IP Application Services Configuration Guide, Cisco IOS XE Release 2

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 NONINFRINGEMENT 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.

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. 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. (1110R)

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.

© 2011 Cisco Systems, Inc. All rights reserved.

CONTENTS

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 Cisco IOS XE 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

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 document.

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 Enhanced Object Tracking

- Benefits of Enhanced Object Tracking, page 4

Feature Design of Enhanced Object Tracking

Enhanced Object Tracking 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 command-line interface (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 also 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 or not the threshold has been met.
- Boolean “and” function--When a tracked list has been assigned a Boolean “and” function, it means that 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, it means that at least one object defined within a subset must be in an up state so that the tracked object can become up.

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
Open Shortest Path First (OSPF)	1
Intermediate System-to-Intermediate System (IS-IS)	10

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. Cisco IOS XE software uses IP SLAs to collect real-time metrics such as response

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

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 *Cisco IOS XENetwork Management Configuration Guide* for more information on how EOT works with EEM.

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, page 5
- Tracking the IP-Routing State of an Interface, page 7
- Tracking IP-Route Reachability, page 8
- Tracking the Threshold of IP-Route Metrics, page 10
- Tracking the State of an IP SLAs Operation, page 12
- Tracking the Reachability of an IP SLAs IP Host, page 14

- Configuring a Tracked List and Boolean Expression, page 15
- Configuring a Tracked List and Threshold Weight, page 17
- Configuring a Tracked List and Threshold Percentage, page 18
- Configuring Track List Defaults, page 20

Tracking the Line-Protocol State of an Interface

Perform this task to track the line-protocol state of an interface.

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. See the Tracking the IP-Routing State of an Interface, page 7 section for more information.

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**track timer interface** *seconds* **Example:** `Router(config)# track timer interface 5`	(Optional) Specifies the interval in which the tracking process polls the tracked object. • The default interval that the tracking process polls interface objects is 1 second.

	Command or Action	Purpose
Step 4	**track** *object-number* **interface** *type number* **line-protocol** **Example:** `Router(config)# track 3 interface GigabitEthernet 1/0/0 line-protocol`	Tracks the line-protocol state of an interface and enters tracking configuration mode.
Step 5	**delay** {**up** *seconds* [**down** *seconds*] | [**up** *seconds*] **down** *seconds*} **Example:** `Router(config-track)# delay up 30`	(Optional) Specifies a period of time (in seconds) to delay communicating state changes of a tracked object.
Step 6	**end** **Example:** `Router(config-track)# end`	Exits to privileged EXEC mode.
Step 7	**show track** *object-number* **Example:** `Router# show track 3`	(Optional) Displays tracking information. • 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:

```
Router# show track 3

Track 3
   Interface GigabitEthernet1/0/0 line-protocol
   Line protocol is Up
     1 change, last change 00:00:05
   Tracked by:
     HSRP GigabitEthernet0/0/0 3
```

Tracking the IP-Routing State of an Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **track timer interface** *seconds*
4. **track** *object-number* **interface** *type number* **ip routing**
5. **delay** {**up** *seconds* [**down** *seconds*] | [**up** *seconds*] **down** *seconds*}
6. **end**
7. **show track** *object-number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**track timer interface** *seconds* **Example:** `Router(config)# track timer interface 5`	(Optional) Specifies the interval in which the tracking process polls the tracked object. • The default interval that the tracking process polls interface objects is 1 second.
Step 4	**track** *object-number* **interface** *type number* **ip routing** **Example:** `Router(config)# track 1 interface GigabitEthernet 1/0/0 ip routing`	Tracks the IP-routing state of an interface and enters tracking configuration mode. • IP-route tracking tracks an IP route in the routing table and the ability of an interface to route IP packets.

