 | The ICMP Unreachable Rate Limiting User Feedback feature enables you to clear and display packets that have been discarded because of an unreachable destination, and to configure a threshold interval for triggering error messages. When message logging is generated, it displays on your console. In Cisco IOS Release 12.4(2)T, this feature was introduced. The following commands were introduced or modified by this feature: clear ip icmp rate-limit , ip icmp rate-limit unreachable , show ip icmp rate-limit . |

| Feature Name | Releases | Feature Information |
|-----------------------------------------|----------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| IP Precedence Accounting | 12.2(21) 12.1(27b)E1 12.1(5)T15 12.2(25)S 12.2(33)SRA 12.2(18)SXF13 12.2(33)SXH1 15.0(1)S | <p>The IP Precedence Accounting feature provides accounting information for IP traffic based on the precedence of any interface. This feature calculates the total packet and byte counts for an interface that receives or sends IP packets and sorts the results based on the IP precedence. This feature is supported on all interfaces and subinterfaces and supports CEF, dCEF, flow, and optimum switching.</p> <p>The following command was introduced by this feature: show interface precedence, ip accounting precedence.</p> |
| Show and Clear Commands for IOS Sockets | 12.4(11)T | <p>The Show and Clear Commands for IOS Sockets feature introduces the show udp, show sockets, and clear sockets commands. These new commands are useful for monitoring and managing the Cisco IOS Socket library.</p> <p>The following commands were introduced or modified by this feature: clear sockets, show sockets, show udp.</p> <p>The following command was replaced by this feature: show ip sockets.</p> |



CHAPTER 3

Configuring IPv4 Broadcast Packet Handling

This module explains what IPv4 broadcast packets are, when they are used, and how to customize your router's configuration for situations when the default behavior for handling IPv4 broadcast packets isn't appropriate.

This module also explains some common scenarios that require customizing IPv4 broadcast packet handling by routers. For example, UDP forwarding of Dynamic Host Configuration Protocol (DHCP) traffic to ensure broadcast packets sent by DHCP clients can reach DHCP servers that are not on the same network segment as the client. Configuration tasks and examples are also provided in this module.

- [Finding Feature Information, on page 53](#)
- [Information About IPv4 Broadcast Packet Handling, on page 53](#)
- [How to Configure IP Broadcast Packet Handling, on page 64](#)
- [Configuration Examples for IP Broadcast Packet Handling, on page 74](#)
- [Additional References for WCCP—Configurable Router ID, on page 75](#)
- [Feature Information for IP Broadcast Packet Handling, on page 76](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

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

Information About IPv4 Broadcast Packet Handling

IP Unicast Address

An IP unicast address is not a broadcast addresses. A packet with an unicast destination IP address is intended for a specific IP host. For example, 172.16.1.1/32. Only the intended host of a unicast packets receives and processes the packet. This term is often used in conjunction with references to types of IP broadcast traffic. For example, a network administrator considering upgrading a router in a network must consider the amount of unicast, multicast, and broadcast traffic because each type of traffic can have a different effect on the performance of the router.

IP Broadcast Address

IP broadcast packets are sent to the destination IP broadcast address 255.255.255.255 (or the older but still occasionally used IP broadcast address of 000.000.000.000). The broadcast destination IP addresses 255.255.255.255 and 000.000.000.000 are used when a packet is intended for every IP-enabled device on a network.



Note Packets that use the broadcast IP address as the destination IP address are known as broadcast packets.

If routers forwarded IP broadcast packets by default, the packets would have to be forwarded out every interface that is enabled for IP because the 255.255.255.255 IP destination address is assumed to be reachable via every IP enabled interface in the router. Forwarding IP broadcast packets out every interface that is enabled for IP would result in what is known as a broadcast storm (network overload due to high levels of broadcast traffic). In order to avoid the IP packet broadcast storm that would be created if a router forwarded packets with a broadcast IP destination address out every IP-enabled interface, the default behavior for a router is to *not* forward broadcast packets. This is a key difference between routing IP traffic at Layer 3 versus bridging it at Layer 2. Layer 2 bridges by default forward IP broadcast traffic out every interface that is in a forwarding state, which can lead to scalability problems.

Some TCP/IP protocols use the IP broadcast address to either communicate with all of the hosts on a network segment or to identify the IP address of a specific host on a network segment. For example:

- Routing Information Protocol (RIP) version 1 sends routing table information using the IP broadcast address so that any other host on the network segment running RIP version 1 can receive and process the updates.
- The Address Resolution Protocol (ARP) is used to determine the Layer 2 MAC address of the host that owns a specific Layer 3 IP address. ARP sends an IP broadcast packet (that is also a Layer 2 broadcast frame) on the local network. All of the hosts on the local network receive the ARP broadcast packet because it is sent to as a Layer 2 broadcast frame. All of the hosts on the local network process the ARP packet because it is sent to the IP broadcast address. Only the host that owns the IP address indicated in the data area of the ARP packet responds to the ARP broadcast packet.

IP Directed Broadcast Address

An IP directed broadcast is intended to reach all hosts on a remote network. A router that needs to send data to a remote IP host when only the IP network address is known uses an IP directed broadcast to reach the remote host. For example, a directed broadcast sent by a host with an IP address of 192.168.100.1 with a destination IP address of 172.16.255.255 is intended only for hosts that are in the 172.16.0.0 address space (hosts that have an IP address that begins with 172.16.0.0).

An IP directed broadcast packet is routed through the network as a unicast packet until it arrives at the target subnet, where it is converted into a Layer 2 broadcast frame (MAC address of FFFF.FFFF.FFFF). Because of the nature of the IP addressing architecture, only the last router in the chain, the one that is connected directly to the target subnet, can conclusively identify a directed broadcast. For example, only a router with an interface connected to a network using an IP address in the 172.16.0.0/16 address space such as 172.16.1.1/16 can determine that a packet sent to 172.16.255.255 is a directed broadcast and convert it to a Layer 2 broadcast that is received by all hosts on the local network. The other routers in the network that are not connected to the 172.16.0.0/16 network forward packets addressed to 172.16.255.255 as if they were for a specific IP host.

All of the hosts on the remote network receive IP directed broadcasts after they are converted to Layer 2 broadcast frames. Ideally only the intended destination host will fully process the IP directed broadcast and respond to it. However, IP directed broadcasts can be used for malicious purposes. For example, IP directed broadcasts are used in "smurf" Denial of Service (DoS) attack and derivatives thereof. In a "smurf" attack, the attacker sends Internet Control Message Protocol (ICMP) echo requests (pings) to a directed broadcast address using the source IP address of the device that is the target of the attack. The target is usually a host inside a company's network such as a web server. The ICMP echo requests are sent to an IP directed broadcast address in the company's network that causes all the hosts on the target subnet to send ICMP echo replies to the device under attack. By sending a continuous stream of such requests, the attacker can create a much larger stream of replies, which can completely inundate the host that is under attack. For information on how IP directed broadcasts are used in DoS attacks, search the Internet for "IP directed broadcasts," "denial of service," and "smurf attacks."

Due to the security implications of allowing a router to forward directed broadcasts and the reduction in applications that require directed broadcasts, IP directed broadcasts are disabled by default in Cisco IOS Release 12.0 and later releases. If your network requires support for IP directed broadcasts, you can enable it on the interfaces that you want to translate the IP directed broadcasts to Layer 2 broadcasts using the **ip directed-broadcast** command. For example, if your router is receiving IP directed broadcasts on Fast Ethernet interface 0/0 for the network address assigned to Fast Ethernet interface 0/1, and you want the IP directed broadcasts to be translated to Layer 2 broadcasts out interface Fast Ethernet interface 0/1, configure the **ip directed-broadcast** command on Fast Ethernet interface 0/1. You can specify an access list to control which IP directed broadcasts are translated to Layer 2 broadcasts. When an access list is specified, only those IP packets permitted by the access list are eligible to be translated from directed broadcasts to Layer 2 broadcasts. For example, if you know that the only legitimate source IP address of any IP directed broadcasts in your network is 192.168.10.2, create an extended IP access list allowing traffic from 192.168.10.2 and assign the access list with the **ip directed-broadcast access-list** command.

IP Directed Broadcasts

IP directed broadcasts are dropped by default. Dropping IP directed broadcasts reduces the risk of DoS attacks.

You can enable forwarding of IP directed broadcasts on an interface where the broadcast becomes a physical broadcast. You enable the translation of directed IP broadcast packets to Layer 2 broadcast frames on the interface that is connected to the IP network that the IP directed broadcast is addressed to. For example, if you need to translate IP directed broadcasts with the IP destination address of 172.16.10.255 to Layer 2 broadcast frames, you enable the translation on the interface that is connected to IP network 172.16.10.0/24.

You can specify an access list to control which directed broadcasts are forwarded. When an access list is specified, only those IP packets permitted by the access list are eligible to be translated from directed broadcasts to physical broadcasts.

IP directed broadcasts are disabled by default in Cisco IOS Release 12.0 and newer releases.

IP Multicast Addresses

IP multicast addresses are intended to reach an arbitrary subset of the hosts on a local network. IP broadcast addresses create a problem because every host must receive and process the data in each packet to determine if it contains information that the host must process further. IP multicast addresses resolve this problem by using well-known IP addresses that a host must be configured to recognize before it will process packets addressed to it. When a host receives an IP multicast packet, the host compares the IP multicast address with the list of multicast addresses it is configured to recognize. If the host is not configured to recognize the IP multicast address, the host ignores the packet instead of processing it further to analyze the data in the packet.

Because the host can ignore the packet it spends less time and fewer resources than it would have had to spend if the packet had been an IP broadcast that had to be processed all the way to the data layer before it was discarded.

The range of IP addresses reserved for Class D multicast addresses is 224.0.0.0 to 239.255.255.255/32 (255.255.255.255).

Most of the TCP/IP routing protocols use IP multicast addresses to send routing updates and other information to hosts on the same local network that are running the same routing protocol. Many other applications such as audio/video streaming over the Internet use IP multicast addresses. For a list of the currently assigned IP multicast addresses see [Internet Multicast Addresses](#).

Information on configuring network devices for IP multicast support is available in the following documentation:

- *Cisco IOS IP Multicast Configuration Guide*
- *Cisco IOS IP Multicast Command Reference*

Early IP Implementations

Several early IP implementations do not use the current broadcast address standard of 255.255.255.255. Instead, they use the old standard, which calls for all zeros (000.000.000.000) instead of all ones to indicate broadcast addresses. Many of these implementations do not recognize an all-1s broadcast address and fail to respond to the broadcast correctly. Others forward all-1s broadcasts by default, which causes a serious network overload known as a *broadcast storm*. Implementations that exhibit these problems include systems based on versions of Berkeley Standard Distribution (BSD) UNIX prior to Version 4.3.

DHCP and IPv4 Broadcast Packets

DHCP requires that the client (host requiring information from the DHCP server) send broadcast packets to find a DHCP server to request configuration information from. If the DHCP server is not on the same network segment as the client that is sending the DHCP broadcasts, the router must be configured to forward the DHCP requests to the appropriate network.

For more information on DHCP, see RFC 2131 *Dynamic Host Configuration Protocol*, at <http://www.ietf.org/rfc/rfc2131.txt>.

UDP Broadcast Packet Forwarding

UDP broadcast packets are used by TCP/IP protocols such as DHCP and applications that need to send the same data to multiple hosts concurrently. Because routers by default do not forward broadcast packets you need to customize your router's configuration if your network has UDP broadcast traffic on it. One option for forwarding UDP broadcast packets is to use the UDP forwarding feature. UDP forwarding rewrites the broadcast IP address of a UDP packet to either a unicast (specific host) IP address or a directed IP broadcast. After the address is rewritten the UDP packet is forwarded by all of the routers in the path to the destination network without requiring additional configuration changes on the other routers.

You can enable forwarding of UDP broadcast packets, such as DHCP requests, to a host, or to multiple hosts on the same target network. When a UDP broadcast packet is forwarded, the destination IP address is rewritten to match the address that you configure. For example, the **ip helper-address 172.16.10.2** command rewrites the IP destination address from 255.255.255.255 to 172.16.10.2.

To enable UDP broadcast packet forwarding to specific host, use a specific host IP address as the helper address when you configure the **ip helper-address address** command. To enable UDP broadcast packet forwarding to a range of hosts to allow for load sharing and redundancy, use an IP directed broadcast address as the helper address when you configure the **ip helper-address address** command.

UDP Broadcast Packet Flooding

You can allow IP broadcasts to be flooded throughout your network in a controlled fashion using the database created by the Layer 2 bridging Spanning Tree Protocol (STP). Enabling this feature also prevents flooding loops. In order to support this capability, the Cisco IOS software on your router must include support for transparent bridging, and transparent bridging must be configured on each interface that is to participate in the flooding. If bridging is not configured on an interface, the interface is still able to receive broadcasts. However, the interface will never forward broadcasts it receives, and the router will never use that interface to send broadcasts received on a different interface.

Packets that are forwarded to a single network address using the IP helper address mechanism can be flooded. Only one copy of the packet is sent on each network segment.

In order to be considered for flooding, packets must meet the following criteria. (These are the same conditions used to consider packet forwarding using IP helper addresses.)

- The packet must be a MAC-level broadcast (FFFF.FFFF.FFFF).
- The packet must be an IP-level broadcast (255.255.255.255).
- The packet must be a Trivial File Transfer Protocol (TFTP), Domain Name System (DNS), Time, NetBIOS, Neighbor Discovery (ND), or BOOTP packet, or a UDP protocol specified by the **ip forward-protocol udp** global configuration command.
- The time-to-live (TTL) value of the packet must be at least two.

If you want to send the flooded UDP packets to a specific host, you can change the Layer 3 IP broadcast address of the flooded UDP packets with the **ip broadcast-address** command in interface configuration mode. The address of the flooded UDP packets can be set to any desired IP address. The source address of the flooded UDP packet is never changed. The TTL value of the flooded UDP packet is decremented.

After a decision has been made to send the datagram out on an interface (and the destination IP address possibly changed), the datagram is handed to the normal IP output routines and is, therefore, subject to access lists if they are present on the output interface.

If no actual bridging is desired, you can configure a type-code bridging filter that will deny all packet types from being bridged. Refer to the "Configuring Transparent Bridging" module of the *Cisco IOS Bridging and IBM Networking Configuration Guide* for more information about using access lists to filter bridged traffic. The Spanning-Tree database is still available to the IP forwarding code to use for the flooding.

IP Broadcast Flooding Acceleration

You can accelerate flooding of UDP datagrams using the spanning-tree algorithm. Used in conjunction with the **ip forward-protocol spanning-tree** command in global configuration mode, this feature boosts the performance of spanning-tree-based UDP flooding by a factor of about four to five times. The feature, called *turbo flooding*, is supported over Ethernet interfaces configured for Advanced Research Projects Agency (ARPA) encapsulated, FDDI, and high-level data link control (HDLC)-encapsulated serial interfaces. However, it is not supported on Token Ring interfaces. As long as the Token Rings and the non-HDLC serial interfaces are not part of the bridge group being used for UDP flooding, turbo flooding will behave normally.

Default UDP Port Numbers

If a helper address is specified and UDP forwarding is enabled, broadcast packets destined to the following port numbers are forwarded by default:

- Time service (port 37)
- IEN-116 Name Service (port 42)
- TACACS service (port 49)
- Domain Naming System (port 53)
- BOOTP client and server packets (ports 67 and 68)
- TFTP (port 69)
- NetBIOS Name Server (port 137)
- NetBIOS Datagram Server (port 138)

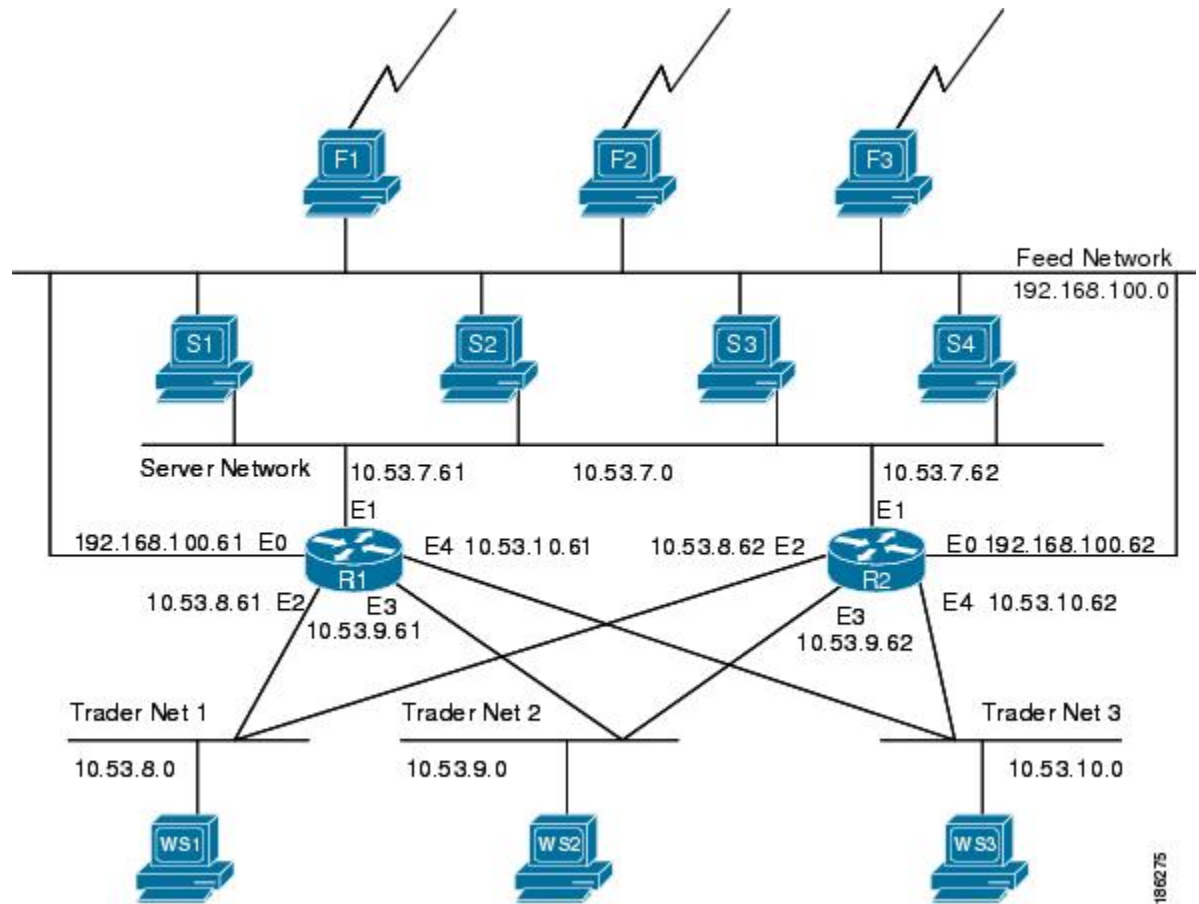
Default IP Broadcast Address

The Cisco IOS software supports sending IP broadcasts on both LANs and WANs. There are several ways to indicate an IP broadcast address. The default is an address consisting of all ones (255.255.255.255), although the software can be configured to generate any form of IP broadcast address such as all zeros (0.0.0.0), and directed broadcasts such as 172.16.255.255. Cisco IOS software can receive and process most IP broadcast addresses.

UDP Broadcast Packet Case Study

This case study is from a trading floor application in a financial company. The workstations (WS1, WS2, and WS3) in the following figure receive financial data from the feed network. The financial data is sent using UDP broadcasts.

Figure 3: Topology that Requires UDP Broadcast Forwarding

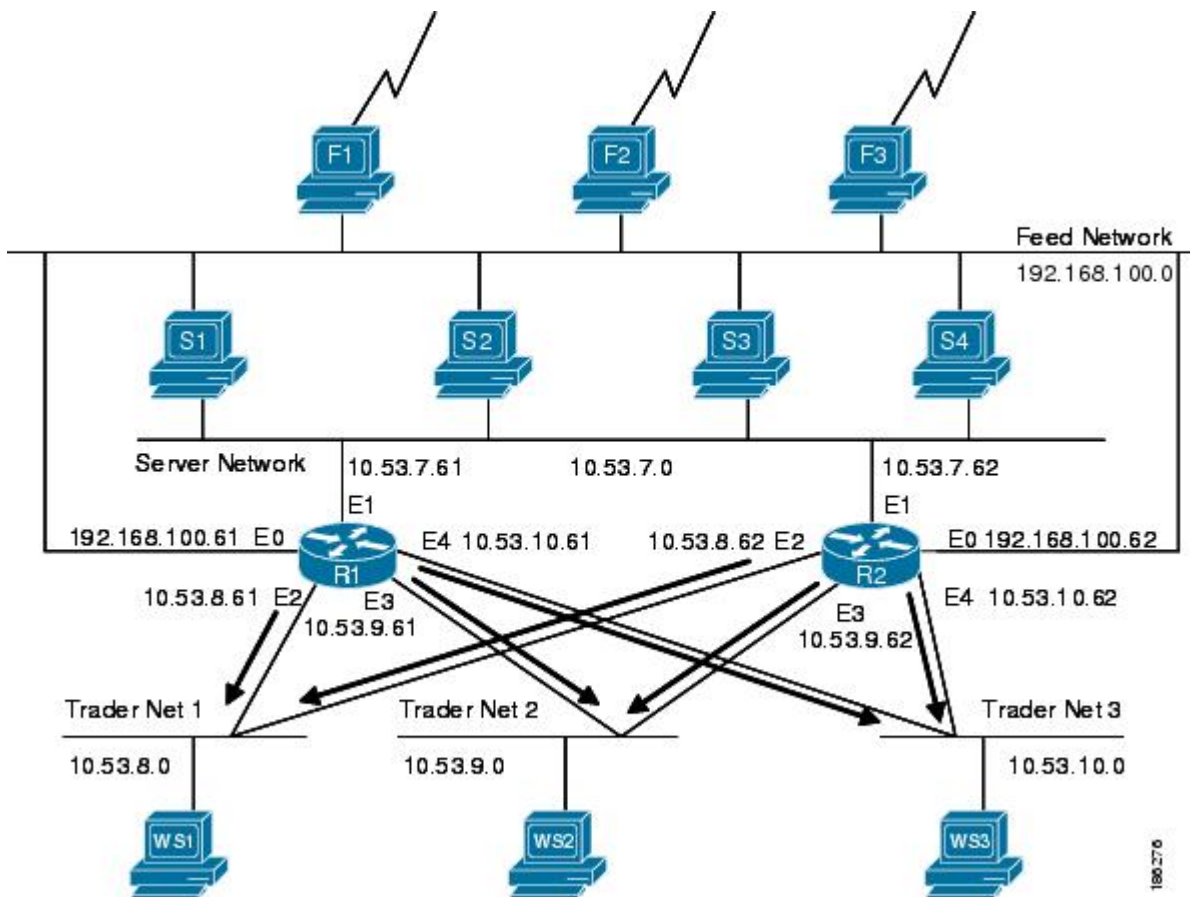


The following sections explain the possible solutions for this application:

UDP Broadcast Packet Forwarding

The first option is UDP broadcast packet using helper addresses. To configure helper addressing, you must specify the **ip helper-address** command on every interface on every router that receives a UDP broadcast that needs to be forwarded. On router 1 and router 2 in the figure below, IP helper addresses can be configured to move data from the server network to the trader networks. However IP helper addressing was determined not to be an optimal solution for this type of topology because each router receives unnecessary broadcasts from the other router, as shown in the figure below.

Figure 4: Flow of UDP Packets



In this case, router 1 receives each broadcast sent by router 2 three times, one for each segment, and router 2 receives each broadcast sent by router 1 three times, one for each segment. When each broadcast is received, the router must analyze it and determine that the broadcast does not need to be forwarded. As more segments are added to the network, the routers become overloaded with unnecessary traffic, which must be analyzed and discarded.

When IP helper addressing is used in this type of topology, no more than one router can be configured to forward UDP broadcasts (unless the receiving applications can handle duplicate broadcasts). This is because duplicate packets arrive on the trader network. This restriction limits redundancy in the design and can be undesirable in some implementations.

To configure routers to send UDP broadcasts bidirectionally in this type of topology, a second **ip helper address** command must be applied to every router interface that receives UDP broadcasts. As more segments and devices are added to the network, more **ip helper address** commands are required to reach them, so the administration of these routers becomes more complex over time.



Note Bidirectional traffic in this topology significantly impacts router performance.

Although IP helper addressing is well-suited to nonredundant, nonparallel topologies that do not require a mechanism for controlling broadcast loops, IP helper addressing does not work well in this topology. To improve performance, the network designers considered four other alternatives:

- Setting the broadcast address on the servers to all ones (255.255.255.255)—This alternative was dismissed because the servers have more than one interface, causing server broadcasts to be sent back onto the feed network. In addition, some workstation implementations do not allow all 1s broadcasts when multiple interfaces are present.
- Setting the broadcast address of the servers to the major network broadcast IP address—This alternative was dismissed because the TCP/IP implementation on the servers does not allow the use of major network IP broadcast addresses when the network is subnetted.
- Eliminating the subnets and letting the workstations use Address Resolution Protocol (ARP) to learn addresses—This alternative was dismissed because the servers cannot quickly learn an alternative route in the event of a primary router failure.
- UDP broadcast packet flooding—This alternative uses the spanning-tree topology created with transparent bridging to forward UDP broadcast packets in a redundant topology while avoiding loops and duplicate broadcast traffic.

UDP Broadcast Packet Flooding

UDP flooding uses the spanning-tree algorithm to forward packets in a controlled manner. Bridging is enabled on each router interface for the sole purpose of building the spanning tree. The spanning tree prevents loops by stopping a broadcast from being forwarded out an interface on which the broadcast was received. The spanning tree also prevents packet duplication by placing certain interfaces in the blocked state (so that no packets are forwarded) and other interfaces in the forwarding state (so that packets that need to be forwarded are forwarded).

Before you can enable UDP flooding, the router must be running software that supports transparent bridging and bridging must be configured on each interface that is to participate in the flooding. If bridging is not configured for an interface, the interface will receive broadcasts, but the router will not forward those broadcasts and will not use that interface as a destination for sending broadcasts received on a different interface.

When configured for UDP flooding, the router uses the destination address specified by the **ip broadcast-address** command on the output interface to assign a destination address to a flooded UDP datagram. Thus, the destination address might change as the datagram propagates through the network. The source address, however, does not change.

With UDP flooding, both routers shown in the figure below use a spanning-tree to control the network topology for the purpose of forwarding broadcasts. The **bridge protocol** command can specify either the **dec** keyword (for the Digital Equipment Corporation (DEC) spanning-tree protocol) or the **ieee** keyword (for the IEEE Ethernet protocol). All routers in the network must enable the same spanning-tree protocol. The **ip forward-protocol spanning-tree** command uses the database created by the **bridge protocol** command. Only one broadcast packet arrives at each segment, and UDP broadcasts can traverse the network in both directions.

Because bridging is enabled only to build the spanning-tree database, use access lists to prevent the spanning-tree from forwarding non-UDP traffic.

The router configuration specifies a path cost for each interface to determine which interface forwards or blocks packets. The default path cost for Ethernet is 100. Setting the path cost for each interface on router 2 to 50 causes the spanning-tree algorithm to place the interfaces in router 2 in forwarding state. Given the higher path cost (100) for the interfaces in router 1, the interfaces in router 1 are in the blocked state and do

not forward the broadcasts. With these interface states, broadcast traffic flows through router 2. If router 2 fails, the spanning-tree algorithm will place the interfaces in router 1 in the forwarding state, and router 1 will forward broadcast traffic.

With one router forwarding broadcast traffic from the server network to the trader networks, you should configure the other router to forward unicast traffic. For that reason, each router enables the ICMP Router Discovery Protocol (IRDP), and each workstation on the trader networks runs the IRDP daemon. On router 1, the **preference** keyword of the **ip irdp** command sets a higher IRDP preference than does the configuration for router 2, which causes each IRDP daemon to use router 1 as its preferred default gateway for unicast traffic forwarding. Users of those workstations can use the **netstat -rn** command to see how the routers are being used.

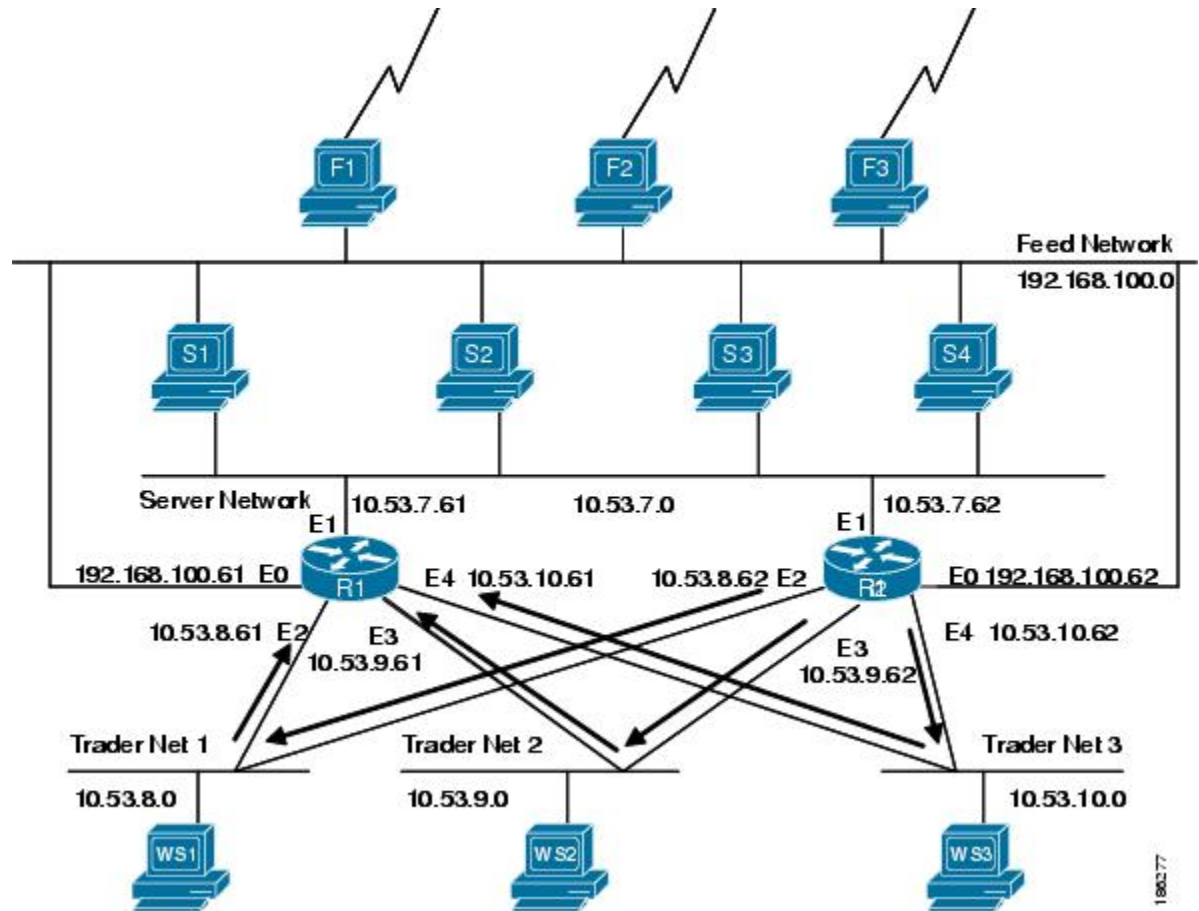
On the routers, the **holdtime**, **maxadvertinterval**, and **minadvertinterval** keywords of the **ip irdp** command reduce the advertising interval from the default so that the IRDP daemons running on the hosts expect to see advertisements more frequently. With the advertising interval reduced, the workstations will adopt router 2 more quickly if router 1 becomes unavailable. With this configuration, when a router becomes unavailable, IRDP offers a convergence time of less than one minute.

IRDP is preferred over the Routing Information Protocol (RIP) and default gateways for the following reasons:

- RIP takes longer to converge.
- Configuration of router 1 as the default gateway on each Sun workstation on the trader networks would allow those Sun workstations to send unicast traffic to router 1, but would not provide an alternative route if router 1 becomes unavailable.

The figure below shows how data flows when the network is configured for UDP flooding.

Figure 5: Data Flow with UDP Flooding and IRDP



Note This topology is broadcast intensive--broadcasts sometimes consume 20 percent of the 10-MB Ethernet bandwidth. However, this is a favorable percentage when compared to the configuration of IP helper addressing, which, in the same network, causes broadcasts to consume up to 50 percent of the 10-MB Ethernet bandwidth.

If the hosts on the trader networks do not support IRDP, Hot Standby Routing Protocol (HSRP), Virtual Router Redundancy Protocol (VRRP), or Gateway Load Balancing Protocol (GLBP) can be used to select which router will handle unicast traffic. These protocols allow the standby router to take over quickly if the primary router becomes unavailable.

Enable turbo flooding on the routers to increase the performance of UDP flooding.



Note Turbo flooding increases the amount of processing that is done at interrupt level, which increases the CPU load on the router. Turbo flooding may not be appropriate on routers that are already under high CPU load or that must also perform other CPU-intensive activities.

How to Configure IP Broadcast Packet Handling

Enabling IP Directed Broadcasts Without an Access List

Perform this task to permit the forwarding of IP directed broadcasts from any source.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *address mask*
5. **ip directed-broadcast**
6. **end**

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Step 1 | enable Example: Router> enable | Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted. |
| Step 2 | configure terminal Example: Device# configure terminal | Enters global configuration mode. |
| Step 3 | interface <i>type number</i> Example: Device(config)# interface fastethernet 0/1 | Specifies an interface and enters interface configuration mode. |
| Step 4 | ip address <i>address mask</i> Example: Device(config-if)# ip address 172.16.10.1 255.255.255.0 | Assigns an IP address to the interface. |
| Step 5 | ip directed-broadcast Example: Device(config-if)# ip directed-broadcast | Enables IP directed broadcasts on the interface. <ul style="list-style-type: none"> • Configure this command on the interface that is connected to the IP network address of the directed broadcast packets. • In this example the directed broadcast packets are addressed to 172.16.10.255. |

| | Command or Action | Purpose |
|--------|---------------------------------------------------------|---------------------------------------------------------------------------|
| Step 6 | end Example: Device(config-if)# end | Exits the current configuration mode and returns to privileged EXEC mode. |

Enabling IP Directed Broadcasts with an Access List

Perform this task to limit the forwarding of IP directed broadcasts by applying an access list to the **ip directed-broadcast** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list 100-199 permit ip source-address mask destination-address mask**
4. **interface type number**
5. **ip address address mask**
6. **ip directed-broadcast access-list**
7. **end**

DETAILED STEPS

| | Command or Action | Purpose |
|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Step 1 | enable Example: Router> enable | Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted. |
| Step 2 | configure terminal Example: Device# configure terminal | Enters global configuration mode. |
| Step 3 | access-list 100-199 permit ip source-address mask destination-address mask Example: Device(config)# access-list 100 permit ip 10.4.9.167 0.0.0.0 172.16.10.0 0.0.0.255 | Creates an access list to limit the IP directed broadcasts that are forwarded. <ul style="list-style-type: none"> • In this example the IP directed broadcasts are sent by the host with the IP address of 10.4.9.167 to the IP directed broadcast address 172.16.10.255. |
| Step 4 | interface type number Example: Device(config)# interface fastethernet 0/0 | Specifies an interface and enters interface configuration mode. |

| | Command or Action | Purpose |
|---------------|--------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Step 5 | ip address <i>address mask</i> Example: <pre>Device(config-if)# ip address 172.16.10.1 255.255.255.0</pre> | Assigns an IP address to the interface. |
| Step 6 | ip directed-broadcast <i>access-list</i> Example: <pre>Device(config-if)# ip directed-broadcast 100</pre> | Enables IP directed broadcasts on the interface for broadcast packets that are allowed by the access list you assigned. Configure this command on the interface that is connected to the IP network address of the directed broadcast packets. <ul style="list-style-type: none"> In this example the directed broadcast packets are addressed to 172.16.10.255. |
| Step 7 | end Example: <pre>Device(config-if)# end</pre> | Exits the current configuration mode and returns to privileged EXEC mode. |

Enabling Forwarding of UDP Broadcast Packets to a Specific Host

Perform this task to enable UDP broadcast packet forwarding to a single host.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip forward-protocol udp**
4. **interface** *type number*
5. **ip address** *address mask*
6. **ip helper-address** *address*
7. **end**

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|---------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------|
| Step 1 | enable Example: <pre>Router> enable</pre> | Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted. |
| Step 2 | configure terminal Example: <pre>Device# configure terminal</pre> | Enters global configuration mode. |

| | Command or Action | Purpose |
|---------------|-----------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Step 3 | ip forward-protocol udp Example: Device(config)# ip forward-protocol udp | Enables forwarding of UDP broadcast packets. |
| Step 4 | interface type number Example: Device(config)# interface fastethernet 0/1 | Specifies an interface and enters interface configuration mode. |
| Step 5 | ip address address mask Example: Device(config-if)# ip address 172.16.10.1 255.255.255.0 | Assigns an IP address to the interface. |
| Step 6 | ip helper-address address Example: Device(config-if)# ip helper-address 172.16.10.2 | Enables an IP helper address for the interface that is receiving the UDP broadcast packets. <ul style="list-style-type: none"> • In this example the IP destination address of the IP UDP broadcast packets is rewritten to 172.16.10.2. |
| Step 7 | end Example: Device(config-if)# end | Exits the current configuration mode and returns to privileged EXEC mode. |

Enabling Forwarding of UDP Broadcast Packets to a Range of Hosts

Perform this task to enable UDP broadcast packet forwarding to a range of hosts to allow for load sharing between the destination hosts and to provide redundancy if one or more of the destination hosts fail.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip forward-protocol udp**
4. **interface type number**
5. **ip address address mask**
6. **ip helper-address address**
7. **exit**
8. **interface type number**
9. **ip address address mask**
10. **ip directed-broadcast**
11. **end**

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Step 1 | enable Example: <pre>Router> enable</pre> | Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted. |
| Step 2 | configure terminal Example: <pre>Device# configure terminal</pre> | Enters global configuration mode. |
| Step 3 | ip forward-protocol udp Example: <pre>Device(config)# ip forward-protocol udp</pre> | Enables forwarding of UDP broadcast packets. |
| Step 4 | interface <i>type number</i> Example: <pre>Device(config)# interface fastethernet 0/0</pre> | Specifies an interface and enters interface configuration mode. |
| Step 5 | ip address <i>address mask</i> Example: <pre>Device(config-if)# ip address 192.168.10.1 255.255.255.0</pre> | Assigns an IP address to the interface. |
| Step 6 | ip helper-address <i>address</i> Example: <pre>Device(config-if)# ip helper-address 172.16.10.255</pre> | Enables an IP helper address for the interface that is receiving the UDP broadcast packets. <ul style="list-style-type: none"> • In this example an IP directed broadcast address is used. The IP destination address of the IP UDP broadcast packets is rewritten to 172.16.10.255. • All of the hosts on the 172.16.10.0/24 network that support the application or service that the UDP broadcast packets are intended for will respond to the UDP broadcast packets. <p>Note This often results in the source of the UDP broadcast packets receiving responses from two or more hosts. In most circumstances the source of the UDP broadcast packets accepts the first response and ignores any subsequent responses. In some situations the source of the UDP broadcast packets cannot handle duplicate responses and reacts by reloading, or other unexpected behavior.</p> |

| | Command or Action | Purpose |
|----------------|---------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------|
| Step 7 | exit Example: Device(config-if)# exit | Returns to global configuration mode. |
| Step 8 | interface type number Example: Device(config)# interface fastethernet 0/1 | Specifies an interface and enters interface configuration mode. |
| Step 9 | ip address address mask Example: Device(config-if)# ip address 172.16.10.1 255.255.255.0 | Assigns an IP address to the interface. |
| Step 10 | ip directed-broadcast Example: Device(config-if)# ip directed-broadcast | Enables IP directed broadcasts on the interface that is transmitting the UDP broadcasts. |
| Step 11 | end Example: Device(config-if)# end | Exits the current configuration mode and returns to privileged EXEC mode. |

Changing the Default IP Broadcast Address for All Interfaces to 0.0.0.0 on Routers Without Nonvolatile Memory

If your router does not have NVRAM, and you need to change the IP broadcast address to 0.0.0.0, you must change the IP broadcast address manually by setting jumpers in the processor configuration register. Setting bit 10 causes the device to use all 0s. Bit 10 interacts with bit 14, which controls the network and host portions of the broadcast address. Setting bit 14 causes the device to include the network and host portions of its address in the broadcast address. The table below shows the combined effect of setting bits 10 and 14.