	Command or Action	Purpose
Step 5	**delay** {**up** *seconds* [**down** *seconds*] | [**up** *seconds*] **down** *seconds*} **Example:** `Router(config-track)# delay up 30`	(Optional) Specifies a period of time (in seconds) to delay communicating state changes of a tracked object.
Step 6	**end** **Example:** `Router(config-track)# end`	Returns to privileged EXEC mode.
Step 7	**show track** *object-number* **Example:** `Router# show track 1`	Displays tracking information. • Use this command to verify the configuration.

Example

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

```
Router# show track 1

Track 1
   Interface GigabitEthernet1/0/0 ip routing
   IP routing is Up
     1 change, last change 00:01:08
   Tracked by:
     HSRP GigabitEthernet0/0/0 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*
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:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**track timer ip route** *seconds* **Example:** `Router(config)# track timer ip route 20`	(Optional) Specifies the interval in which the tracking process polls the tracked object. • The default interval that the tracking process polls IP-route objects is 15 seconds.
Step 4	**track** *object-number* **ip route** *ip-address*/*prefix-length* **reachability** **Example:** `Router(config)# track 4 ip route 10.16.0.0/16 reachability`	Tracks the reachability of an IP route and enters tracking configuration mode.
Step 5	**delay** {**up** *seconds* [**down** *seconds*] | [**up** *seconds*] **down** *seconds*} **Example:** `Router(config-track)# delay up 30`	(Optional) Specifies a period of time (in seconds) to delay communicating state changes of a tracked object.
Step 6	**ip vrf** *vrf-name* **Example:** `Router(config-track)# ip vrf VRF2`	(Optional) Configures a VPN routing and forwarding (VRF) table.
Step 7	**end** **Example:** `Router(config-track)# end`	Returns to privileged EXEC mode.

	Command or Action	Purpose
Step 8	**show track** *object-number* **Example:** `Router# show track 4`	(Optional) Displays tracking information. • 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:

```
Router# 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 GigabitEthernet0/1
   Tracked by:
     HSRP GigabitEthernet1/0/3 4
```

Tracking the Threshold of IP-Route Metrics

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **track timer ip route** *seconds*
4. **track resolution ip route** {**eigrp** *resolution-value* | **isis** *resolution-value* | **ospf** *resolution-value* | **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:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**track timer ip route** *seconds* **Example:** `Router(config)# track timer ip route 20`	(Optional) Specifies the interval in which the tracking process polls the tracked object. • The default interval that the tracking process polls IP-route objects is 15 seconds.
Step 4	**track resolution ip route** {**eigrp** *resolution-value* | **isis** *resolution-value* | **ospf** *resolution-value* | **static** *resolution-value* **Example:** `Router(config)# track resolution ip route eigrp 300`	(Optional) Specifies resolution parameters for a tracked object. • Use this command to change the default metric resolution values.
Step 5	**track** *object-number* **ip route** *ip-address/prefix-length* **metric threshold** **Example:** `Router(config)# track 6 ip route 10.16.0.0/16 metric threshold`	Tracks the scaled metric value of an IP route to determine if it is above or below a threshold. • The default down value is 255, which equates to an inaccessible route. • The default up value is 254.
Step 6	**delay** {**up** *seconds* [**down** *seconds*] | [**up***seconds*] **down** *seconds*} **Example:** `Router(config-track)# delay up 30`	(Optional) Specifies a period of time (in seconds) to delay communicating state changes of a tracked object.
Step 7	**ip vrf** *vrf-name* **Example:** `Router(config-track)# ip vrf VRF1`	(Optional) Configures a VRF table.

	Command or Action	Purpose
Step 8	**threshold metric** {**up** *number* [**down** *number*] | **down** *number* [**up** *number*]} **Example:** `Router(config-track)# threshold metric up 254 down 255`	(Optional) Sets a metric threshold other than the default value.
Step 9	**end** **Example:** `Router(config-track)# end`	Exits to privileged EXEC mode.
Step 10	**show track** *object-number* **Example:** `Router# show track 6`	(Optional) Displays tracking information. • Use this command to verify the configuration.

Example

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

```
Router# 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 GigabitEthernet0/1/1
   Tracked by:
     HSRP GigabitEthernet1/0/0 6
```

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:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**track** *object-number* **ip sla** *operation-number* **state** **Example:** `Router(config)# track 2 ip sla 4 state`	Tracks the state of an IP SLAs object and enters tracking configuration mode. **Note** Effective with Cisco IOS XE Release 2.4 the **track rtr** command was replaced by the **track ip sla** command.
Step 4	**delay** {**up** *seconds* [**down** *seconds*] | [**up** *seconds*] **down** *seconds*} **Example:** `Router(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:** `Router(config-track)# end`	Exits to privileged EXEC mode.
Step 6	**show track** *object-number* **Example:** `Router# show track 2`	(Optional) Displays tracking information. • Use this command to verify the configuration.

Example

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

```
Router# 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 (millisecs) 4
Tracked by:
  HSRP GigabitEthernet1/0/0 2
```

Tracking the Reachability of an IP SLAs IP Host

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **track** *object-number* **ip sla** *operation-number* **reachability**
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:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**track** *object-number* **ip sla** *operation-number* **reachability** **Example:** `Router(config)# track 2 ip sla 4 reachability`	Tracks the reachability of an IP SLAs IP host and enters tracking configuration mode. **Note** Effective with Cisco IOS XE Release 2.4 the **track rtr** command was replaced by the **track ip sla** command.
Step 4	**delay** {**up** *seconds* [**down** *seconds*]	[**up** *seconds*] **down** *seconds*} **Example:** `Router(config-track)# delay up 30 down 10`	(Optional) Specifies a period of time (in seconds) to delay communicating state changes of a tracked object.

	Command or Action	Purpose
Step 5	**end** **Example:** `Router(config-track)# end`	Exits to privileged EXEC mode.
Step 6	**show track** *object-number* **Example:** `Router# show track 3`	(Optional) Displays tracking information. • Use this command to verify the configuration.

Example

The following example shows whether the route is reachable:

```
Router# 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 GigabitEthernet1/0/0 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, page 17* section and the *Configuring a Tracked List and Threshold Percentage, page 18* section.

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:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**track** *track-number* **list boolean** {**and**	**or**} **Example:** `Router(config)# track 100 list boolean and`	Configures a tracked list object and enters tracking configuration mode.
Step 4	**object** *object-number* [**not**] **Example:** `Router(config-track)# object 3 not`	Specifies the object to be tracked. • The*object-number* 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** *seconds* [**down** *seconds*]	[**up** *seconds*] **down** *seconds*} **Example:** `Router(config-track)# delay up 3`	(Optional) Specifies a tracking delay in seconds between up and down states.

	Command or Action	Purpose
Step 6	**end** **Example:** `Router(config-track)# end`	Returns to privileged EXEC mode.

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, page 15* section and the *Configuring a Tracked List and Threshold Percentage, page 18* section.

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:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**track** *track-number* **list threshold weight** **Example:** `Router(config)# track 100 list threshold weight`	Configures a tracked list object and enters tracking configuration mode. The keywords are as follows: • **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** *object-number* [**weight** *weight-number*] **Example:** `Router(config-track)# object 3 weight 30`	Specifies the object to be tracked. The *object-number* 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** *number* **down** *number* | **up** *number* | **down** *number*} **Example:** `Router(config-track)# threshold weight up 30`	Specifies the threshold weight. • **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** *seconds* [**down** *seconds*] | [**up** *seconds*] **down** *seconds*} **Example:** `Router(config-track)# delay up 3`	(Optional) Specifies a tracking delay in seconds between up and down states.
Step 7	**end** **Example:** `Router(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, page 15* section and the*Configuring a Tracked List and Threshold Weight, page 17* section.

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

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:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**track** *track-number* **list threshold percentage** **Example:** `Router(config)# track 100 list threshold percentage`	Configures a tracked list object and enters tracking configuration mode. The keywords are as follows: • **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** *object-number* **Example:** `Router(config-track)# object 3`	Specifies the object to be tracked. • The *object-number* argument has a valid range from 1 to 500. There is no default.