Table 5: Configuration Register Settings for Broadcast Address Destination

| Bit 14 | Bit 10 | Address (<net><host>) |
|--------|--------|-----------------------|
| Out | Out | <ones><ones> |
| Out | In | <zeros><zeros> |
| In | In | <net><zeros> |
| In | Out | <net><ones> |

For additional information on setting the hardware jumpers on your router, see the hardware documentation that was supplied with your router.

Changing the Default IP Broadcast Address for All Interfaces to 0.0.0.0 on Routers with Nonvolatile Memory

Cisco IOS-based routers with NVRAM have software configuration registers that allow you to modify several behaviors of the router such as where it looks for images to load, what IP broadcast address it uses, and the console line speed. The factory default value for the configuration register is 0x2102 where *0X* indicates this a hexadecimal number. The **config-register** command is used to modify the settings of the software configuration registers.

Information on configuring other behaviors with the software configuration registers using the **config-register** command is available in the following documentation:

- "Loading and Managing System Images" chapter of the *Cisco IOS Configuration Fundamentals Configuration Guide*
- *Cisco IOS Configuration Fundamentals Command Reference*



Caution

You need to be very careful when you change the software configuration registers on your router because if you inadvertently alter the console port line speed, you will not be able to configure the router with a terminal server on the console port unless you know the speed that you set for the console port, and you know how to change the line speed for your terminal application. If your router is configured for alternate access to the CLI such as using Telnet or a web browser, you can use this method to log in to the router and change the software configuration register back to 0x2102.

Perform this task to set the IP broadcast address on every interface to 0.0.0.0 while maintaining the remainder of the default values for the software configuration register settings.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **config-register** *value*
4. **end**

DETAILED STEPS

| | Command or Action | Purpose |
|--------|----------------------------------------------------|--------------------------------------------------------------------------------------------------------------------|
| Step 1 | enable Example: Router> enable | Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted. |
| Step 2 | configure terminal Example: | Enters global configuration mode. |

| | Command or Action | Purpose |
|---------------|--------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | Device# configure terminal | |
| Step 3 | config-register <i>value</i> Example: Device(config)# config-register 0x2502 | Sets the IP broadcast address to 0.0.0.0 on every interface while maintaining the remainder of the default values for the other software configuration register settings. |
| Step 4 | end Example: Device(config)# end | Exits the current configuration mode and returns to privileged EXEC mode. |

Changing the IP Broadcast Address to Any IP Address on One or More Interfaces in a Router

Perform this task if you network requires an IP broadcast address other than 255.255.255.255 or 0.0.0.0, or you want to change the IP broadcast address to 0.0.0.0 on a subset of the interfaces on the router instead of on all of the interfaces on the router.

SUMMARY STEPS

1. enable
2. configure terminal
3. interface *type number*
4. ip address *address mask*
5. ip broadcast-address *address*
6. end

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------|
| Step 1 | enable Example: Router> enable | Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted. |
| Step 2 | configure terminal Example: Device# configure terminal | Enters global configuration mode. |
| Step 3 | interface <i>type number</i> Example: Device(config)# interface fastethernet 0/1 | Specifies an interface and enters interface configuration mode. |

| | Command or Action | Purpose |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| Step 4 | ip address <i>address mask</i> Example: <pre>Device(config-if)# ip address 172.16.10.1 255.255.255.0</pre> | Assigns an IP address to the interface. |
| Step 5 | ip broadcast-address <i>address</i> Example: <pre>Device(config-if)# ip broadcast-address 172.16.10.255</pre> | Specifies the IP broadcast address <ul style="list-style-type: none"> • In this example IP broadcasts are sent to 172.16.10.255. |
| Step 6 | end Example: <pre>Device(config-if)# end</pre> | Exits the current configuration mode and returns to privileged EXEC mode. |

Configuring UDP Broadcast Packet Flooding

Before you begin

The version of Cisco IOS software on your router must support transparent bridging.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **bridge** *number* **protocol ieee**
4. **ip forward-protocol spanning-tree**
5. **ip forward-protocol turbo-flood**
6. **ip forward-protocol udp**
7. **interface** *type number*
8. **ip address** *address mask*
9. **bridge-group** *number*
10. **interface** *type number*
11. **ip address** *address mask*
12. **bridge-group** *number*
13. **end**

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|----------------------------------|--------------------------------------------------------------------------------------------------------------------|
| Step 1 | enable Example: | Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted. |

| | Command or Action | Purpose |
|----------------|-------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------|
| | Router> enable | |
| Step 2 | configure terminal Example: Device# configure terminal | Enters global configuration mode. |
| Step 3 | bridge <i>number</i> protocol ieee Example: Device(config)# bridge 1 protocol ieee | Enables spanning-tree bridging and specifies the bridging protocol. |
| Step 4 | ip forward-protocol spanning-tree Example: Device(config)# ip forward-protocol spanning-tree | Enables using the spanning-tree forwarding table to flood broadcast packets. |
| Step 5 | ip forward-protocol turbo-flood Example: Device(config)# ip forward-protocol turbo-flood | (Optional) Enables fast forwarding of broadcast packets using the spanning-tree forwarding table. |
| Step 6 | ip forward-protocol udp Example: Device(config)# ip forward-protocol udp | Enables forwarding of UDP broadcasts. |
| Step 7 | interface <i>type number</i> Example: Device(config)# interface fastethernet 0/0 | Specifies an interface and enters interface configuration mode. |
| Step 8 | ip address <i>address mask</i> Example: Device(config-if)# ip address 192.168.10.1 255.255.255.0 | Assigns an IP address to the interface. |
| Step 9 | bridge-group <i>number</i> Example: Device(config-if)# bridge-group 1 | Places the interface in the spanning-tree bridge group specified. |
| Step 10 | interface <i>type number</i> Example: Device(config-if)# interface fastethernet 0/1 | Specifies an interface and enters interface configuration mode. |

| | Command or Action | Purpose |
|----------------|------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------|
| Step 11 | ip address <i>address mask</i> Example: Device(config-if)# ip address 172.16.10.1 255.255.255.0 | Assigns an IP address to the interface. |
| Step 12 | bridge-group <i>number</i> Example: Device(config-if)# bridge-group 1 | Places the interface in the spanning-tree bridge group specified. |
| Step 13 | end Example: Device(config-if)# end | Exits the current configuration mode and returns to privileged EXEC mode. |

Configuration Examples for IP Broadcast Packet Handling

Example: Enabling IP Directed Broadcasts with an Access List

The following example shows how to enable IP directed broadcasts with an access list to control the directed broadcasts that are forwarded.

```
Device(config)# access-list 100 permit ip 10.4.9.167 0.0.0.0 172.16.10.0 0.0.0.255
Device(config)# interface fastethernet 0/0
Device(config-if)# ip address 172.16.10.1 255.255.255.0
Device(config-if)# ip directed-broadcast 100
```

Example: Configuring UDP Broadcast Packet Flooding

```
Device(config)# bridge 1 protocol ieee
Device(config)# ip forward-protocol spanning-tree
Device(config)# ip forward-protocol turbo-flood
Device(config)# ip forward-protocol udp
Device(config)# interface fastethernet 0/0
Device(config-if)# ip address 192.168.10.1 255.255.255.0
Device(config-if)# bridge-group 1
Device(config)# interface fastethernet 0/1
Device(config-if)# ip address 172.16.10.1 255.255.255.0
Device(config-if)# bridge-group 1
```

Additional References for WCCP—Configurable Router ID

Related Documents

| Related Topic | Document Title |
|-----------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Cisco IOS commands | Cisco IOS Master Commands List, All Releases |
| WCCP commands | Cisco IOS IP Application Services Command Reference |
| Currently assigned IP multicast addresses | <i>Internet Multicast Addresses</i> http://www.iana.org/assignments/multicast-addresses |
| Configuration fundamentals configuration tasks | <i>Cisco IOS Configuration Fundamentals Configuration Guide</i> |
| Configuration fundamentals commands | <i>Cisco IOS Configuration Fundamentals Command Reference</i> |
| Cisco IOS bridging and IBM networking configuration tasks | <i>Cisco IOS Bridging and IBM Networking Configuration Guide</i> |
| Cisco IOS bridging and IBM networking commands | <i>Cisco IOS Bridging and IBM Networking Command Reference</i> |
| Cisco IOS IP multicast configuration tasks | <i>Cisco IOS IP Multicast Configuration Guide</i> |
| Cisco IOS IP Multicast commands | <i>Cisco IOS IP Multicast Command Reference</i> |

Standards

| Standard | Title |
|-----------------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| IEEE Spanning-Tree Bridging | 802.1D MAC Bridges http://www.ieee802.org/1/pages/802.1D-2003.html |

MIBs

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

RFCs

| RFC | Title |
|----------|------------------------------------------------------------------------------------------------------------------------------------|
| RFC 1812 | <i>Requirements for IP Version 4 Routers</i> http://www.ietf.org/rfc/rfc1812.txt |
| RFC 2131 | <i>Dynamic Host Configuration Protocol</i> http://www.ietf.org/rfc/rfc2131.txt |

Technical Assistance

| Description | Link |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|
| The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password. | http://www.cisco.com/cisco/web/support/index.html |

Feature Information for IP Broadcast Packet Handling

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

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

Table 6: Feature Information for IP Broadcast Packet Handling

| Feature Name | Releases | Feature Information |
|--------------------------------------|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| IP Directed Broadcasts | 10.0 | Enables the translation of a directed broadcast to physical broadcasts. The following command was introduced or modified by this feature: ip directed-broadcast . |
| UDP Broadcast Packet Forwarding | 10.0 | Enables the forwarding of UDP broadcast packets. The following commands were introduced or modified by this feature: ip forward-protocol , ip helper-address . |
| Flooding Packets Using spanning-tree | 10.0 | Enables the forwarding of UDP broadcast packets using the spanning-tree forwarding table. The following commands were introduced or modified by this feature: ip forward-protocol spanning-tree , ip forward-protocol turbo-flood . |
| Specifying an IP Broadcast Address | 10.0 | Specifies the IP broadcast address for an interface. The following command was introduced or modified by this feature: ip broadcast-address . |



CHAPTER 4

Configuring TCP

TCP is a protocol that specifies the format of data and acknowledgments used in data transfer. TCP is a connection-oriented protocol because participants must establish a connection before data can be transferred. By performing flow control and error correction, TCP guarantees reliable, in-sequence delivery of packets. TCP is considered a reliable protocol because it will continue to request an IP packet that is dropped or received out of order until it is received. This module explains concepts related to TCP and how to configure TCP in a network.

- [Finding Feature Information, on page 77](#)
- [Prerequisites for TCP, on page 77](#)
- [Restrictions for TCP, on page 78](#)
- [Information About TCP, on page 78](#)
- [How to Configure TCP, on page 83](#)
- [Configuration Examples for TCP, on page 92](#)
- [Additional References, on page 96](#)
- [Feature Information for TCP, on page 97](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

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

Prerequisites for TCP

TCP Time Stamp, TCP Selective Acknowledgment, and TCP Header Compression

Because TCP time stamps are always sent and echoed in both directions and the time-stamp value in the header is always changing, TCP header compression will not compress the outgoing packet. To allow TCP header compression over a serial link, the TCP time-stamp option is disabled. If you want to use TCP header compression over a serial line, TCP time stamp and TCP selective acknowledgment must be disabled. Both

features are disabled by default. Use the **no ip tcp selective-ack** command to disable the TCP selective acknowledgment once it is enabled.

Restrictions for TCP

The TCP Keepalive timer parameters can be configured only on vty and TTY applications.

Information About TCP

TCP Services

TCP provides reliable transmission of data in an IP environment. TCP corresponds to the transport layer (Layer 4) of the Open Systems Interconnection (OSI) reference model. Among the services that TCP provides are stream data transfer, reliability, efficient flow control, full-duplex operation, and multiplexing.

With stream data transfer, TCP delivers an unstructured stream of bytes that are identified by sequence numbers. This service benefits applications because they do not have to divide data into blocks before handing it off to TCP. Instead, TCP groups bytes into segments and passes them to IP for delivery.

TCP offers reliability by providing connection-oriented, end-to-end reliable packet delivery through an internetwork. It does this by sequencing bytes with a forwarding acknowledgment number that indicates to the destination the next byte that the source expects to receive. Bytes that are not acknowledged within a specified time period are retransmitted. The reliability mechanism of TCP allows devices to handle lost, delayed, duplicate, or misread packets. A timeout mechanism allows devices to detect lost packets and request retransmission.

TCP offers efficient flow control, which means that the receiving TCP process indicates the highest sequence number that it can receive without overflowing its internal buffers when sending acknowledgments back to the source.

TCP offers full-duplex operation, and TCP processes can both send and receive data at the same time.

TCP multiplexing allows numerous simultaneous upper-layer conversations to be multiplexed over a single connection.

TCP Connection Establishment

To use reliable transport services, TCP hosts must establish a connection-oriented session with one another. Connection establishment is performed by using a “three-way handshake” mechanism.

A three-way handshake synchronizes both ends of a connection by allowing both sides to agree upon the initial sequence numbers. This mechanism guarantees that both sides are ready to transmit data. The three-way handshake is necessary so that packets are not transmitted or retransmitted during session establishment or after session termination.

Each host randomly chooses a sequence number, which is used to track bytes within the stream that the host is sending. The three-way handshake proceeds in the following manner:

- The first host (Host A) initiates a connection by sending a packet with the initial sequence number (X) and the synchronize/start (SYN) bit set to indicate a connection request.

- The second host (Host B) receives the SYN, records the sequence number X, and replies by acknowledging (ACK) the SYN (with an ACK = X + 1). Host B includes its own initial sequence number (SEQ = Y). An ACK = 20 means that the host has received bytes 0 through 19 and expects byte 20 next. This technique is called forward acknowledgment.
- Host A acknowledges all bytes that Host B has sent with a forward acknowledgment indicating the next byte Host A expects to receive (ACK = Y + 1). Data transfer can then begin.

TCP Connection Attempt Time

You can set the amount of time the software will wait before attempting to establish a TCP connection. The connection attempt time is a host parameter and pertains to traffic that originated at the device and not to traffic going through the device. To set the TCP connection attempt time, use the **ip tcp synwait-time** command in global configuration mode. The default is 30 seconds.

TCP Selective Acknowledgment

The TCP Selective Acknowledgment feature improves performance if multiple packets are lost from one TCP window of data.

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

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

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

TCP selective acknowledgment is used only when multiple packets are dropped within one TCP window. There is no performance impact when the feature is enabled but not used. Use the **ip tcp selective-ack** command in global configuration mode to enable TCP selective acknowledgment.

Refer to RFC 2018 for more details about TCP selective acknowledgment.

TCP Time Stamp

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

Refer to RFC 1323 for more details on TCP time stamps.

TCP Maximum Read Size

The maximum number of characters that TCP reads from the input queue for Telnet and relogin at one time is very large (the largest possible 32-bit positive number) by default. To change the TCP maximum read size value, use the **ip tcp chunk-size** command in global configuration mode.



Note We do not recommend that you change this value.

TCP Path MTU Discovery

Path MTU Discovery is a method for maximizing the use of the available bandwidth in the network between endpoints of a TCP connection, which is described in RFC 1191. IP Path MTU Discovery allows a host to dynamically discover and cope with differences in the maximum allowable maximum transmission unit (MTU) size of the various links along the path. Sometimes a device is unable to forward a datagram because it requires fragmentation (the packet is larger than the MTU that you set for the interface with the **interface** configuration command), but the “do not fragment” (DF) bit is set. The intermediate gateway sends a “Fragmentation needed and DF bit set” Internet Control Message Protocol (ICMP) message to the sending host, alerting the host to the problem. On receiving this message, the host reduces its assumed path MTU and consequently sends a smaller packet that will fit the smallest packet size of all links along the path.

By default, TCP Path MTU Discovery is disabled. Existing connections are not affected irrespective of whether this feature is enabled or disabled.

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

Use the **ip tcp path-mtu-discovery** global configuration command to enable Path MTU Discovery for connections initiated by the device when the device is acting as a host.

For more information about Path MTU Discovery, refer to the “Configuring IP Services” module of the *IP Application Services Configuration Guide*.

TCP Window Scaling

The TCP Window Scaling feature adds support for the Window Scaling option in RFC 1323, *TCP Extensions for High Performance*. A larger window size is recommended to improve TCP performance in network paths with large bandwidth-delay product characteristics that are called Long Fat Networks (LFNs). The TCP Window Scaling enhancement provides LFN support.

The window scaling extension expands the definition of the TCP window to 32 bits and then uses a scale factor to carry this 32-bit value in the 16-bit window field of the TCP header. The window size can increase to a scale factor of 14. Typical applications use a scale factor of 3 when deployed in LFNs.

The TCP Window Scaling feature complies with RFC 1323. The maximum window size was increased to 1,073,741,823 bytes. The larger scalable window size will allow TCP to perform better over LFNs. Use the **ip tcp window-size** command in global configuration mode to configure the TCP window size.

TCP Sliding Window

A TCP sliding window provides an efficient use of network bandwidth because it enables hosts to send multiple bytes or packets before waiting for an acknowledgment.

In TCP, the receiver specifies the current window size in every packet. Because TCP provides a byte-stream connection, window sizes are expressed in bytes. A window is the number of data bytes that the sender is allowed to send before waiting for an acknowledgment. Initial window sizes are indicated at connection setup, but might vary throughout the data transfer to provide flow control. A window size of zero means “Send no data.” The default TCP window size is 4128 bytes. We recommend that you keep the default value unless your device is sending large packets (greater than 536 bytes). Use the **ip tcp window-size** command to change the default window size.

In a TCP sliding-window operation, for example, the sender might have a sequence of bytes to send (numbered 1 to 10) to a receiver who has a window size of five. The sender then places a window around the first five bytes and transmits them together. The sender then waits for an acknowledgment.

The receiver responds with an ACK = 6, indicating that it has received bytes 1 to 5 and is expecting byte 6 next. In the same packet, the receiver indicates that its window size is 5. The sender then moves the sliding window five bytes to the right and transmits bytes 6 to 10. The receiver responds with an ACK = 11, indicating that it is expecting sequenced byte 11 next. In this packet, if the receiver indicates that its window size is 0, the sender cannot send any more bytes until the receiver sends another packet with a window size greater than 0.

TCP Outgoing Queue Size

The default TCP outgoing queue size per connection is five segments if the connection has a TTY associated with it (such as a Telnet connection). If no TTY connection is associated with a connection, the default queue size is 20 segments. Use the **ip tcp queuemax** command to change the five-segment default value.

TCP Congestion Avoidance

The TCP Congestion Avoidance feature enables the monitoring of acknowledgment packets to the TCP sender when multiple packets are lost in a single window of data. Previous to introduction of this feature, the sender would exit Fast-Recovery mode, wait for three or more duplicate acknowledgment packets before retransmitting the next unacknowledged packet, or wait for the retransmission timer to start slowly. This delay could lead to performance issues.

Implementation of RFC 2581 and RFC 3782 addresses the modifications to the Fast-Recovery algorithm that incorporates a response to partial acknowledgments received during Fast Recovery, improving performance in situations where multiple packets are lost in a single window of data.

This feature is an enhancement to the existing Fast Recovery algorithm. No commands are used to enable or disable this feature.

The output of the **debug ip tcp transactions** command has been enhanced to monitor acknowledgment packets by showing the following conditions:

- TCP entering Fast Recovery mode.
- Duplicate acknowledgments being received during Fast Recovery mode.
- Partial acknowledgments being received.

TCP Explicit Congestion Notification

The TCP Explicit Congestion Notification (ECN) feature allows an intermediate router to notify end hosts of impending network congestion. It also provides enhanced support for TCP sessions associated with applications, such as Telnet, web browsing, and transfer of audio and video data that are sensitive to delay or packet loss. The benefit of this feature is the reduction of delay and packet loss in data transmissions. Use the **ip tcp ecn** command in global configuration mode to enable TCP ECN.

TCP MSS Adjustment

The TCP MSS Adjustment feature enables the configuration of the maximum segment size (MSS) for transient packets that traverse a device, specifically TCP segments with the SYN bit set. Use the **ip tcp adjust-mss** command in interface configuration mode to specify the MSS value on the intermediate device of the SYN packets to avoid truncation.

When a host (usually a PC) initiates a TCP session with a server, the host negotiates the IP segment size by using the MSS option field in the TCP SYN packet. The value of the MSS field is determined by the MTU configuration on the host. The default MSS value for a PC is 1500 bytes.

The PPP over Ethernet (PPPoE) standard supports a Maximum Transmission Unit (MTU) of only 1492 bytes. The disparity between the host and PPPoE MTU size can cause the device in between the host and the server to drop 1500-byte packets and terminate TCP sessions over the PPPoE network. Even if the path MTU (which detects the correct MTU across the path) is enabled on the host, sessions may be dropped because system administrators sometimes disable ICMP error messages that must be relayed from the host for path MTU to work.

The **ip tcp adjust-mss** command helps prevent TCP sessions from being dropped by adjusting the MSS value of the TCP SYN packets.

The **ip tcp adjust-mss** command is effective only for TCP connections passing through the device.

In most cases, the optimum value for the *max-segment-size* argument of the **ip tcp adjust-mss** command is 1452 bytes. This value plus the 20-byte IP header, the 20-byte TCP header, and the 8-byte PPPoE header add up to a 1500-byte packet that matches the MTU size for the Ethernet link.

See the “Configuring the MSS Value and MTU for Transient TCP SYN Packets” section for configuration instructions.

IPv6 TCP Traffic

Due to the differences in the network layer (IP) headers between IPv4 and IPv6, extra overhead such as tunnel headers may be added during the IPv6 traffic path and this may cause IP fragmentation. IPv6 path MTU (PMTU) detects the MTU and then the sender does IPv6 fragmentation.

The **ipv6 tcp adjust-mss** command allows the TCP MSS Adjustment feature to be enabled on IPv6 traffic.

TCP Applications Flags Enhancement

The TCP Applications Flags Enhancement feature enables the user to display additional flags with reference to TCP applications. There are two types of flags: status and option. The status flags indicate the status of TCP connections such as passive open, active open, retransmission timeout, and app closed for listening. The additional flags indicate the state of set options such as whether a VPN routing and forwarding instance (VRF) is set, whether a user is idle, and whether a keepalive timer is running. Use the **show tcp** command to display TCP application flags.

TCP Show Extension

The TCP Show Extension feature introduces the capability to display addresses in IP format instead of the hostname format and to display the VRF table associated with the connection. To display the status for all endpoints with addresses in IP format, use the **show tcp brief numeric** command.

TCP MIB for RFC 4022 Support

The TCP MIB for RFC 4022 Support feature introduces support for RFC 4022, *Management Information Base for the Transmission Control Protocol (TCP)*. RFC 4022 is an incremental change of the TCP MIB to improve the manageability of TCP.

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>

TCP Keepalive Timer

The TCP Keepalive Timer feature provides a mechanism to identify dead connections.

When a TCP connection on a routing device is idle for too long, the device sends a TCP keepalive packet to the peer with only the Acknowledgment (ACK) flag turned on. If a response packet (a TCP ACK packet) is not received after the device sends a specific number of probes, the connection is considered dead and the device initiating the probes frees resources used by the TCP connection.

The following parameters are used to configure TCP keepalive:

- TCP Keepalive idle time—The value of this parameter indicates the time for which a TCP connection can be idle before the connection initiates keepalive probes.
- TCP Keepalive retries—The value of this parameter is the number of unacknowledged probes that a device can send before declaring the connection as dead and tearing it down.
- TCP Keepalive interval—The time between subsequent probe retries.

How to Configure TCP

Configuring TCP Performance Parameters

Before you begin

Both sides of the network link must be configured to support window scaling or the default of 65,535 bytes will be applied as the maximum window size. To support Explicit Congestion Notification (ECN), the remote peer must be ECN-enabled because the ECN capability is negotiated during a three-way handshake with the remote peer.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **ip tcp synwait-time** *seconds*
4. **ip tcp path-mtu-discovery** [**age-timer** {*minutes* | **infinite**}]
5. **ip tcp selective-ack**
6. **ip tcp timestamp**
7. **ip tcp chunk-size** *characters*
8. **ip tcp window-size** *bytes*
9. **ip tcp ecn**
10. **ip tcp queuemax** *packets*
11. **end**

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Step 1 | enable Example: Device> enable | Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted. |
| Step 2 | configure terminal Example: Device# configure terminal | Enters global configuration mode. |
| Step 3 | ip tcp synwait-time <i>seconds</i> Example: Device(config)# ip tcp synwait-time 60 | (Optional) Sets the amount of time the Cisco software will wait before attempting to establish a TCP connection. <ul style="list-style-type: none">• The default is 30 seconds. |
| Step 4 | ip tcp path-mtu-discovery [age-timer { <i>minutes</i> infinite }] Example: Device(config)# ip tcp path-mtu-discovery age-timer 11 | (Optional) Enables Path MTU Discovery. <ul style="list-style-type: none">• age-timer —Time interval, in minutes, TCP reestimates the Maximum Transmission Unit (MTU) with a larger Maximum Segment Size (MSS). The default is 10 minutes. The maximum is 30 minutes.• infinite—Disables the age timer. |
| Step 5 | ip tcp selective-ack Example: Device(config)# ip tcp selective-ack | (Optional) Enables TCP selective acknowledgment. |
| Step 6 | ip tcp timestamp Example: Device(config)# ip tcp timestamp | (Optional) Enables the TCP time stamp. |
| Step 7 | ip tcp chunk-size <i>characters</i> Example: | (Optional) Sets the TCP maximum read size for Telnet or rlogin. |

| | Command or Action | Purpose |
|----------------|------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | Device(config)# ip tcp chunk-size 64000 | Note We do not recommend that you change this value. |
| Step 8 | ip tcp window-size bytes Example: Device(config)# ip tcp window-size 75000 | (Optional) Sets the TCP window size. <ul style="list-style-type: none"> The <i>bytes</i> argument can be set to an integer from 68 to 1073741823. To enable window scaling to support Long Flat Networks (LFNs), the TCP window size must be more than 65535. The default window size is 4128 if window scaling is not configured. Note With CSCsw45317, the <i>bytes</i> argument can be set to an integer from 68 to 1073741823. |
| Step 9 | ip tcp ecn Example: Device(config)# ip tcp ecn | (Optional) Enables ECN for TCP. |
| Step 10 | ip tcp queuemax packets Example: Device(config)# ip tcp queuemax 10 | (Optional) Sets the TCP outgoing queue size. |
| Step 11 | end Example: Device(config)# end | Exits to privileged EXEC mode. |

Configuring the MSS Value and MTU for Transient TCP SYN Packets

Perform this task to configure the maximum size segment (MSS) for transient packets that traverse a device, specifically TCP segments with the SYN bit set, and to configure the MTU size of IP packets.

If you are configuring the **ip mtu** command on the same interface as the **ip tcp adjust-mss** command, we recommend that you use the following commands and values:

- **ip tcp adjust-mss 1452**
- **ip mtu 1492**

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip tcp adjust-mss** *max-segment-size*
5. **ip mtu** *bytes*
6. **end**

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Step 1 | enable Example: Device> enable | Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted. |
| Step 2 | configure terminal Example: Device# configure terminal | Enters global configuration mode. |
| Step 3 | interface <i>type number</i> Example: Device(config)# interface GigabitEthernet 1/0/0 | Configures an interface type and enters interface configuration mode. |
| Step 4 | ip tcp adjust-mss <i>max-segment-size</i> Example: Device(config-if)# ip tcp adjust-mss 1452 | Adjusts the MSS value of TCP SYN packets going through a device. <ul style="list-style-type: none"> • The <i>max-segment-size</i> argument is the maximum segment size, in bytes. The range is from 500 to 1460. |
| Step 5 | ip mtu <i>bytes</i> Example: Device(config-if)# ip mtu 1492 | Sets the MTU size of IP packets, in bytes, sent on an interface. |
| Step 6 | end Example: Device(config-if)# end | Exits to global configuration mode. |

Configuring the MSS Value for IPv6 Traffic

Perform this task to configure the maximum size segment (MSS) for transient packets that traverse a device, specifically TCP segments with the DF bit set in IPv6 network layer (IP) header.

SUMMARY STEPS

1. enable
2. configure terminal
3. interface *type number*
4. ipv6 tcp adjust-mss *max-segment-size*
5. end

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|-------------------|-------------------------------|
| Step 1 | enable | Enables privileged EXEC mode. |

| | Command or Action | Purpose |
|---------------|----------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | Example: Device> enable | <ul style="list-style-type: none"> Enter your password if prompted. |
| Step 2 | configure terminal Example: Device# configure terminal | Enters global configuration mode. |
| Step 3 | interface <i>type number</i> Example: Device(config)# interface GigabitEthernet 1/0/0 | Configures an interface type and enters interface configuration mode. |
| Step 4 | ipv6 tcp adjust-mss <i>max-segment-size</i> Example: Device(config-if)# ipv6 tcp adjust-mss 1452 | Adjusts the MSS value of TCP DF packets going through a device. <ul style="list-style-type: none"> The <i>max-segment-size</i> argument is the maximum segment size, in bytes. The range is from 40 to 1940. |
| Step 5 | end Example: Device(config-if)# end | Exits interface configuration mode and returns to privileged EXEC mode. |

Verifying TCP Performance Parameters

SUMMARY STEPS

1. **show tcp** [*line-number*] [*tcb address*]
2. **show tcp brief** [**all** | **numeric**]
3. **debug ip tcp transactions**
4. **debug ip tcp congestion**

DETAILED STEPS

Step 1 **show tcp** [*line-number*] [*tcb address*]

Displays the status of TCP connections. The arguments and keyword are as follows:

- *line-number*—(Optional) Absolute line number of the Telnet connection status.
- *tcb*—(Optional) Transmission control block (TCB) of the Explicit Congestion Notification (ECN)-enabled connection.
- *address*—(Optional) TCB hexadecimal address. The valid range is from 0x0 to 0xFFFFFFFF.

The following sample output from the **show tcp tcb** command displays detailed information about an ECN-enabled connection that uses a hexadecimal address format:

Example:

```
Device# show tcp tcb 0x62CD2BB8
```

```

Connection state is LISTEN, I/O status: 1, unread input bytes: 0
Connection is ECN enabled
Local host: 10.10.10.1, Local port: 179
Foreign host: 10.10.10.2, Foreign port: 12000
Enqueued packets for retransmit: 0, input: 0 mis-ordered: 0 (0 bytes)
Event Timers (current time is 0x4F31940):
Timer          Starts      Wakeups          Next
Retrans        0           0                0x0
TimeWait       0           0                0x0
AckHold        0           0                0x0
SendWnd        0           0                0x0
KeepAlive      0           0                0x0
GiveUp         0           0                0x0
PmtuAger       0           0                0x0
DeadWait       0           0                0x0
iss:           0 snduna:       0 sndnxt:        0   sndwnd:        0
irs:           0 rcvnxt:       0 rcvwnd:        4128 delrcvwnd:     0
SRTT: 0 ms, RTTO: 2000 ms, RTV: 2000 ms, KRTT: 0 ms
minRTT: 60000 ms, maxRTT: 0 ms, ACK hold: 200 ms
Flags: passive open, higher precedence, retransmission timeout
TCB is waiting for TCP Process (67)
Datagrams (max data segment is 516 bytes):
Rcvd: 6 (out of order: 0), with data: 0, total data bytes: 0
Sent: 0 (retransmit: 0, fastretransmit: 0), with data: 0, total data
bytes: 0

```

Cisco Software Modularity

The following sample output from the **show tcp tcb** command displays a Software Modularity image:

Example:

```
Device# show tcp tcb 0x1059C10
```

```

Connection state is ESTAB, I/O status: 0, unread input bytes: 0
Local host: 10.4.2.32, Local port: 23
Foreign host: 10.4.2.39, Foreign port: 11000
VRF table id is: 0
Current send queue size: 0 (max 65536)
Current receive queue size: 0 (max 32768) mis-ordered: 0 bytes
Event Timers (current time is 0xB9ACB9):
Timer          Starts      Wakeups          Next (msec)
Retrans        6           0                0
SendWnd        0           0                0
TimeWait       0           0                0
AckHold        8           4                0
KeepAlive      11          0                7199992
PmtuAger       0           0                0
GiveUp         0           0                0
Throttle       0           0                0
irs:   1633857851 rcvnxt: 1633857890 rcvadv: 1633890620 rcvwnd: 32730
iss:   4231531315 snduna: 4231531392 sndnxt: 4231531392 sndwnd: 4052
sndmax: 4231531392 sndcwnd: 10220
SRTT: 84 ms, RTTO: 650 ms, RTV: 69 ms, KRTT: 0 ms
minRTT: 0 ms, maxRTT: 200 ms, ACK hold: 200 ms
Keepalive time: 7200 sec, SYN wait time: 75 sec
Giveup time: 0 ms, Retransmission retries: 0, Retransmit forever: FALSE
State flags: none
Feature flags: Nagle
Request flags: none
Window scales: rcv 0, snd 0, request rcv 0, request snd 0
Timestamp option: recent 0, recent age 0, last ACK sent 0
Datagrams (in bytes): MSS 1460, peer MSS 1460, min MSS 1460, max MSS 1460
Rcvd: 14 (out of order: 0), with data: 10, total data bytes: 38
Sent: 10 (retransmit: 0, fastretransmit: 0), with data: 5, total data bytes: 76

```

```
Header prediction hit rate: 72 %
Socket states: SS_ISCONNECTED, SS_PRIV
Read buffer flags: SB_WAIT, SB_SEL, SB_DEL_WAKEUP
Read notifications: 4
Write buffer flags: SB_DEL_WAKEUP
Write notifications: 0
Socket status: 0
```

Step 2 **show tcp brief [all | numeric]**

(Optional) Displays addresses in IP format.

Use the **show tcp brief** command to display a concise description of TCP connection endpoints. Use the optional **all** keyword to display the status for all endpoints with addresses in a Domain Name System (DNS) hostname format. If this keyword is not used, endpoints in the LISTEN state are not shown. Use the optional **numeric** keyword to display the status for all endpoints with addresses in IP format.

Note If the **ip domain-lookup** command is enabled on the device, and you execute the **show tcp brief** command, the response time of the device to display the output will be very slow. To get a faster response, you should disable the **ip domain-lookup** command.

The following is sample output from the **show tcp brief** command while a user is connected to the system by using Telnet:

Example:

```
Device# show tcp brief
```

| TCB | Local Address | Foreign Address | (state) |
|----------|---------------------|----------------------|---------|
| 609789AC | Device.cisco.com.23 | cider.cisco.com.3733 | ESTAB |

The following example shows the IP activity after the **numeric** keyword is used to display addresses in IP format:

Example:

```
Device# show tcp brief numeric
```

| TCB | Local Address | Foreign Address | (state) |
|----------|-----------------|-----------------|---------|
| 6523A4FC | 10.1.25.3.11000 | 10.1.25.3.23 | ESTAB |
| 65239A84 | 10.1.25.3.23 | 10.1.25.3.11000 | ESTAB |
| 653FCBBC | *.1723 *.* | LISTEN | |

Step 3 **debug ip tcp transactions**

Use the **debug ip tcp transactions** command to display information about significant TCP transactions such as state changes, retransmissions, and duplicate packets. The TCP/IP network isolated above the data link layer might encounter performance issues. The **debug ip tcp transactions** command can be useful in debugging these performance issues.

The following is sample output from the **debug ip tcp transactions** command:

Example:

```
Device# debug ip tcp transactions
```

```
TCP: sending SYN, seq 168108, ack 88655553
TCP0: Connection to 10.9.0.13:22530, advertising MSS 966
TCP0: state was LISTEN -> SYNRCVD [23 -> 10.9.0.13(22530)]
TCP0: state was SYNSENT -> SYNRCVD [23 -> 10.9.0.13(22530)]
TCP0: Connection to 10.9.0.13:22530, received MSS 956
TCP0: restart retransmission in 5996
TCP0: state was SYNRCVD -> ESTAB [23 -> 10.9.0.13(22530)]
TCP2: restart retransmission in 10689
TCP2: restart retransmission in 10641
TCP2: restart retransmission in 10633
```

```
TCP2: restart retransmission in 13384 -> 10.0.0.13(16151)]
TCP0: restart retransmission in 5996 [23 -> 10.0.0.13(16151)]
```

The following line from the **debug ip tcp transactions** command sample output shows that TCP has entered Fast Recovery mode:

Example:

```
fast re-transmit - sndcwnd - 512, snd_last - 33884268765
```

The following lines from the **debug ip tcp transactions** command sample output show that a duplicate acknowledgment is received when TCP is in Fast Recovery mode (first line) and a partial acknowledgment has been received (second line):

Example:

```
TCP0:ignoring second congestion in same window sndcwn - 512, snd_1st - 33884268765
TCP0:partial ACK received sndcwnd:338842495
```

Step 4 debug ip tcp congestion

Use the **debug ip tcp congestion** command to display information about TCP congestion events. The TCP/IP network isolated above the data link layer might encounter performance issues. The **debug ip tcp congestion** command can be used to debug these performance issues. The command also displays information related to variations in the TCP send window, congestion window, and congestion threshold window.