	Command or Action	Purpose
Step 5	**threshold percentage** {**up** *number* [**down** *number*] | **down** *number* [**up** *number*]} **Example:** `Router(config-track)# threshold percentage up 30`	Specifies the threshold percentage. • **up** *number*—Valid range is from 1 to 100. • **down** *number* —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** *seconds* [**down** *seconds*] | [**up** *seconds*] **down** *seconds*} **Example:** `Router(config-track)# delay up 3`	(Optional) Specifies a tracking delay in seconds between up and down states.
Step 7	**end** **Example:** `Router(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:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**track** *track-number* **Example:** `Router(config)# track 3`	Enters tracking configuration mode.
Step 4	**default** {**delay** | **object** *object-number* | **threshold percentage**} **Example:** `Router(config-track)# default delay`	Specifies a default delay value for a tracked list, a default object, and default threshold parameters for a tracked list. • **delay** —Reverts to the default delay. • **object** *object-number*—Specifies a default object for the track list. The valid range is from 1 to 1000. • **threshold percentage**—Specifies a default threshold percentage.
Step 5	**end** **Example:** `Router(config-track)# 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

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

Router B Configuration

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

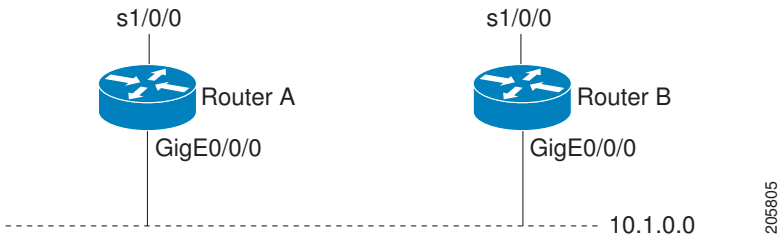
Example: Interface IP Routing

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

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

```
Router(config-if)# standby 1 priority 110
Router(config-if)# standby 1 track 100 decrement 10
```

Router B Configuration

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

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

Router B Configuration

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

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

Router B Configuration

```
Router(config)# track 100 ip route 10.2.2.0/24 metric threshold
```

```
!
Router(config)# interface GigabitEthernet0/0/0
Router(config-if)# ip address 10.1.1.22 255.255.255.0
Router(config-if)# standby 1 preempt
Router(config-if)# standby 1 ip 10.1.1.1
Router(config-if)# standby 1 priority 105
Router(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 in Cisco IOS XE releases prior to Cisco IOS XE Release 2.4:

```
Router(config)# ip sla 1
Router(config-ip-sla)# icmp-echo 10.51.12.4
Router(config-ip-sla-echo)# timeout 1000
Router(config-ip-sla-echo)# frequency 3
Router(config-ip-sla-echo)# threshold 2
Router(config-ip-sla-echo)# request-data-size 1400
Router(config-ip-sla-echo)# exit
Router(config)# ip sla schedule 1 start-time now life forever
Router(config-ip-sla)# exit
Router(config)# track 2 rtr 1 state
Router(config)# track 3 rtr 1 reachability
Router(config-track)# exit
Router(config)# interface GigabitEthernet0/1/0
Router(config-if)# ip address 10.21.0.4 255.255.0.0
Router(config-if)# no shutdown
Router(config-if)# standby 3 ip 10.21.0.10d
Router(config-if)# standby 3 priority 120
Router(config-if)# standby 3 preempt
Router(config-if)# standby 3 track 2 decrement 10
Router(config-if)# standby 3 track 3 decrement 10
```

The following example shows how to configure IP host tracking for IP SLAs operation 1 in Cisco IOS XE Release 2.4 and later releases:

```
Router(config)# ip sla 1
Router(config-ip-sla)# icmp-echo 10.51.12.4
Router(config-ip-sla-echo)# timeout 1000
Router(config-ip-sla-echo)# frequency 3
Router(config-ip-sla-echo)# threshold 2
Router(config-ip-sla-echo)# request-data-size 1400
Router(config-ip-sla-echo)# exit
Router(config)# ip sla schedule 1 start-time now life forever
Router(config-ip-sla)# exit
Router(config)# track 2 ip sla 1 state
Router(config)# track 3 ip sla 1 reachability
Router(config-track)# exit
Router(config)# interface gigabitethernet0/1/1
Router(config-if)# ip address 10.21.0.4 255.255.0.0
Router(config-if)# no shutdown
Router(config-if)# standby 3 ip 10.21.0.10d
Router(config-if)# standby 3 priority 120
Router(config-if)# standby 3 preempt
Router(config-if)# standby 3 track 2 decrement 10
Router(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:

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

```
Router(config)# track 100 list boolean and
Router(config-track)# object 1
Router(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:

```
Router(config)# track 1 interface GigabitEthernet2/0/0 line-protocol
Router(config)# track 2 interface GigabitEthernet2/1/0 line-protocol
Router(config-track)# exit
Router(config)# track 101 list boolean or
Router(config-track)# object 1
Router(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):

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

Example: Threshold Weight for a Tracked List

In the following example, three GigabitEtherent 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:

```
Router(config)# track 1 interface GigabitEthernet2/0/0 line-protocol
Router(config)# track 2 interface GigabitEthernet2/1/0 line-protocol
Router(config)# track 3 interface GigabitEthernet2/2/0 line-protocol
Router(config-track)# exit
Router(config)# track 100 list threshold weight
Router(config-track)# object 1 weight 20
Router(config-track)# object 2 weight 20
Router(config-track)# object 3 weight 20
Router(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 >= 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.

```
Router(config)# track 4 list threshold weight
Router(config-track)# object 1 weight 15
Router(config-track)# object 2 weight 20
Router(config-track)# object 3 weight 30
Router(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.

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

Additional References

Related Documents

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

Standards

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

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco 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 Release 2.1	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 XE 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 of IP SLAs Operations	Cisco IOS XE Release 2.1	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--Object Tracking List	Cisco IOS XE Release 2.1	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
FHRP--EOT Deprecation of **rtr** Keyword	Cisco IOS XE Release 2.4	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--Enhanced Object Tracking Integration with Embedded Event Manager	Cisco IOS XE Release 2.1	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**.

Glossary

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

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.

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.

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.

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. 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. (1110R)

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.

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 command reference master index, or search online.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 document.

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 Cisco IOS XE 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 also can 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

Cisco IOS XE 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 Cisco IOS XE 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 Internet Message Control Protocol (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 Cisco IOS XE 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 Cisco IOS XE 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.

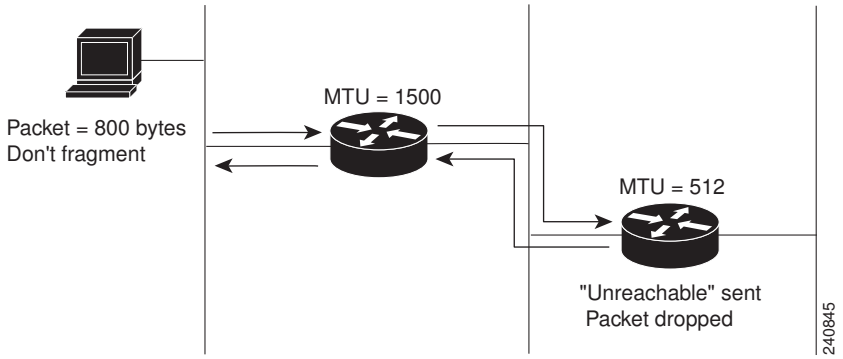
The objective of a DoS attack is to deprive a user or organization access to services or resources. If a Website is compromised by a DoS attack, millions of users could be denied access to the site. DoS attacks do not typically result in intrusion or the illegal theft of information. Instead of providing 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 Cisco IOS XE 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. TheCisco IOS XE 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” bit of the datagram is set, the datagram would be dropped because the 512-byte router is unable to forward it. All packets larger than 512 bytes are dropped in this case. The second router returns an ICMP destination unreachable message to the source of the datagram with its Code field indicating “Fragmentation needed and DF set.” To support IP Path MTU Discovery, it would also include the MTU of the next hop network link in the low-order bits of an unused header field.