The following is sample output from the **debug ip tcp congestion** command:

Example:

```
Device# debug ip tcp congestion

*May 20 22:49:49.091: Setting New Reno as congestion control algorithm
*May 22 05:21:47.281: Advance cwnd by 12
*May 22 05:21:47.281: TCP85FD0C10: sndcwnd: 1472
*May 22 05:21:47.285: Advance cwnd by 3
*May 22 05:21:47.285: TCP85FD0C10: sndcwnd: 1475
*May 22 05:21:47.285: Advance cwnd by 3
*May 22 05:21:47.285: TCP85FD0C10: sndcwnd: 1478
*May 22 05:21:47.285: Advance cwnd by 9
*May 22 05:21:47.285: TCP85FD0C10: sndcwnd: 1487
*May 20 22:50:32.559: [New Reno] sndcwnd: 8388480 ssthresh: 65535 snd_mark: 232322
*May 20 22:50:32.559: 10.168.10.10:42416 <---> 10.168.30.11:49100 congestion window changes
*May 20 22:50:32.559: cwnd from 8388480 to 2514841, ssthresh from 65535 to 2514841
```

For Cisco TCP, New Reno is the default congestion control algorithm. However, an application can also use Binary Increase Congestion Control (BIC) as the congestion control algorithm. The following is sample output from the **debug ip tcp congestion** command using BIC:

Example:

```
Device# debug ip tcp congestion

*May 22 05:21:42.281: Setting BIC as congestion control algorithm
*May 22 05:21:47.281: Advance cwnd by 12
*May 22 05:21:47.281: TCP85FD0C10: sndcwnd: 1472
*May 22 05:21:47.285: Advance cwnd by 3
*May 22 05:21:47.285: TCP85FD0C10: sndcwnd: 1475
*May 22 05:21:47.285: Advance cwnd by 3
*May 22 05:21:47.285: TCP85FD0C10: sndcwnd: 1478
*May 22 05:21:47.285: Advance cwnd by 9
*May 22 05:21:47.285: TCP85FD0C10: sndcwnd: 1487
*May 20 22:50:32.559: [BIC] sndcwnd: 8388480 ssthresh: 65535 bic_last_max_cwnd: 0 last_cwnd: 8388480
```

```
*May 20 22:50:32.559: 10.168.10.10:42416 <---> 10.168.30.11:49100 congestion window changes
*May 20 22:50:32.559: cwnd from 8388480 to 2514841, ssthresh from 65535 to 2514841
*May 20 22:50:32.559: bic_last_max_cwnd changes from 0 to 8388480
```

Configuring Keepalive Parameters

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip tcp keepalive interval** *seconds*
4. **ip tcp keepalive retries** *number-of-retries*
5. **end**
6. **show running-config**

DETAILED STEPS

| | Command or Action | Purpose |
|--------|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|
| Step 1 | enable Example: Device> enable | Enables privileged EXEC mode. • Enter your password if prompted. |
| Step 2 | configure terminal Example: Device# configure terminal | Enables global configuration mode. |
| Step 3 | ip tcp keepalive interval <i>seconds</i> Example: Device(config)# ip tcp keepalive interval 23 | Configures the keepalive interval. |
| Step 4 | ip tcp keepalive retries <i>number-of-retries</i> Example: Device(config)# ip tcp keepalive retries 5 | Configures the number of unacknowledged probes that can be sent before declaring the connection as dead. |
| Step 5 | end Example: Device(config)# end | Exits global configuration mode. |
| Step 6 | show running-config Example: Device# show running-config | (Optional) Displays the running configuration. |

Configuration Examples for TCP

Example: Verifying the Configuration of TCP ECN

The following example shows how to verify whether TCP ECN is configured:

```
Device# show running-config

Building configuration...
.
.
.
ip tcp ecn ! ECN is configured.
.
.
.
```

The following example shows how to verify whether TCP is ECN-enabled on a specific connection (local host):

```
Device# show tcp tcb 123456A

!Local host
!
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Connection is ECN Enabled
Local host: 10.1.25.31, Local port: 11002
Foreign host: 10.1.25.34, Foreign port: 23
```

The following example shows how to display concise information about one address:

```
Device# show tcp brief

!
TCB           Local address           Foreign Address          (state)
609789C       Router.example.com.23    cider.example.com.3733  ESTAB
```

The following example shows how to enable IP TCP ECN debugging:

```
Device# debug ip tcp ecn
!
TCP ECN debugging is on
!
Device# telnet 10.1.25.31

Trying 10.1.25.31 ...
!
01:43:19: 10.1.25.35:11000 <---> 10.1.25.31:23   out ECN-setup SYN
01:43:21: 10.1.25.35:11000 <---> 10.1.25.31:23   congestion window changes
01:43:21: cwnd from 1460 to 1460, ssthresh from 65535 to 2920
01:43:21: 10.1.25.35:11000 <---> 10.1.25.31:23   in non-ECN-setup SYN-ACK
```

Before a TCP connection can use ECN, a host sends an ECN-setup SYN (synchronization) packet to a remote end that contains an Echo Congestion Experience (ECE) and Congestion window reduced (CWR) bit set in the header. Setting the ECE and CWR bits indicates to the remote end that the sending TCP is ECN capable,

rather than an indication of congestion. The remote end sends an ECN-setup SYN-ACK (acknowledgment) packet to the sending host.

In this example the “out ECN-setup SYN” text means that a SYN packet with the ECE and CWR bit set was sent to the remote end. The “in non-ECN-setup SYN-ACK” text means that the remote end did not favorably acknowledge the ECN request and, therefore, the session is not ECN capable.

The following output shows that ECN capabilities are enabled at both ends. In response to the ECN-setup SYN, the other end favorably replied with an ECN-setup SYN-ACK message. This connection is now ECN capable for the rest of the session.

```
Device# telnet 10.10.10.10

Trying 10.10.10.10 ... Open
Password required, but none set
!
1d20h: 10.1.25.34:11003 <---> 10.1.25.35:23   out ECN-setup SYN
1d20h: 10.1.25.34:11003 <---> 10.1.25.35:23   in ECN-setup SYN-ACK
```

The following example shows how to verify that the hosts are connected:

```
Device# show debugging
!
TCP:
  TCP Packet debugging is on
  TCP ECN debugging is on
!
Device# telnet 10.1.25.234
!
Trying 10.1.25.234 ...
!
00:02:48: 10.1.25.31:11001 <---> 10.1.25.234:23   out ECN-setup SYN
00:02:48: tcp0: O CLOSED 10.1.25.234:11001 10.1.25.31:23 seq 1922220018
          OPTS 4 ECE CWR SYN WIN 4128
00:02:50: 10.1.25.31:11001 <---> 10.1.25.234:23   congestion window changes
00:02:50: cwnd from 1460 to 1460, ssthresh from 65535 to 2920
00:02:50: tcp0: R SYNSENT 10.1.25.234:11001 10.1.25.31:23 seq 1922220018
          OPTS 4 ECE CWR SYN WIN 4128
00:02:54: 10.1.25.31:11001 <---> 10.1.25.234:23   congestion window changes
00:02:54: cwnd from 1460 to 1460, ssthresh from 2920 to 2920
00:02:54: tcp0: R SYNSENT 10.1.25.234:11001 10.1.25.31:23 seq 1922220018
          OPTS 4 ECE CWR SYN WIN 4128
00:03:02: 10.1.25.31:11001 <---> 10.1.25.234:23   congestion window changes
00:03:02: cwnd from 1460 to 1460, ssthresh from 2920 to 2920
00:03:02: tcp0: R SYNSENT 10.1.25.234:11001 10.1.25.31:23 seq 1922220018
          OPTS 4 ECE CWR SYN WIN 4128
00:03:18: 10.1.25.31:11001 <---> 10.1.25.234:23   SYN with ECN disabled
00:03:18: 10.1.25.31:11001 <---> 10.1.25.234:23   congestion window changes
00:03:18: cwnd from 1460 to 1460, ssthresh from 2920 to 2920
00:03:18: tcp0: O SYNSENT 10.1.25.234:11001 10.1.25.31:23 seq 1922220018
          OPTS 4 SYN WIN 4128
00:03:20: 10.1.25.31:11001 <---> 10.1.25.234:23   congestion window changes
00:03:20: cwnd from 1460 to 1460, ssthresh from 2920 to 2920
00:03:20: tcp0: R SYNSENT 10.1.25.234:11001 10.1.25.31:23 seq 1922220018
          OPTS 4 SYN WIN 4128
00:03:24: 10.1.25.31:11001 <---> 10.1.25.234:23   congestion window changes
00:03:24: cwnd from 1460 to 1460, ssthresh from 2920 to 2920
00:03:24: tcp0: R SYNSENT 10.1.25.234:11001 10.1.25.31:23 seq 1922220018
          OPTS 4 SYN WIN 4128
00:03:32: 10.1.25.31:11001 <---> 10.1.25.234:23   congestion window changes
00:03:32: cwnd from 1460 to 1460, ssthresh from 2920 to 2920
00:03:32: tcp0: R SYNSENT 10.1.25.234:11001 10.1.25.31:23 seq 1922220018
```

```

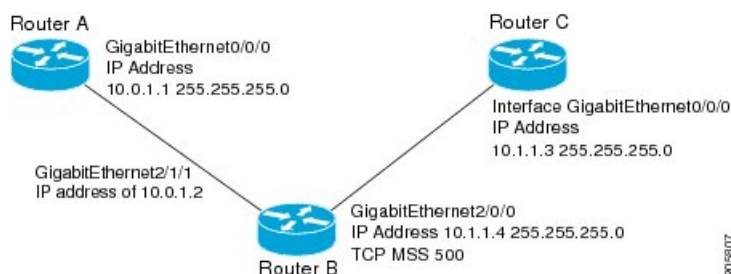
OPTS 4 SYN WIN 4128
!Connection timed out; remote host not responding

```

Example: Configuring the TCP MSS Adjustment

The following example shows how to configure and verify the interface adjustment value for the example topology displayed in the figure below:

Figure 6: Example Topology for TCP MSS Adjustment



Configure the interface adjustment value on router B:

```

Router_B(config)# interface GigabitEthernet 2/0/0
Router_B(config-if)# ip tcp adjust-mss 500

```

Telnet from router A to router C with B having the Maximum Segment Size (MSS) adjustment configured:

```

Router_A# telnet 192.168.1.1

```

```

Trying 192.168.1.1... Open

```

Observe the debug output from router C:

```

Router_C# debug ip tcp transactions

Sep 5 18:42:46.247: TCP0: state was LISTEN -> SYNRCVD [23 -> 10.0.1.1(38437)]
Sep 5 18:42:46.247: TCP: tcb 32290C0 connection to 10.0.1.1:38437, peer MSS 500, MSS is 500
Sep 5 18:42:46.247: TCP: sending SYN, seq 580539401, ack 6015751
Sep 5 18:42:46.247: TCP0: Connection to 10.0.1.1:38437, advertising MSS 500
Sep 5 18:42:46.251: TCP0: state was SYNRCVD -> ESTAB [23 -> 10.0.1.1(38437)]

```

The MSS gets adjusted to 500 on Router B as configured.

The following example shows the configuration of a Point-to-Point Protocol over Ethernet (PPPoE) client with the MSS value set to 1452:

```

Device(config)# vpdn enable
Device(config)# no vpdn logging
Device(config)# vpdn-group 1
Device(config-vpdn)# request-dialin
Device(config-vpdn-req-in)# protocol pppoe
Device(config-vpdn-req-in)# exit
Device(config-vpdn)# exit
Device(config)# interface GigabitEthernet 0/0/0
Device(config-if)# ip address 192.168.100.1.255.255.255.0
Device(config-if)# ip tcp adjust-mss 1452
Device(config-if)# ip nat inside
Device(config-if)# exit

```

```

Device(config)# interface ATM 0
Device(config-if)# no ip address
Device(config-if)# no atm ilmi-keepalive
Device(config-if)# pvc 8/35
Device(config-if)# pppoe client dial-pool-number 1
Device(config-if)# dsl equipment-type CPE
Device(config-if)# dsl operating-mode GSHDSL symmetric annex B
Device(config-if)# dsl linerate AUTO
Device(config-if)# exit
Device(config)# interface Dialer 1
Device(config-if)3 ip address negotiated
Device(config-if)# ip mtu 1492
Device(config-if)# ip nat outside
Device(config-if)# encapsulation ppp
Device(config-if)# dialer pool 1
Device(config-if)# dialer-group 1
Device(config-if)# ppp authentication pap callin
Device(config-if)# ppp pap sent-username sohodyn password 7 141B1309000528
Device(config-if)# ip nat inside source list 101 Dialer1 overload
Device(config-if)# exit
Device(config)# ip route 0.0.0.0.0.0.0.0 Dialer1
Device(config)# access-list permit ip 192.168.100.0.0.0.0.255 any

```

The following example shows the configuration of interface adjustment value for IPv6 traffic:

```

Device> enable
Device# configure terminal
Device(config)# interface GigabitEthernet 0/0/0
Device(config)# ipv6 tcp adjust-mss 1452
Device(config)# end

```

Example: Configuring the TCP Application Flags Enhancement

The following output shows the flags (status and option) displayed using the **show tcp** command:

```

Device# show tcp
.
.
.
Status Flags: passive open, active open, retransmission timeout
App closed
Option Flags: vrf id set
IP Precedence value: 6
.
.
.
SRTT: 273 ms, RTTO: 490 ms, RTV: 217 ms, KRRT: 0 ms
minRTT: 0 ms, maxRTT: 300 ms, ACK hold: 200 ms

```

Example: Displaying Addresses in IP Format

The following example shows the IP activity by using the **numeric** keyword to display the addresses in IP format:

```

Device# show tcp brief numeric

```

| TCB | Local Address | Foreign Address | (state) |
|----------|-----------------|-----------------|---------|
| 6523A4FC | 10.1.25.3.11000 | 10.1.25.3.23 | ESTAB |

```

65239A84      10.1.25.3.23      10.1.25.3.11000      ESTAB
653FCBEC      *.1723 *.* LISTEN

```

Example: Configuring Keepalive Parameters

The following example shows how to configure TCP keepalive parameters.

```

Device# configure terminal
Device(config)# ip tcp keepalive interval 2
Device(config)# ip tcp keepalive retries 5

```

The following is a sample output of the **show running-config** command:

```

Device# show running-config

ip tcp keepalive retries 5
ip tcp keepalive interval 2

```

Additional References

Related Documents

| Related Topic | Document Title |
|----------------------------------|--------------------------------------------------------------|
| Cisco IOS commands | Cisco IOS Master Commands List, All Releases |
| IP Application Services commands | IP Application Services Command Reference |

Standards and RFCs

| Standard/RFC | Title |
|--------------|-----------------------------------------------------------------------------------------|
| RFC 793 | Transmission Control Protocol |
| RFC 1191 | Path MTU discovery |
| RFC 1323 | TCP Extensions for High Performance |
| RFC 2018 | TCP Selective Acknowledgment Options |
| RFC 2581 | TCP Congestion Control |
| RFC 3168 | The Addition of Explicit Congestion Notification (ECN) to IP |
| RFC 3782 | The NewReno Modification to TCP's Fast Recovery Algorithm |
| RFC 4022 | Management Information Base for the Transmission Control Protocol (TCP) |

MIBs

| MIB | MIBs Link |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CISCO-TCP-MIB | To locate and download MIBs for selected platforms, Cisco software 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 Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password. | http://www.cisco.com/cisco/web/support/index.html |

Feature Information for TCP

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

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

Table 7: Feature Information for TCP

| Feature Name | Releases | Feature Information |
|-----------------------------------|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TCP Application Flags Enhancement | 12.2(31)SB2 12.4(2)T | The TCP Applications Flags Enhancement feature enables the user to display additional flags with reference to TCP applications. There are two types of flags: status and option. The status flags indicate the status of TCP connections such as retransmission timeouts, application closed, and synchronized (SYNC) handshakes for listening. The additional flags indicate the state of set options such as whether a VPN routing and forwarding instance (VRF) is set, whether a user is idle, and whether a keepalive timer is running. The following command was modified by this feature: show tcp . |

| Feature Name | Releases | Feature Information |
|--------------------------------------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TCP Congestion Avoidance | 12.3(7)T | <p>The TCP Congestion Avoidance feature enables the monitoring of acknowledgment packets to the TCP sender when multiple packets are lost in a single window of data. Before this feature was introduced, the sender would exit Fast-Recovery mode, wait for three or more duplicate acknowledgment packets before retransmitting the next unacknowledged packet, or wait for the retransmission timer to start slowly. This delay could lead to performance issues.</p> <p>Implementation of RFC 2581 and RFC 3782 addresses the modifications to the Fast-Recovery algorithm that incorporates a response to partial acknowledgments received during Fast Recovery, improving performance in situations where multiple packets are lost in a single window of data.</p> <p>This feature is an enhancement to the existing Fast Recovery algorithm. No commands are used to enable or disable this feature.</p> <p>The output of the debug ip tcp transactions command monitors acknowledgment packets by displaying the following conditions:</p> <ul style="list-style-type: none"> • TCP entering Fast Recovery mode. • Duplicate acknowledgments being received during Fast Recovery mode. • Partial acknowledgments being received. <p>The following command was modified by this feature: debug ip tcp transactions.</p> |
| TCP Explicit Congestion Notification | 12.3(7)T | <p>The TCP Explicit Congestion Notification (ECN) feature allows an intermediate router to notify end hosts of impending network congestion. It also provides enhanced support for TCP sessions associated with applications such as Telnet, web browsing, and transfer of audio and video data, that are sensitive to delay or packet loss. The benefit of this is the reduction of delay and packet loss in data transmissions.</p> <p>The following commands were introduced or modified by this feature: debug ip tcp ecn, ip tcp ecn, show debugging, show tcp.</p> |
| TCP MIB for RFC4022 Support | 12.2(33)XN | <p>The TCP MIB for RFC 4022 Support feature introduces support for RFC 4022, <i>Management Information Base for the Transmission Control Protocol (TCP)</i>. RFC 4022 is an incremental change of the TCP MIB to improve the manageability of TCP.</p> <p>There are no new or modified commands for this feature.</p> |

| Feature Name | Releases | Feature Information |
|---------------------|---------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TCP MSS Adjust | 12.2(4)T 12.2(8)T 12.2(18)ZU2 12.2(28)SB 12.2(33)SRA 12.2(33)SXH 15.0(1)S | <p>The TCP MSS Adjust feature enables the configuration of the maximum segment size (MSS) for transient packets that traverse a device, specifically TCP segments in the SYN bit set.</p> <p>In 12.2(4)T, this feature was introduced.</p> <p>In 12.2(8)T, the command that was introduced by this feature was changed from ip adjust-mss to ip tcp adjust-mss.</p> <p>In 12.2(28)SB and 12.2(33)SRA, this feature was enhanced to be configurable on subinterfaces.</p> <p>The following command was introduced by this feature: ip tcp adjust-mss.</p> |
| TCP Show Extension | 12.2(31)SB2 12.4(2)T | <p>The TCP Show Extension feature introduces the capability to display addresses in IP format instead of hostname format and to display the VRF table associated with the connection.</p> <p>The following command was modified by this feature: show tcp brief.</p> |
| TCP Window Scaling | 12.2(8)T 12.2(31)SB2 | <p>The TCP Window Scaling feature adds support for the Window Scaling option in RFC 1323. A larger window size is recommended to improve TCP performance in network paths with large bandwidth, long-delay characteristics that are called Long Fat Networks (LFNs). This TCP Window Scaling enhancement provides that support.</p> <p>The following command was introduced or modified by this feature: ip tcp window-size.</p> |
| TCP Keepalive Timer | 15.2(4)M | <p>The TCP Keepalive Timer feature introduces the capability to identify dead connections between multiple routing devices.</p> <p>The following command was introduced or modified by this feature: ip tcp keepalive.</p> |



CHAPTER 5

Configuring UDP Forwarding Support for IP Redundancy Virtual Router Groups

User Datagram Protocol (UDP) forwarding is a feature used in Cisco IOS software to forward broadcast and multicast packets received for a specific IP address. Virtual Router Group (VRG) support, implemented with the Hot Standby Routing Protocol (HSRP), allows a set of routers to be grouped as a logical router that answers to a well-known IP address. The UDP Forwarding Support for IP Redundancy Virtual Router Groups feature enables UDP forwarding to be VRG aware; this results in packets getting forwarded only to the active router in the VRG.

This module explains the concepts related UDP forwarding and VRG support and describes how to configure UDP forwarding support for IP Redundancy Virtual Router Groups in a network.

- [Finding Feature Information, on page 101](#)
- [Prerequisites for UDP Forwarding Support for IP Redundancy Virtual Router Groups, on page 102](#)
- [Information About UDP Forwarding Support for IP Redundancy Virtual Router Groups, on page 102](#)
- [How to Configure UDP Forwarding Support for IP Redundancy Virtual Router Groups, on page 102](#)
- [Configuration Examples for UDP Forwarding Support for IP Redundancy Virtual Router Groups, on page 104](#)
- [Additional References, on page 104](#)
- [Feature Information for UDP Forwarding Support for IP Redundancy Virtual Router Groups, on page 105](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

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

Prerequisites for UDP Forwarding Support for IP Redundancy Virtual Router Groups

- The UDP Forwarding Support for Virtual Router Groups feature is available only on platforms that support VRGs.

Information About UDP Forwarding Support for IP Redundancy Virtual Router Groups

Benefits of the UDP Forwarding Support for Virtual Router Groups Feature

Forwarding is limited to the active router in the VRG instead of all routers within the VRG. Prior to the implementation of this feature, the only VRG support was HSRP. Within a VRG that is formed by HSRP, the forwarding of UDP-based broadcast and multicast packets is done by all the routers within the VRG. This process can cause some DHCP servers to operate incorrectly. The UDP Forwarding Support for VRGs feature limits forwarding to the active router in the VRG.

VRG awareness is achieved with IP Redundancy Service (IRS). The IRS application programming interface (API) provides notification updates of a specific VRG, addition and deletion of a VRG, and querying of the current state of a VRG. A state change notification is provided to avoid the performance impact of querying the state of the VRG each time it is needed. The UDP forwarding code caches the VRG state for each required helper address that is defined. Each time the UDP forwarding code needs to execute, it checks the current state of the VRG associated with the helper address and forwards packets only to VRGs that are active.

How to Configure UDP Forwarding Support for IP Redundancy Virtual Router Groups

Configuring UDP Forwarding Support for IP Redundancy Virtual Router Groups

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no shutdown**
5. **ip address** *ip-address mask*
6. **ip helper-address** *address* **redundancy** *vrg-name*
7. **standby** *group-number* **ip** *ip-address*
8. **standby** *group-number* **name** *group-name*
9. **end**

DETAILED STEPS

| | Command or Action | Purpose |
|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------|
| Step 1 | enable Example: <pre>Router> enable</pre> | Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted. |
| Step 2 | configure terminal Example: <pre>Router# configure terminal</pre> | Enters global configuration mode. |
| Step 3 | interface <i>type number</i> Example: <pre>Router(config)# interface fastethernet 0/0</pre> | Specifies an interface and enters interface configuration mode. |
| Step 4 | no shutdown Example: <pre>Router(config-if)# no shutdown</pre> | Restarts a disabled interface. |
| Step 5 | ip address <i>ip-address mask</i> Example: <pre>Router(config-if)# ip address 172.16.10.1 255.255.255.0</pre> | Sets a primary address for the interface. |
| Step 6 | ip helper-address <i>address redundancy vrg-name</i> Example: <pre>Router(config-if)# ip helper-address 10.1.1.1 redundancy vrg1</pre> | Enables UDP forwarding support for the VRG. |
| Step 7 | standby <i>group-number ip ip-address</i> Example: <pre>Router(config-if)# standby 1 ip 172.16.10.254</pre> | Activates HSRP. |
| Step 8 | standby <i>group-number name group-name</i> Example: <pre>Router(config-if)# standby 1 name vrg1</pre> | Configures the name of the standby group. |
| Step 9 | end Example: <pre>Router(config-if)# end</pre> | Exits the current configuration mode and returns to privileged EXEC mode. |

Configuration Examples for UDP Forwarding Support for IP Redundancy Virtual Router Groups

Example: Configuring UDP Forwarding Support for IP Redundancy Virtual Router Groups

The following example shows how to configure UDP Forwarding Support for IP Redundancy Virtual Router Groups:

```
Router(config)# interface fastethernet 0/0
Router(config-if)# no shutdown
Router(config-if)# ip address 172.16.10.1 255.255.255.0
Router(config-if)# ip helper-address 10.1.1.1 redundancy vrg1
Router(config-if)# standby 1 ip 172.16.10.254
Router(config-if)# standby 1 name vrg1
Router(config-if)# end
```

Additional References

Related Documents

| Related Topic | Document Title |
|------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------|
| Cisco IOS commands | Cisco IOS Master Commands List, All Releases |
| IP application services commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples | <i>Cisco IOS IP Application Services Command Reference</i> |

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 | To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs |

RFCs

| RFC | Title |
|--------------------------------------------------------------------------------------------|-------|
| No new or modified RFCs are supported, and support for existing RFCs has not been modified | — |

Technical Assistance

| Description | Link |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|
| The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password. | http://www.cisco.com/cisco/web/support/index.html |

Feature Information for UDP Forwarding Support for IP Redundancy Virtual Router Groups

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

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

Table 8: Feature Information for UDP Forwarding Support for IP Redundancy Virtual Router Groups

| Feature Name | Releases | Feature Information |
|---------------------------------------------------------------|-----------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| UDP Forwarding Support for IP Redundancy Virtual Router Group | Cisco IOS XE 3.1.0SG 12.2(50)SY 12.2(15)T 15.0(1)SY 15.2(1)S | UDP forwarding is a feature used in Cisco IOS software to forward broadcast and multicast packets received for a specific IP address. Virtual Router Group (VRG) support is implemented with the Hot Standby Routing Protocol (HSRP) and it allows a set of routers to be grouped as a logical router that answers to a well-known IP address. The UDP Forwarding Support for IP Redundancy Virtual Router Groups feature enables UDP forwarding to be VRG aware, resulting in forwarding only to the active router in the VRG. The following command was introduced or modified: ip helper-address . |



CHAPTER 6

Configuring WCCP

The Web Cache Communication Protocol (WCCP) is a Cisco-developed content-routing technology that intercepts IP packets and redirects those packets to a destination other than that specified in the IP packet. Typically the packets are redirected from their destination web server on the Internet to a content engine that is local to the client. In some WCCP deployment scenarios, redirection of traffic may also be required from the web server to the client. WCCP enables you to integrate content engines into your network infrastructure.

Cisco IOS Release 12.1 and later releases allow the use of either WCCP Version 1 (WCCPv1) or Version 2 (WCCPv2).

The tasks in this document assume that you have already configured content engines on your network. For specific information on hardware and network planning associated with Cisco Content Engines and WCCP, see the Cisco Content Engines documentation at the following URL:

<http://www.cisco.com/univercd/cc/td/doc/product/webscale/content/index.htm>

- [Finding Feature Information, on page 107](#)
- [Prerequisites for WCCP, on page 108](#)
- [Restrictions for WCCP, on page 108](#)
- [Information About WCCP, on page 109](#)
- [How to Configure WCCP, on page 121](#)
- [Configuration Examples for WCCP, on page 131](#)
- [Additional References, on page 136](#)
- [Feature Information for WCCP, on page 138](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

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

Prerequisites for WCCP

- To use WCCP, IP must be configured on the interface connected to the Internet and another interface must be connected to the content engine.
- The interface connected to the content engine must be a Fast Ethernet or Gigabit Ethernet interface.

Restrictions for WCCP

General

The following limitations apply to Web Cache Communication Protocol Version1 (WCCPv1) and WCCP Version 2 (WCCPv2):

- WCCP works only with IPv4 networks.
- WCCP bypasses Network Address Translation (NAT) when Cisco Express Forwarding is enabled.

WCCPv1

- WCCPv1 supports the redirection of HTTP (TCP port 80) traffic only.
- WCCPv1 does not allow multiple routers to be attached to a cluster of content engines.

WCCPv2

- WCCP works only with IPv4 networks.
- For routers servicing a multicast cluster, the Time To Live (TTL) value must be set at 15 or fewer.
- Service groups can comprise up to 32 content engines and 32 routers.
- All content engines in a cluster must be configured to communicate with all routers servicing the cluster.
- Multicast addresses must be from 224.0.0.0 to 239.255.255.255.

WCCP VRF Support

- In Cisco IOS Release 12.2(33)SRE, this feature is supported only on Cisco 7200 NPE-G2 and Cisco 7304-NPE-G100 routers.

WCCP Layer 2 Forwarding and Return

In a Cisco Wide Area Application Services (WAAS) and Cisco IOS XE firewall configuration, all packets processed by a Wide Area Application Engine (WAE) device must go over the Cisco IOS XE firewall in both directions to support the Web Cache Coordination Protocol (WCCP) generic routing encapsulation (GRE) redirect. This situation occurs when a Layer 2 redirect is not available. If a Layer 2 redirect is configured on the WAE, the system defaults to the GRE redirect to continue to function.

The client device and a WAE device or a cache engine cannot be connected to a Cisco device with the same interface and WCCP redirect configured on the interface.

The following two configurations are supported:

- For WCCP Layer 2 return, the client and WAE are connected to a Cisco device with same interface and WCCP output is configured on the interface.
- For WCCP Layer 2 return, the client and WAE are connected to a Cisco device with same physical interface but in different VLANs and sub-interfaces.

Cisco 7600 Series Routers Access Control Lists

When WCCP is using the mask assignment, any redirect list is merged with the mask information from the appliance and the resulting merged ACL is passed down to the Cisco 7600 series router hardware. Only Permit or Deny ACL entries from the redirect list in which the protocol is IP or exactly matches the service group protocol are merged with the mask information from the appliance.

The following restrictions apply to the redirect-list ACL:

- The ACL must be an IPv4 simple or extended ACL.
- Only individual source or destination port numbers may be specified; port ranges cannot be specified.
- The only valid matching criteria in addition to individual source or destination port numbers are **dscp** or **tos**.
- The use of **fragments**, **time-range**, or **options** keywords, or any TCP flags is not permitted.
- If the redirect ACL does not meet the restrictions shown, the system will log the following error message:

```
WCCP-3-BADACE: Service <service group>, invalid access-list entry (seq:<sequence>, reason:<reason>)
```

- WCCP continues to redirect packets, but the redirection is carried out in software until the access list is adjusted.

Information About WCCP

WCCP Overview

WCCP uses Cisco Content Engines (or other content engines running WCCP) to localize traffic patterns in the network, enabling content requests to be fulfilled locally. Traffic localization reduces transmission costs and download time.

WCCP enables routing platforms to transparently redirect content requests. With transparent redirection, users can fulfill content requests locally without configuring their browsers to use a web proxy. Instead, they can use the target URL to request content, and have their requests automatically redirected to a content engine. The word "transparent" in this case means that the end user does not know that a requested file (such as a web page) came from the content engine instead of from the originally specified server.

A content engine receiving a request attempts to service it from its own local cache. If the requested information is not present, the content engine issues its own request to the originally targeted server to get the required

information. A content engine retrieving the requested information forwards it to the requesting client and caches it to fulfill future requests, thus maximizing download performance and substantially reducing transmission costs.

WCCP enables a series of content engines, called a content engine cluster, to provide content to a router or multiple routers. Network administrators can easily scale their content engines to manage heavy traffic loads through these clustering capabilities. Cisco clustering technology enables each cluster member to work in parallel, resulting in linear scalability. Clustering content engines greatly improves the scalability, redundancy, and availability of your caching solution. You can cluster up to 32 content engines to scale to your desired capacity.

Layer 2 Forwarding Redirection and Return

WCCP uses either generic routing encapsulation (GRE) or Layer 2 (L2) to redirect or return IP traffic. When WCCP forwards traffic via GRE, the redirected packets are encapsulated within a GRE header. The packets also have a WCCP redirect header. When WCCP forwards traffic using L2, the original MAC header of the IP packet is overwritten and replaced with the MAC header for the WCCP client.

Using L2 as a forwarding method allows direct forwarding to the content engine without further lookup. Layer 2 redirection requires that the router and content engines are directly connected, that is, on the same IP subnetwork.

When WCCP returns traffic via GRE, the returned packets are encapsulated within a GRE header. The destination IP address is the address of the router and the source address is the address of the WCCP client. When WCCP returns traffic via L2, the original IP packet is returned without any added header information. The router to which the packet is returned will recognize the source of the packet and prevent redirection.

The WCCP redirection method does not have to match the return method.

L2 forwarding, return, or redirection are typically used for hardware-accelerated platforms. Depending on your release, L2 forwarding, return, and redirection can also be used for software-switching platforms.

For content engines running Application and Content Networking System (ACNS) software, use the **wccp custom-web-cache** command with the **l2-redirect** keyword to configure L2 redirection. For content engines running Cisco Wide Area Application Services (WAAS) software, use the **wccp tcp-promiscuous** command with the **l2-redirect** keyword to configure L2 redirection.



Note Before configuring a GRE tunnel, configure a loopback interface (that is not attached to a VRF) with an IP address so that the internally created tunnel interface is enabled for IPv4 forwarding by unnumbering itself to this dummy loopback interface. You do not need to configure a loopback interface if the system has at least one interface that is not attached to a VRF and that is configured with an IPv4 address.

For information about Cisco ACNS commands used to configure Cisco Content Engines, see the [Cisco ACNS Software Command Reference](#).

For more information about WAAS commands used to configure Cisco Content Engines, see the [Cisco Wide Area Application Services Command Reference](#).

WCCP Mask Assignment

The WCCP Mask Assignment feature enables mask assignment as the load-balancing method (instead of the default hash assignment method) for a WCCP service.

For content engines running Application and Content Networking System (ACNS) software, use the **wccp custom-web-cache** command with the **mask-assign** keyword to configure mask assignment. For content engines running Cisco Wide Area Application Services (WAAS) software, use the **wccp tcp-promiscuous** command with the **mask-assign** keyword to configure mask assignment.

For information about Cisco ACNS commands used to configure Cisco Content Engines, see the [Cisco ACNS Software Command Reference](#).

For more information about WAAS commands used to configure Cisco Content Engines, see the [Cisco Wide Area Application Services Command Reference](#).

Hardware Acceleration

Cisco 7600 series routers provide WCCP Layer 2 Policy Feature Card (PFC) redirection hardware acceleration. Hardware acceleration allows Cisco Content Engines to perform a L2 MAC address rewrite redirection method when directly connected to a compatible router.

Redirection processing is accelerated in the routing hardware, which is more efficient than L3 redirection with Generic Routing Encapsulation (GRE). L2 redirection takes place on the router, and is not visible to the Multilayer Switch Feature Card (MSFC). The WCCP L2 PFC redirection feature requires no configuration on the MSFC. The **show ip wccp {service-number | web-cache} detail** command displays which redirection method is in use for each content engine.

In order for the router to make complete use of hardware redirection, the content engine must be configured with L2 redirection and mask assignment.

Use the **ip wccp web-cache accelerated** command on hardware-based platforms to enforce the use of L2 redirection and mask assignment. Using this command configures the router to form a service group and redirect packets with an appliance only if the appliance is configured for L2 and mask assignment.

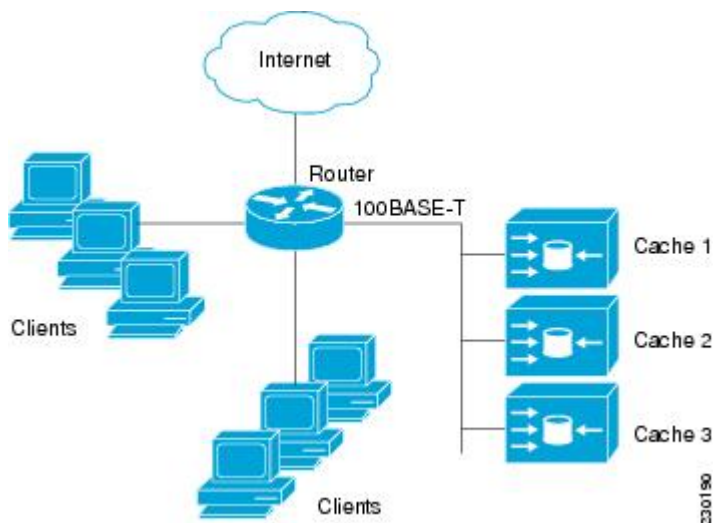
The following guidelines apply to WCCP Layer 2 PFC redirection:

- The WCCP Layer 2 PFC redirection feature sets the IP flow mask to full-flow mode.
- You can configure the Cisco Cache Engine software Release 2.2 or later releases to use the WCCP Layer 2 PFC redirection feature.
- L2 redirection takes place on the PFC and is not visible to the MSFC. The **show ip wccp {service-number | web-cache} detail** command on the MSFC displays statistics for only the first packet of an L2 redirected flow, which provides an indication of how many flows, rather than packets, are using L2 redirection. You can view information about L2 redirected flows by entering the **show platform flow ip** command. The PFC3 provides hardware acceleration for GRE. If you use WCCP Layer 3 redirection with GRE, there is hardware support for encapsulation, but the PFC3 does not provide hardware support for decapsulation of WCCP GRE traffic.

WCCPv1 Configuration

With WCCPv1, only a single router services a cluster. In this scenario, this router is the device that performs all the IP packet redirection. The figure below illustrates the WCCPv1 configuration.

Figure 7: WCCPv1 Configuration



Content is not duplicated on the content engines. The benefit of using multiple content engines is that you can scale a caching solution by clustering multiple physical content engines to appear as one logical cache.

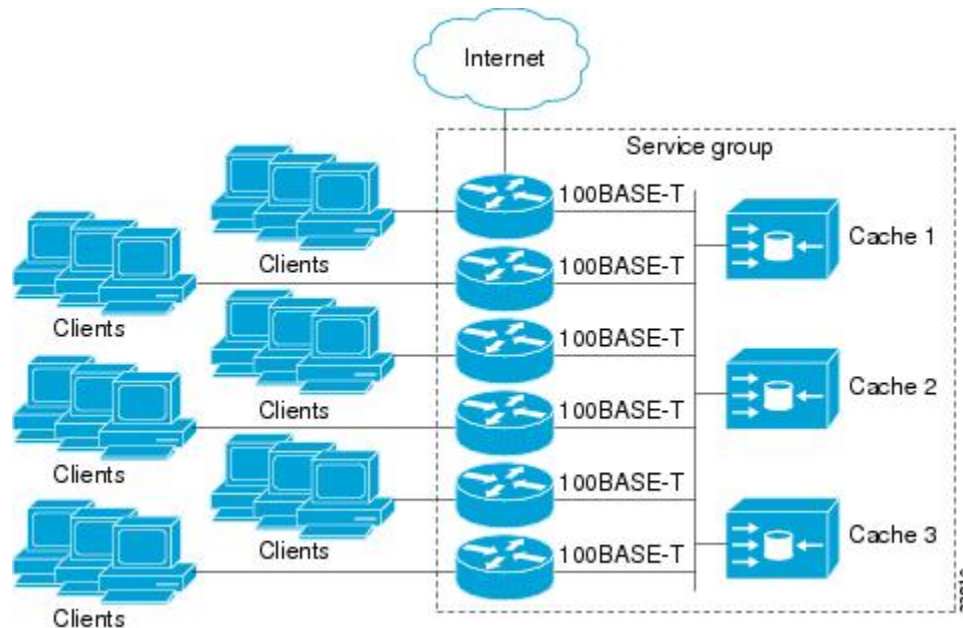
The following sequence of events details how WCCPv1 configuration works:

1. Each content engine is configured by the system administrator with the IP address of the control router. Up to 32 content engines can connect to a single control router.
2. The content engines send their IP addresses to the control router using WCCP, indicating their presence. Routers and content engines communicate to each other via a control channel; this channel is based on UDP port 2048.
3. This information is used by the control router to create a cluster view (a list of caches in the cluster). This view is sent to each content engine in the cluster, essentially making all the content engines aware of each other. A stable view is established after the membership of the cluster remains the same for a certain amount of time.
4. When a stable view has been established, one content engine is elected as the lead content engine. (The lead is defined as the content engine seen by all the content engines in the cluster with the lowest IP address). This lead content engine uses WCCP to indicate to the control router how IP packet redirection should be performed. Specifically, the lead content engine designates how redirected traffic should be distributed across the content engines in the cluster.

WCCPv2 Configuration

Multiple routers can use WCCPv2 to service a content engine cluster. In WCCPv1, only one router could redirect content requests to a cluster. The figure below illustrates a sample configuration using multiple routers.

Figure 8: Cisco Content Engine Network Configuration Using WCCPv2



The subset of content engines within a cluster and routers connected to the cluster that are running the same service is known as a service group. Available services include TCP and UDP redirection.

In WCCPv1, the content engines were configured with the address of the single router. WCCPv2 requires that each content engine be aware of all the routers in the service group. To specify the addresses of all the routers in a service group, you must choose one of the following methods:

- **Unicast**—A list of router addresses for each of the routers in the group is configured on each content engine. In this case the address of each router in the group must be explicitly specified for each content engine during configuration.
- **Multicast**—A single multicast address is configured on each content engine. In the multicast address method, the content engine sends a single-address notification that provides coverage for all routers in the service group. For example, a content engine could indicate that packets should be sent to a multicast address of 224.0.0.100, which would send a multicast packet to all routers in the service group configured for group listening using WCCP (see the `ip wccp group-listen` or the `ipv6 wccp group-listen` interface configuration command for details).

The multicast option is easier to configure because you need only specify a single address on each content engine. This option also allows you to add and remove routers from a service group dynamically, without needing to reconfigure the content engines with a different list of addresses each time.

The following sequence of events details how WCCPv2 configuration works:

1. Each content engine is configured with a list of routers.
2. Each content engine announces its presence and a list of all routers with which it has established communications. The routers reply with their view (list) of content engines in the group.
3. When the view is consistent across all content engines in the cluster, one content engine is designated as the lead and sets the policy that the routers need to deploy in redirecting packets.

WCCPv2 Support for Services Other Than HTTP

WCCPv2 allows redirection of traffic other than HTTP (TCP port 80 traffic), including a variety of UDP and TCP traffic. WCCPv2 supports the redirection of packets intended for other ports, including those used for proxy-web cache handling, File Transfer Protocol (FTP) caching, FTP proxy handling, web caching for ports other than 80, and Real Audio, video, and telephony applications.

To accommodate the various types of services available, WCCPv2 introduced the concept of multiple *service groups*. Service information is specified in the WCCP configuration commands using dynamic services identification numbers (such as 98) or a predefined service keyword (such as **web-cache**). This information is used to validate that service group members are all using or providing the same service.

The content engines in a service group specify traffic to be redirected by protocol (TCP or UDP) and up to eight source or destination ports. Each service group has a priority status assigned to it. The priority of a dynamic service is assigned by the content engine. The priority value is in the range of 0 to 255 where 0 is the lowest priority. The predefined web-cache service has an assigned priority of 240.

WCCPv2 Support for Multiple Routers

WCCPv2 allows multiple routers to be attached to a cluster of cache engines. The use of multiple routers in a service group allows for redundancy, interface aggregation, and distribution of the redirection load. WCCPv2 supports up to 32 routers per service group. Each service group is established and maintained independently.

WCCPv2 MD5 Security

WCCPv2 provides optional authentication that enables you to control which routers and content engines become part of the service group using passwords and the Hashed Message Authentication Code—Message Digest (HMAC MD5) standard. Shared-secret MD5 one-time authentication (set using the **ip wccp [password [0 | 7] password]** global configuration command) enables messages to be protected against interception, inspection, and replay.

WCCPv2 Web Cache Packet Return

If a content engine is unable to provide a requested object it has cached due to error or overload, the content engine will return the request to the router for onward transmission to the originally specified destination server. WCCPv2 provides a check on packets that determines which requests have been returned from the content engine unserved. Using this information, the router can then forward the request to the originally targeted server (rather than attempting to resend the request to the content engine cluster). This process provides error handling transparency to clients.

Typical reasons why a content engine would reject packets and initiate the packet return feature include the following:

- Instances when the content engine is overloaded and has no room to service the packets
- Instances when the content engine is filtering for certain conditions that make caching packets counterproductive (for example, when IP authentication has been turned on)

WCCPv2 Load Distribution

WCCPv2 can be used to adjust the load being offered to individual content engines to provide an effective use of the available resources while helping to ensure high quality of service (QoS) to the clients. WCCPv2 allows the designated content engine to adjust the load on a particular content engine and balance the load across the content engines in a cluster. WCCPv2 uses three techniques to perform load distribution:

- Hot spot handling—Allows an individual hash bucket to be distributed across all the content engines. Prior to WCCPv2, information from one hash bucket could go to only one content engine.
- Load balancing—Allows the set of hash buckets assigned to a content engine to be adjusted so that the load can be shifted from an overwhelmed content engine to other members that have available capacity.
- Load shedding—Enables the router to selectively redirect the load to avoid exceeding the capacity of a content engine.

The use of these hashing parameters prevents one content engine from being overloaded and reduces the potential for bottlenecks.

WCCP VRF Support

The WCCP VRF Support feature enhances the WCCPv2 protocol by implementing support for virtual routing and forwarding (VRF).

The WCCP VRF Support feature allows service groups to be configured on a per-VRF basis in addition to those defined globally.

Along with the service identifier, the VRF of WCCP protocol packets arriving at the router is used to associate cache-engines with a configured service group.

The same VRF must have the interface on which redirection is applied, the interface which is connected to cache engine, and the interface on which the packet would have left if it had not been redirected.

WCCP VRF Tunnel Interfaces

In releases that support the WCCP VRF Support feature, the use of GRE redirection results in the creation of new tunnel interfaces. You can display these tunnel interfaces by entering the **show ip interface brief | include tunnel** command:

```
Device# show ip interface brief | include tunnel

Tunnel0      172.16.0.1      YES unset  up
Tunnel1      172.16.0.1      YES unset  up
Tunnel2      172.16.0.1      YES unset  up
Tunnel3      172.16.0.1      YES unset  up
Device#
```

The tunnel interfaces are automatically created in order to process outgoing GRE-encapsulated traffic for WCCP. The tunnel interfaces appear when a content engine connects and requests GRE redirection. The tunnel interfaces are not created directly by WCCP, but are created indirectly via a tunnel application programming interface (API). WCCP does not have direct knowledge of the tunnel interfaces, but can redirect packets to them, resulting in the appropriate encapsulation being applied to the packets. After the appropriate encapsulation is applied, the packet is then sent to the content engine.



Note The tunnel interfaces are not used to connect with incoming WCCP GRE return packets.

One tunnel is created for each service group that is using GRE redirection. One additional tunnel is created to provide an IP address that allows the other tunnel group interfaces to be unnumbered but still enabled for IPv4.

You can confirm the connection between the tunnels and WCCP by entering the **show tunnel groups wccp** command:

```
Device# show tunnel groups wccp

WCCP : service group 0 in "Default", ver v2, assgnmnt: hash-table
      intf: Tunnel0, locally sourced
WCCP : service group 317 in "Default", ver v2, assgnmnt: hash-table
      intf: Tunnel3, locally sourced
WCCP : service group 318 in "Default", ver v2, assgnmnt: hash-table
      intf: Tunnel2, locally sourced
```

You can display additional information about each tunnel interface by entering the **show tunnel interface interface-number** command:

```
Device# show tunnel interface t0

Tunnel0
  Mode:multi-GRE/IP, Destination UNKNOWN, Source 10.1.1.80
  Application ID 2: WCCP : service group 0 in "Default", ver v2, assgnmnt: hash-table
  Linestate - current up
  Internal linestate - current up, evaluated up

Device# show tunnel interface t1

Tunnel1
  Mode:multi-GRE/IP, Destination UNKNOWN, Source 172.16.0.1
  Application ID 2: unspecified
  Linestate - current up
  Internal linestate - current up, evaluated up

Device# show tunnel interface t2

Tunnel2
  Mode:multi-GRE/IP, Destination UNKNOWN, Source 10.1.1.80
  Application ID 2: WCCP : service group 318 in "Default", ver v2, assgnmnt: hash-table
  Linestate - current up
  Internal linestate - current up, evaluated up

Device# show tunnel interface t3

Tunnel3
  Mode:multi-GRE/IP, Destination UNKNOWN, Source 10.1.1.80
  Application ID 2: WCCP : service group 317 in "Default", ver v2, assgnmnt: hash-table
  Linestate - current up
  Internal linestate - current up, evaluated up
Device#
```

Note that the service group number shown in the examples is the internal tunnel representation of the WCCP service group number. Group 0 is the web-cache service. To determine the dynamic services, subtract 256 from the displayed service group number to convert to the WCCP service group number. For interfaces that are used for redirection, the source address shown is the WCCP router ID.

You can display information about the connected content engines and encapsulation, including software packet counters, by entering the **show adjacency [tunnel-interface] [encapsulation] [detail] [internal]** command:

```
Device# show adjacency t0
```

```
Protocol Interface          Address
IP          Tunnel0        10.1.1.82(3)
```

```
Device# show adjacency t0 encapsulation
```

```
Protocol Interface          Address
IP          Tunnel0        10.1.1.82(3)
```

```
Encap length 28
4500000000000000FF2F7D2B1E010150
1E0101520000883E00000000
Provider: TUNNEL
Protocol header count in macstring: 3
  HDR 0: ipv4
    dst: static, 10.1.1.82
    src: static, 10.1.1.80
    prot: static, 47
    ttl: static, 255
    df: static, cleared
    per packet fields: tos ident tl chksm
  HDR 1: gre
    prot: static, 0x883E
    per packet fields: none
  HDR 2: wccpv2
    dyn: static, cleared
    sgID: static, 0
    per packet fields: alt altB priB
```

```
Device# show adjacency t0 detail
```

```
Protocol Interface          Address
IP          Tunnel0        10.1.1.82(3)
                                connectionid 1
                                0 packets, 0 bytes
                                epoch 0
                                sourced in sev-epoch 1
                                Encap length 28
                                4500000000000000FF2F7D2B1E010150
                                1E0101520000883E00000000
                                Tun endpt
                                Next chain element:
                                IP adj out of Ethernet0/0, addr 10.1.1.82
```

```
Device# show adjacency t0 internal
```

```
Protocol Interface          Address
IP          Tunnel0        10.1.1.82(3)
                                connectionid 1
                                0 packets, 0 bytes
                                epoch 0
                                sourced in sev-epoch 1
                                Encap length 28
                                4500000000000000FF2F7D2B1E010150
                                1E0101520000883E00000000
                                Tun endpt
                                Next chain element:
                                IP adj out of Ethernet0/0, addr 10.1.1.82
                                parent oce 0x4BC76A8
                                frame originated locally (Null0)
                                L3 mtu 17856
```

```

Device#
Flags (0x2808C4)
Fixup enabled (0x40000000)
  GRE WCCP redirection
HWIDB/IDB pointers 0x55A13E0/0x35F5A80
IP redirect disabled
Switching vector: IPv4 midchain adj oce
IP Tunnel stack to 10.1.1.82 in Default (0x0)
  nh tracking enabled: 10.1.1.82/32
  IP adj out of Ethernet0/0, addr 10.1.1.82
Adjacency pointer 0x4BC74D8
Next-hop 10.1.1.82

```

WCCP Bypass Packets

WCCP intercepts IP packets and redirects those packets to a destination other than the destination that is specified in the IP header. Typically the packets are redirected from a web server on the Internet to a web cache that is local to the destination.

Occasionally a web cache cannot manage the redirected packets appropriately and returns the packets unchanged to the originating router. These packets are called bypass packets and are returned to the originating router using either Layer 2 forwarding without encapsulation (L2) or encapsulated in generic routing encapsulation (GRE). The router decapsulates and forwards the packets normally. The VRF associated with the ingress interface (or the global table if there is no VRF associated) is used to route the packet to the destination.

GRE is a tunneling protocol developed by Cisco that encapsulates packet types from a variety of protocols inside IP tunnels, creating a virtual point-to-point link over an IP network.

WCCP Closed Services and Open Services

In applications where packets are intercepted and redirected by a Cisco IOS router to external WCCP client devices, it may be necessary to block the packets for the application when a WCCP client device is not available. This blocking is achieved by configuring a WCCP closed service. When a WCCP service is configured as closed, the packets that fulfill the services, but do not have an active client device, are discarded.

By default, WCCP operates as an open service, wherein communication between clients and servers proceeds normally in the absence of an intermediary device.

The **ip wccp service-list** or the **ipv6 wccp service-list** command can be used for both closed-mode and open-mode services. Use the **service-list** keyword and *service-access-list* argument to register an application protocol type or port number. Use the **mode** keyword to select an open or closed service.

WCCP Outbound ACL Check

When WCCP is enabled for redirection on an ingress interface, the packets are redirected by WCCP and instead egress on an interface other than the destination that is specified in the IP header. The packets are still subject to ACLs configured on the ingress interface. However, redirection can cause the packets to bypass the ACL configured on the original egress interface. Packets that would have been dropped because of the ACL configured on the original egress interface can be sent out on the redirect egress interface, which poses a possible security problem. Enabling the WCCP Outbound ACL check feature ensures that redirected packets are subject to any ACL conditions configured on the original egress interface.

WCCP Service Groups

WCCP is a component of Cisco IOS software that redirects traffic with defined characteristics from its original destination to an alternative destination. The typical application of WCCP is to redirect traffic bound for a remote web server to a local web cache to improve response time and optimize network resource usage.

The nature of the selected traffic for redirection is defined by service groups (see figure below) specified on content engines and communicated to routers by using WCCP. The maximum number of service groups allowed across all VRFs is 256.

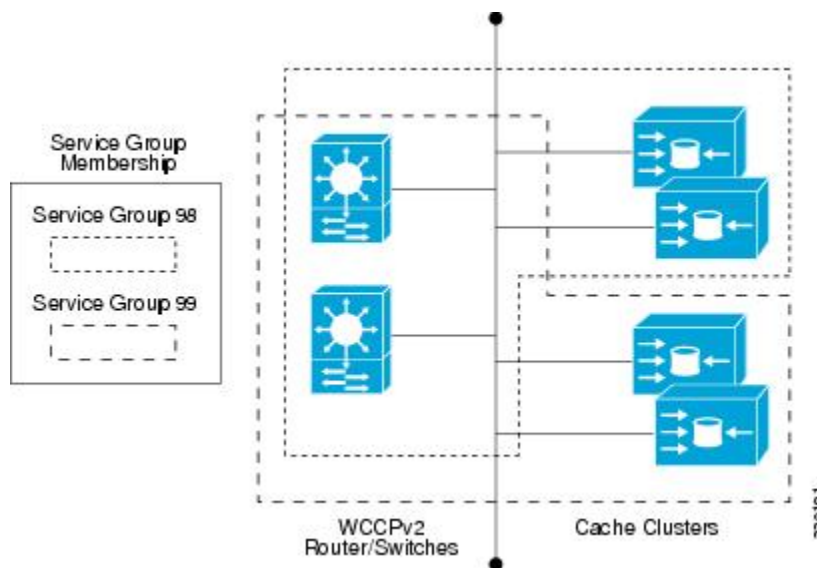
WCCPv2 supports up to 32 routers per service group. Each service group is established and maintained independently.

WCCPv2 uses service groups based on logical redirection services, deployed for intercepting and redirecting traffic. The standard service is web cache, which intercepts TCP port 80 (HTTP) traffic and redirects that traffic to the content engines. This service is referred to as a *well-known service*, because the characteristics of the web cache service are known by both the router and content engines. A description of a well-known service is not required beyond a service identification. To specify the standard web cache service, use the **ip wccp** or the **ipv6 wccp** command with the **web-cache** keyword.



Note More than one service can run on a router at the same time, and routers and content engines can be part of multiple service groups at the same time.

Figure 9: WCCP Service Groups



The dynamic services are defined by the content engines; the content engine instructs the router which protocol or ports to intercept, and how to distribute the traffic. The router itself does not have information on the characteristics of the dynamic service group’s traffic, because this information is provided by the first content engine to join the group. In a dynamic service, up to eight ports can be specified within a single protocol.

Cisco Content Engines, for example, use dynamic service 99 to specify a reverse-proxy service. However, other content engine devices may use this service number for some other service.

WCCP—Check All Services

An interface may be configured with more than one WCCP service. When more than one WCCP service is configured on an interface, the precedence of a service depends on the relative priority of the service compared to the priority of the other configured services. Each WCCP service has a priority value as part of its definition. When an interface is configured with more than one WCCP service, the precedence of the packets is matched against service groups in priority order.



Note The priority of a WCCP service group cannot be configured via Cisco IOS software.

With the **ip wccp check services all** or the **ipv6 wccp check services all** command, WCCP can be configured to check all configured services for a match and perform redirection for those services if appropriate. The caches to which packets are redirected can be controlled by a redirect ACL and by the service priority. The **ip wccp check services all** commands must be configured at global level to support multiple WCCP services.

If no WCCP services are configured with a redirect ACL, the services are considered in priority order until a service is found that matches the IP packet. If no services match the packet, the packet is not redirected. If a service matches the packet and the service has a redirect ACL configured, then the IP packet will be checked against the ACL. If the packet is rejected by the ACL, the packet will not be passed down to lower priority services unless the **ip wccp check services all** or the **ipv6 wccp check services all** command is configured. When the **ip wccp check services all** or the **ipv6 wccp check services all** command is configured, WCCP will continue to attempt to match the packet against any remaining lower priority services configured on the interface.

WCCP Interoperability with NAT

To redirect traffic using WCCP to a router running WAAS software that is also configured with NAT, enable the **ip nat inside** or the **ipv6 nat inside** command on the WAAS interface. If you are not able to configure the **ip nat inside** or the **ipv6 nat inside** command on the WAAS interface, disable Cisco Express Forwarding. You must also update the WCCP redirect ACL to include a private address to ensure that pretranslated traffic is redirected.

WCCP Troubleshooting Tips

CPU usage may be very high when WCCP is enabled. The WCCP counters enable a determination of the bypass traffic directly on the router and can indicate whether the cause is high CPU usage due to enablement of WCCP. In some situations, 10 percent bypass traffic may be normal; in other situations, 10 percent may be high. However, any figure above 25 percent should prompt a closer investigation of what is occurring in the web cache.

If the counters suggest that the level of bypass traffic is high, the next step is to examine the bypass counters in the content engine and determine why the content engine is choosing to bypass the traffic. You can log in to the content engine console and use the CLI to investigate further. The counters allow you to determine the percent of traffic being bypassed.

You can use the **clear ipv6 wccp service-id** command to remove the IPv6 WCCP statistics (counts) maintained on the router for a particular service.

You can use the **clear wccp** command to remove all (IPv4 and IPv6) WCCP statistics (counts) maintained on the router for a particular service.

You can use the **show ipv6 wccp** command to display the IPv6 WCCP global statistics (counts).

You can use the **show wccp** command to display all (IPv4 and IPv6) WCCP global statistics (counts).

How to Configure WCCP

The following configuration tasks assume that you have already installed and configured the content engines you want to include in your network. You must configure the content engines in the cluster before configuring WCCP functionality on your routers or switches. Refer to the [Cisco Cache Engine User Guide](#) for content engine configuration and setup tasks.

Configuring WCCP

Perform this task to configure WCCP.

Until you configure a WCCP service using the **ip wccp {web-cache | service-number}** global configuration command, WCCP is disabled on the device. The first use of a form of the **ip wccp** command enables WCCP. By default WCCPv2 is used for services, but you can use WCCPv1 functionality instead. To change the running version of WCCP from Version 2 to Version 1, or to return to WCCPv2 after an initial change, use the **ip wccp version** command in global configuration mode.

If a function is not allowed in WCCPv1, an error prompt will be printed to the screen. For example, if WCCPv1 is running on the device and you try to configure a dynamic service, the following message will be displayed: “WCCP V1 only supports the web-cache service.” The **show ip wccp EXEC** command will display the WCCP protocol version number that is running on your device.

Use the **ip wccp web-cache password** command to set a password for a device and the content engines in a service group. MD5 password security requires that each device and content engine that wants to join a service group be configured with the service group password. The password must be up to eight characters in length. Each content engine or device in the service group will authenticate the security component in a received WCCP packet immediately after validating the WCCP message header. Packets failing authentication will be discarded.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip wccp version {1 | 2}**
4. **ip wccp {web-cache | service-number} [group-address multicast-address] [redirect-list access-list] [group-list access-list] [password password [0 | 7]]**
5. **interface type number**
6. **ip wccp {web-cache | service-number} redirect {in | out}**
7. **exit**
8. **interface type number**
9. **ip wccp redirect exclude in**

DETAILED STEPS

| | Command or Action | Purpose |
|--------|-------------------|-------------------------------|
| Step 1 | enable | Enables privileged EXEC mode. |

| | Command or Action | Purpose |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | Example: Device> enable | <ul style="list-style-type: none"> Enter your password if prompted. |
| Step 2 | configure terminal Example: Device# configure terminal | Enters global configuration mode. |
| Step 3 | ip wccp version {1 2} Example: Device(config)# ip wccp version 2 | Specifies which version of WCCP to configure on a device. <ul style="list-style-type: none"> WCCPv2 is the default running version. |
| Step 4 | ip wccp {web-cache service-number} [group-address multicast-address] [redirect-list access-list] [group-list access-list] [password password [0 7]] Example: Device(config)# ip wccp web-cache password pwd | Specifies a web-cache or dynamic service to enable on a device, specifies the IP multicast address used by the service group, specifies any access lists to use, specifies whether to use MD5 authentication, and enables the WCCP service. <ul style="list-style-type: none"> Note The password length must not exceed 8 characters. |
| Step 5 | interface type number Example: Device(config)# interface GigabitEthernet 0/0 | Targets an interface number for which the web cache service will run, and enters interface configuration mode. |
| Step 6 | ip wccp {web-cache service-number} redirect {in out} Example: Device(config-if)# ip wccp web-cache redirect in | Enables packet redirection on an outbound or inbound interface using WCCP. <ul style="list-style-type: none"> As indicated by the out and in keyword options, redirection can be specified for outbound interfaces or inbound interfaces. |
| Step 7 | exit Example: Device(config-if)# exit | Exits interface configuration mode. |
| Step 8 | interface type number Example: Device(config)# interface GigabitEthernet 0/2/0 | Targets an interface number on which to exclude traffic for redirection, and enters interface configuration mode. |
| Step 9 | ip wccp redirect exclude in Example: Device(config-if)# ip wccp redirect exclude in | (Optional) Excludes traffic on the specified interface from redirection. |

Configuring Closed Services

Perform this task to specify the number of service groups for WCCP, to configure a service group as a closed or open service, and to optionally specify a check of all services.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Enter one of the following commands:
 - **ip wccp [vrf vrf-name] service-number [service-list service-access-list mode {open | closed}]**
 - or
 - **ip wccp [vrf vrf-name] web-cache mode {open | closed}**
4. **ip wccp check services all**
5. **ip wccp [vrf vrf-name] {web-cache | service-number}**
6. **exit**

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Step 1 | enable Example: Device> enable | Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted. |
| Step 2 | configure terminal Example: Device# configure terminal | Enters global configuration mode. |
| Step 3 | Enter one of the following commands: <ul style="list-style-type: none"> • ip wccp [vrf vrf-name] service-number [service-list service-access-list mode {open closed}] • or • ip wccp [vrf vrf-name] web-cache mode {open closed} Example: Device(config)# ip wccp 90 service-list 120 mode closed or Device(config)# ip wccp web-cache mode closed | Configures a dynamic WCCP service as closed or open. or Configures a web-cache service as closed or open. Note When configuring the web-cache service as a closed service, you cannot specify a service access list. Note When configuring a dynamic WCCP service as a closed service, you must specify a service access list. |
| Step 4 | ip wccp check services all Example: | (Optional) Enables a check of all WCCP services. |

| | Command or Action | Purpose |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | Device(config)# ip wccp check services all | <ul style="list-style-type: none"> Use this command to configure WCCP to check the other configured services for a match and perform redirection for those services if appropriate. The caches to which packets are redirected can be controlled by the redirect ACL and not just the service description. <p>Note The ip wccp check services all command is a global WCCP command that applies to all services and is not associated with a single service.</p> |
| Step 5 | ip wccp [<i>vrf vrf-name</i>] { web-cache <i>service-number</i> } Example: Device(config)# ip wccp 201 | Specifies the WCCP service identifier. <ul style="list-style-type: none"> You can specify the standard web-cache service or a dynamic service number from 0 to 255. The maximum number of services that can be specified is 256. |
| Step 6 | exit Example: Device(config)# exit | Exits to privileged EXEC mode. |

Registering a Router to a Multicast Address

If you decide to use the multicast address option for your service group, you must configure the router to listen for the multicast broadcasts on an interface.

For network configurations where redirected traffic needs to traverse an intervening router, the router being traversed must be configured to perform IP multicast routing. You must configure the following two components to enable traversal over an intervening router:

- Enable IP multicast routing using the **ip multicast-routing** global configuration command.
- Enable the interfaces to which the cache engines will connect to receive multicast transmissions using the **ip wccp group-listen** interface configuration command.

SUMMARY STEPS

- enable**
- configure terminal**
- ip multicast-routing** [*vrf vrf-name*] [**distributed**]
- ip wccp** [*vrf vrf-name*] {**web-cache** | *service-number*} **group-address** *multicast-address*
- interface** *type number*
- ip pim** {**sparse-mode** | **sparse-dense-mode** | **dense-mode** [**proxy-register** {**list** *access-list* | **route-map** *map-name*}]}
- ip wccp** [*vrf vrf-name*] {**web-cache** | *service-number*} **group-listen**

DETAILED STEPS

| | Command or Action | Purpose |
|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Step 1 | enable Example: Device> enable | Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted. |
| Step 2 | configure terminal Example: Device# configure terminal | Enters global configuration mode. |
| Step 3 | ip multicast-routing [vrf vrf-name] [distributed] Example: Device(config)# ip multicast-routing | Enables IP multicast routing. |
| Step 4 | ip wccp [vrf vrf-name] {web-cache service-number} group-address multicast-address Example: Device(config)# ip wccp 99 group-address 239.1.1.1 | Specifies the multicast address for the service group. |
| Step 5 | interface type number Example: Device(config)# interface ethernet 0/0 | Enables the interfaces to which the content engines will connect to receive multicast transmissions for which the web cache service will run, and enters interface configuration mode. |
| Step 6 | ip pim {sparse-mode sparse-dense-mode dense-mode [proxy-register {list access-list route-map map-name}]} Example: Device(config-if)# ip pim dense-mode | (Optional) Enables Protocol Independent Multicast (PIM) on an interface. Note To ensure correct operation of the ip wccp group-listen command on Cisco 7600 series routers, you must enter the ip pim command in addition to the ip wccp group-listen command. |
| Step 7 | ip wccp [vrf vrf-name] {web-cache service-number} group-listen Example: Device(config-if)# ip wccp 99 group-listen | Configures an interface to enable or disable the reception of IP multicast packets for WCCP. |

Using Access Lists for a WCCP Service Group

Perform this task to configure the device to use an access list to determine which traffic should be directed to which content engines.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list access-list-number remark remark**
4. **access-list access-list-number permit {source [source-wildcard] | any} [log]**
5. **access-list access-list-number remark remark**
6. **access-list access-list-number deny {source [source-wildcard] | any} | [log]**
7. Repeat some combination of Steps 3 through 6 until you have specified the sources on which you want to base your access list.
8. **ip wccp [vrf vrf-name] web-cache group-list access-list**
9. **ip wccp [vrf vrf-name] web-cache redirect-list access-list**

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Step 1 | <p>enable</p> <p>Example:</p> <pre>Device> enable</pre> | <p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> • Enter your password if prompted. |
| Step 2 | <p>configure terminal</p> <p>Example:</p> <pre>Device# configure terminal</pre> | <p>Enters global configuration mode.</p> |
| Step 3 | <p>access-list access-list-number remark remark</p> <p>Example:</p> <pre>Device(config)# access-list 1 remark Give access to user1</pre> | <p>(Optional) Adds a user-friendly comment about an access list entry.</p> <ul style="list-style-type: none"> • A remark of up to 100 characters can precede or follow an access list entry. |
| Step 4 | <p>access-list access-list-number permit {source [source-wildcard] any} [log]</p> <p>Example:</p> <pre>Device(config)# access-list 1 permit 172.16.5.22 0.0.0.0</pre> | <p>Creates an access list that enables or disables traffic redirection to the cache engine and permits the specified source based on a source address and wildcard mask.</p> <ul style="list-style-type: none"> • Every access list needs at least one permit statement; it does not need to be the first entry. • Standard IP access lists are numbered 1 to 99 or 1300 to 1999. • If the <i>source-wildcard</i> is omitted, a wildcard mask of 0.0.0.0 is assumed, meaning match on all bits of the source address. • Optionally use the keyword any as a substitute for the <i>source source-wildcard</i> to specify the source and source wildcard of 0.0.0.0 255.255.255.255. |

| | Command or Action | Purpose |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | <ul style="list-style-type: none"> In this example, host 172.16.5.22 is allowed to pass the access list. |
| Step 5 | access-list <i>access-list-number</i> remark <i>remark</i> Example: Device(config)# access-list 1 remark Give access to user1 | (Optional) Adds a user-friendly comment about an access list entry. <ul style="list-style-type: none"> A remark of up to 100 characters can precede or follow an access list entry. |
| Step 6 | access-list <i>access-list-number</i> deny { <i>source</i> [<i>source-wildcard</i>] any } [log] Example: Device(config)# access-list 1 deny 172.16.7.34 0.0.0.0 | Denies the specified source based on a source address and wildcard mask. <ul style="list-style-type: none"> If the <i>source-wildcard</i> is omitted, a wildcard mask of 0.0.0.0 is assumed, meaning match on all bits of the source address. Optionally use the abbreviation <i>any</i> as a substitute for the <i>source source-wildcard</i> to specify the source and source wildcard of 0.0.0.0 255.255.255.255. In this example, host 172.16.7.34 is denied passing the access list. |
| Step 7 | Repeat some combination of Steps 3 through 6 until you have specified the sources on which you want to base your access list. | Remember that all sources not specifically permitted are denied by an implicit deny statement at the end of the access list. |
| Step 8 | ip wccp [<i>vrf vrf-name</i>] web-cache group-list <i>access-list</i> Example: Device(config) ip wccp web-cache group-list 1 | Indicates to the device from which IP addresses of content engines to accept packets. |
| Step 9 | ip wccp [<i>vrf vrf-name</i>] web-cache redirect-list <i>access-list</i> Example: Device(config)# ip wccp web-cache redirect-list 1 | (Optional) Disables caching for certain clients. |

Enabling the WCCP Outbound ACL Check



Note When all redirection is performed in the hardware, the mode of redirection will change when outbound ACL checking is enabled. The first packet is switched in software to allow the extra ACL check to be performed before a shortcut is installed.

SUMMARY STEPS

1. enable

2. **configure terminal**
3. **ip wccp** [*vrf vrf-name*] {**web-cache** | *service-number*} [**group-address** *multicast-address*] [**redirect-list** *access-list*] [**group-list** *access-list*] [**password** *password*]
4. **ip wccp check acl outbound**
5. **exit**

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Step 1 | enable Example: Device> enable | Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted. |
| Step 2 | configure terminal Example: Device# configure terminal | Enters global configuration mode. |
| Step 3 | ip wccp [<i>vrf vrf-name</i>] { web-cache <i>service-number</i> } [group-address <i>multicast-address</i>] [redirect-list <i>access-list</i>] [group-list <i>access-list</i>] [password <i>password</i>] Example: Device(config)# ip wccp web-cache | Enables the support for a Cisco content engine service group or any content engine service group and configures a redirect ACL list or group ACL. Note The web-cache keyword is for WCCP version 1 and version 2 and the <i>service-number</i> argument is for WCCP version 2 only. |
| Step 4 | ip wccp check acl outbound Example: Device(config)# ip wccp check acl outbound | Checks the access control list (ACL) for egress interfaces for packets redirected by WCCP. |
| Step 5 | exit Example: Device(config)# exit | Exits global configuration. |

Enabling WCCP Interoperability with NAT

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip nat inside**
5. **ip wccp** *service-number* **redirect in**
6. **exit**

7. **interface** *type number*
8. **ip nat outside**
9. **ip wccp** *service-number* **redirect in**
10. **exit**
11. **interface** *type number*
12. **ip nat inside**
13. **ip wccp redirect exclude in**

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|-------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Step 1 | enable Example: Router> enable | Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted. |
| Step 2 | configure terminal Example: Router# configure terminal | Enters global configuration mode. |
| Step 3 | interface <i>type number</i> Example: Router(config)# interface ethernet 1 | Specifies an interface on which to enable NAT and enters interface configuration mode. <ul style="list-style-type: none"> • This is the LAN-facing interface. |
| Step 4 | ip nat inside Example: Router(config-if)# ip nat inside | Designates that traffic originating from or destined for the interface is subject to NAT and indicates that the interface is connected to the inside network (the network subject to NAT translation). |
| Step 5 | ip wccp <i>service-number</i> redirect in Example: Router(config-if)# ip wccp 61 redirect in | Enables packet redirection on an inbound interface using WCCP. |
| Step 6 | exit Example: Router(config-if)# exit | Exits interface configuration mode and returns to global configuration mode. |
| Step 7 | interface <i>type number</i> Example: Router(config)# interface ethernet 2 | Specifies an interface on which to enable NAT and enters interface configuration mode. <ul style="list-style-type: none"> • This is the WAN-facing interface. |

| | Command or Action | Purpose |
|----------------|----------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Step 8 | ip nat outside Example: Router(config-if)# ip nat outside | Designates that traffic originating from or destined for the interface is subject to NAT and indicates that the interface is connected to the outside network. |
| Step 9 | ip wccp <i>service-number</i> redirect in Example: Router(config-if)# ip wccp 62 redirect in | Enables packet redirection on an inbound interface using WCCP. |
| Step 10 | exit Example: Router(config-if)# exit | Exits interface configuration mode and returns to global configuration mode. |
| Step 11 | interface <i>type number</i> Example: Router(config)# interface ethernet 3 | Specifies an interface on which to enable NAT and enters interface configuration mode. <ul style="list-style-type: none"> • This is the WAAS-facing interface. |
| Step 12 | ip nat inside Example: Router(config-if)# ip nat inside | Designates that traffic originating from or destined for the interface is subject to NAT and indicates that the interface is connected to the inside network (the network subject to NAT translation). |
| Step 13 | ip wccp redirect exclude in Example: Router(config-if)# ip wccp redirect exclude in | Configures an interface to exclude packets received on an interface from being checked for redirection.. |

Verifying and Monitoring WCCP Configuration Settings

SUMMARY STEPS

1. enable
2. show ip wccp [web-cache *service-number*] [detail view]
3. show ip interface
4. more system:running-config

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|----------------------------------|--------------------------------------------------------------------------------------------------------------------|
| Step 1 | enable Example: | Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted. |

| | Command or Action | Purpose |
|---------------|--------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | Device> enable | |
| Step 2 | show ip wccp [web-cache [service-number] [detail view] Example: Device# show ip wccp 24 detail | Displays global information related to WCCP, including the protocol version running, the number of content engines in the router service group, which content engine group is allowed to connect to the router, and which access list is being used. <ul style="list-style-type: none"> • <i>service-number</i>—(Optional) Dynamic number of the web-cache service group being controlled by the content engine. The range is from 0 to 99. For web caches that use Cisco Content Engines, the reverse proxy service is indicated by a value of 99. • web-cache—(Optional) statistics for the web-cache service. • detail—(Optional) other members of a particular service group or web cache that have or have not been detected. • view—(Optional) information about a router or all web caches. |
| Step 3 | show ip interface Example: Device# show ip interface | Displays status about whether any ip wccp redirection commands are configured on an interface; for example, “Web Cache Redirect is enabled / disabled.” |
| Step 4 | more system:running-config Example: Device# more system:running-config | (Optional) Displays contents of the running configuration file (equivalent to the show running-config command). |

Configuration Examples for WCCP

Example: Changing the Version of WCCP on a Router

The following example shows how to change the WCCP version from the default of WCCPv2 to WCCPv1, and enabling the web-cache service in WCCPv1:

```
Device# show ip wccp

% WCCP version 2 is not enabled
Device# configure terminal

Device(config)# ip wccp version 1
```

```

Device(config)# end
Device# show ip wccp

% WCCP version 1 is not enabled
Device# configure terminal

Device(config)# ip wccp web-cache
Device(config)# end
Device# show ip wccp

Global WCCP information:
  Router information:
    Router Identifier:          10.4.9.8
    Protocol Version:          1.0
  .
  .
  .

```

Example: Configuring a General WCCPv2 Session

```

Device# configure terminal
Device(config)# ip wccp web-cache group-address 224.1.1.100 password password
Device(config)# ip wccp source-interface GigabitEthernet 0/1/0
Device(config)# ip wccp check services all
! Configures a check of all WCCP services.
Device(config)# interface GigabitEthernet 0/1/0
Device(config-if)# ip wccp web-cache redirect in
Device(config-if)# exit
Device(config)# interface GigabitEthernet 0/2/0
Device(config-if)# ip wccp redirect exclude in
Device(config-if)# exit

```

Example: Setting a Password for a Router and Content Engines

```

Router# configure terminal
Router(config)# ip wccp web-cache password password1

```

Example: Configuring a Web Cache Service

```

Device# configure terminal
Device(config)# ip wccp web-cache
Device(config)# interface GigabitEthernet 0/1/0
Device(config-if)# ip wccp web-cache redirect in
Device(config-if)# exit
Device# copy running-config startup-config

```

The following example shows how to configure a session in which redirection of HTTP traffic arriving on Gigabit Ethernet interface 0/1/0 is enabled:

```

Device# configure terminal
Device(config)# interface GigabitEthernet 0/1/0
Device(config-if)# ip wccp web-cache redirect in
Device(config-if)# exit
Device# show ip interface GigabitEthernet 0/1/0
.

```



```

.
.
WCCP Redirect inbound is enabled
WCCP Redirect exclude is disabled
.
.
.

```

Example: Running a Reverse Proxy Service

The following example assumes that you are configuring a service group using Cisco cache engines, which use dynamic service 99 to run a reverse proxy service:

```

Router# configure terminal
Router(config)# ip wccp 99
Router(config)# interface gigabitethernet 0/1/0
Router(config-if)# ip wccp 99 redirect out

```

Example: Registering a Router to a Multicast Address

```

Device# configure terminal
Device(config)# ip wccp web-cache group-address 224.1.1.100
Device(config)# interface gigabitethernet 0/1/0
Device(config-if)# ip wccp web cache group-listen

```

The following example shows a router configured to run a reverse proxy service, using the multicast address of 224.1.1.1. Redirection applies to packets outgoing via Gigabit Ethernet interface 0/1/0:

```

Device# configure terminal
Device(config)# ip wccp 99 group-address 224.1.1.1
Device(config)# interface gigabitethernet 0/1/0
Device(config-if)# ip wccp 99 redirect out

```

Example: Using Access Lists

To achieve better security, you can use a standard access list to notify the device which IP addresses are valid addresses for a content engine attempting to register with the current device. The following example shows a standard access list configuration session where the access list number is 10 for some sample hosts:

```

Device(config)# access-list 10 permit host 10.1.1.1
Device(config)# access-list 10 permit host 10.1.1.2
Device(config)# access-list 10 permit host 10.1.1.3
Device(config)# ip wccp web-cache group-list 10

```

To disable caching for certain clients, servers, or client/server pairs, you can use WCCP access lists. The following example shows that any requests coming from 10.1.1.1 to 10.3.1.1 will bypass the cache, and that all other requests will be serviced normally:

```

Device(config)# ip wccp web-cache redirect-list 120
Device(config)# access-list 120 deny tcp host 10.1.1.1 any
Device(config)# access-list 120 deny tcp any host 10.3.1.1
Device(config)# access-list 120 permit ip any any

```

The following example configures a device to redirect web-related packets received via Gigabit Ethernet interface 0/1/0, destined to any host except 209.165.200.224:

```
Device(config)# access-list 100 deny ip any host 209.165.200.224
Device(config)# access-list 100 permit ip any any
Device(config)# ip wccp web-cache redirect-list 100
Device(config)# interface gigabitethernet 0/1/0
Device(config-if)# ip wccp web-cache redirect in
```

Example: WCCP Outbound ACL Check Configuration

The following configuration example shows that the access list prevents traffic from network 10.0.0.0 leaving Gigabit Ethernet interface 0/1/0. Because the outbound ACL check is enabled, WCCP does not redirect that traffic. WCCP checks packets against the ACL before they are redirected.

```
Device(config)# ip wccp web-cache
Device(config)# ip wccp check acl outbound
Device(config)# interface gigabitethernet 0/1/0
Device(config-if)# ip access-group 10 out
Device(config-if)# exit
Device(config)# ip wccp web-cache redirect-list redirect-out
Device(config)# access-list 10 deny 10.0.0.0 0.255.255.255
Device(config)# access-list 10 permit any
```

If the outbound ACL check is disabled, the HTTP packets from network 10.0.0.0 would be redirected to a web cache. Users with that network address could retrieve web pages even though the network administrator wanted to prevent it.