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

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

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

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 Cisco IOS XE 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.

IP MAC Accounting

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

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*
5. **no ip unreachables**
6. **no ip redirects**
7. **no ip mask-reply**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**no ip source-route** **Example:** `Router(config)# no ip source-route`	Disables IP source routing.
Step 4	**interface** *type/number* **Example:** `Router(config)# interface GigabitEthernet 0/0/0`	Specifies the interface to configure and enters interface configuration mode.

	Command or Action	Purpose
Step 5	**no ip unreachables** **Example:** `Router(config-if)# no ip unreachables`	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 Cisco IOS XE software send unreachable messages.
Step 6	**no ip redirects** **Example:** `Router(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:** `Router(config-if)# no ip mask-reply`	Disables the sending of ICMP mask reply messages.

Setting the MTU Packet Size

All interfaces have a default MTU packet size. You can adjust the IP MTU size so that the Cisco IOS XE 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*
4. **ip mtu** *bytes*
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type/number* **Example:** `Router(config)# interface GigabitEthernet 0/0/0`	Specifies the interface to configure and enters interface configuration mode.
Step 4	**ip mtu** *bytes* **Example:** `Router(config-if)# ip mtu 300`	Sets the IP MTU packet size for an interface.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits 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. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip accounting-threshold** *threshold* **Example:** `Router(config)# ip accounting-threshold 500`	(Optional) Sets the maximum number of accounting entries to be created.
Step 4	**ip accounting-list** *ip-address wildcard* **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** *count* **Example:** `Router(config)# ip accounting-transits 100`	(Optional) Controls the number of transit records that will be stored in the IP accounting database.

	Command or Action	Purpose
Step 6	**interface** *type number* **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. • 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

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

SUMMARY STEPS

1. **clear ip traffic**
2. **clear ip accounting checkpoint**
3. **show ip accounting checkpoint** [**output-packets** |**access-violations**]
4. **show interface** [*type number*] **mac**
5. **show ip redirects**
6. (Optional) **show ip sockets**
7. **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
```

Step 3 **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 4 **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 GigabitEthernet 0/0/0 mac

GigabitEthernet0/0/0
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 5 **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

The following is sample output from 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 6 (Optional) **show ip sockets**

To display IP socket information, and to verify that the socket being used is opening correctly, use the **show ip sockets** command. If there is a local and remote endpoint, a connection is established with the ports indicated.

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

Example:

```
Router# show ip sockets

Proto    Remote         Port      Local           Port  In Out Stat TTY OutputIF
 17      10.0.0.0         0       172.16.186.193  67    0   0    1   0
 17      172.16.191.135   514     172.16.191.129  1811  0   0    0   0
 17      172.16.135.20    514     172.16.191.1    4125  0   0    0   0
 17      172.16.207.163   49      172.16.186.193  49    0   0    9   0
 17      10.0.0.0         123     172.16.186.193  123   0   0    1   0
 88      10.0.0.0         0       172.16.186.193  202   0   0    0   0
 17      172.16.96.59     32856   172.16.191.1    161   0   0    1   0
 17     --listen--             --any--        496  0    0   1    0
```

Step 7 **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
 Frags: 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 GigabitEthernet 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 also will disable IP Path MTU Discovery, because path discovery works by having the Cisco IOS XE 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.

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

Example: Setting the MTU Packet Size

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