Example: Verifying WCCP Settings

The following example shows how to verify your configuration changes by using the **more system:running-config** command in privileged EXEC mode. The following example shows that both the web cache service and dynamic service 99 are enabled on the device:

```
Device# more system:running-config

Building configuration...
Current configuration:
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service udp-small-servers
service tcp-small-servers
!
hostname router4
!
enable secret 5 $1$nSVy$faliJsVQXVPW.KuCxZNTh1
enable password password1
!
ip subnet-zero
ip wccp web-cache
ip wccp 99
ip domain-name cisco.com
ip name-server 10.1.1.1
ip name-server 10.1.1.2
```

```

ip name-server 10.1.1.3
!
!
interface GigabitEthernet0/1/1
ip address 10.3.1.2 255.255.255.0
no ip directed-broadcast
ip wccp web-cache redirect in
ip wccp 99 redirect in
no ip route-cache
no ip mroute-cache
!
interface GigabitEthernet0/1/0
ip address 10.4.1.1 255.255.255.0
no ip directed-broadcast
ip wccp 99 redirect in
no ip route-cache
no ip mroute-cache
!
interface Serial0
no ip address
no ip directed-broadcast
no ip route-cache
no ip mroute-cache
shutdown
!
interface Serial1
no ip address
no ip directed-broadcast
no ip route-cache
no ip mroute-cache
shutdown
!
ip default-gateway 10.3.1.1
ip classless
ip route 0.0.0.0 0.0.0.0 10.3.1.1
no ip http server
!
!
line con 0
transport input none
line aux 0
transport input all
line vty 0 4
password password1
login
!
end

```

The following example shows how to display global statistics related to WCCP:

```

Device# show ip wccp web-cache detail

WCCP Client information:
WCCP Client ID:      10.1.1.2
Protocol Version:    2.0
State:               Usable
Redirection:         L2
Packet Return:       L2
Packets Redirected:  0
Connect Time:        00:20:34
Assignment:          MASK
Mask SrcAddr  DstAddr  SrcPort DstPort

```

```

-----
0000: 0x00000000 0x00001741 0x0000 0x0000
Value SrcAddr  DstAddr  SrcPort  DstPort  CE-IP
-----
0000: 0x00000000 0x00000000 0x0000 0x0000 0x3C010102 (10.1.1.2)
0001: 0x00000000 0x00000001 0x0000 0x0000 0x3C010102 (10.1.1.2)
0002: 0x00000000 0x00000040 0x0000 0x0000 0x3C010102 (10.1.1.2)
0003: 0x00000000 0x00000041 0x0000 0x0000 0x3C010102 (10.1.1.2)
0004: 0x00000000 0x00000100 0x0000 0x0000 0x3C010102 (10.1.1.2)
0005: 0x00000000 0x00000101 0x0000 0x0000 0x3C010102 (10.1.1.2)
0006: 0x00000000 0x00000140 0x0000 0x0000 0x3C010102 (10.1.1.2)

```

For more information about the **show ip wccp web-cache** command, see the *Cisco IOS IP Application Services Command Reference*.

Example: Enabling WCCP Interoperability with NAT

```

Router(config)# interface ethernet1 ! This is the LAN-facing interface
Router(config-if)# ip nat inside
Router(config-if)# ip wccp 61 redirect in
Router(config-if)# exit
Router(config)# interface ethernet2 ! This is the WAN-facing interface
Router(config-if)# ip nat outside
Router(config-if)# ip wccp 62 redirect in
Router(config-if)# exit
Router(config)# interface ethernet3 ! This is the WAAS-facing interface
Router(config-if)# ip nat inside
Router(config-if)# ip wccp redirect exclude in

```

Additional References

Related Documents

| Related Topic | Document Title |
|-------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Cisco IOS commands | Cisco IOS Master Commands List, All Releases |
| Cisco ACNS software configuration information | <ul style="list-style-type: none"> • <i>Cisco ACNS Software Caching Configuration Guide, Release 4.2</i> • Cisco ACNS Software listing page on Cisco.com |
| IP access list overview, configuration tasks, and commands | <i>Cisco IOS Security Command Reference</i> |
| IP addressing and services commands and configuration tasks | <ul style="list-style-type: none"> • <i>Cisco IOS IP Addressing Services Configuration Guide</i> • <i>Cisco IOS IP Addressing Services Command Reference</i> |

| Related Topic | Document Title |
|-----------------------------------------------------------------------------------------------------------------|------------------------------------------------------------|
| WCCP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples | <i>Cisco IOS IP Application Services Command Reference</i> |

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. | To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs |

RFCs

| RFC | Title |
|---------------------------------------------------------------------------------------------|-------|
| No new or modified RFCs are supported, and support for existing RFCs has not been modified. | — |

Technical Assistance

| Description | Link |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|
| The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password. | http://www.cisco.com/cisco/web/support/index.html |

Feature Information for WCCP

Table 9: Feature Information for WCCP

| Feature Name | Releases | Feature Information |
|-------------------------------------|-----------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| WCCP Bypass Counters | 12.3(7)T 12.2(25)S | The WCCP Bypass Counters feature allows you to display a count of packets that have been bypassed by a web cache and returned to the originating router to be forwarded normally. The show ip wccp command was modified by this feature. |
| WCCP Closed Services | 12.4(11)T | The WCCP Closed Services feature permits WCCP services to be configured so that WCCP always intercepts traffic for such services but, if no WCCP client (such as a content engine) has registered to receive this traffic, packets are discarded. This behavior supports Application-Oriented Network Services (AONS) applications, which require traffic to be transparently intercepted using WCCP but do not want the packets to be forwarded to their destination if the WCCP client is unavailable to perform its processing. (This is contrary to the traditional use of WCCP to assist caches where the absence of a cache does not change the behavior as observed by the user.) The ip wccp command was modified by this feature. |
| WCCP Increased Services | 12.3(14)T 12.2(33)SRA 12.2(33)SXH | The WCCP Increased Services feature increases the number of services supported by WCCP to a maximum of 256 across all VRFs. The following commands were modified by this feature: ip wccp , ip wccp check services all , ip wccp outbound-acl-check , show ip wccp . |
| WCCP Layer 2 Redirection/Forwarding | 12.4(20)T | The WCCP Layer 2 Redirection/Forwarding feature allows directly connected Cisco content engines to use Layer 2 redirection, which is more efficient than Layer 3 redirection via GRE encapsulation. You can configure a directly connected Cache Engine to negotiate use of the WCCP Layer 2 Redirection/Forwarding feature. The WCCP Layer 2 Redirection/Forwarding feature requires no configuration on the router or switch. There are no new or modified commands associated with this feature. |
| WCCP L2 Return | 12.4(20)T | The WCCP L2 Return feature allows content engines to return packets to WCCP routers directly connected at Layer 2 by swapping the source and destination MAC addresses rather than tunneling packets back to the router inside a Layer 3 GRE tunnel. There are no new or modified commands associated with this feature. |

| Feature Name | Releases | Feature Information |
|----------------------------------------|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| WCCP Mask Assignment | 12.4(20)T | <p>The WCCP Mask Assignment feature introduces support for ACNS/WAAS devices using mask assignment as a cache engine assignment method.</p> <p>There are no new or modified commands associated with this feature.</p> |
| WCCP Outbound ACL Check | 12.3(7)T 12.2(25)S | <p>The WCCP Outbound ACL Check feature enables you to ensure that traffic redirected by WCCP at an input interface is subjected to the outbound ACL checks that may be configured on the output interface prior to redirection.</p> <p>This feature is supported by Web Cache Communication Protocol (WCCP) Version 1 and Version 2.</p> <p>The following commands were introduced or modified by this feature: ip wccp, ip wccp check acl outbound.</p> |
| WCCP Redirection on Inbound Interfaces | 12.1(3)T 15.0(1)S | <p>The WCCP Redirection on Inbound Interfaces feature enables interfaces to be configured for input redirection for a particular WCCP service. When this feature is enabled on an interface, all packets arriving at that interface are compared against the specified WCCP service. If the packets match, they will be redirected.</p> <p>The following commands were introduced or modified by this feature: ip wccp redirect-list.</p> |
| WCCP Version 2 | 12.0(3)T 15.0(1)S | <p>The WCCP Version 2 feature provides several enhancements and features to the WCCP protocol, including:</p> <ul style="list-style-type: none"> • The ability of multiple routers to service a content engine cluster. • Redirection of traffic other than HTTP (TCP port 80 traffic), including a variety of UDP and TCP traffic. • Optional authentication that enables you to control which routers and content engines become part of the service group using passwords and the HMAC MD5 standard. • A check on packets that determines which requests have been returned from the content engine unserved. • Load adjustments for individual content engines to provide an effective use of the available resources while helping to ensure high quality of service (QoS) to the clients. <p>The following commands were introduced or modified by this feature: clear ip wccp, ip wccp, ip wccp group-listen, ip wccp redirect, ip wccp redirect exclude in, ip wccp version, show ip wccp.</p> |

| Feature Name | Releases | Feature Information |
|------------------|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| WCCP VRF Support | 15.0(1)M 12.2(33)SRE | <p>The WCCP VRF Support feature provides enhancements to the existing WCCPv2 protocol which support VRF awareness.</p> <p>In Cisco IOS Release 12.2(33)SRE, this feature is supported only on Cisco 7200 NPE-G2 and Cisco 7304-NPE-G100 routers.</p> <p>The following commands were introduced or modified by this feature: clear ip wccp, debug ip wccp, ip wccp, ip wccp group-listen, ip wccp redirect, show ip wccp.</p> |



CHAPTER 7

Object Tracking: IPv6 Route Tracking

The Object Tracking: IPv6 Route Tracking feature expands the Enhanced Object Tracking (EOT) functionality to allow the tracking of IPv6 routes.

- [Finding Feature Information, on page 141](#)
- [Restrictions for Object Tracking: IPv6 Route Tracking, on page 141](#)
- [Information About Object Tracking: IPv6 Route Tracking, on page 142](#)
- [How to Configure Object Tracking: IPv6 Route Tracking, on page 142](#)
- [Configuration Examples for Object Tracking: IPv6 Route Tracking, on page 146](#)
- [Additional References for Object Tracking: IPv6 Route Tracking, on page 147](#)
- [Feature Information for Object Tracking: IPv6 Route Tracking, on page 148](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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

Restrictions for Object Tracking: IPv6 Route Tracking

Object Tracking: IPv6 Route Tracking is not Stateful Switchover (SSO)-aware and cannot be used with Hot Standby Router Protocol (HSRP), Virtual Router Redundancy Protocol (VRRP), or Gateway Load Balancing Protocol (GLBP) in SSO mode.

Information About Object Tracking: IPv6 Route Tracking

Enhanced Object Tracking and IPv6 Route Tracking

Enhanced Object Tracking (EOT) provides complete separation between the objects to be tracked and the action to be taken by a client when a tracked object changes. Thus, several clients such as Hot Standby Router Protocol (HSRP), Virtual Router Redundancy Protocol (VRRP), or Gateway Load Balancing Protocol (GLBP) can register interest with a tracking process, track the same object, and each take different a action when the object changes.

Each tracked object is identified by a unique number that is specified on the tracking CLI. Client processes use this number to track a specific object.

A tracking process periodically polls tracked objects and notes any change in value. The changes in the tracked object are communicated to interested client processes, either immediately or after a specified delay. The object values are reported as either up or down.

The Object Tracking: IPv6 Route Tracking feature expands EOT functionality to allow the tracking of IPv6 routes.

How to Configure Object Tracking: IPv6 Route Tracking

Tracking the IPv6-Routing State of an Interface

SUMMARY STEPS

1. **track timer interface** *{seconds | msec milliseconds}*
2. **track object-number interface type number ipv6 routing**
3. **carrier-delay**
4. **delay** *{up seconds [down seconds] | [up seconds] down seconds}*
5. **end**
6. **show track object-number**

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Step 1 | track timer interface <i>{seconds msec milliseconds}</i> Example: <pre>Device(config)# track timer interface 5</pre> | (Optional) Specifies the interval that a tracking process polls the tracked interface. <ul style="list-style-type: none"> • The default interval that the tracking process polls interface objects is 1 second. Note All polling frequencies can be configured down to 500 milliseconds, overriding the minimum 1-second interval configured using the msec keyword and <i>milliseconds</i> argument. |

| | Command or Action | Purpose |
|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Step 2 | track <i>object-number</i> interface <i>type number</i> ipv6 routing Example: <pre>Device(config)# track 1 interface GigabitEthernet 0/0/1 ipv6 routing</pre> | Tracks the IPv6-routing state of an interface and enters tracking configuration mode. <ul style="list-style-type: none"> IPv6-route tracking tracks an IPv6 route in the routing table and the ability of an interface to route IPv6 packets. |
| Step 3 | carrier-delay Example: <pre>Device(config-track)# carrier-delay</pre> | (Optional) Enables enhanced object tracking to consider the carrier-delay timer when tracking the status of an interface. |
| Step 4 | delay { up <i>seconds</i> [down <i>seconds</i>] [up <i>seconds</i>] down <i>seconds</i> } Example: <pre>Device(config-track)# delay up 30</pre> | (Optional) Specifies a period of time (in seconds) to delay communicating state changes of a tracked object. Note The up keyword specifies the time to delay the notification of an up event. The down keyword specifies the time to delay the notification of a down event. |
| Step 5 | end Example: <pre>Device(config-track)# end</pre> | Returns to privileged EXEC mode. |
| Step 6 | show track <i>object-number</i> Example: <pre>Device# show track 1</pre> | Displays tracking information. <ul style="list-style-type: none"> Use this command to verify the configuration. |

Tracking the Threshold of IPv6-Route Metrics

SUMMARY STEPS

- track timer ipv6 route** {*seconds* | **msec** *milliseconds*}
- track resolution ipv6 route** {**bgp** | **eigrp** | **isis** | **ospf** | **static** } *resolution-value*
- track** *object-number* **ipv6 route** *ipv6-address/prefix-length* **metric threshold**
- delay** {**up** *seconds* [**down** *seconds*] | [**up** *seconds*] **down** *seconds*}
- ipv6 vrf** *vrf-name*
- threshold metric** {**up** *number* [**down** *number*] | **down** *number* [**up** *number*] }
- end**
- show track** *object-number*

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Step 1 | <p>track timer ipv6 route {seconds msec milliseconds}</p> <p>Example:</p> <pre>Device(config)# track timer ipv6 route 20</pre> | <p>(Optional) Specifies the interval that a tracking process polls the tracked object.</p> <ul style="list-style-type: none"> The default interval that the tracking process polls IPv6-route objects is 15 seconds. <p>Note All polling frequencies can be configured down to 500 milliseconds, overriding the minimum 1-second interval configured using the msec keyword and <i>milliseconds</i> argument.</p> |
| Step 2 | <p>track resolution ipv6 route {bgp eigrp isis ospf static} resolution-value</p> <p>Example:</p> <pre>Device(config)# track resolution ipv6 route eigrp 300</pre> | <p>(Optional) Specifies resolution parameters for a tracked object.</p> <ul style="list-style-type: none"> Use this command to change the default metric resolution values. |
| Step 3 | <p>track object-number ipv6 route ipv6-address/prefix-length metric threshold</p> <p>Example:</p> <pre>Device(config)# track 6 ipv6 route 2001:DB8:0:ABCD::1/10 metric threshold</pre> | <p>Tracks the scaled metric value of an IPv6 route to determine if it is above or below a threshold and enters tracking configuration mode.</p> <ul style="list-style-type: none"> The default down value is 255, which equates to an inaccessible route. The default up value is 254. |
| Step 4 | <p>delay {up seconds [down seconds] [up seconds] down seconds}</p> <p>Example:</p> <pre>Device(config-track)# delay up 30</pre> | <p>(Optional) Specifies a period of time (in seconds) to delay communicating state changes of a tracked object.</p> <p>Note The up keyword specifies the time to delay the notification of an up event. The down keyword specifies the time to delay the notification of a down event.</p> |
| Step 5 | <p>ipv6 vrf vrf-name</p> <p>Example:</p> <pre>Device(config-track)# ipv6 vrf VRF1</pre> | <p>(Optional) Tracks an IPv6 route in a specific VPN virtual routing and forwarding (VRF) table.</p> |
| Step 6 | <p>threshold metric {up number [down number] down number [up number]}</p> <p>Example:</p> | <p>(Optional) Sets a metric threshold other than the default value.</p> |

| | Command or Action | Purpose |
|---------------|-----------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <pre>Device(config-track)# threshold metric up 254 down 255</pre> | <p>Note The up keyword specifies the up threshold. The state is up if the scaled metric for that route is less than or equal to the up threshold. The default up threshold is 254. The down keyword specifies the down threshold. The state is down if the scaled metric for that route is greater than or equal to the down threshold. The default down threshold is 255.</p> |
| Step 7 | <p>end</p> <p>Example:</p> <pre>Device(config-track)# end</pre> | Returns to privileged EXEC mode. |
| Step 8 | <p>show track object-number</p> <p>Example:</p> <pre>Device# show track 6</pre> | <p>(Optional) Displays tracking information.</p> <ul style="list-style-type: none"> • Use this command to verify the configuration. |

Tracking IPv6-Route Reachability

Perform this task to track the reachability of an IPv6 route. A tracked object is considered up when a routing table entry exists for the route and the route is accessible.

SUMMARY STEPS

1. **track timer ipv6 route** {seconds | msec milliseconds}
2. **track object-number ip route ip-address/prefix-length reachability**
3. **delay** {up seconds [down seconds] | [up seconds] down seconds}
4. **ipv6 vrf vrf-name**
5. **end**
6. **show track object-number**

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Step 1 | <p>track timer ipv6 route {seconds msec milliseconds}</p> <p>Example:</p> <pre>Device(config)# track timer ipv6 route 20</pre> | <p>(Optional) Specifies the interval that a tracking process polls the tracked object.</p> <ul style="list-style-type: none"> • The default interval that the tracking process polls IPv6-route objects is 15 seconds. <p>Note All polling frequencies can be configured down to 500 milliseconds, overriding the minimum 1-second interval configured using the msec keyword and <i>milliseconds</i> argument.</p> |

| | Command or Action | Purpose |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Step 2 | track <i>object-number</i> ip route <i>ip-address/prefix-length</i> reachability Example: <pre>Device(config)# track 4 ipv6 route 2001:DB8:0:AB82::1/10 reachability</pre> | Tracks the reachability of an IPv6 route and enters tracking configuration mode. |
| Step 3 | delay { up <i>seconds</i> [down <i>seconds</i>] [up <i>seconds</i>] down <i>seconds</i> } Example: <pre>Device(config-track)# delay up 30</pre> | (Optional) Specifies a period of time (in seconds) to delay communicating state changes of a tracked object. Note The up keyword specifies the time to delay the notification of an up event. The down keyword specifies the time to delay the notification of a down event. |
| Step 4 | ipv6 vrf <i>vrf-name</i> Example: <pre>Device(config-track)# ipv6 vrf VRF2</pre> | (Optional) Configures a VPN virtual routing and forwarding (VRF) table. |
| Step 5 | end Example: <pre>Device(config-track)# end</pre> | Returns to privileged EXEC mode. |
| Step 6 | show track <i>object-number</i> Example: <pre>Device# show track 4</pre> | (Optional) Displays tracking information. <ul style="list-style-type: none"> • Use this command to verify the configuration. |

Configuration Examples for Object Tracking: IPv6 Route Tracking

Example: Tracking the IPv6-Routing State of an Interface

The following example shows how to configure tracking for IPv6 routing on the GigabitEthernet 0/0/1 interface:

```
Device(config)# track timer interface 5
Device(config)# track 1 interface GigabitEthernet 0/0/1 ipv6 routing
Device(config-track)# carrier-delay
Device(config-track)# delay up 30
Device(config-track)# end
```

Example: Tracking the Threshold of IPv6-Route Metrics

The following example shows how to configure tracking for IPv6 metric thresholds:

```
Device(config)# track timer ipv6 route 20
Device(config)# track resolution ipv6 route eigrp 300
Device(config)# track 6 ipv6 route 2001:DB8:0:ABCD::1/10 metric threshold
Device(config-track)# delay up 30
Device(config-track)# ipv6 vrf VRF1
Device(config-track)# threshold metric up 254 down 255
Device(config-track)# end
```

Example: Tracking IPv6-Route Reachability

The following example shows how to configure tracking for IPv6-route reachability:

```
Device(config)# track timer ipv6 route 20
Device(config)# track 4 ipv6 route 2001:DB8:0:AB82::1/10 reachability
Device(config-track)# delay up 30
Device(config-track)# ipv6 vrf VRF2
Device(config-track)# end
```

Additional References for Object Tracking: IPv6 Route Tracking

Related Documents

| Related Topic | Document Title |
|----------------------------------|-------------------------------------------------------------|
| Cisco IOS commands | Cisco IOS Master Command List, All Releases |
| Object tracking | <i>Configuring Enhanced Object Tracking</i> |
| IP Application Services commands | <i>Cisco IOS IP Application Services Command Reference</i> |

Technical Assistance

| Description | Link |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------|
| <p>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</p> <p>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.</p> <p>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</p> | http://www.cisco.com/support |

Feature Information for Object Tracking: IPv6 Route Tracking

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

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

Table 10: Feature Information for Object Tracking: IPv6 Route Tracking

| Feature Name | Releases | Feature Information |
|--------------------------------------|----------------------------|---------------------------------------------------------------------------------------------------------|
| Object Tracking: IPv6 Route Tracking | Cisco IOS Release 15.3(3)M | This feature expands Enhanced Object Tracking (EOT) functionality to allow the tracking of IPv6 routes. |



CHAPTER 8

IPv6 Static Route Support for Object Tracking

The IPv6 Static Route Support for Object Tracking feature allows an IPv6 static route to be associated with a tracked-object. A static route is only inserted into the routing information base (RIB) when the tracked object is reachable.

This module provides an overview of the feature and explains how to configure it.

- [Finding Feature Information, on page 149](#)
- [Information About IPv6 Static Route Support for Object Tracking, on page 149](#)
- [How to Configure IPv6 Static Route Support for Object Tracking, on page 150](#)
- [Configuration Examples for IPv6 Static Route Support for Object Tracking, on page 152](#)
- [Additional References for IPv6 Static Route Support for Object Tracking, on page 152](#)
- [Feature Information for IPv6 Static Route Support for Object Tracking , on page 153](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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

Information About IPv6 Static Route Support for Object Tracking

IPv6 Static Route Support for Object Tracking Overview

Object tracking allows you to track specific objects on a device, such as the interface line protocol state, IP routing, and route reachability, and to take action when the tracked object's state changes. Tracking allows software clients to register interest in the behavior of an object, and receive notifications of changes. This object represents the state of the system functionality such as the status of an interface (up or down), the existence of an IP prefix in the Routing Information Base (RIB) and so on.

An IPv6 static route creates a tracked object-context for each tracked object. Tracked object contexts are stored in an AVL list that is maintained by the IPv6 static route and indexed by the object number. A

tracked-object context is removed from the AVL list when the object is no longer associated with any IPv6 static routes. All IPv6 static routes associated with a tracked object is linked to the tracked object context by an indirect list. An IPv6 static route becomes a client of the tracked objects, and this allows the IPv6 static route to track the state of a tracked object. The **ipv6 route** command allows an IPv6 static route to be associated with a tracked object.

Routing Table Insertion

An IPv6 static route associated with a tracked-object is inserted into the IPv6 routing table if the state of the tracked-object is up and all other routing-table-insertion criteria are met.

The IPv6 Static Route Object Tracking feature uses the IPv6 static deferred state check mechanism to insert or delete a static route into or from the Routing Information Base (RIB). A change in the state of the tracked object is signaled from tracked objects and this causes IPv6 static to insert all IPv6 static routes associated with the tracked object into the state check queue (unless they are already in it). A separate process removes IPv6 static routes from the state check queue and determines whether these routes should be inserted into the RIB or removed from the RIB using the RIB insertion criteria.

Routing Table Insertion Criteria

The following insertion criteria must be met for an IPv6 static route to be inserted into the IPv6 routing table:

1. Interface is up.
2. Next-hop address is not the device's own address.
3. Next-hop address .
4. Next-hop address is resolved.
5. Bidirectional Forwarding Detection (BFD) session is up, if BFD tracking is configured.



Note

An IPv6 static route can be associated with a tracked object and a BFD session. Both tracked object and BFD session state must be up before the IPv6 static route is inserted in the routing table.

6. Tracked object state is up.

An IPv6 static route in the routing table is removed if any of the insertion criteria becomes false.

How to Configure IPv6 Static Route Support for Object Tracking

Configuring the IPv6 Static Routing Support for Object Tracking

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 route vrf table-name-id ipv6-prefix {interface-type interface-number [next-hop-ipv6-address] | next-hop-ipv6-address} [admin-distance [multicast-vrf-distance]] [multicast] [nexthop-vrf table-name-id] [unicast] [tag tag-value] [track object-number] namestatic-route }**

4. **end**
5. **show track *object-number***
6. **show ipv6 static vrf *id***

DETAILED STEPS

| | Command or Action | Purpose |
|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Step 1 | enable Example: <pre>Device> enable</pre> | Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted. |
| Step 2 | configure terminal Example: <pre>Device# configure terminal</pre> | Enters global configuration mode. |
| Step 3 | ipv6 route vrf <i>table-name-id</i> <i>ipv6-prefix</i> {<i>interface-type</i> <i>interface-number</i> [<i>next-hop-ipv6-address</i>] <i>next-hop-ipv6-address</i>} [<i>admin-distance</i> [<i>multicast-vrf-distance</i>]] [multicast] [nexthop-vrf <i>table-name-id</i>] [unicast] [tag <i>tag-value</i>] [track <i>object-number</i>] [namestatic-route]} Example: <pre>Device(config)# ipv6 route vrf 3 2001:DB8:1:2::/64 GigabitEthernet0/0 2001:DB8:3:4::1 track 42</pre> | Establishes static IPv6 routes for all VPN routing and forwarding (VRF) instance tables or a specific VRF table for IPv4 or IPv6 address. <ul style="list-style-type: none"> • Configure the IPv6 static route object tracking to the static route configuration by using the track <i>object-number</i> command. |
| Step 4 | end Example: <pre>Device(config)# end</pre> | Exits global configuration mode and returns to privileged EXEC mode. |
| Step 5 | show track <i>object-number</i> Example: <pre>Device# show track 42</pre> | Displays information about objects that are tracked by the tracking process. |
| Step 6 | show ipv6 static vrf <i>id</i> Example: <pre>Device(config)# show ipv6 static vrf 3</pre> | Displays static routes that are added to the routing-table, and the reasons if a static route is not added. |

Example

The following is sample output from the **show track** command:

```
Device# show track 42
```

```

Track 42
  IP route 10.21.12.0 255.255.255.0 reachability
  Reachability is Down (no ip route), delayed Up (1 sec remaining) (connected)
    1 change, last change 00:00:24
  Delay up 20 secs, down 10 secs
  First-hop interface is unknown (was GigabitEthernet1/0)
  Tracked by:
    HSRP GigabitEthernet0/0 3

```

Configuration Examples for IPv6 Static Route Support for Object Tracking

Example: IPv6 Static Route Object Tracking

The following example associates the static route 2001:DB8:1:2::/64 with the state of tracked-object number 42. The static route is inserted in the IPv6 routing table if the state of tracked-object number 42 is up.

```

Device> enable
Device# configure terminal
Device(config)# ipv6 route vrf 3 2001:DB8:1:2::/64 GigabitEthernet0/0 2001:DB8:3:4::1 track
42
Device(config)# end

```

Additional References for IPv6 Static Route Support for Object Tracking

Related Documents

| Related Topic | Document Title |
|----------------------------------|---------------------------------------------------------------------|
| Cisco IOS commands | Cisco IOS Master Command List, All Releases |
| IP Application Services commands | Cisco IOS IP Application Services Command Reference |
| Object tracking | Configuring Enhanced Object Tracking |

Technical Assistance

| Description | Link |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------|
| <p>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</p> <p>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.</p> <p>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</p> | http://www.cisco.com/support |

Feature Information for IPv6 Static Route Support for Object Tracking

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

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

Table 11: Feature Information for IPv6 Static Route Support for Object Tracking

| Feature Name | Releases | Feature Information |
|-----------------------------------------------|----------|------------------------------------------------------------------------------------------------------------------------|
| IPv6 Static Route Support for Object Tracking | 15.4(1)T | This feature expands Enhanced Object Tracking (EOT) functionality to allow the object tracking for IPv6 static routes. |



CHAPTER 9

Stream Control Transmission Protocol

Stream Control Transmission Protocol (SCTP) is a reliable datagram-oriented IP transport protocol specified by RFC 2960. It provides the layer between an SCTP user application and an unreliable end-to-end datagram service such as IP. The basic service offered by SCTP is the reliable transfer of user datagrams between peer SCTP users. It performs this service within the context of an association between two SCTP hosts. SCTP is connection-oriented, but SCTP association is a broader concept than the Transmission Control Protocol (TCP) connection, for example.

SCTP provides the means for each SCTP endpoint to provide its peer with a list of transport addresses, such as address and UDP port combinations. This list is provided during association startup and shows the transport addresses through which the endpoint can be reached and from which messages originate. The SCTP association includes transfer over all the possible source and destination combinations that might be generated from the two endpoint lists (also known as multihoming).

SCTP is not explicitly configured on routers, but it underlies several Cisco applications. The commands described in this document are useful for troubleshooting when SCTP issues are suspected as the cause of problems.

- [Finding Feature Information, on page 155](#)
- [Prerequisites for SCTP, on page 156](#)
- [Information About SCTP, on page 156](#)
- [How to Configure SCTP, on page 158](#)
- [Configuration Examples for SCTP, on page 174](#)
- [Additional References, on page 175](#)
- [Feature Information for SCTP, on page 176](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

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

Prerequisites for SCTP

- Cisco IOS Release 12.2(2)MB, Cisco IOS Release 12.2(4)T, or a later release.

Information About SCTP

Benefits of SCTP

SCTP provides the following services and features:

- Acknowledged reliable nonduplicated transfer of user data
- Application-level segmentation to conform to the maximum transmission unit (MTU) size
- Sequenced delivery of user datagrams within multiple streams
- Optional multiplexing of user datagrams into SCTP datagrams
- Enhanced reliability through support of multihoming at either end or both ends of the association
- Congestion avoidance and resistance to flooding and masquerade attacks

SCTP Release 2

SCTP Release 2 introduced updated output for the following SCTP commands:

- **show ip sctp association parameters**
- **show ip sctp association statistics**

SCTP Show Clear CLI Enhancements

The SCTP Show/Clear CLI Enhancements feature provides access to additional SCTP information that can help with troubleshooting potential problems. These enhancements also make the updated SCTP **show** and **clear** commands consistent with the CLI of other transport protocols.

The following commands were replaced by this feature:

clear ip sctp statistics , **show ip sctp association list**, **show ip sctp association parameters**, **show ip sctp association statistics**, **show ip sctp errors**, **show ip sctp instances**, **show ip sctp statistics**

The following commands were introduced by this feature:

clear sctp statistics , **show sctp association**, **show sctp association list**, **show sctp association parameters**, **show sctp association statistics**, **show sctp errors**, **show sctp instance**, **show sctp instances**, **show sctp instances**, **show sctp statistics**

SCTP Stream Reset

SCTP Release 4 introduced the SCTP stream reset feature. The SCTP stream reset feature enables SCTP to reset stream transport sequence numbers and all stream sequence numbers. Applications that use SCTP may reset a stream and restart the numbering sequence of the stream at zero. A notification to the upper layer that the stream numbering sequence has been reset is sent by the application. Resetting an SCTP stream enables applications to reuse streams for different purposes while preserving the stream sequence number for the application so that message flows can be tracked. Without the SCTP stream reset feature, reusing streams results in message numbers larger than expected. SCTP stream reset enables SCTP to:

- Dynamically reset a peer's outbound streams
- Dynamically reset a local host's outbound stream
- Dynamically reset specific numbered streams

The resetting of a stream occurs upon request from an upper layer application. This capability is enabled by default in Cisco IOS Release 12.4(15)T and later releases.

SCTP Authentication

SCTP Release 4 introduced the SCTP Authentication feature. The SCTP Authentication feature enables SCTP to:

- Set up a dynamic shared association key with no shared secret
- Allow a shared secret to be combined with an association key
- Use the shared association secret to authenticate chunks
- Negotiate which chunk types must be authenticated

The SCTP Authentication feature enables applications to use these optional extensions. Use the **ip sctp authenticate** command to define chunks that the client requires be authenticated.

SCTP Authentication procedures use either Message Digest 5 (MD5) or Secure Hash Algorithm 1 (SHA-1), which can be memory and CPU intensive. Enabling SCTP Authentication on DATA chunks could impact CPU utilization when a large number of authenticated chunks are sent.

You can define the types of SCTP chunks the client requires be authenticated. Optionally, you can use the **ip sctp asconf** command to configure SCTP to send an Address Configuration Change (ASCONF) chunk automatically in response to an IP address change in an SCTP stream, or to check that the endpoint supports authentication before sending the ASCONF chunk.

The table below lists the SCTP chunk types and numbers that can be authenticated by entering the **ip sctp authenticate** command in global configuration mode.

Table 12: SCTP Authentication Chunk Types

| SCTP Chunk Type | SCTP Chunk Number | Description |
|-------------------|-------------------|---------------------------------------------|
| abort association | 0x06 | ABORT chunk. |
| asconf | 0xc1 | ASCONF. Address configuration change chunk. |
| asconf-ack | 0x80 | ASCONF-ACK. ASCONF acknowledgement. |

| SCTP Chunk Type | SCTP Chunk Number | Description |
|-----------------|-------------------|-----------------------------------------------------------------------------|
| cookie-ack | 0x0b | COOKIE acknowledgment chunk. |
| cookie-echo | 0x0a | COOKIE-ECHO chunk. |
| data | 0x00 | DATA chunk. |
| fwd-tsn | 0xc0 | FWD-CUM-TSN chunk. Forwarded cumulative transmission sequence number chunk. |
| heartbeat | 0x04 | HEARTBEAT request chunk. |
| heartbeat-ack | 0x05 | HEARTBEAT acknowledgement chunk. |
| packet-drop | 0x81 | PACKET-DROP chunk. |
| sack | 0x03 | Selective acknowledgment chunk. |
| shutdown | 0x07 | SHUTDOWN chunk. |
| shutdown-ack | 0x08 | SHUTDOWN acknowledgment chunk. |
| stream-reset | 0x82 | STREAM-RESET chunk. |

SCTP Add-IP

SCTP Release 4 introduced the SCTP Add-IP feature. The SCTP Add-IP feature enables the ability to add or delete an IP address for an endpoint of an existing SCTP association and to communicate this change to the remote end. An ADD-IP chunk is sent to the remote end adding or removing the redundant server addresses of the association. The ADD-IP chunk also deletes all addresses of a failed host from an association.

In Cisco IOS software, adding or deleting an IP address from an SCTP association can be triggered programmatically by the application, or automatically in response to an IP address change on the router.

The SCTP Add-IP feature also enables an application to programmatically set the primary address for an SCTP association.

The SCTP Add-IP feature introduces two new SCTP chunk types, the Address Configuration (ASCONF) chunk, and the Address Acknowledgement (ASCONF-ACK) chunk. The ASCONF chunk is used by the sender in an SCTP stream to communicate to the remote endpoint that the SCTP stream contains a configuration change request that must be acknowledged. The receiving endpoint of an ASCONF chunk uses the ASCONF-ACK chunk to acknowledge receipt of the ASCONF chunk.

The **ip asconf auto** command is used to configure an application to automatically send an ASCONF chunk in response to an IP Address change on the endpoint and to configure SCTP to either accept the ASCONF chunk automatically, or to require the chunk be authenticated.

How to Configure SCTP

SCTP is not explicitly configured on routers, but it underlies several Cisco applications. This section contains the following tasks:

Configuring SCTP Authentication Parameters

SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `ip sctp authenticate {chunk-type | chunk-number}`
4. `ip sctp asconf {authenticate check | auto}`

DETAILED STEPS

| | Command or Action | Purpose |
|--------|---------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Step 1 | enable Example: Router> enable | Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted. |
| Step 2 | configure terminal Example: Router# configure terminal | Enters global configuration mode. |
| Step 3 | ip sctp authenticate {<i>chunk-type</i> <i>chunk-number</i>} Example: Router(config)# ip sctp authenticate sack | Defines Stream Control Transmission Protocol (SCTP) data chunks that the client requires be authenticated. |
| Step 4 | ip sctp asconf {authenticate check auto} Example: Router(config)# ip sctp asconf auto | (Optional) Configures SCTP to send ASCONF chunks automatically in response to an IP address change on a router, or to check that the endpoint supports authentication before sending the ASCONF chunk. |

Displaying Information About SCTP Associations and Parameters

To display information about SCTP associations and instances, use the following commands.



Note SCTP commands that display statistical information show only the information that is available since the last time a **clear sctp statistics** command was executed. The **clear sctp statistics** command clears all SCTP statistics, both those compiled for individual associations and those compiled overall.

SUMMARY STEPS

1. `show sctp association list`
2. `show sctp association parameters`
3. `show sctp association statistics`

4. **show sctp errors**
5. **show sctp instance**
6. **show sctp instances**
7. **show sctp statistics**

DETAILED STEPS

Step 1 show sctp association list

The **show sctp association list** command provides the current SCTP association and instance identifiers, the current state of SCTP associations, and the local and remote port numbers and addresses that are used in the associations. The following example shows three current associations that are in the established state. Each association belongs to the same instance, as noted by their instance identifiers.

Example:

```
Router# show sctp association list

*** SCTP Association List ***
AssocID:0, Instance ID:0
Current state:ESTABLISHED
Local port:8989, Adrs:10.1.0.2 10.2.0.2
Remote port:8989, Adrs:10.6.0.4 10.5.0.4
AssocID:1, Instance ID:0
Current state:ESTABLISHED
Local port:8989, Adrs:10.1.0.2 10.2.0.2
Remote port:8990, Adrs:10.6.0.4 10.5.0.4
AssocID:2, Instance ID:0
Current state:ESTABLISHED
Local port:8989, Adrs:10.1.0.2 10.2.0.2
Remote port:8991, Adrs:10.6.0.4 10.5.0.4
```

Step 2 show sctp association parameters

The **show sctp association parameters** command provides information to determine the stability of SCTP associations, dynamically calculated statistics about destinations, and values to assess network congestion. This command also displays parameter values for the specified association.

The association configuration section displays information similar to that in the **show sctp association list** command, including association identifiers, state, and local and remote port and address information. The current primary destination is also displayed.

The following sample output shows the IP SCTP association parameters for association 0:

Example:

```
Router# show sctp association parameters 0

** SCTP Association Parameters **
AssocID: 0 Context: 0 InstanceID: 1
Assoc state: ESTABLISHED Uptime: 19:05:57.425
Local port: 8181
Local addresses: 10.1.0.3 10.2.0.3
Remote port: 8181
Primary dest addr: 10.5.0.4
Effective primary dest addr: 10.5.0.4
Destination addresses:
10.5.0.4: State: ACTIVE
```

```

Heartbeats: Enabled Timeout: 30000 ms
RTO/RTT/SRTT: 1000/16/38 ms TOS: 0 MTU: 1500
cwnd: 5364 ssthresh: 3000 outstand: 768
Num retrans: 0 Max retrans: 5 Num times failed: 0
10.6.0.4: State: ACTIVE
Heartbeats: Enabled Timeout: 30000 ms
RTO/RTT/SRTT: 1000/4/7 ms TOS: 0 MTU: 1500
cwnd: 3960 ssthresh: 3000 outstand: 0
Num retrans: 0 Max retrans: 5 Num times failed: 0
Local vertag: 9A245CD4 Remote vertag: 2A08D122
Num inbound streams: 10 outbound streams: 10
Max assoc retrans: 5 Max init retrans: 8
CumSack timeout: 200 ms Bundle timeout: 100 ms
Min RTO: 1000 ms Max RTO: 60000 ms
LocalRwnd: 18000 Low: 13455 RemoteRwnd: 15252 Low: 13161
Congest levels: 0 current level: 0 high mark: 325

```

Step 3 show sctp association statistics

This command shows only the information that has become available since the last time a **clear sctp statistics** command was executed for an SCTP association.

The following sample output shows the statistics accumulated for SCTP association 0:

Example:

```

Router# show sctp association statistics 0

** SCTP Association Statistics **
AssocID/InstanceID: 0/1
Current State: ESTABLISHED
Control Chunks
  Sent: 623874 Rcvd: 660227
Data Chunks Sent
  Total: 14235644 Retransmitted: 60487
  Ordered: 6369678 Unordered: 6371263
  Avg bundled: 18 Total Bytes: 640603980
Data Chunks Rcvd
  Total: 14496585 Discarded: 1755575
  Ordered: 6369741 Unordered: 6371269
  Avg bundled: 18 Total Bytes: 652346325
  Out of Seq TSN: 3069353
ULP Dgrams
  Sent: 12740941 Ready: 12740961 Rcvd: 12740941

```

Step 4 show sctp errors

The **show sctp errors** command displays all errors across all associations that have been logged since the last time that the SCTP statistics were cleared with the **clear sctp statistics** command. If no errors have been logged, this is indicated in the output. The following sample output shows a session that has SCTP errors:

Example:

```

Router# show sctp errors

** SCTP Error Statistics **
Invalid verification tag:      5
Communication Lost:           64
Destination Address Failed:    3
Unknown INIT params rcvd:     16
Invalid cookie signature:      5
Expired cookie:                1

```

```
Peer restarted:          1
No Listening instance:    2
```

Step 5 **show sctp instance**

The **show sctp instance** command displays information for the currently configured instance with the ID specified in the command syntax. The instance number, local port, and address information are displayed. The instance state is either *available* or *deletion pending*. An instance enters the deletion pending state when a request is made to delete it but there are currently established associations for that instance. The instance cannot be deleted immediately and instead enters the pending state. No new associations are allowed in this instance, and when the last association is terminated or fails, the instance is deleted.

The default inbound and outbound stream numbers (shown in the example output in the next section) are used for establishing incoming associations, the maximum number of associations allowed for this instance is shown, and a snapshot of each existing association is shown, if any exists.

The following sample output displays information for SCTP instance 0. In this example, instance 0 is using local port 1000 and has three current associations.

Example:

```
Router# show sctp instance 0

Instance ID:0 Local port:1000 State:available
Local addr:10.1.0.2 10.2.0.2
Default streams inbound:1  outbound:1
Current associations: (max allowed:200)
  AssocID:0 State:ESTABLISHED Remote port:8989
    Dest addr:10.6.0.4 10.5.0.4
  AssocID:1 State:ESTABLISHED Remote port:8990
    Dest addr:10.6.0.4 10.5.0.4
  AssocID:2 State:ESTABLISHED Remote port:8991
    Dest addr:10.6.0.4 10.5.0.4
```

Step 6 **show sctp instances**

The **show sctp instances** command displays information for each of the currently configured instances. The instance number, local port, and address information are displayed. The instance state is either *available* or *deletion pending*. An instance enters the deletion pending state when a request is made to delete it but there are currently established associations for that instance. The instance cannot be deleted immediately and instead enters the pending state. No new associations are allowed in this instance, and when the last association is terminated or fails, the instance is deleted.

The default inbound and outbound stream numbers are used for establishing incoming associations, the maximum number of associations allowed for this instance is shown, and a snapshot of each existing association is shown, if any exists.

Note When you enter the **show sctp instances** command, you must type the complete word **instances** in the command syntax. If you try to enter an abbreviated form of this word, there will be a partial match that identifies the **show sctp instance** *instance-id* command.

The following sample output shows available IP SCTP instances. In this example, two current instances are active and available. The first is using local port 8989, and the second is using 9191. Instance identifier 0 has three current associations, and instance identifier 1 has no current associations.

Example:

```
Router# show sctp instances

*** Sctp Instances ***
Instance ID:0 Local port:8989
```

```
Instance state:available
Local addr:10.1.0.2 10.2.0.2
Default streams inbound:1  outbound:1
Current associations: (max allowed:6)
  AssocID:0  State:ESTABLISHED  Remote port:8989
    Dest addr:10.6.0.4 10.5.0.4
  AssocID:1  State:ESTABLISHED  Remote port:8990
    Dest addr:10.6.0.4 10.5.0.4
  AssocID:2  State:ESTABLISHED  Remote port:8991
    Dest addr:10.6.0.4 10.5.0.4
Instance ID:1  Local port:9191
Instance state:available
Local addr:10.1.0.2 10.2.0.2
Default streams inbound:1  outbound:1
No current associations established for this instance.
Max allowed:6
```

Step 7 **show sctp statistics**

The **show sctp statistics** command displays the overall SCTP statistics accumulated since the last **clear sctp statistics** command. It includes numbers for all currently established associations, and for any that have been terminated. The statistics indicated are similar to those shown for individual associations. The following sample output shows SCTP statistics:

Example:

```
Router# show sctp statistics

*** SCTP Overall Statistics ***
Total Chunks Sent:          2097
Total Chunks Rcvd:         2766
Data Chunks Rcvd In Seq:   538
Data Chunks Rcvd Out of Seq: 0
Total Data Chunks Sent:    538
Total Data Chunks Rcvd:    538
Total Data Bytes Sent:     53800
Total Data Bytes Rcvd:     53800
Total Data Chunks Discarded: 0
Total Data Chunks Retrans: 0
Total SCTP Dgrams Sent:    1561
Total SCTP Dgrams Rcvd:    2228
Total ULP Dgrams Sent:     538
Total ULP Dgrams Ready:    538
Total ULP Dgrams Rcvd:     538
```

Troubleshooting SCTP Associations and Parameters

This section describes the debug commands available for troubleshooting SCTP associations and parameters.

In a live system, the debug commands for performance, state, signal, and warnings are the most useful. These debug commands show any association or destination address failures and can be used to monitor the stability of any established associations.

Debug commands other than those for performance, state, signal, and warnings can generate a great deal of output and therefore can cause associations to fail. These commands should be used only in test environments or when there are very low amounts of traffic.



Caution Many SCTP debug commands should be used with extreme caution or not at all in live systems, depending on the amount of traffic, because the extra messages they generate may cause associations to fail. This caution is repeated in descriptions of the commands that may cause disruption to live systems.



Note SCTP debug commands display information for all current SCTP associations and cannot be limited to particular associations.

SUMMARY STEPS

1. `debug ip sctp api`
2. `debug ip sctp congestion`
3. `debug ip sctp init`
4. `debug ip sctp multihome`
5. `debug ip sctp performance`
6. `debug ip sctp rcvchunks`
7. `debug ip sctp rto`
8. `debug ip sctp segments`
9. `debug ip sctp segmentv`
10. `debug ip sctp signal`
11. `debug ip sctp state`
12. `debug ip sctp sndchunks`
13. `debug ip sctp timer`
14. `debug ip sctp warnings`

DETAILED STEPS

Step 1 `debug ip sctp api`

The `debug ip sctp api` command shows all SCTP calls to the application programming interface (API) that are being executed and the parameters associated with these calls.

Caution The `debug ip sctp api` command should not be used in a live system that has any significant amount of traffic running because it can generate a lot of traffic, which can cause associations to fail.

The following is sample output for this command:

Example:

```
Router# debug ip sctp api

*Mar 1 00:31:14.211: SCTP: sctp_send: Assoc ID: 1
*Mar 1 00:31:14.211: SCTP:                stream num: 10
*Mar 1 00:31:14.211: SCTP:                bptr: 62EE332C, dptr: 4F7B598
*Mar 1 00:31:14.211: SCTP:                datalen: 100
*Mar 1 00:31:14.211: SCTP:                context: 1
*Mar 1 00:31:14.211: SCTP:                lifetime: 0
*Mar 1 00:31:14.211: SCTP:                unorder flag: FALSE
```



```
*Mar 1 00:31:14.211: SCTP:          bundle flag: TRUE
*Mar 1 00:31:14.211: SCTP: sctp_send successful return
*Mar 1 00:31:14.211: SCTP: sctp_receive: Assoc ID: 1
*Mar 1 00:31:14.215: SCTP:          max data len: 100
.
.
.
```

Step 2 debug ip sctp congestion

The **debug ip sctp congestion** command displays various events related to calculating the current congestion parameters, including congestion window (cwnd) values per destination address and local and remote receiver window (rwnd) parameters. Information is displayed when bundling and sending data chunks, indicating the current cwnd and rwnd values and remote rwnd values, thus showing when data can or cannot be sent or bundled. When chunks are acknowledged by the remote peer, the number of bytes outstanding and remote rwnd values are updated.

Information is also displayed when new chunks are received, thus decreasing the local rwnd space, and when chunks are freed because the upper-layer protocol (ULP) is receiving datagrams from SCTP and thus freeing local rwnd space. The following is sample output for this command:

Example:

```
Router# debug ip sctp congestion

SCTP: Assoc 0: Slow start 10.6.0.4, cwnd 3000
SCTP: Assoc 0: Data chunks rcvd, local rwnd 7800
SCTP: Assoc 0: Free chunks, local rwnd 9000
SCTP: Assoc 0: Data chunks rcvd, local rwnd 8200
SCTP: Assoc 0: Add Sack, local a_rwnd 8200
SCTP: Assoc 0: Free chunks, local rwnd 9000
SCTP: Assoc 0: Data chunks rcvd, local rwnd 7800
SCTP: Assoc 0: Data chunks rcvd, local rwnd 7000
SCTP: Assoc 0: Add Sack, local a_rwnd 7000
SCTP: Assoc 0: Free chunks, local rwnd 9000
SCTP: Assoc 0: Bundle for 10.5.0.4, rem rwnd 14000, cwnd 19500, outstand 0
SCTP: Assoc 0: Bundled 12 chunks, remote rwnd 12800, outstand 1200
SCTP: Assoc 0: Bundling data, next chunk dataLen (100) > remaining mtu size
SCTP: Assoc 0: Bundle for 10.5.0.4, rem rwnd 12800, cwnd 19500, outstand 1200
SCTP: Assoc 0: Bundled 12 chunks, remote rwnd 11600, outstand 2400
SCTP: Assoc 0: Bundling data, next chunk dataLen (100) > remaining mtu size
SCTP: Assoc 0: Bundle for 10.5.0.4, rem rwnd 11600, cwnd 19500, outstand 2400
SCTP: Assoc 0: Bundled 12 chunks, remote rwnd 10400, outstand 3600
SCTP: Assoc 0: Bundling data, next chunk dataLen (100) > remaining mtu size
SCTP: Assoc 0: Bundle for 10.5.0.4, rem rwnd 10400, cwnd 19500, outstand 3600
SCTP: Assoc 0: Bundled 4 chunks, remote rwnd 10000, outstand 4000
SCTP: Assoc 0: No additional chunks waiting.
SCTP: Assoc 0: Data chunks rcvd, local rwnd 7800
SCTP: Assoc 0: Data chunks rcvd, local rwnd 7000
SCTP: Assoc 0: Add Sack, local a_rwnd 7000
SCTP: Assoc 0: Chunk A22F3B45 ack'd, dest 10.5.0.4, outstanding 3900
SCTP: Assoc 0: Chunk A22F3B46 ack'd, dest 10.5.0.4, outstanding 3800
SCTP: Assoc 0: Chunk A22F3B47 ack'd, dest 10.5.0.4, outstanding 3700
SCTP: Assoc 0: Chunk A22F3B48 ack'd, dest 10.5.0.4, outstanding 3600
SCTP: Assoc 0: Chunk A22F3B49 ack'd, dest 10.5.0.4, outstanding 3500
SCTP: Assoc 0: Chunk A22F3B4A ack'd, dest 10.5.0.4, outstanding 3400
SCTP: Assoc 0: Chunk A22F3B4B ack'd, dest 10.5.0.4, outstanding 3300
SCTP: Assoc 0: Chunk A22F3B4C ack'd, dest 10.5.0.4, outstanding 3200
SCTP: Assoc 0: Chunk A22F3B4D ack'd, dest 10.5.0.4, outstanding 3100
SCTP: Assoc 0: Chunk A22F3B4E ack'd, dest 10.5.0.4, outstanding 3000
SCTP: Assoc 0: Chunk A22F3B4F ack'd, dest 10.5.0.4, outstanding 2900
SCTP: Assoc 0: Chunk A22F3B50 ack'd, dest 10.5.0.4, outstanding 2800
SCTP: Assoc 0: Chunk A22F3B51 ack'd, dest 10.5.0.4, outstanding 2700
SCTP: Assoc 0: Chunk A22F3B52 ack'd, dest 10.5.0.4, outstanding 2600
```

```
SCTP: Assoc 0: Chunk A22F3B53 ack'd, dest 10.5.0.4, outstanding 2500
SCTP: Assoc 0: Chunk A22F3B54 ack'd, dest 10.5.0.4, outstanding 2400
SCTP: Assoc 0: Chunk A22F3B55 ack'd, dest 10.5.0.4, outstanding 2300
SCTP: Assoc 0: Chunk A22F3B56 ack'd, dest 10.5.0.4, outstanding 2200
```

Step 3 debug ip sctp init

The **debug ip sctp init** command shows datagrams and other information related to the initializing of new associations. All initialization chunks are shown, including the INIT, INIT_ACK, COOKIE_ECHO, and COOKIE_ACK chunks. This debug command can be used to see the chunks associated with any initialization sequence, but does not display data chunks sent once the association is established. Therefore, it is safe to use in a live system that has traffic flowing when you have trouble with associations that fail and have to be reestablished.

Example:

```
Router# debug ip sctp init

*Mar 1 00:53:07.279: SCTP Test: Attempting to open assoc to remote port 8787...assoc ID is 0
*Mar 1 00:53:07.279: SCTP: Process Assoc Request
*Mar 1 00:53:07.279: SCTP: Assoc 0: dest addr list:
*Mar 1 00:53:07.279: SCTP:                addr 10.5.0.4
*Mar 1 00:53:07.279: SCTP:                addr 10.6.0.4
*Mar 1 00:53:07.279:
...
*Mar 1 00:53:13.279: SCTP: Assoc 0: Send Init
*Mar 1 00:53:13.279: SCTP:        INIT_CHUNK, len 42
*Mar 1 00:53:13.279: SCTP:        Initiate Tag: B4A10C4D, Initial TSN: B4A10C4D, rwnd 9000
*Mar 1 00:53:13.279: SCTP:        Streams Inbound: 13, Outbound: 13
*Mar 1 00:53:13.279: SCTP:        IP Addr: 10.1.0.2
*Mar 1 00:53:13.279: SCTP:        IP Addr: 10.2.0.2
*Mar 1 00:53:13.279: SCTP:        Supported addr types: 5
*Mar 1 00:53:13.307: SCTP: Process Init
*Mar 1 00:53:13.307: SCTP:        INIT_CHUNK, len 42
*Mar 1 00:53:13.307: SCTP:        Initiate Tag: 3C2D8327, Initial TSN: 3C2D8327, rwnd 18000
*Mar 1 00:53:13.307: SCTP:        Streams Inbound: 13, Outbound: 13
*Mar 1 00:53:13.307: SCTP:        IP Addr: 10.5.0.4
*Mar 1 00:53:13.307: SCTP:        IP Addr: 10.6.0.4
*Mar 1 00:53:13.307: SCTP:        Supported addr types: 5
*Mar 1 00:53:13.307: SCTP: Assoc 0: Send InitAck
*Mar 1 00:53:13.307: SCTP:        INIT_ACK_CHUNK, len 124
*Mar 1 00:53:13.307: SCTP:        Initiate Tag: B4A10C4D, Initial TSN: B4A10C4D, rwnd 9000
*Mar 1 00:53:13.307: SCTP:        Streams Inbound: 13, Outbound: 13
*Mar 1 00:53:13.307: SCTP:        Responder cookie len 88
*Mar 1 00:53:13.307: SCTP:        IP Addr: 10.1.0.2
*Mar 1 00:53:13.307: SCTP:        IP Addr: 10.2.0.2
*Mar 1 00:53:13.311: SCTP: Assoc 0: Process Cookie
*Mar 1 00:53:13.311: SCTP:        COOKIE_ECHO_CHUNK, len 88
*Mar 1 00:53:13.311: SCTP: Assoc 0: dest addr list:
*Mar 1 00:53:13.311: SCTP:                addr 10.5.0.4
*Mar 1 00:53:13.311: SCTP:                addr 10.6.0.4
*Mar 1 00:53:13.311:
*Mar 1 00:53:13.311: SCTP: Instance 0 dest addr list:
*Mar 1 00:53:13.311: SCTP:                addr 10.5.0.4
*Mar 1 00:53:13.311: SCTP:                addr 10.6.0.4
*Mar 1 00:53:13.311:
*Mar 1 00:53:13.311: SCTP: Assoc 0: Send CookieAck
*Mar 1 00:53:13.311: SCTP:        COOKIE_ACK_CHUNK
```

Step 4 debug ip sctp multihome

The **debug ip sctp multihome** command shows the source and destination of datagrams in order to monitor the use of the multihome addresses. More than one IP address parameter can be included in an INIT chunk when the INIT sender

is multihomed. Datagrams should mostly be sent to the primary destination addresses unless the network is experiencing problems, in which case the datagrams can be sent to the secondary addresses.

Caution The **debug ip sctp multihome** command generates one debug line for each datagram sent or received. It should be used with extreme caution in a live network.

The following is sample output for this command:

Example:

```
Router# debug ip sctp multihome

SCTP: Rcvd s=10.5.0.4 8787, d=10.1.0.2 8787, len 1404
SCTP: Rcvd s=10.5.0.4 8787, d=10.1.0.2 8787, len 476
SCTP: Sent: Assoc 0: s=10.1.0.2 8787, d=10.5.0.4 8787, len 28
SCTP: Assoc 0: Send Data to dest 10.5.0.4
SCTP: Sent: Assoc 0: s=10.1.0.2 8787, d=10.5.0.4 8787, len 1404
SCTP: Sent: Assoc 0: s=10.1.0.2 8787, d=10.5.0.4 8787, len 1404
SCTP: Sent: Assoc 0: s=10.1.0.2 8787, d=10.5.0.4 8787, len 1404
SCTP: Sent: Assoc 0: s=10.1.0.2 8787, d=10.5.0.4 8787, len 476
SCTP: Rcvd s=10.5.0.4 8787, d=10.1.0.2 8787, len 28
SCTP: Rcvd s=10.5.0.4 8787, d=10.1.0.2 8787, len 28
SCTP: Rcvd s=10.5.0.4 8787, d=10.1.0.2 8787, len 1404
SCTP: Rcvd s=10.5.0.4 8787, d=10.1.0.2 8787, len 1404
SCTP: Sent: Assoc 0: s=10.1.0.2 8787, d=10.5.0.4 8787, len 28
SCTP: Rcvd s=10.5.0.4 8787, d=10.1.0.2 8787, len 1404
SCTP: Rcvd s=10.5.0.4 8787, d=10.1.0.2 8787, len 476
SCTP: Sent: Assoc 0: s=10.1.0.2 8787, d=10.5.0.4 8787, len 28
SCTP: Assoc 0: Send Data to dest 10.5.0.4
SCTP: Sent: Assoc 0: s=10.1.0.2 8787, d=10.5.0.4 8787, len 1404
SCTP: Sent: Assoc 0: s=10.1.0.2 8787, d=10.5.0.4 8787, len 1404
SCTP: Sent: Assoc 0: s=10.1.0.2 8787, d=10.5.0.4 8787, len 1404
SCTP: Sent: Assoc 0: s=10.1.0.2 8787, d=10.5.0.4 8787, len 476
SCTP: Rcvd s=10.6.0.4 8787, d=10.2.0.2 8787, len 44
SCTP: Sent: Assoc 0: s=10.2.0.2 8787, d=10.6.0.4 8787, len 44
SCTP: Rcvd s=10.5.0.4 8787, d=10.1.0.2 8787, len 28
SCTP: Rcvd s=10.5.0.4 8787, d=10.1.0.2 8787, len 28
SCTP: Rcvd s=10.5.0.4 8787, d=10.1.0.2 8787, len 1404
SCTP: Rcvd s=10.5.0.4 8787, d=10.1.0.2 8787, len 1404
SCTP: Sent: Assoc 0: s=10.1.0.2 8787, d=10.5.0.4 8787, len 28
SCTP: Rcvd s=10.5.0.4 8787, d=10.1.0.2 8787, len 1404
SCTP: Rcvd s=10.5.0.4 8787, d=10.1.0.2 8787, len 476
```

Step 5 **debug ip sctp performance**

The **debug ip sctp performance** command reveals the average number of chunks and datagrams being sent and received per second. Once enabled, the **debug ip sctp performance** command displays this information once every 10 seconds. Note that the averages are cumulative since the last time the statistics were cleared and so may not accurately reflect the number of datagrams and chunks currently being sent and received.

In the following example, when the performance debug was first enabled, it showed a very low rate of traffic. However, it was expected that these numbers were not accurate, so a **clear ip sctp** command was executed. The average numbers adjusted quickly to reflect the accurate amount of flowing traffic.

Example:

```
Router# debug ip sctp performance

SCTP Sent: SCTP Dgrams 5, Chunks 28, Data Chunks 29, ULP Dgrams 29
SCTP Rcvd: SCTP Dgrams 7, Chunks 28, Data Chunks 29, ULP Dgrams 29
Chunks Discarded: 0, Retransmitted 0
```

```

SCTP Sent: SCTP Dgrams 6, Chunks 29, Data Chunks 30, ULP Dgrams 30
SCTP Rcvd: SCTP Dgrams 7, Chunks 29, Data Chunks 30, ULP Dgrams 30
Chunks Discarded: 0, Retransmitted 0
SCTP Sent: SCTP Dgrams 6, Chunks 29, Data Chunks 31, ULP Dgrams 31
SCTP Rcvd: SCTP Dgrams 7, Chunks 30, Data Chunks 31, ULP Dgrams 31
Chunks Discarded: 0, Retransmitted 0
SCTP Sent: SCTP Dgrams 6, Chunks 30, Data Chunks 31, ULP Dgrams 31
SCTP Rcvd: SCTP Dgrams 7, Chunks 31, Data Chunks 32, ULP Dgrams 31
Chunks Discarded: 0, Retransmitted 0
SCTP Sent: SCTP Dgrams 6, Chunks 31, Data Chunks 32, ULP Dgrams 32
SCTP Rcvd: SCTP Dgrams 7, Chunks 32, Data Chunks 32, ULP Dgrams 32
Chunks Discarded: 0, Retransmitted 0

```

```
Router# clear ip sctp statistics
```

```

SCTP Sent: SCTP Dgrams 30, Chunks 210, Data Chunks 199, ULP Dgrams 201
SCTP Rcvd: SCTP Dgrams 30, Chunks 208, Data Chunks 198, ULP Dgrams 198
Chunks Discarded: 0, Retransmitted 0
SCTP Sent: SCTP Dgrams 30, Chunks 210, Data Chunks 199, ULP Dgrams 200
SCTP Rcvd: SCTP Dgrams 30, Chunks 209, Data Chunks 199, ULP Dgrams 199
Chunks Discarded: 0, Retransmitted 0
SCTP Sent: SCTP Dgrams 30, Chunks 211, Data Chunks 200, ULP Dgrams 199
SCTP Rcvd: SCTP Dgrams 30, Chunks 209, Data Chunks 198, ULP Dgrams 198
Chunks Discarded: 0, Retransmitted 0

```

Step 6 debug ip sctp rcvchunks

The **debug ip sctp rcvchunks** command displays information about chunks that are received. It shows the stream number, sequence number, chunk length, and chunk transmission sequence number (TSN) for each chunk received, and whether the chunk is for a new datagram or is part of a datagram that is already being reassembled. The command output shows whether the datagram is complete after receiving this chunk or not and, if it is complete, whether it is in sequence within the specified stream and can be delivered to the ULP. It shows the SACKs that are sent back to the remote, indicating the cumulative TSN acknowledged, the number of fragments included, and that the datagram is received by the ULP.

Caution The **debug ip sctp rcvchunks** command generates multiple debug lines for each chunk received. It should be used with extreme caution in a live network.

In the following example, a segmented datagram is received in two chunks, for stream 0 and sequence number 0. The length of the first chunk is 1452, and the second is 1 byte. The first chunk indicates that it is for a new datagram, but the second chunk indicates that it is part of an existing datagram that is already being reassembled. When the first chunk is processed, it is noted to be in sequence, but is not complete and so cannot be delivered yet. When the second chunk is received, the datagram is both in sequence and complete. The application receives the datagram, and a SACK is shown to acknowledge that both chunks were received with no missing chunks indicated (that is, with no fragments).

Example:

```

Router# debug ip sctp rcvchunks

SCTP: Assoc 0: New chunk (0/0/1452/2C33D822) for new dgram (0)
SCTP: Assoc 0: dgram (0) is in seq
SCTP: Assoc 0: Add Sack Chunk, CumTSN=2C33D822, numFrgs=0
SCTP: Assoc 0: New chunk (0/0/1/2C33D823) for existing dgram (0)
SCTP: Assoc 0: dgram (0) is complete
SCTP: Assoc 0: ApplRecv chunk 0/0/1452/2C33D822
SCTP: Assoc 0: ApplRecv chunk 0/0/1/2C33D823
SCTP: Assoc 0: Add Sack Chunk, CumTSN=2C33D823, numFrgs=0

```

Step 7 debug ip sctp rto

The **debug ip sctp rto** command shows any adjustments that are made to the retransmission (retrans) timeout value due either to retransmission of data chunks or to unacknowledged heartbeats.

Caution The **debug ip sctp rto** command can generate a great deal of output. It should be used with extreme caution in a live network.

In the following example, there is only one destination address available. Each time the chunk needs to be retransmitted, the retransmission timeout (RTO) value is doubled.

Example:

```
Router# debug ip sctp rto

SCTP: Assoc 0: destaddr 10.5.0.4, retrans timeout on chunk 942BAC55
SCTP: Assoc 0: destaddr 10.5.0.4, rto backoff 2000 ms
SCTP: Assoc 0: destaddr 10.5.0.4, retrans timeout on chunk 942BAC55
SCTP: Assoc 0: destaddr 10.5.0.4, rto backoff 4000 ms
SCTP: Assoc 0: destaddr 10.5.0.4, retrans timeout on chunk 942BAC55
SCTP: Assoc 0: destaddr 10.5.0.4, rto backoff 8000 ms
SCTP: Assoc 0: destaddr 10.5.0.4, retrans timeout on chunk 942BAC55
SCTP: Assoc 0: destaddr 10.5.0.4, rto backoff 16000 ms
SCTP: Assoc 0: destaddr 10.5.0.4, retrans timeout on chunk 942BAC55
SCTP: Assoc 0: destaddr 10.5.0.4, rto backoff 32000 ms
```

Step 8 **debug ip sctp segments**

The **debug ip sctp segments** output shows every datagram that is sent or received and the chunks that are contained in each. The segment debug command has two forms: simple and verbose. This is the simple form of the segment output, and it shows basic information for each chunk type. See the **debug ip sctp segmentv** command for the verbose form of this output.

Caution The **debug ip sctp segments** command generates several lines of output for each datagram sent or received. It should be used with extreme caution in a live network.

The following output shows an example in which an association is established, a few heartbeats are sent, the remote endpoint fails, and the association is restarted.

Example:

```
Router# debug ip sctp segments

SCTP: Sent: Assoc 0: s=10.1.0.2 8787, d=10.5.0.4 8787, len 56
SCTP: INIT_CHUNK, Tag: 3C72A02A, TSN: 3C72A02A
SCTP: Recv: Assoc 0: s=10.5.0.4 8787, d=10.1.0.2 8787, len 56
SCTP: INIT_CHUNK, Tag: 13E5AD6C, TSN: 13E5AD6C
SCTP: Sent: Assoc NULL: s=10.1.0.2 8787, d=10.5.0.4 8787, len 136
SCTP: INIT_ACK_CHUNK, Tag: 3C72A02A, TSN: 3C72A02A
SCTP: Recv: Assoc 0: s=10.5.0.4 8787, d=10.1.0.2 8787, len 100
SCTP: COOKIE_ECHO_CHUNK, len 88
SCTP: Sent: Assoc NULL: s=10.1.0.2 8787, d=10.5.0.4 8787, len 16
SCTP: COOKIE_ACK_CHUNK
SCTP: Sent: Assoc 0: s=10.1.0.2 8787, d=10.5.0.4 8787, len 52
SCTP: HEARTBEAT_CHUNK
SCTP: Sent: Assoc 0: s=10.1.0.2 8787, d=10.5.0.4 8787, len 52
SCTP: HEARTBEAT_CHUNK
SCTP: Sent: Assoc 0: s=10.1.0.2 8787, d=10.5.0.4 8787, len 52
SCTP: HEARTBEAT_CHUNK
SCTP: Recv: Assoc 0: s=10.5.0.4 8787, d=10.1.0.2 8787, len 56
SCTP: INIT_CHUNK, Tag: 4F2D8235, TSN: 4F2D8235
SCTP: Sent: Assoc NULL: s=10.1.0.2 8787, d=10.5.0.4 8787, len 136
SCTP: INIT_ACK_CHUNK, Tag: 7DD7E424, TSN: 7DD7E424
```

```

SCTP: Recv: Assoc 0: s=10.5.0.4 8787, d=10.1.0.2 8787, len 100
SCTP: COOKIE_ECHO_CHUNK, len 88
SCTP: Sent: Assoc NULL: s=10.1.0.2 8787, d=10.5.0.4 8787, len 16
SCTP: COOKIE_ACK_CHUNK
SCTP: Recv: Assoc 0: s=10.5.0.4 8787, d=10.1.0.2 8787, len 144
SCTP: SACK_CHUNK, TSN ack: 7DD7E423, rwnd 18000, num frags 0
SCTP: DATA_CHUNK, 4/0/100/4F2D8235
SCTP: Sent: Assoc 0: s=10.1.0.2 8787, d=10.5.0.4 8787, len 28
SCTP: SACK_CHUNK, TSN ack: 4F2D8235, rwnd 8900, num frags 0
SCTP: Sent: Assoc 0: s=10.1.0.2 8787, d=10.5.0.4 8787, len 128
SCTP: DATA_CHUNK, 4/0/100/7DD7E424
SCTP: Recv: Assoc 0: s=10.5.0.4 8787, d=10.1.0.2 8787, len 28
SCTP: SACK_CHUNK, TSN ack: 7DD7E424, rwnd 17900, num frags 0
SCTP: Recv: Assoc 0: s=10.6.0.4 8787, d=10.2.0.2 8787, len 44
SCTP: HEARTBEAT_CHUNK
SCTP: Sent: Assoc 0: s=10.2.0.2 8787, d=10.6.0.4 8787, len 44
SCTP: HEARTBEAT_ACK_CHUNK
SCTP: Recv: Assoc 0: s=10.5.0.4 8787, d=10.1.0.2 8787, len 128
SCTP: DATA_CHUNK, 7/0/100/4F2D8236
SCTP: Sent: Assoc 0: s=10.1.0.2 8787, d=10.5.0.4 8787, len 144
SCTP: SACK_CHUNK, TSN ack: 4F2D8236, rwnd 9000, num frags 0
SCTP: DATA_CHUNK, 7/0/100/7DD7E425
SCTP: Recv: Assoc 0: s=10.5.0.4 8787, d=10.1.0.2 8787, len 28
SCTP: SACK_CHUNK, TSN ack: 7DD7E424, rwnd 18000, num frags 0
SCTP: Recv: Assoc 0: s=10.5.0.4 8787, d=10.1.0.2 8787, len 28
SCTP: SACK_CHUNK, TSN ack: 7DD7E425, rwnd 17900, num frags 0
SCTP: Recv: Assoc 0: s=10.5.0.4 8787, d=10.1.0.2 8787, len 128
SCTP: DATA_CHUNK, 4/1/100/4F2D8237

```

Step 9 debug ip sctp segmentv

The **debug ip sctp segmentv** command output shows every datagram that is sent or received and the chunks that are contained in each. This is the verbose form of the output, and it shows detailed information for each chunk type (see the **debug ip sctp segments** command for the simple form output).

Caution The **debug ip sctp segmentv** command generates multiple lines of output for each datagram sent and received. It should be used with extreme caution in a live network.

The following output shows an example in which an association is established, a few heartbeats are sent, the remote endpoint fails, and the association is restarted.

Example:

```

Router# debug ip sctp segmentv

SCTP: Sent: Assoc 0: s=10.1.0.2 8787, d=10.5.0.4 8787, len 56, ver tag 0
SCTP: INIT_CHUNK, len 42
SCTP: Initiate Tag: B131ED6A, Initial TSN: B131ED6A, rwnd 9000
SCTP: Streams Inbound: 13, Outbound: 13
SCTP: IP Addr: 10.1.0.2
SCTP: IP Addr: 10.2.0.2
SCTP: Supported addr types: 5
SCTP: Recv: Assoc 0: s=10.5.0.4 8787, d=10.1.0.2 8787, len 56, ver tag 0
SCTP: INIT_CHUNK, len 42
SCTP: Initiate Tag: 5516B2F3, Initial TSN: 5516B2F3, rwnd 18000
SCTP: Streams Inbound: 13, Outbound: 13
SCTP: IP Addr: 10.5.0.4
SCTP: IP Addr: 10.6.0.4
SCTP: Supported addr types: 5
SCTP: Sent: Assoc NULL: s=10.1.0.2 8787, d=10.5.0.4 8787, len 136, ver tag 5516B2F3
SCTP: INIT_ACK_CHUNK, len 124
SCTP: Initiate Tag: B131ED6A, Initial TSN: B131ED6A, rwnd 9000

```

```

SCTP:      Streams Inbound: 13, Outbound: 13
SCTP:      Responder cookie len 88
SCTP:      IP Addr: 10.1.0.2
SCTP:      IP Addr: 10.2.0.2
SCTP: Recv: Assoc 0: s=10.5.0.4 8787, d=10.1.0.2 8787, len 100, ver tag B131ED6A
SCTP:      COOKIE_ECHO_CHUNK, len 88
SCTP: Sent: Assoc NULL: s=10.1.0.2 8787, d=10.5.0.4 8787, len 16, ver tag 5516B2F3
SCTP:      COOKIE_ACK_CHUNK
SCTP: Recv: Assoc 0: s=10.5.0.4 8787, d=10.1.0.2 8787, len 144, ver tag B131ED6A
SCTP:      SACK_CHUNK, len 16
SCTP:      TSN ack: (0xB131ED69)
SCTP:      Rcv win credit: 18000
SCTP:      Num frags: 0
SCTP:      DATA_CHUNK, flags 3, chunkLen 116
SCTP:      DATA_CHUNK, 0/0/100/5516B2F3
SCTP: Sent: Assoc 0: s=10.1.0.2 8787, d=10.5.0.4 8787, len 28, ver tag 5516B2F3
SCTP:      SACK_CHUNK, len 16
SCTP:      TSN ack: (0x5516B2F3)
SCTP:      Rcv win credit: 8900
SCTP:      Num frags: 0
SCTP: Sent: Assoc 0: s=10.1.0.2 8787, d=10.5.0.4 8787, len 128, ver tag 5516B2F3
SCTP:      DATA_CHUNK, flags 3, chunkLen 116
SCTP:      DATA_CHUNK, 0/0/100/B131ED6A
SCTP: Recv: Assoc 0: s=10.6.0.4 8787, d=10.2.0.2 8787, len 44, ver tag B131ED6A
SCTP:      HEARTBEAT_CHUNK
SCTP: Sent: Assoc 0: s=10.2.0.2 8787, d=10.6.0.4 8787, len 44, ver tag 5516B2F3
SCTP:      HEARTBEAT_ACK_CHUNK
SCTP: Recv: Assoc 0: s=10.5.0.4 8787, d=10.1.0.2 8787, len 28, ver tag B131ED6A
SCTP:      SACK_CHUNK, len 16

```

Step 10 debug ip sctp signal

The **debug ip sctp signal** command shows signals that are sent from SCTP to the application or ULP. These signals inform the ULP of state transitions for associations or destination addresses. There is also a signal sent to the ULP when new data is available to be received, but this signal is not shown in the example output below because it occurs infrequently. This debug command can be used to see if the current associations are stable. Because it does not generate output except on state transitions, it is safe to use in a live environment. It still should be used with caution, however, depending on the number of associations being handled by the system and the stability of the network.

Step 11 debug ip sctp state

The **debug ip sctp state** command is often used at the same time as the **debug ip sctp signal** command. Using the two commands together gives good insight into the stability of associations.

In the following example, a new association is requested and established. The peer then restarts the association and notes that the association failed and is being reestablished. The local peer then indicates that the association has failed because it has tried to retransmit the specified chunk more than the maximum number of times without success. As a result, the association fails (because of communication loss) and is terminated. The ULP requests that the association be attempted again, and this attempt succeeds. A shutdown is then received from the remote peer, and the local peer enters the shutdown acknowledge sent state, which is followed by the association being terminated. Again, another association attempt is made and succeeds.

Example:

```

Router# debug ip sctp signal
Router# debug ip sctp state

<new assoc attempt>
00:20:08: SCTP: Assoc 0: state CLOSED -> COOKIE_WAIT
00:20:15: SCTP: Assoc 0: state COOKIE_WAIT -> ESTABLISHED
00:20:15: SCTP: Assoc 0: Sent ASSOC_UP signal for CONFIGD_ASSOC

```

```

00:21:03: SCTP: Assoc 0: Restart rcvd from peer
00:21:03: SCTP: Assoc 0: Sent ASSOC_RESTART signal
00:21:04: SCTP: Assoc 0: chunk 62EA7F40 retransmitted more than max times, failing assoc
00:21:04: SCTP: Assoc 0: Sent ASSOC_FAILED signal, reason: SCTP_COMM_LOST
00:21:04: SCTP: Assoc 0: Sent ASSOC_TERMINATE signal
00:21:04: SCTP: Assoc 0: state ESTABLISHED -> CLOSED
<new assoc attempt>
00:21:04: SCTP: Assoc 0: state CLOSED -> COOKIE_WAIT
00:21:04: SCTP: Assoc 0: state COOKIE_WAIT -> COOKIE_ECHOED
00:21:04: SCTP: Assoc 0: state COOKIE_ECHOED -> ESTABLISHED
00:21:04: SCTP: Assoc 0: Sent ASSOC_UP signal for CONFIGD_ASSOC
00:21:04: SCTP: Assoc 0: Sent TERMINATE_PENDING signal
00:21:04: SCTP: Assoc 0: state ESTABLISHED -> SHUTDOWN_ACKSENT
00:21:04: SCTP: Assoc 0: Sent ASSOC_TERMINATE signal
00:21:04: SCTP: Assoc 0: state SHUTDOWN_ACKSENT -> CLOSED
<new assoc attempt>
00:21:04: SCTP: Assoc 0: state CLOSED -> COOKIE_WAIT
00:21:04: SCTP: Assoc 0: state COOKIE_WAIT -> COOKIE_ECHOED
00:21:04: SCTP: Assoc 0: state COOKIE_ECHOED -> ESTABLISHED
00:21:04: SCTP: Assoc 0: Sent ASSOC_UP signal for CONFIGD_ASSOC

```

Step 12 debug ip sctp sndchunks

The **debug ip sctp sndchunks** command shows the following types of information about all chunks that are being sent to remote SCTP peers:

- Application send requests from the local SCTP peer
- Chunks being bundled and sent to the remote peer
- Processing of the SACKs from the remote peer, indicating which chunks were successfully received
- Chunks that are marked for retransmission

Caution The **debug ip sctp sndchunks** command generates large amounts of data if there is any significant amount of traffic flowing. It should be used with extreme caution in live networks.

Example:

```
Router# debug ip sctp sndchunks
```

```

SCTP: Assoc 0: ApplSend, chunk: 0/10412/100/A23134F8 to 10.5.0.4
SCTP: Assoc 0: ApplSend, chunk: 5/10443/100/A23134F9 to 10.5.0.4
SCTP: Assoc 0: ApplSend, chunk: 5/10448/100/A231355C to 10.5.0.4
SCTP: Assoc 0: Set oldest chunk for dest 10.5.0.4 to TSN A23134F8
SCTP: Assoc 0: Bundling data, added 0/10412/100/A23134F8, outstanding 100
SCTP: Assoc 0: Bundling data, added 5/10443/100/A23134F9, outstanding 200
SCTP: Assoc 0: Bundling data, added 4/10545/100/A23134FA, outstanding 300
SCTP: Assoc 0: Bundling data, added 10/10371/100/A23134FB, outstanding 400
SCTP: Assoc 0: Bundling data, added 11/10382/100/A23134FC, outstanding 500
SCTP: Assoc 0: Process Sack Chunk, CumTSN=A231350F, numFrgs=0
SCTP: Assoc 0: Reset oldest chunk on addr 10.5.0.4 to A2313510
SCTP: Assoc 0: Process Sack Chunk, CumTSN=A2313527, numFrgs=0
SCTP: Assoc 0: Reset oldest chunk on addr 10.5.0.4 to A2313528
SCTP: Assoc 0: Process Sack Chunk, CumTSN=A231353F, numFrgs=0
SCTP: Assoc 0: Reset oldest chunk on addr 10.5.0.4 to A2313540
SCTP: Assoc 0: Process Sack Chunk, CumTSN=A2313557, numFrgs=0
SCTP: Assoc 0: Reset oldest chunk on addr 10.5.0.4 to A2313558
SCTP: Assoc 0: ApplSend, chunk: 10/10385/100/A23135BE to 10.5.0.4
SCTP: Assoc 0: ApplSend, chunk: 8/10230/100/A23135BF to 10.5.0.4
SCTP: Assoc 0: ApplSend, chunk: 5/10459/100/A23135C0 to 10.5.0.4
SCTP: Assoc 0: ApplSend, chunk: 4/10558/100/A23135C1 to 10.5.0.4

```



```

SCTP: Assoc 0: Set oldest chunk for dest 10.5.0.4 to TSN A231355D
SCTP: Assoc 0: Bundling data, added 5/10449/100/A231355D, outstanding 100
SCTP: Assoc 0: Bundling data, added 3/10490/100/A231355E, outstanding 200
SCTP: Assoc 0: Process Sack Chunk, CumTSN=A23135A4, numFrgs=0
SCTP: Assoc 0: Reset oldest chunk on addr 10.5.0.4 to A23135A5
SCTP: Assoc 0: Process Sack Chunk, CumTSN=A23135BC, numFrgs=0
SCTP: Assoc 0: Reset oldest chunk on addr 10.5.0.4 to A23135BD
SCTP: Assoc 0: Process Sack Chunk, CumTSN=A23135C1, numFrgs=0
SCTP: Assoc 0: ApplSend, chunk: 5/10460/100/A23135C2 to 10.5.0.4
SCTP: Assoc 0: ApplSend, chunk: 5/10461/100/A23135C3 to 10.5.0.4
SCTP: Assoc 0: ApplSend, chunk: 11/10403/100/A2313626 to 10.5.0.4
SCTP: Assoc 0: Set oldest chunk for dest 10.5.0.4 to TSN A23135C2
SCTP: Assoc 0: Bundling data, added 5/10460/100/A23135C2, outstanding 100
SCTP: Assoc 0: Bundling data, added 5/10461/100/A23135C3, outstanding 200
SCTP: Assoc 0: Bundling data, added 5/10462/100/A23135C4, outstanding 300
SCTP: Assoc 0: Bundling data, added 4/10559/100/A23135C5, outstanding 400
SCTP: Assoc 0: Bundling data, added 4/10560/100/A23135C6, outstanding 500
SCTP: Assoc 0: Bundled 12 chunk(s) in next dgram to 10.5.0.4
SCTP: Assoc 0: Bundling data, added 1/10418/100/A2313622, outstanding 9700
SCTP: Assoc 0: Bundling data, added 3/10502/100/A2313623, outstanding 9800
SCTP: Assoc 0: Bundling data, added 7/10482/100/A2313624, outstanding 9900
SCTP: Assoc 0: Bundling data, added 3/10503/100/A2313625, outstanding 10000
SCTP: Assoc 0: Bundling data, added 11/10403/100/A2313626, outstanding 10100
SCTP: Assoc 0: Bundled 5 chunk(s) in next dgram to 10.5.0.4
SCTP: Assoc 0: Mark chunk A23135C2 for retrans
SCTP: Assoc 0: Mark chunk A23135C3 for retrans
SCTP: Assoc 0: Mark chunk A23135C4 for retrans
SCTP: Assoc 0: Mark chunk A23135C5 for retrans
SCTP: Assoc 0: Mark chunk A23135C6 for retrans
SCTP: Assoc 0: Mark chunk A23135C7 for retrans
SCTP: Assoc 0: Mark chunk A23135C8 for retrans
SCTP: Assoc 0: Mark chunk A23135C9 for retrans
SCTP: Assoc 0: Mark chunk A23135CA for retrans
SCTP: Assoc 0: Bundled 6 chunk(s) in next dgram to 10.6.0.4
SCTP: Assoc 0: Mark chunk A23135C2 for retrans
SCTP: Assoc 0: Mark chunk A23135C3 for retrans
SCTP: Assoc 0: Mark chunk A23135C4 for retrans

```

Step 13 debug ip sctp timer

The **debug ip sctp timer** command displays information about all started, stopped, and triggering SCTP timers. After they have been started, many SCTP timers are not restarted until they expire or are stopped. For these timers, the first call succeeds in starting the timer, and subsequent calls do nothing until the timer either expires or is stopped. For example, the retransmission timer is started when the first chunk is sent, but then is not started again for subsequent chunks when there is outstanding data.

Caution The **debug ip sctp timer** command generates a significant amount of output. It should be used with extreme caution in a live network.

The following example shows output from the **debug ip sctp timer** command:

Example:

```

Router# debug ip sctp timer

SCTP: Assoc 0: Starting CUMSACK timer
SCTP: Timer already started, not restarting
SCTP: Assoc 0: Starting CUMSACK timer
SCTP: Timer already started, not restarting
SCTP: Assoc 0: Timer BUNDLE triggered
SCTP: Assoc 0: Starting RETRANS timer for destaddr 10.5.0.4
SCTP: Assoc 0: Starting RETRANS timer for destaddr 10.5.0.4

```

```

SCTP: Timer already started, not restarting
SCTP: Assoc 0: Starting RETRANS timer for destaddr 10.5.0.4
SCTP: Timer already started, not restarting
SCTP: Assoc 0: Starting RETRANS timer for destaddr 10.5.0.4
SCTP: Timer already started, not restarting
SCTP: Assoc 0: Stopping RETRANS timer for destaddr 10.5.0.4
SCTP: Assoc 0: Starting RETRANS timer for destaddr 10.5.0.4
SCTP: Assoc 0: Stopping RETRANS timer for destaddr 10.5.0.4
SCTP: Assoc 0: Starting CUMSACK timer
SCTP: Timer already started, not restarting
SCTP: Assoc 0: Starting CUMSACK timer
SCTP: Timer already started, not restarting
SCTP: Assoc 0: Starting CUMSACK timer
SCTP: Timer already started, not restarting
SCTP: Assoc 0: Starting CUMSACK timer
SCTP: Assoc 0: Starting CUMSACK timer
SCTP: Timer already started, not restarting
SCTP: Assoc 0: Starting CUMSACK timer
SCTP: Timer already started, not restarting
SCTP: Assoc 0: Stopping CUMSACK timer
SCTP: Assoc 0: Starting CUMSACK timer
SCTP: Assoc 0: Starting CUMSACK timer
SCTP: Assoc 0: Starting CUMSACK timer
SCTP: Timer already started, not restarting

```

Step 14 debug ip sctp warnings

The **debug ip sctp warnings** command displays information on any unusual situation that is encountered. These situations may or may not indicate problems, depending on the particulars of the situation. Below are some examples of events or conditions that are flagged as warnings.

Example:

```

Router# debug ip sctp warnings

SCTP: Assoc 0: No cookie in InitAck, discarding
SCTP: Assoc 0: Incoming INIT_ACK: inbound streams req'd 15, allowed 13
SCTP: Assoc 0: Incoming INIT_ACK request: outbound streams req'd 13, allowed 1
SCTP: Assoc 0: Remote verification tag in init ack is zero, discarding
SCTP: Remote verification tag in init is zero, discarding
SCTP: Assoc 0: Rwnd less than min allowed (1500) in incoming INITACK, rcvd 0
SCTP: Assoc 0: Rwnd less than min allowed (1500) in incoming INITACK, rcvd 1499
SCTP: Rwnd in INIT too small (0), discarding
SCTP: Rwnd in INIT too small (1499), discarding
SCTP: Unknown INIT param 16537 (0x4099), length 8
SCTP: Assoc 0: Unknown INITACK param 153 (0x99), length 8
SCTP: Assoc 0: No cookie in InitAck, discarding
SCTP: Assoc 0: No cookie in InitAck, discarding
SCTP: Processing INIT, invalid param len 0, discarding...
SCTP: Assoc 0: Processing INITACK, invalid param len 0, discarding...

```

Configuration Examples for SCTP

Example: Defining SCTP Authentication Parameters

The following example shows how to define SCTP data chunks that the client requires be authenticated and to configure SCTP to automatically send ASCONF chunks in response to IP address changes on the router:

```

Router# configure terminal
Router(config)# ip sctp authenticate data
Router(config)# ip sctp asconf auto

```

Additional References

Related Documents

| Related Topic | Document Title |
|------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Cisco IOS commands | Cisco IOS Master Commands List, All Releases |
| Applications that make use of SCTP | <ul style="list-style-type: none"> • PRI Backhaul Using the Stream Control Transmission Protocol and the ISDN Q.921 User Adaptation Layer • Netflow Reliable Export with SCTP section in the <i>Cisco IOS Netflow Configuration Guide</i> |
| SCTP commands | <i>Cisco IOS IP Application Services Command Reference</i> |

Standards

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

MIBs

| MIB | MIBs Link |
|-----------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| No new 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 software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs |

RFCs

| RFC | Title |
|----------|-------------------------------------------------------------|
| RFC 2960 | Stream Control Transmission Protocol (SCTP) |

Technical Assistance

| Description | Link |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------|
| <p>The Cisco Support and Documentation website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</p> <p>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.</p> <p>Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.</p> | <p>http://www.cisco.com/cisco/web/support/index.html</p> |

Feature Information for SCTP

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

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

Table 13: Feature Information for SCTP

| Feature Name | Releases | Feature Information |
|-----------------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SCTP, Release 1 | 12.2(4)T | <p>Stream Control Transmission Protocol (SCTP) is a reliable datagram-oriented IP transport protocol specified by RFC 2960.</p> <p>In 12.2(4)T, this feature was introduced.</p> <p>The following commands were introduced or modified by this feature: clear ip sctp statistics, debug ip sctp api, debug ip sctp congestion, debug ip sctp init, debug ip sctp multihome, debug ip sctp performance, debug ip sctp rcvchunks, debug ip sctp rto, debug ip sctp segments, debug ip sctp segmentv, debug ip sctp signal, debug ip sctp sndchunks, debug ip sctp state, debug ip sctp timer, debug ip sctp warnings, show ip sctp association list, show ip sctp association parameters, show ip sctp association statistics, show ip sctp errors, show ip sctp instances, show ip sctp statistics.</p> |
| SCTP Release 2 | 12.2(8)T | <p>SCTP Release 2 introduced updated output for SCTP commands.</p> <p>The following commands were modified by this feature: show ip sctp association paramters, show ip sctp association statistics.</p> |

| Feature Name | Releases | Feature Information |
|----------------------------------|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SCTP Show/Clear CLI Enhancements | 12.4(11)T | <p>The SCTP Show/Clear CLI Enhancements feature provides access to additional SCTP information that can help with troubleshooting potential problems. These enhancements also make the updated SCTP show and clear commands consistent with the CLI of other transport protocols.</p> <p>The following commands were replaced by this feature: clear ip sctp statistics, show ip sctp association list, show ip sctp association parameters, show ip sctp association statistics, show ip sctp errors, show ip sctp instances, show ip sctp statistics.</p> <p>The following commands were introduced by this feature: clear sctp statistics, show sctp association, show sctp association list, show sctp association parameters, show sctp association statistics, show sctp errors, show sctp instance, show sctp instances, show sctp instances, show sctp statistics.</p> |
| SCTP Release 4, Phase 1 | 12.4(15)T | <p>SCTP Release 4 introduced the SCTP Stream Reset and Authentication features. SCTP Release 4 introduced the SCTP Stream Reset and Authentication features.</p> <p>SCTP stream reset feature enables SCTP to reset stream transport sequence numbers and all stream sequence numbers. SCTP stream reset enables SCTP to:</p> <ul style="list-style-type: none"> • Dynamically reset a peer's outbound streams • Dynamically reset a local host's outbound stream • Dynamically reset specific numbered streams <p>SCTP Authentication enables SCTP to:</p> <ul style="list-style-type: none"> • Set up a dynamic shared association key with no shared secret • Allow a shared secret to be combined with an association key • Use the shared association secret to authenticate chunks • Negotiate which chunk types must be authenticated |
| SCTP Release 4, Phase 2 | 12.4(20)T | <p>As of Cisco IOS Release 12.4(20)T, SCTP Release 4 introduced the SCTP Add-IP feature.</p> <p>The SCTP Add-IP feature enables the ability to add or delete an IP address for an endpoint of an existing SCTP association and to communicate this change to the remote end. An ADD-IP chunk is sent to the remote end adding or removing the redundant server addresses of the association. The ADD-IP chunk also deletes all addresses of a failed host from an association. The SCTP Add-IP feature also enables an application to programmatically set the primary address for an SCTP association.</p> <p>The following commands are new or modified in this release: ip sctp authenticate, ip sctp asconf.</p> |



CHAPTER 10

WCCP—Configurable Router ID

The WCCP—Configurable Router ID feature enables the configuration of a Web Cache Communication Protocol (WCCP) source interface. The IP address of this configured source interface is then used as the preferred WCCP router ID and WCCP GRE source address. When a WCCP router ID is manually configured, router IDs are no longer automatically generated when the current router ID is no longer valid and the router ID does not change when another IP address is added to the system.

- [Finding Feature Information, on page 179](#)
- [Restrictions for WCCP—Configurable Router ID, on page 179](#)
- [Information About WCCP—Configurable Router ID, on page 180](#)
- [How to Configure WCCP—Configurable Router ID, on page 180](#)
- [Configuration Examples for WCCP—Configurable Router ID, on page 181](#)
- [Additional References for WCCP—Configurable Router ID, on page 181](#)
- [Feature Information for WCCP—Configurable Router ID, on page 182](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

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

Restrictions for WCCP—Configurable Router ID

The following restriction apply to this feature:

- Do not configure the Web Cache Control Protocol (WCCP) router ID as the tunnel source, if multipoint generic routing encapsulation (GRE) tunnels are configured on a router, because this configuration may cause the traffic over this tunnel to fail.

Information About WCCP—Configurable Router ID

WCCP—Configurable Router ID Overview

WCCP uses a router ID in its control messages that a WCCP client can use to uniquely identify a particular WCCP server. The router ID is an IP address and is used as the source address of any WCCP-generated Generic Routing Encapsulation (GRE) frames. Prior to the WCCP—Configurable Router ID feature, WCCP selected a router ID using an automatic mechanism; the highest reachable IP address on the system (or the highest loopback IP address, if there is one) was used as the WCCP router ID. The highest IP address on the system is not always the best choice as the router ID or as the source address of GRE frames. A change in addressing information on the system may cause the WCCP router ID to change unexpectedly. During this changeover period, WCCP clients briefly advertise the existence of two routers (the old router ID and the new router ID) and GRE frames are sourced from a different address.

The WCCP—Configurable Router ID feature enables you to define a WCCP source interface from which the router ID will be obtained. The IP address of this configured source interface is then used as the preferred WCCP router ID and WCCP GRE source address. When a WCCP router ID is manually configured, the router ID does not change when another IP address is added to the system. The router ID changes only when a new router ID is manually configured using the `ip wccp source-interface` or the `ipv6 wccp source-interface` command, or when the address on the manually configured interface is no longer valid.

How to Configure WCCP—Configurable Router ID

Configuring a Preferred WCCP Router ID

SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `ip wccp [vrf vrf-name] source-interface source-interface`
4. `exit`

DETAILED STEPS

| | Command or Action | Purpose |
|--------|-----------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------|
| Step 1 | <code>enable</code> Example: Device> <code>enable</code> | Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted. |
| Step 2 | <code>configure terminal</code> Example: Device# <code>configure terminal</code> | Enters global configuration mode. |

| | Command or Action | Purpose |
|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------|
| Step 3 | ip wccp [vrf vrf-name] source-interface source-interface Example: <pre>Device(config)# ip wccp source-interface GigabitEthernet 0/0/0</pre> | Configures a preferred WCCP router ID. |
| Step 4 | exit Example: <pre>Device(config)# exit</pre> | Exits global configuration mode and returns to privileged EXEC mode. |

Configuration Examples for WCCP—Configurable Router ID

Example: Configuring a Preferred WCCP Router ID

The following example displays the configuration for a preferred WCCP router ID:

```
! Configure a preferred WCCP router ID
ip wccp source-interface GigabitEthernet 0/0/0
```

Additional References for WCCP—Configurable Router ID

Related Documents

| Related Topic | Document Title |
|-----------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Cisco IOS commands | Cisco IOS Master Commands List, All Releases |
| WCCP commands | Cisco IOS IP Application Services Command Reference |
| Currently assigned IP multicast addresses | <i>Internet Multicast Addresses</i> http://www.iana.org/assignments/multicast-addresses |
| Configuration fundamentals configuration tasks | <i>Cisco IOS Configuration Fundamentals Configuration Guide</i> |
| Configuration fundamentals commands | <i>Cisco IOS Configuration Fundamentals Command Reference</i> |
| Cisco IOS bridging and IBM networking configuration tasks | <i>Cisco IOS Bridging and IBM Networking Configuration Guide</i> |
| Cisco IOS bridging and IBM networking commands | <i>Cisco IOS Bridging and IBM Networking Command Reference</i> |

| Related Topic | Document Title |
|--------------------------------------------|---------------------------------------------------|
| Cisco IOS IP multicast configuration tasks | <i>Cisco IOS IP Multicast Configuration Guide</i> |
| Cisco IOS IP Multicast commands | <i>Cisco IOS IP Multicast Command Reference</i> |

Standards

| Standard | Title |
|-----------------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| IEEE Spanning-Tree Bridging | 802.1D MAC Bridges http://www.ieee802.org/1/pages/802.1D-2003.html |

MIBs

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

RFCs

| RFC | Title |
|----------|------------------------------------------------------------------------------------------------------------------------------------|
| RFC 1812 | <i>Requirements for IP Version 4 Routers</i> http://www.ietf.org/rfc/rfc1812.txt |
| RFC 2131 | <i>Dynamic Host Configuration Protocol</i> http://www.ietf.org/rfc/rfc2131.txt |

Technical Assistance

| Description | Link |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|
| The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password. | http://www.cisco.com/cisco/web/support/index.html |

Feature Information for WCCP—Configurable Router ID

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

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

Table 14: Feature Information for WCCP—Configurable Router ID

| Feature Name | Releases | Feature Information |
|-----------------------------|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| WCCP—Configurable Router ID | 15.1(1)SG 15.2(3)T | <p>The WCCP—Configurable Router ID feature enables the configuration of a Web Cache Communication Protocol (WCCP) source interface. The IP address of this configured source interface is then used as the preferred WCCP router ID and WCCP GRE source address. When a WCCP router ID is manually configured, router IDs are no longer automatically generated when the current router ID is no longer valid and the router ID does not change when another IP address is added to the system.</p> <p>The following command was added: ip wccp source-interface.</p> |



CHAPTER 11

WCCP—Fast Timers

The Web Cache Communication Protocol (WCCP)—Fast Timers feature enables WCCP to establish redirection using a configurable message interval when a WCCP client is added to a service group or when a WCCP client fails.

The WCCP message interval capability introduced by the WCCP-Fast Timers feature defines the transmission interval that WCCP clients and WCCP routers use when sending keepalive messages and defines a scaling factor used when calculating the timeout value. The WCCP router uses the timeout value to determine if a WCCP client is no longer available and to redirect traffic as a result.

- [Finding Feature Information, on page 185](#)
- [Information About WCCP—Fast Timers, on page 185](#)
- [How to Configure WCCP—Fast Timers, on page 186](#)
- [Configuration Examples for WCCP—Fast Timers, on page 187](#)
- [Additional References For IP Services, on page 187](#)
- [Feature Information for WCCP—Fast Timers, on page 188](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

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

Information About WCCP—Fast Timers

WCCP—Fast Timers Overview

The WCCP—Fast Timers feature enables WCCP to establish redirection using a configurable message interval when a WCCP client is added to a service group or when a WCCP client fails. WCCP routers and WCCP clients exchange keepalive messages at a fixed interval. Prior to the introduction of the WCCP—Fast Timers feature, the WCCP message interval was fixed at 10 seconds. The WCCP—Fast Timers feature enables use of message intervals ranging from 0.5 seconds to 60 seconds and a timeout value scaling factor of 1 to 5. The

default is 10 seconds. The timer interval is driven by the WCCP client which is being redirected to. The WCCP clients must support variable message interval timers in order for the WCCP—Fast Timers feature to function correctly.

The WCCP message interval capability introduced by the WCCP—Fast Timers feature defines the transmission interval that WCCP clients and WCCP routers use when sending keepalive messages and defines a scaling factor used when calculating the timeout value. The WCCP router uses the timeout value to determine if a WCCP client is no longer available and to redirect traffic as a result. The WCCP router enforces a single message interval per service group. WCCP clients with incompatible message intervals are prevented from joining a service group. If a default message interval that is smaller than the default 10 seconds is used, CPU usage will increase.

You can use the **show ip wccp service *service-number* detail** command to display information about the message interval.

How to Configure WCCP—Fast Timers

Displaying WCCP—Fast Timers Information

SUMMARY STEPS

1. **enable**
2. **show ip wccp [[*service-number*] [detail]]**

DETAILED STEPS

| | Command or Action | Purpose |
|--------|---------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Step 1 | enable Example: Device> enable | Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted. |
| Step 2 | show ip wccp [[<i>service-number</i>] [detail]] Example: Device# show ip wccp 24 detail | Displays WCCP client information that includes the message interval information. <ul style="list-style-type: none"> • The message interval is the fixed time interval between successive keepalive messages sent from a WCCP client to a WCCP router. The default time interval is 10 seconds. If the default time interval is configured, the "Message Interval" field is not displayed. <p>Note You configure the time interval on the WCCP client device. Details of the client configuration are specific to each type of client, so you should consult the documentation of your WCCP client device. Client devices may choose not support the full range of settings that are supported by the router.</p> |

Configuration Examples for WCCP—Fast Timers

Example: Displaying WCCP-Fast Timers Information

The following example displays WCCP client information that includes the message interval information:

```
Device# show ip wccp 91 detail

WCCP Client information:
WCCP Client ID: 10.1.1.14
Protocol Version: 2.0
State: Usable
Redirection: GRE
Packet Return: GRE
Assignment: MASK

Message Interval: 2.500 seconds (2.354 since last message)
Client timeout: 15 seconds
Assignment timeout: 25 seconds
Packets Redirected: 0
Connect Time: 00:01:56
Bypassed Packets
Process: 0
CEF: 0
```

Additional References For IP Services

Related Documents

| Related Topic | Document Title |
|----------------------------------|---------------------------------------------------------------------|
| Cisco IOS commands | Cisco IOS Master Commands List, All Releases |
| IP application services commands | Cisco IOS IP Application Services Command Reference |

Standards and RFCs

| Standard | Title |
|----------|------------------------------------------------|
| RFC 1256 | ICMP Router Discovery Messages |

Technical Assistance

| Description | Link |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|
| The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password. | http://www.cisco.com/cisco/web/support/index.html |

Feature Information for WCCP—Fast Timers

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

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

Table 15: Feature Information for WCCP—Fast Timers

| Feature Name | Releases | Feature Information |
|------------------|-----------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| WCCP—Fast Timers | 15.1(1)SG 15.2(3)T Cisco IOS XE Release 3.3SG | <p>The Web Cache Communication Protocol (WCCP)—Fast Timers feature enables WCCP to establish redirection using a configurable message interval when a WCCP client is added to a service group or when a WCCP client fails.</p> <p>The WCCP message interval capability introduced by the WCCP-Fast Timers feature defines the transmission interval that WCCP clients and WCCP routers use when sending keepalive messages and defines a scaling factor used when calculating the timeout value. The WCCP router uses the timeout value to determine if a WCCP client is no longer available and to redirect traffic as a result.</p> <p>The following command were modified: show ip wccp.</p> |



CHAPTER 12

WCCPv2—IPv6 Support

This feature introduces support for Web Cache Communication Protocol version 2 (WCCPv2) in an IPv6 environment.

WCCP is a Cisco-developed content-routing technology that intercepts IP packets and redirects those packets to a destination other than that specified in the IP packet. Typically the packets are redirected from their destination web server on the Internet to a content engine that is local to the client. In some WCCP deployment scenarios, redirection of traffic may also be required from the web server to the client. WCCP enables you to integrate content engines into your network infrastructure.

Multiple routers can use WCCPv2 to service a content engine cluster. In WCCPv1, only one router can redirect content requests to a cluster.

- [Finding Feature Information, on page 189](#)
- [Prerequisites for WCCPv2—IPv6 Support, on page 189](#)
- [Restrictions for WCCPv2—IPv6 Support, on page 190](#)
- [Information About WCCPv2—IPv6 Support, on page 190](#)
- [How to Configure WCCPv2—IPv6 Support, on page 201](#)
- [Configuration Examples for WCCPv2—IPv6 Support, on page 211](#)
- [Additional References, on page 216](#)
- [Feature Information for WCCPv2—IPv6 Support, on page 216](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

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

Prerequisites for WCCPv2—IPv6 Support

- IPv6 must be configured on the interface used for redirection and on the interface facing the content engine.

- The interface connected to the content engine must be a Fast Ethernet or Gigabit Ethernet interface.

Restrictions for WCCPv2—IPv6 Support

WCCPv2

- WCCP bypasses Network Address Translation (NAT) when Cisco Express Forwarding is enabled.
- For routers servicing a multicast cluster, the Time To Live (TTL) value must be set at 15 or lower.
- Service groups can comprise up to 32 content engines and 32 routers.
- All content engines in a cluster must be configured to communicate with all routers servicing the cluster.
- Multicast addresses must be in the range from 224.0.0.0 to 239.255.255.255.

Layer 2 Forwarding and Return

- Layer 2 redirection requires that content engines be directly connected to an interface on each WCCP router. Unless multicast IP addresses are used, WCCP configuration of the content engine must reference the directly connected interface IP address of the WCCP router and not a loopback IP address or any other IP address configured on the WCCP router.

Information About WCCPv2—IPv6 Support

WCCP Overview

WCCP uses Cisco Content Engines (or other content engines running WCCP) to localize traffic patterns in the network, enabling content requests to be fulfilled locally. Traffic localization reduces transmission costs and download time.

WCCP enables routing platforms to transparently redirect content requests. With transparent redirection, users can fulfill content requests locally without configuring their browsers to use a web proxy. Instead, they can use the target URL to request content, and have their requests automatically redirected to a content engine. The word "transparent" in this case means that the end user does not know that a requested file (such as a web page) came from the content engine instead of from the originally specified server.

A content engine receiving a request attempts to service it from its own local cache. If the requested information is not present, the content engine issues its own request to the originally targeted server to get the required information. A content engine retrieving the requested information forwards it to the requesting client and caches it to fulfill future requests, thus maximizing download performance and substantially reducing transmission costs.

WCCP enables a series of content engines, called a content engine cluster, to provide content to a router or multiple routers. Network administrators can easily scale their content engines to manage heavy traffic loads through these clustering capabilities. Cisco clustering technology enables each cluster member to work in parallel, resulting in linear scalability. Clustering content engines greatly improves the scalability, redundancy, and availability of your caching solution. You can cluster up to 32 content engines to scale to your desired capacity.

Layer 2 Forwarding Redirection and Return

WCCP uses either generic routing encapsulation (GRE) or Layer 2 (L2) to redirect or return IP traffic. When WCCP forwards traffic via GRE, the redirected packets are encapsulated within a GRE header. The packets also have a WCCP redirect header. When WCCP forwards traffic using L2, the original MAC header of the IP packet is overwritten and replaced with the MAC header for the WCCP client.

Using L2 as a forwarding method allows direct forwarding to the content engine without further lookup. Layer 2 redirection requires that the router and content engines are directly connected, that is, on the same IP subnetwork.

When WCCP returns traffic via GRE, the returned packets are encapsulated within a GRE header. The destination IP address is the address of the router and the source address is the address of the WCCP client. When WCCP returns traffic via L2, the original IP packet is returned without any added header information. The router to which the packet is returned will recognize the source of the packet and prevent redirection.

The WCCP redirection method does not have to match the return method.

L2 forwarding, return, or redirection are typically used for hardware-accelerated platforms. Depending on your release, L2 forwarding, return, and redirection can also be used for software-switching platforms.

For content engines running Application and Content Networking System (ACNS) software, use the **wccp custom-web-cache** command with the **l2-redirect** keyword to configure L2 redirection. For content engines running Cisco Wide Area Application Services (WAAS) software, use the **wccp tcp-promiscuous** command with the **l2-redirect** keyword to configure L2 redirection.



Note Before configuring a GRE tunnel, configure a loopback interface (that is not attached to a VRF) with an IP address so that the internally created tunnel interface is enabled for IPv4 forwarding by unnumbering itself to this dummy loopback interface. You do not need to configure a loopback interface if the system has at least one interface that is not attached to a VRF and that is configured with an IPv4 address.

For information about Cisco ACNS commands used to configure Cisco Content Engines, see the [Cisco ACNS Software Command Reference](#).

For more information about WAAS commands used to configure Cisco Content Engines, see the [Cisco Wide Area Application Services Command Reference](#).

WCCP Mask Assignment

The WCCP Mask Assignment feature enables mask assignment as the load-balancing method (instead of the default hash assignment method) for a WCCP service.

For content engines running Application and Content Networking System (ACNS) software, use the **wccp custom-web-cache** command with the **mask-assign** keyword to configure mask assignment. For content engines running Cisco Wide Area Application Services (WAAS) software, use the **wccp tcp-promiscuous** command with the **mask-assign** keyword to configure mask assignment.

For information about Cisco ACNS commands used to configure Cisco Content Engines, see the [Cisco ACNS Software Command Reference](#).

For more information about WAAS commands used to configure Cisco Content Engines, see the [Cisco Wide Area Application Services Command Reference](#).

WCCP Hash Assignment

The Cisco ASR 1000 Series Aggregation Services Routers support hash assignment for IPv6 load balance across different content engines, but does not support mask assignment. However, it supports both hash assignment and mask assignment for IPv4.

For content engines running the Cisco Application and Content Networking System (ACNS) software, use the **wccp custom-web-cache** command with the **hash-assign** keyword to configure hash assignment. For content engines running Cisco Wide Area Application Services (WAAS) software, use the **wccp tcp-promiscuous** command with the **hash-assign** keyword to configure hash assignment.

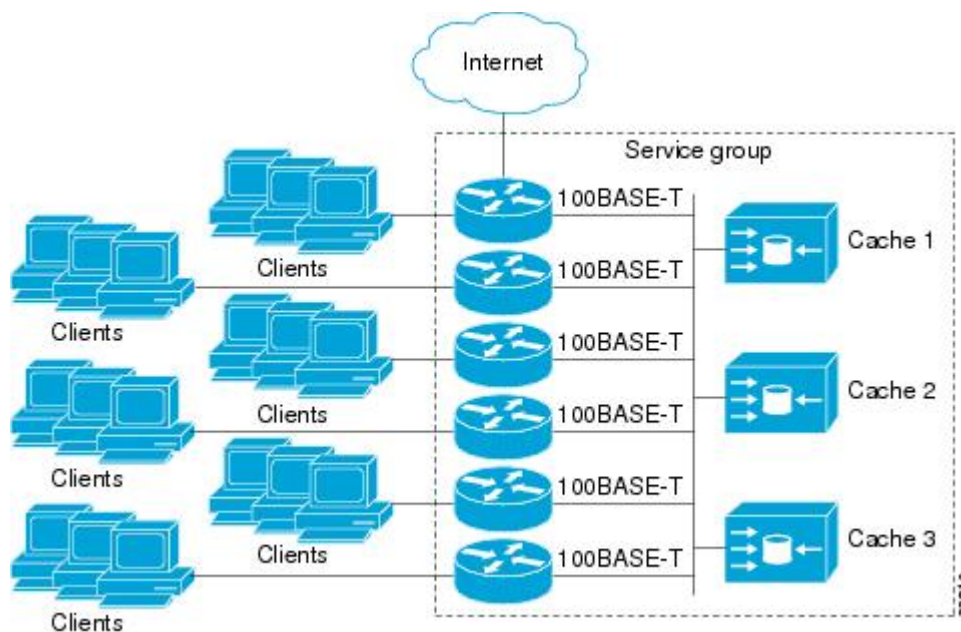
For information about Cisco ACNS commands used to configure Cisco Content Engines, see the [Cisco ACNS Software Command Reference](#).

For more information about WAAS commands used to configure Cisco Content Engines, see the [Cisco Wide Area Application Services Command Reference](#).

WCCPv2 Configuration

Multiple routers can use WCCPv2 to service a content engine cluster. In WCCPv1, only one router could redirect content requests to a cluster. The figure below illustrates a sample configuration using multiple routers.

Figure 10: Cisco Content Engine Network Configuration Using WCCPv2



The subset of content engines within a cluster and routers connected to the cluster that are running the same service is known as a service group. Available services include TCP and UDP redirection.

In WCCPv1, the content engines were configured with the address of the single router. WCCPv2 requires that each content engine be aware of all the routers in the service group. To specify the addresses of all the routers in a service group, you must choose one of the following methods:

- Unicast—A list of router addresses for each of the routers in the group is configured on each content engine. In this case the address of each router in the group must be explicitly specified for each content engine during configuration.
- Multicast—A single multicast address is configured on each content engine. In the multicast address method, the content engine sends a single-address notification that provides coverage for all routers in the service group. For example, a content engine could indicate that packets should be sent to a multicast address of 224.0.0.100, which would send a multicast packet to all routers in the service group configured for group listening using WCCP (see the **ip wccp group-listen** or the **ipv6 wccp group-listen** interface configuration command for details).

The multicast option is easier to configure because you need only specify a single address on each content engine. This option also allows you to add and remove routers from a service group dynamically, without needing to reconfigure the content engines with a different list of addresses each time.

The following sequence of events details how WCCPv2 configuration works:

1. Each content engine is configured with a list of routers.
2. Each content engine announces its presence and a list of all routers with which it has established communications. The routers reply with their view (list) of content engines in the group.
3. When the view is consistent across all content engines in the cluster, one content engine is designated as the lead and sets the policy that the routers need to deploy in redirecting packets.

WCCPv2 Support for Services Other Than HTTP

WCCPv2 allows redirection of traffic other than HTTP (TCP port 80 traffic), including a variety of UDP and TCP traffic. WCCPv2 supports the redirection of packets intended for other ports, including those used for proxy-web cache handling, File Transfer Protocol (FTP) caching, FTP proxy handling, web caching for ports other than 80, and Real Audio, video, and telephony applications.

To accommodate the various types of services available, WCCPv2 introduced the concept of multiple *service groups*. Service information is specified in the WCCP configuration commands using dynamic services identification numbers (such as 98) or a predefined service keyword (such as **web-cache**). This information is used to validate that service group members are all using or providing the same service.

The content engines in a service group specify traffic to be redirected by protocol (TCP or UDP) and up to eight source or destination ports. Each service group has a priority status assigned to it. The priority of a dynamic service is assigned by the content engine. The priority value is in the range of 0 to 255 where 0 is the lowest priority. The predefined web-cache service has an assigned priority of 240.

WCCPv2 Support for Multiple Routers

WCCPv2 allows multiple routers to be attached to a cluster of cache engines. The use of multiple routers in a service group allows for redundancy, interface aggregation, and distribution of the redirection load. WCCPv2 supports up to 32 routers per service group. Each service group is established and maintained independently.

WCCPv2 MD5 Security

WCCPv2 provides optional authentication that enables you to control which routers and content engines become part of the service group using passwords and the Hashed Message Authentication Code—Message

Digest (HMAC MD5) standard. Shared-secret MD5 one-time authentication (set using the `ip wccp [password [0 | 7] password]` global configuration command) enables messages to be protected against interception, inspection, and replay.

WCCPv2 Web Cache Packet Return

If a content engine is unable to provide a requested object it has cached due to error or overload, the content engine will return the request to the router for onward transmission to the originally specified destination server. WCCPv2 provides a check on packets that determines which requests have been returned from the content engine unserved. Using this information, the router can then forward the request to the originally targeted server (rather than attempting to resend the request to the content engine cluster). This process provides error handling transparency to clients.

Typical reasons why a content engine would reject packets and initiate the packet return feature include the following:

- Instances when the content engine is overloaded and has no room to service the packets
- Instances when the content engine is filtering for certain conditions that make caching packets counterproductive (for example, when IP authentication has been turned on)

WCCPv2 Load Distribution

WCCPv2 can be used to adjust the load being offered to individual content engines to provide an effective use of the available resources while helping to ensure high quality of service (QoS) to the clients. WCCPv2 allows the designated content engine to adjust the load on a particular content engine and balance the load across the content engines in a cluster. WCCPv2 uses three techniques to perform load distribution:

- Hot spot handling—Allows an individual hash bucket to be distributed across all the content engines. Prior to WCCPv2, information from one hash bucket could go to only one content engine.
- Load balancing—Allows the set of hash buckets assigned to a content engine to be adjusted so that the load can be shifted from an overwhelmed content engine to other members that have available capacity.
- Load shedding—Enables the router to selectively redirect the load to avoid exceeding the capacity of a content engine.

The use of these hashing parameters prevents one content engine from being overloaded and reduces the potential for bottlenecking.

WCCP VRF Support

The WCCP VRF Support feature enhances the WCCPv2 protocol by implementing support for virtual routing and forwarding (VRF).

The WCCP VRF Support feature allows service groups to be configured on a per-VRF basis in addition to those defined globally.

Along with the service identifier, the VRF of WCCP protocol packets arriving at the router is used to associate cache-engines with a configured service group.

The same VRF must have the interface on which redirection is applied, the interface which is connected to cache engine, and the interface on which the packet would have left if it had not been redirected.

IPv6 WCCP Tunnel Interface

The use of GRE redirection results in the creation of new tunnel interfaces. You can display these tunnel interfaces by entering the **show ipv6 interface brief | include tunnel** command:

```
Device# show ipv6 interface brief | include tunnel

Tunnel0          2001::DB8:1::1    YES unset up
Tunnel1          2001::DB8:1::1    YES unset up
Tunnel2          2001::DB8:1::1    YES unset up
Tunnel3          2001::DB8:1::1    YES unset up
Device#
```

The tunnel interfaces are automatically created in order to process outgoing GRE-encapsulated traffic for WCCP. The tunnel interfaces appear when a content engine connects and requests GRE redirection. The tunnel interfaces are not created directly by WCCP, but are created indirectly via a tunnel application programming interface (API). WCCP does not have direct knowledge of the tunnel interfaces, but can redirect packets to them, resulting in the appropriate encapsulation being applied to the packets. After the appropriate encapsulation is applied, the packet is then sent to the content engine.



Note The tunnel interfaces are not used to connect with incoming WCCP GRE return packets.

One tunnel is created for each service group that is using GRE redirection. One additional tunnel is created to provide an IP address that allows the other tunnel group interfaces to be unnumbered but still enabled for IPv6.

You can confirm the connection between the tunnels and WCCP by entering the **show tunnel groups wccp** command:

```
Device# show tunnel groups wccp

WCCP : service group 0 in "Default", ver v2, assgnmnt: hash-table
      intf: Tunnel0, locally sourced
WCCP : service group 317 in "Default", ver v2, assgnmnt: hash-table
      intf: Tunnel3, locally sourced
WCCP : service group 318 in "Default", ver v2, assgnmnt: hash-table
      intf: Tunnel2, locally sourced
```

You can display additional information about each tunnel interface by entering the **show tunnel interface interface-number** command:

```
Device# show tunnel interface t0

Tunnel0
  Mode:multi-GRE/IP, Destination UNKNOWN, Source 2001::DB8:1::2
  Application ID 2: WCCP : service group 0 in "Default", ver v2, assgnmnt: hash-table
  Linestate - current up
  Internal linestate - current up, evaluated up

Device# show tunnel interface t1

Tunnel1
  Mode:multi-GRE/IP, Destination UNKNOWN, Source 2001::DB8:1::1
  Application ID 2: unspecified
  Linestate - current up
  Internal linestate - current up, evaluated up
```

```
Device# show tunnel interface t2
```

```
Tunnel2
  Mode:multi-GRE/IP, Destination UNKNOWN, Source 2001::DB8:1::1
  Application ID 2: WCCP : service group 318 in "Default", ver v2, assgnmnt: hash-table
  Linestate - current up
  Internal linestate - current up, evaluated up
```

```
Device# show tunnel interface t3
```

```
Tunnel3
  Mode:multi-GRE/IP, Destination UNKNOWN, Source 2001::DB8:1::1
  Application ID 2: WCCP : service group 317 in "Default", ver v2, assgnmnt: hash-table
  Linestate - current up
  Internal linestate - current up, evaluated up
Device#
```

Note that the service group number shown in the examples is the internal tunnel representation of the WCCP service group number. Group 0 is the web-cache service. To determine the dynamic services, subtract 256 from the displayed service group number to convert to the WCCP service group number. For interfaces that are used for redirection, the source address shown is the WCCP router ID.

You can display information about the connected content engines and encapsulation, including software packet counters, by entering the **show adjacency** [*tunnel-interface*] [**encapsulation**] [**detail**] [**internal**] command:

```
Device# show adjacency t0
```

```
Protocol Interface          Address
IP          Tunnel0         2001::DB8:1::1(3)
```

```
Device# show adjacency t0 encapsulation
```

```
Protocol Interface          Address
IPV6          Tunnell       2001:DB8:1::11(2)
  Encap length 48
  60000000000002FFF20010DB801000000
  000000000000000120010DB800010000
  00000000000000110000883E00000000
  Provider: TUNNEL
IPV6          Tunnell       2001:DB8:1::12(2)
  Encap length 48
  60000000000002FFF20010DB801000000
  000000000000000120010DB800010000
  000000000000000120000883E000000000
  Provider: TUNNEL
```

```
Device# show adjacency t0 detail
```

```
Protocol Interface          Address
IPV6          Tunnell       2001:DB8:1::11(2)
                                0 packets, 0 bytes
                                epoch 0
                                sourced in sev-epoch 22
                                Encap length 48
                                60000000000002FFF20010DB801000000
                                000000000000000120010DB800010000
                                000000000000000110000883E000000000
                                Tun endpt
                                Next chain element:
                                punt
```

```
Device# show adjacency t0 internal
```



```

Protocol Interface          Address
IPV6      Tunnel1          2001:DB8:1::11 (2)
                                0 packets, 0 bytes
                                epoch 0
                                sourced in sev-epoch 22
                                Encap length 48
                                6000000000002FFF20010DB801000000
                                000000000000000120010DB800010000
                                00000000000000110000883E00000000
                                Tun endpt
                                Next chain element:
                                  punt
                                  parent oce 0x68C55B00
                                  frame originated locally (Null0)
                                L3 mtu 0
                                Flags (0x2808C6)
                                Fixup disabled
                                HWIDB/IDB pointers 0x200900DC/0x20090D98
                                IP redirect disabled
                                Switching vector: IPv6 midchain adjacency oce
                                Next-hop cannot be inferred
                                IP Tunnel stack to 2001:DB8:1::11 in Default (0x0)

Device#

```

WCCP Bypass Packets

WCCP intercepts IP packets and redirects those packets to a destination other than the destination that is specified in the IP header. Typically the packets are redirected from a web server on the Internet to a web cache that is local to the destination.

Occasionally a web cache cannot manage the redirected packets appropriately and returns the packets unchanged to the originating router. These packets are called bypass packets and are returned to the originating router using either Layer 2 forwarding without encapsulation (L2) or encapsulated in generic routing encapsulation (GRE). The router decapsulates and forwards the packets normally. The VRF associated with the ingress interface (or the global table if there is no VRF associated) is used to route the packet to the destination.

GRE is a tunneling protocol developed by Cisco that encapsulates packet types from a variety of protocols inside IP tunnels, creating a virtual point-to-point link over an IP network.

WCCP Closed Services and Open Services

In applications where packets are intercepted and redirected by a Cisco IOS router to external WCCP client devices, it may be necessary to block the packets for the application when a WCCP client device is not available. This blocking is achieved by configuring a WCCP closed service. When a WCCP service is configured as closed, the packets that fulfill the services, but do not have an active client device, are discarded.

By default, WCCP operates as an open service, wherein communication between clients and servers proceeds normally in the absence of an intermediary device.

The **ip wccp service-list** or the **ipv6 wccp service-list** command can be used for both closed-mode and open-mode services. Use the **service-list** keyword and *service-access-list* argument to register an application protocol type or port number. Use the **mode** keyword to select an open or closed service.

WCCP Outbound ACL Check

When WCCP is enabled for redirection on an ingress interface, the packets are redirected by WCCP and instead egress on an interface other than the destination that is specified in the IP header. The packets are still subject to ACLs configured on the ingress interface. However, redirection can cause the packets to bypass the ACL configured on the original egress interface. Packets that would have been dropped because of the ACL configured on the original egress interface can be sent out on the redirect egress interface, which poses a possible security problem. Enabling the WCCP Outbound ACL check feature ensures that redirected packets are subject to any ACL conditions configured on the original egress interface.

WCCP Service Groups

WCCP is a component of Cisco IOS software that redirects traffic with defined characteristics from its original destination to an alternative destination. The typical application of WCCP is to redirect traffic bound for a remote web server to a local web cache to improve response time and optimize network resource usage.

The nature of the selected traffic for redirection is defined by service groups (see figure below) specified on content engines and communicated to routers by using WCCP. The maximum number of service groups allowed across all VRFs is 256.

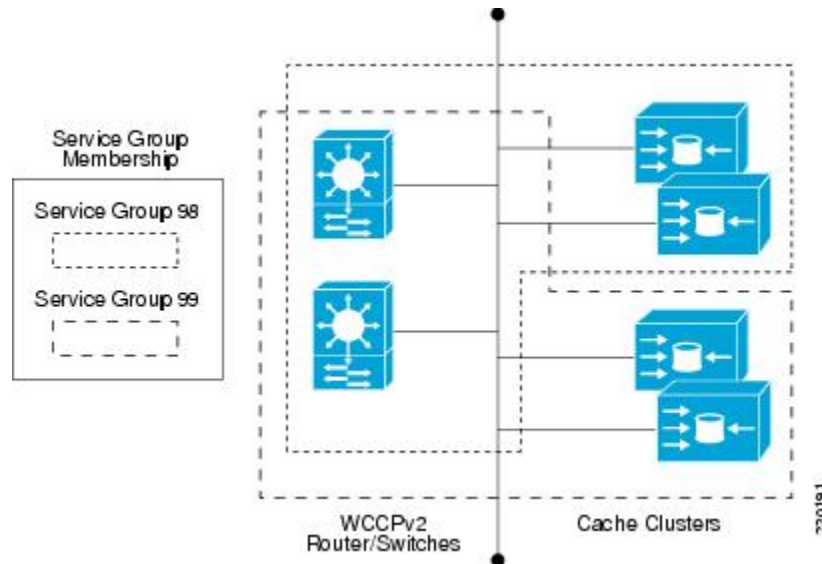
WCCPv2 supports up to 32 routers per service group. Each service group is established and maintained independently.

WCCPv2 uses service groups based on logical redirection services, deployed for intercepting and redirecting traffic. The standard service is web cache, which intercepts TCP port 80 (HTTP) traffic and redirects that traffic to the content engines. This service is referred to as a *well-known service*, because the characteristics of the web cache service are known by both the router and content engines. A description of a well-known service is not required beyond a service identification. To specify the standard web cache service, use the **ip wccp** or the **ipv6 wccp** command with the **web-cache** keyword.



Note More than one service can run on a router at the same time, and routers and content engines can be part of multiple service groups at the same time.

Figure 11: WCCP Service Groups



The dynamic services are defined by the content engines; the content engine instructs the router which protocol or ports to intercept, and how to distribute the traffic. The router itself does not have information on the characteristics of the dynamic service group's traffic, because this information is provided by the first content engine to join the group. In a dynamic service, up to eight ports can be specified within a single protocol.

Cisco Content Engines, for example, use dynamic service 99 to specify a reverse-proxy service. However, other content engine devices may use this service number for some other service.

WCCP—Check All Services

An interface may be configured with more than one WCCP service. When more than one WCCP service is configured on an interface, the precedence of a service depends on the relative priority of the service compared to the priority of the other configured services. Each WCCP service has a priority value as part of its definition. When an interface is configured with more than one WCCP service, the precedence of the packets is matched against service groups in priority order.



Note The priority of a WCCP service group cannot be configured via Cisco IOS software.

With the **ip wccp check services all** or the **ipv6 wccp check services all** command, WCCP can be configured to check all configured services for a match and perform redirection for those services if appropriate. The caches to which packets are redirected can be controlled by a redirect ACL and by the service priority. The **ip wccp check services all** commands must be configured at global level to support multiple WCCP services.

If no WCCP services are configured with a redirect ACL, the services are considered in priority order until a service is found that matches the IP packet. If no services match the packet, the packet is not redirected. If a service matches the packet and the service has a redirect ACL configured, then the IP packet will be checked against the ACL. If the packet is rejected by the ACL, the packet will not be passed down to lower priority services unless the **ip wccp check services all** or the **ipv6 wccp check services all** command is configured. When the **ip wccp check services all** or the **ipv6 wccp check services all** command is configured, WCCP

will continue to attempt to match the packet against any remaining lower priority services configured on the interface.

WCCP Interoperability with NAT

To redirect traffic using WCCP to a router running WAAS software that is also configured with NAT, enable the **ip nat inside** or the **ipv6 nat inside** command on the WAAS interface. If you are not able to configure the **ip nat inside** or the **ipv6 nat inside** command on the WAAS interface, disable Cisco Express Forwarding. You must also update the WCCP redirect ACL to include a private address to ensure that pretranslated traffic is redirected.

WCCP—Configurable Router ID Overview

WCCP uses a router ID in its control messages that a WCCP client can use to uniquely identify a particular WCCP server. The router ID is an IP address and is used as the source address of any WCCP-generated Generic Routing Encapsulation (GRE) frames. Prior to the WCCP—Configurable Router ID feature, WCCP selected a router ID using an automatic mechanism; the highest reachable IP address on the system (or the highest loopback IP address, if there is one) was used as the WCCP router ID. The highest IP address on the system is not always the best choice as the router ID or as the source address of GRE frames. A change in addressing information on the system may cause the WCCP router ID to change unexpectedly. During this changeover period, WCCP clients briefly advertise the existence of two routers (the old router ID and the new router ID) and GRE frames are sourced from a different address.

The WCCP—Configurable Router ID feature enables you to define a WCCP source interface from which the router ID will be obtained. The IP address of this configured source interface is then used as the preferred WCCP router ID and WCCP GRE source address. When a WCCP router ID is manually configured, the router ID does not change when another IP address is added to the system. The router ID changes only when a new router ID is manually configured using the **ip wccp source- interface** or the **ipv6 wccp source- interface** command, or when the address on the manually configured interface is no longer valid.

WCCP Troubleshooting Tips

CPU usage may be very high when WCCP is enabled. The WCCP counters enable a determination of the bypass traffic directly on the router and can indicate whether the cause is high CPU usage due to enablement of WCCP. In some situations, 10 percent bypass traffic may be normal; in other situations, 10 percent may be high. However, any figure above 25 percent should prompt a closer investigation of what is occurring in the web cache.

If the counters suggest that the level of bypass traffic is high, the next step is to examine the bypass counters in the content engine and determine why the content engine is choosing to bypass the traffic. You can log in to the content engine console and use the CLI to investigate further. The counters allow you to determine the percent of traffic being bypassed.

You can use the **clear ipv6 wccp service-id** command to remove the IPv6 WCCP statistics (counts) maintained on the router for a particular service.

You can use the **clear wccp** command to remove all (IPv4 and IPv6) WCCP statistics (counts) maintained on the router for a particular service.

You can use the **show ipv6 wccp** command to display the IPv6 WCCP global statistics (counts).

You can use the **show wccp** command to display all (IPv4 and IPv6) WCCP global statistics (counts).

How to Configure WCCPv2—IPv6 Support

Configuring a General WCCPv2—IPv6 Session

Perform this task to configure a general IPv6 WCCPv2 session.

Until you configure a WCCP service using the **ipv6 wccp {web-cache | service-number}** global configuration command, WCCP is disabled on the router. The first use of a form of the **ipv6 wccp** command enables WCCP. By default WCCPv2 is used for services.

Using the **ipv6 wccp web-cache password** command, you can set a password for a router and the content engines in a service group. MD5 password security requires that each router and content engine that wants to join a service group be configured with the service group password. The password can be up to eight characters in length. Each content engine or router in the service group will authenticate the security component in a received WCCP packet immediately after validating the WCCP message header. Packets failing authentication will be discarded.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 wccp [vrf vrf-name] source-interface source-interface**
4. **ipv6 wccp [vrf vrf-name] { web-cache | service-number} [group-address group-address] [redirect-list access-list] [group-list access-list] [password password [0 | 7]]**
5. **interface type number**
6. **ipv6 wccp [vrf vrf-name] {web-cache | service-number} redirect {out | in}**
7. **exit**
8. **interface type number**
9. **ipv6 wccp redirect exclude in**

DETAILED STEPS

| | Command or Action | Purpose |
|--------|--------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------|
| Step 1 | enable Example: Device> enable | Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted. |
| Step 2 | configure terminal Example: Device# configure terminal | Enters global configuration mode. |
| Step 3 | ipv6 wccp [vrf vrf-name] source-interface source-interface Example: | Configures a preferred WCCP router ID. |

| | Command or Action | Purpose |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | Device(config)# ipv6 wccp source-interface GigabitEthernet 0/0/0 | |
| Step 4 | ipv6 wccp [<i>vrf vrf-name</i>] { web-cache <i>service-number</i> } [group-address <i>group-address</i>] [redirect-list <i>access-list</i>] [group-list <i>access-list</i>] [password <i>password</i> [0 7]] Example: Device(config)# ipv6 wccp web-cache password password1 | Specifies a web-cache or dynamic service to enable on the router, specifies a VRF-name to associate with the service group, specifies the IP multicast address used by the service group, specifies any access lists to use, specifies whether to use MD5 authentication, and enables the WCCP service. |
| Step 5 | interface <i>type number</i> Example: Device(config)# interface GigabitEthernet 0/0/0 | Targets an interface number for which the web cache service will run, and enters interface configuration mode. |
| Step 6 | ipv6 wccp [<i>vrf vrf-name</i>] { web-cache <i>service-number</i> } redirect { out in } Example: Device(config-if)# ipv6 wccp web-cache redirect in | Enables packet redirection on an outbound or inbound interface using WCCP. <ul style="list-style-type: none"> As indicated by the out and in keyword options, redirection can be specified for outbound interfaces or inbound interfaces. |
| Step 7 | exit Example: Device(config-if)# exit | Exits interface configuration mode. |
| Step 8 | interface <i>type number</i> Example: Device(config)# interface GigabitEthernet 0/2/0 | Targets an interface number on which to exclude traffic for redirection, and enters interface configuration mode. |
| Step 9 | ipv6 wccp redirect exclude in Example: î Device(config-if)# ipv6 wccp redirect exclude in | (Optional) Excludes traffic on the specified interface from redirection. |

Configuring Services for WCCPv2—IPv6

Perform this task to specify the number of service groups for WCCP, to configure a service group as a closed or open service, and to optionally specify a check of all services.

SUMMARY STEPS

1. enable

2. **configure terminal**
3. Enter one of the following commands:
 - **ipv6 wccp [vrf vrf-name] service-number [service-list service-access-list mode {open | closed}]**
 - **ipv6 wccp [vrf vrf-name] web-cache mode {open | closed}**
4. **ipv6 wccp check services all**
5. **ipv6 wccp [vrf vrf-name] {web-cache | service-number}**
6. **exit**

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Step 1 | enable Example: <pre>Device> enable</pre> | Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted. |
| Step 2 | configure terminal Example: <pre>Device# configure terminal</pre> | Enters global configuration mode. |
| Step 3 | Enter one of the following commands: <ul style="list-style-type: none"> • ipv6 wccp [vrf vrf-name] service-number [service-list service-access-list mode {open closed}] • ipv6 wccp [vrf vrf-name] web-cache mode {open closed} Example: <pre>Device(config)# ipv6 wccp 90 service-list 120 mode closed</pre> or <pre>Device(config)# ipv6 wccp web-cache mode closed</pre> | Configures a dynamic WCCP service as closed or open. or Configures a web-cache service as closed or open. <p>Note When configuring the web-cache service as a closed service, you cannot specify a service access list.</p> <p>Note When configuring a dynamic WCCP service as a closed service, you must specify a service access list.</p> |
| Step 4 | ipv6 wccp check services all Example: <pre>Device(config)# ipv6 wccp check services all</pre> | (Optional) Enables a check of all WCCP services. <ul style="list-style-type: none"> • Use this command to configure WCCP to check the other configured services for a match and perform redirection for those services if appropriate. The caches to which packets are redirected can be controlled by the redirect ACL and not just the service description. <p>Note The ipv6 wccp check services all command is a global WCCP command that applies to all services and is not associated with a single service.</p> |

| | Command or Action | Purpose |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Step 5 | ipv6 wccp [<i>vrf vrf-name</i>] { <i>web-cache</i> <i>service-number</i> } Example: <pre>Device(config)# ipv6 wccp 201</pre> | Specifies the WCCP service identifier. <ul style="list-style-type: none"> You can specify the standard web-cache service or a dynamic service number from 0 to 255. The maximum number of services that can be specified is 256. |
| Step 6 | exit Example: <pre>Device(config)# exit</pre> | Exits to privileged EXEC mode. |

Registering a Router to a Multicast Address for WCCPv2— IPv6

If you decide to use the multicast address option for your service group, you must configure the router to listen for the multicast broadcasts on an interface.

For network configurations where redirected traffic needs to traverse an intervening router, the router being traversed must be configured to perform IP multicast routing. You must configure the following two components to enable traversal over an intervening router:

- Enable IP multicast routing using the **ipv6 multicast-routing** global configuration command.
- Enable the interfaces to which the cache engines will connect to receive multicast transmissions using the **ipv6 wccp group-listen** interface configuration command.

SUMMARY STEPS

- enable**
- configure terminal**
- ipv6 multicast-routing** [*vrf vrf-name*] [*distributed*]
- ipv6 wccp** [*vrf vrf-name*] {*web-cache* | *service-number*} **group-address** *multicast-address*
- interface** *type number*
- ip pim** {*sparse-mode* | *sparse-dense-mode* | *dense-mode* [*proxy-register* {*list access-list* | *route-map map-name*}]}
- ipv6 wccp** [*vrf vrf-name*] {*web-cache* | *service-number*} **group-listen**

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------|
| Step 1 | enable Example: <pre>Device> enable</pre> | Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted. |

| | Command or Action | Purpose |
|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Step 2 | configure terminal Example: <pre>Device# configure terminal</pre> | Enters global configuration mode. |
| Step 3 | ipv6 multicast-routing [vrf vrf-name] [distributed] Example: <pre>Device(config)# ipv6 multicast-routing</pre> | Enables IP multicast routing. |
| Step 4 | ipv6 wccp [vrf vrf-name] {web-cache service-number} group-address multicast-address Example: <pre>Device(config)# ipv6 wccp 99 group-address FF15::8000:1</pre> | Specifies the multicast address for the service group. |
| Step 5 | interface type number Example: <pre>Device(config)# interface ethernet 0/0</pre> | Enables the interfaces to which the content engines will connect to receive multicast transmissions for which the web cache service will run, and enters interface configuration mode. |
| Step 6 | ip pim {sparse-mode sparse-dense-mode dense-mode [proxy-register {list access-list route-map map-name}]} Example: <pre>Device(config-if)# ip pim dense-mode</pre> | (Optional) Enables Protocol Independent Multicast (PIM) on an interface. Note To ensure correct operation of the ipv6 wccp group-listen command, you must enter the ip pim command in addition to the ipv6 wccp group-listen command. |
| Step 7 | ipv6 wccp [vrf vrf-name] {web-cache service-number} group-listen Example: <pre>Device(config-if)# ipv6 wccp 99 group-listen</pre> | Configures an interface to enable or disable the reception of IP multicast packets for WCCP. |

Using Access Lists for WCCPv2—IPv6 Service Group

Perform this task to configure the router to use an access list to determine which traffic should be directed to which content engines.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list access-list-number remark remark**
4. **access-list access-list-number permit {source [source-wildcard] | any} [log]**

5. **access-list** *access-list-number* **remark** *remark*
6. **access-list** *access-list-number* **deny** {*source* [*source-wildcard*] | **any**} [**log**]
7. Repeat some combination of Steps 3 through 6 until you have specified the sources on which you want to base your access list.
8. **ipv6 wccp** [*vrf vrf-name*] **web-cache group-list** *access-list*
9. **ipv6 wccp** [*vrf vrf-name*] **web-cache redirect-list** *access-list*

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Step 1 | enable Example: Device> enable | Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted. |
| Step 2 | configure terminal Example: Device# configure terminal | Enters global configuration mode. |
| Step 3 | access-list <i>access-list-number</i> remark <i>remark</i> Example: Device(config)# access-list 1 remark Give access to user1 | (Optional) Adds a user-friendly comment about an access list entry. <ul style="list-style-type: none"> • A remark of up to 100 characters in length can precede or follow an access list entry. |
| Step 4 | access-list <i>access-list-number</i> permit { <i>source</i> [<i>source-wildcard</i>] any } [log] Example: Device(config)# access-list 1 permit 172.16.5.22 0.0.0.0 | Creates an access list that enables or disables traffic redirection to the cache engine and permits the specified source based on a source address and wildcard mask. <ul style="list-style-type: none"> • Every access list needs at least one permit statement; it does not need to be the first entry. • Standard IP access lists are numbered 1 to 99 or 1300 to 1999. • If the <i>source-wildcard</i> string is omitted, a wildcard mask of 0.0.0.0 is assumed, meaning match on all bits of the source address. • Optionally use the keyword any as a substitute for the <i>source source-wildcard</i> to specify the source and source wildcard of 0.0.0.0 255.255.255.255. • In this example, host 172.16.5.22 is allowed to pass the access list. |
| Step 5 | access-list <i>access-list-number</i> remark <i>remark</i> Example: | (Optional) Adds a user-friendly comment about an access list entry. <ul style="list-style-type: none"> • A remark of up to 100 characters can precede or follow an access list entry. |

| | Command or Action | Purpose |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | Device(config)# access-list 1 remark Give access to user1 | |
| Step 6 | <p>access-list <i>access-list-number</i> deny {<i>source</i> [<i>source-wildcard</i>] any} [log]</p> <p>Example:</p> <pre>Device(config)# access-list 1 deny 172.16.7.34 0.0.0.0</pre> | <p>Denies the specified source based on a source address and wildcard mask.</p> <ul style="list-style-type: none"> • If the <i>source-wildcard</i> string is omitted, a wildcard mask of 0.0.0.0 is assumed, meaning match on all bits of the source address. • Optionally use the abbreviation any as a substitute for the <i>source source-wildcard</i> to specify the source and source wildcard of 0.0.0.0 255.255.255.255. • In this example, host 172.16.7.34 is denied passing the access list. |
| Step 7 | Repeat some combination of Steps 3 through 6 until you have specified the sources on which you want to base your access list. | Remember that all sources not specifically permitted are denied by an implicit deny statement at the end of the access list. |
| Step 8 | <p>ipv6 wccp [<i>vrf vrf-name</i>] web-cache group-list <i>access-list</i></p> <p>Example:</p> <pre>Device(config) ipv6 wccp web-cache group-list 1</pre> | Indicates to the router from which IP addresses of content engines to accept packets. |
| Step 9 | <p>ipv6 wccp [<i>vrf vrf-name</i>] web-cache redirect-list <i>access-list</i></p> <p>Example:</p> <pre>Router(config)# ipv6 wccp web-cache redirect-list 1</pre> | (Optional) Disables caching for certain clients. |

Enabling the WCCP—IPv6 Outbound ACL Check



Note When all redirection is performed in the hardware, the mode of redirection will change when outbound ACL checking is enabled. The first packet is switched in software to allow the extra ACL check to be performed before a shortcut is installed.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 wccp** [*vrf vrf-name*] {**web-cache** | *service-number*} [**group-address** *multicast-address*] [**redirect-list** *access-list*] [**group-list** *access-list*] [**password** *password*]
4. **ipv6 wccp check acl outbound**

5. exit

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| Step 1 | enable Example: Device> enable | Enables privileged EXEC mode. • Enter your password if prompted. |
| Step 2 | configure terminal Example: Device# configure terminal | Enters global configuration mode. |
| Step 3 | ipv6 wccp [<i>vrf vrf-name</i>] { <i>web-cache</i> <i>service-number</i> } [<i>group-address multicast-address</i>] [<i>redirect-list access-list</i>] [<i>group-list access-list</i>] [<i>password password</i>] Example: Device(config)# ipv6 wccp web-cache | Enables support for a Cisco content engine service group or any content engine service group and configures a redirect ACL list or group ACL. |
| Step 4 | ipv6 wccp check acl outbound Example: Device(config)# ipv6 wccp check acl outbound | Checks the access control list (ACL) for egress interfaces for packets redirected by WCCP. |
| Step 5 | exit Example: Device(config)# exit | Exits global configuration. |

Enabling WCCPv2—IPv6 Interoperability with NAT

SUMMARY STEPS

1. enable
2. configure terminal
3. interface *type number*
4. ipv6 nat inside
5. ipv6 wccp *service-number* redirect in
6. exit
7. interface *type number*
8. ipv6 nat outside
9. ipv6 wccp *service-number* redirect in
10. exit

11. `interface type number`
12. `ipv6 nat inside`
13. `ipv6 wccp redirect exclude in`

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|---------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Step 1 | enable Example: Device> enable | Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted. |
| Step 2 | configure terminal Example: Device# configure terminal | Enters global configuration mode. |
| Step 3 | interface type number Example: Device(config)# interface ethernet 1 | Specifies an interface on which to enable NAT and enters interface configuration mode. <ul style="list-style-type: none"> • This is the LAN-facing interface. |
| Step 4 | ipv6 nat inside Example: Device(config-if)# ipv6 nat inside | Designates that traffic originating from or destined for the interface is subject to NAT and indicates that the interface is connected to the inside network (the network subject to NAT translation). |
| Step 5 | ipv6 wccp service-number redirect in Example: Device(config-if)# ipv6 wccp 61 redirect in | Enables packet redirection on an inbound interface using WCCP. |
| Step 6 | exit Example: Device(config-if)# exit | Exits interface configuration mode and returns to global configuration mode. |
| Step 7 | interface type number Example: Device(config)# interface ethernet 2 | Specifies an interface on which to enable NAT and enters interface configuration mode. <ul style="list-style-type: none"> • This is the WAN-facing interface. |
| Step 8 | ipv6 nat outside Example: Device(config-if)# ipv6 nat outside | Designates that traffic originating from or destined for the interface is subject to NAT and indicates that the interface is connected to the outside network. |

| | Command or Action | Purpose |
|----------------|-------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Step 9 | ipv6 wccp service-number redirect in Example: Device(config-if)# ipv6 wccp 62 redirect in | Enables packet redirection on an inbound interface using WCCP. |
| Step 10 | exit Example: Device(config-if)# exit | Exits interface configuration mode and returns to global configuration mode. |
| Step 11 | interface type number Example: Device(config)# interface ethernet 3 | Specifies an interface on which to enable NAT and enters interface configuration mode. <ul style="list-style-type: none"> • This is the WAAS-facing interface. |
| Step 12 | ipv6 nat inside Example: Device(config-if)# ipv6 nat inside | Designates that traffic originating from or destined for the interface is subject to NAT and indicates that the interface is connected to the inside network (the network subject to NAT translation). |
| Step 13 | ipv6 wccp redirect exclude in Example: Device(config-if)# ipv6 wccp redirect exclude in | Configures an interface to exclude packets received on an interface from being checked for redirection. |

Verifying and Monitoring WCCPv2—IPv6 Configuration Settings

SUMMARY STEPS

1. enable
2. show ipv6 wccp [vrf vrf-name] [service-number | web-cache] [detail | view]
3. show ipv6 interface
4. more system:running-config

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Step 1 | enable Example: Device> enable | Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted. |
| Step 2 | show ipv6 wccp [vrf vrf-name] [service-number web-cache] [detail view] Example: | (Optional) Displays global information related to WCCP, including the protocol version currently running, the number of content engines in the router service group, which content |

| | Command or Action | Purpose |
|---------------|--------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | Device# show ipv6 wccp 24 detail | <p>engine group is allowed to connect to the router, and which access list is being used. The argument and keywords are as follows:</p> <ul style="list-style-type: none"> • <i>service-number</i>—(Optional) Dynamic number of the web-cache service group being controlled by the content engine. The range is from 0 to 99. For web caches that use Cisco Content Engines, the reverse proxy service is indicated by a value of 99. • web-cache—(Optional) Statistics for the web-cache service. • detail—(Optional) Other members of a particular service group or web cache that have or have not been detected. • view—(Optional) Information about a router or all web caches. |
| Step 3 | show ipv6 interface Example: Device# show ipv6 interface | (Optional) Displays status about whether any ip wccp redirection commands are configured on an interface; for example, “Web Cache Redirect is enabled / disabled.” |
| Step 4 | more system:running-config Example: Device# more system:running-config | (Optional) Displays contents of the currently running configuration file (equivalent to the show running-config command). |

Configuration Examples for WCCPv2—IPv6 Support

Example: Configuring a General WCCPv2—IPv6 Session

```

Device# configure terminal
Device(config)# ipv6 wccp web-cache password password1
Device(config)# ipv6 wccp source-interface GigabitEthernet 0/1/0
Device(config)# ipv6 wccp check services all
    Configures a check of all WCCP services.
Device(config)# interface GigabitEthernet 0/1/0
Device(config-if)# ipv6 wccp web-cache redirect in
Device(config-if)# exit
Device(config)# interface GigabitEthernet 0/2/0
Device(config-if)# ipv6 wccp redirect exclude in
Device(config-if)# exit

```

Example: WCCPv2—IPv6—Setting a Password for a Router and Content Engines

```
Device# configure terminal
Device(config)# ipv6 wccp web-cache password password1
```

Example: WCCPv2—IPv6—Configuring a Web Cache Service

```
Device# configure terminal
Device(config)# ipv6 wccp web-cache
Device(config)# interface GigabitEthernet 0/1/0
Device(config-if)# ipv6 wccp web-cache redirect in
Device(config-if)# exit
Device# copy running-config startup-config
```

The following example shows how to configure a session in which redirection of HTTP traffic arriving on Gigabit Ethernet interface 0/1/0 is enabled:

```
Device# configure terminal
Device(config)# interface GigabitEthernet 0/1/0
Device(config-if)# ipv6 wccp web-cache redirect in
Device(config-if)# exit
Device# show ip interface GigabitEthernet 0/1/0
.
.
.
WCCP Redirect inbound is enabled
WCCP Redirect exclude is disabled
.
.
.
```

Example: WCCPv2—IPv6—Running a Reverse Proxy Service

The following example assumes that you are configuring a service group using Cisco cache engines, which use dynamic service 99 to run a reverse proxy service:

```
Device# configure terminal
Device(config)# ipv6 wccp 99
Device(config)# interface GigabitEthernet 0/1/0
Device(config-if)# ipv6 wccp 99 redirect out
```

Example: WCCPv2—IPv6—Registering a Router to a Multicast Address

```
Device# configure terminal
Device(config)# ipv6 wccp web-cache group-address 224.1.1.100
Device(config)# interface gigabitethernet 0/1/0
Device(config-if)# ipv6 wccp web cache group-listen
```

The following example shows a device configured to run a reverse proxy service, using the multicast address of 224.1.1.1. Redirection applies to packets outgoing via Gigabit Ethernet interface 0/1/0:


```
Device# configure terminal
Device(config)# ipv6 wccp 99 group-address 224.1.1.1
Device(config)# interface gigabitethernet 0/1/0
Device(config-if)# ipv6 wccp 99 redirect out
```

Example: WCCPv2—IPv6—Using Access Lists for a WCCPv2 IPv6 Service Group

To achieve better security, you can use a standard access list to notify the device which IP addresses are valid addresses for a content engine attempting to register with the current device. The following example shows a standard access list configuration session where the access list number is 10 for some sample hosts:

```
Device(config)# access-list 10 permit host 10.1.1.1
Device(config)# access-list 10 permit host 10.1.1.2
Device(config)# access-list 10 permit host 10.1.1.3
Device(config)# ipv6 wccp web-cache group-list 10
```

To disable caching for certain clients, servers, or client/server pairs, you can use WCCP access lists. The following example shows that any requests coming from 10.1.1.1 to 10.3.1.1 will bypass the cache, and that all other requests will be serviced normally:

```
Device(config)# ipv6 wccp web-cache redirect-list 120
Device(config)# access-list 120 deny tcp host 10.1.1.1 any
Device(config)# access-list 120 deny tcp any host 10.3.1.1
Device(config)# access-list 120 permit ip any any
```

The following example configures a device to redirect web-related packets received via Gigabit Ethernet interface 0/1/0, destined to any host except 209.165.200.224:

```
Device(config)# access-list 100 deny ip any host 209.165.200.224
Device(config)# access-list 100 permit ip any any
Device(config)# ipv6 wccp web-cache redirect-list 100
Device(config)# interface gigabitethernet 0/1/0
Device(config-if)# ipv6 wccp web-cache redirect in
```

Example: WCCPv2—IPv6—Configuring Outbound ACL Check

The following configuration example shows that the access list prevents traffic from network 10.0.0.0 leaving Gigabit Ethernet interface 0/1/0. Because the outbound ACL check is enabled, WCCP does not redirect that traffic. WCCP checks packets against the ACL before they are redirected.

```
Device(config)# ipv6 wccp web-cache
Device(config)# ipv6 wccp check acl outbound
Device(config)# interface gigabitethernet 0/1/0
Device(config-if)# ip access-group 10 out
Device(config-if)# exit
Device(config)# ipv6 wccp web-cache redirect-list redirect-out
Device(config)# access-list 10 deny 10.0.0.0 0.255.255.255
Device(config)# access-list 10 permit any
```

If the outbound ACL check is disabled, the HTTP packets from network 10.0.0.0 would be redirected to a web cache. Users with that network address could retrieve web pages even though the network administrator wanted to prevent it.

Example: WCCPv2—IPv6—Enabling WCCP Interoperability with NAT

```

Device(config)# interface ethernet1 ! This is the LAN-facing interface
Device(config-if)# ipv6 nat inside
Device(config-if)# ipv6 wccp 61 redirect in
Device(config-if)# exit
Device(config)# interface ethernet2 ! This is the WAN-facing interface
Device(config-if)# ipv6 nat outside
Device(config-if)# ipv6 wccp 62 redirect in
Device(config-if)# exit
Device(config)# interface ethernet3 ! This is the WAAS-facing interface
Device(config-if)# ipv6 nat inside
Device(config-if)# ipv6 wccp redirect exclude in

```

Example: WCCPv2—IPv6—Verifying WCCP Settings

The following example shows how to verify your configuration changes by using the **more system:running-config** command in privileged EXEC mode. The following example shows that both the web cache service and dynamic service 99 are enabled on the device:

```

Device# more system:running-config

Building configuration...
Current configuration:
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service udp-small-servers
service tcp-small-servers
!
hostname router4
!
enable secret 5 $1$nSVy$faliJsVQXVPW.KuCxZNTTh1
enable password password1
!
ip subnet-zero
ipv6 wccp web-cache
ipv6 wccp 99
ip domain-name cisco.com
ip name-server 10.1.1.1
ip name-server 10.1.1.2
ip name-server 10.1.1.3
!
!
!
interface GigabitEthernet0/1/1
ip address 10.3.1.2 255.255.255.0
no ip directed-broadcast
ipv6 wccp web-cache redirect in
ipv6 wccp 99 redirect in
no ip route-cache
no ip mroute-cache
!
interface GigabitEthernet0/1/0
ip address 10.4.1.1 255.255.255.0
no ip directed-broadcast

```

```

ipv6 wccp 99 redirect in
no ip route-cache
no ip mroute-cache
!
interface Serial0
no ip address
no ip directed-broadcast
no ip route-cache
no ip mroute-cache
shutdown
!
interface Serial1
no ip address
no ip directed-broadcast
no ip route-cache
no ip mroute-cache
shutdown
!
ip default-gateway 10.3.1.1
ip classless
ip route 0.0.0.0 0.0.0.0 10.3.1.1
no ip http server
!
!
!
line con 0
transport input none
line aux 0
transport input all
line vty 0 4
password password1
login
!
end

```

The following example shows how to display global statistics related to WCCP:

```
Device# show ipv6 wccp web-cache detail
```

```

WCCP Client information:
WCCP Client ID:      10.1.1.2
Protocol Version:    2.0
State:               Usable
Redirection:         L2
Packet Return:       L2
Packets Redirected:  0
Connect Time:        00:20:34
Assignment:          MASK
Mask  SrcAddr  DstAddr  SrcPort  DstPort
----  -
0000: 0x00000000 0x00001741 0x0000  0x0000
Value SrcAddr  DstAddr  SrcPort  DstPort  CE-IP
----  -
0000: 0x00000000 0x00000000 0x0000  0x0000 0x3C010102 (10.1.1.2)
0001: 0x00000000 0x00000001 0x0000  0x0000 0x3C010102 (10.1.1.2)
0002: 0x00000000 0x00000040 0x0000  0x0000 0x3C010102 (10.1.1.2)
0003: 0x00000000 0x00000041 0x0000  0x0000 0x3C010102 (10.1.1.2)
0004: 0x00000000 0x00000100 0x0000  0x0000 0x3C010102 (10.1.1.2)
0005: 0x00000000 0x00000101 0x0000  0x0000 0x3C010102 (10.1.1.2)
0006: 0x00000000 0x00000140 0x0000  0x0000 0x3C010102 (10.1.1.2)

```

For more information about the **show ip wccp web-cache** command, see the *Cisco IOS IP Application Services Command Reference* document.

Additional References

Related Documents

| Related Topic | Document Title |
|-----------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Cisco IOS commands | <i>Cisco IOS Master Commands List, All Releases</i> |
| IP addressing and services commands and configuration tasks | <ul style="list-style-type: none"> • <i>Cisco IOS IP Addressing Services Configuration Guide</i> • <i>Cisco IOS IP Addressing Services Command Reference</i> |
| WCCP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples | <i>Cisco IOS IP Application Services Command Reference</i> |

Technical Assistance

| Description | Link |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|
| The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password. | http://www.cisco.com/cisco/web/support/index.html |

Feature Information for WCCPv2—IPv6 Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

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

Table 16: Feature Information for WCCPv2 —IPv6 Support

| Feature Name | Releases | Feature Information |
|---------------------|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| WCCPv2—IPv6 Support | 15.1(1)SY1 15.2(3)T | <p>This feature introduces support for Web Cache Communication Protocol version 2 (WCCPv2) in an IPv6 environment.</p> <p>WCCP is a Cisco-developed content-routing technology that intercepts IP packets and redirects those packets to a destination other than that specified in the IP packet.</p> <p>Multiple routers can use WCCPv2 to service a content engine cluster. In WCCPv1, only one router could redirect content requests to a cluster.</p> <p>The following commands were added: clear ipv6 wccp, clear wccp, debug ipv6 wccp, debug wccp, ipv6 wccp, ipv6 wccp check acl outbound, ipv6 wccp check services all, ipv6 wccp group-listen, ipv6 wccp redirect, ipv6 wccp redirect exclude in, ipv6 wccp source-interface, show ipv6 wccp, show ipv6 wccp global counters, show wccp, show wccp global counters, show platform software wccp <i>service-number</i> ipv6 counters, show platform software wccp rp active <i>service-number</i> ipv6, show platform software wccp fp active <i>service-number</i> ipv6, show platform hardware qfp active feature wccp service id <i>service-number</i> ipv6.</p> |

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco's trademarks can be found at www.cisco.com/go/trademarks. Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1005R)