```
Router(config)# interface GigabitEthernet 0/0/0
Router(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:

```
Router(config)# interface GigabitEthernet 0/0/0
Router(config-if)# ip accounting mac-address input
Router(config-if)# ip accounting mac-address output
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IP addressing services configuration tasks	*Cisco IOS XE IP Addressing Services Configuration Guide*
IP application services configuration tasks	*Cisco IOS XE IP Application Services Configuration Guide*
IP application services commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS XE IP Application Services Command Reference*

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 or modified MIBs are supported by this feature, and support for existing standards 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 791	*Internet Protocol*
RFC 792	*Internet Control Message Protocol*
RFC 1191	*Path MTU discovery*

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	Cisco IOS XE Release 2.1	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. The following command was introduced by this feature: **clear ip traffic**.

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. 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. (1110R)

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.

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. It is considered a reliable protocol because if an IP packet is dropped or received out of order, TCP will request the correct packet until it receives it. This module explains the concepts related to TCP and describes how to configure TCP in a network.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 document.

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 TCP selective acknowledgment once it is enabled.

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 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 identified by sequence numbers. This service benefits applications because they do not have to chop 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 the source expects to receive. Bytes 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 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 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 initial sequence numbers. This mechanism also guarantees that both sides are ready to transmit data and know that

the other side also is ready to transmit. 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 used to track bytes within the stream it is sending. Then, 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 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 the SYN (with an ACK = X + 1). Host B includes its own initial sequence number (SEQ = Y). An ACK = 20 means the host has received bytes 0 through 19 and expects byte 20 next. This technique is called forward acknowledgment.
- Host A acknowledges all bytes Host B sent with a forward acknowledgment indicating the next byte Host A expects to receive (ACK = Y + 1). Data transfer then can begin.

TCP Connection Attempt Time

You can set the amount of time the Cisco IOS XE software will wait to attempt to establish a TCP connection. Because the connection attempt time is a host parameter, it does not pertain to traffic going through the device, just to traffic originated at the device. To set the TCP connection attempt time, use the **ip tcp synwait-time** command in global configuration mode. The default is 30 seconds.

TCP Selective Acknowledgment

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

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

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

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

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 detailed information 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 detailed information on TCP time stamps.

TCP Maximum Read Size

The maximum number of characters that TCP reads from the input queue for Telnet and rlogin at one time is a very large number (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.

We do not recommend that you change this value.

TCP Path MTU Discovery

Path MTU Discovery is a method for maximizing the use of available bandwidth in the network between the endpoints of a TCP connection, 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 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 **interface** configuration command), but the "don't 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 it to the problem. Upon receiving this ICMP message, the host reduces its assumed path MTU and consequently sends a smaller packet that will fit the smallest packet size of all the links along the path.

By default, TCP Path MTU Discovery is disabled. Existing connections are not affected when 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 router when it is acting as a host.

For more information about Path MTU Discovery, refer to the "Configuring IP Services" chapter of the *Cisco IOS XEIP 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 that support.

The window scaling extension in Cisco IOS XE software 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 more 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 you keep the default value unless you know your router 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 transmit bytes 6 to 10. The receiver responds with an ACK = 11, indicating that it is expecting sequenced byte 11 next. In this packet, the receiver might indicate that its window size is 0 (because, for example, its internal buffers are full). At this point, 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 5 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 5-segment default value.

TCP MSS Adjustment

The TCP MSS Adjustment feature enables the configuration of the maximum segment size (MSS) for transient packets that traverse a router, 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 router of the SYN packets to avoid truncation.

When a host (usually a PC) initiates a TCP session with a server, it 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 an MTU of only 1492 bytes. The disparity between the host and PPPoE MTU size can cause the router 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 the ICMP error messages that must be relayed from the host in order 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 router.

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.

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 retransmission timeouts, application closed, and synchronized (SYNC) handshakes for listen. 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 hostname format and to display the VRF table associated with the connection. To display the status for all endpoints with the 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

Zero-Field TCP Packets

Prior to Cisco IOS XE Release 2.5, when a zero-field TCP packet is received on the router, the TCP packet counter is incremented.

In Cisco IOS XE Release 2.5 and later releases, when a zero-field TCP packet is received on the router, the TCP packet counter is not incremented.

When a zero-field TCP packet is received, it is displayed as 0 under the TCP statistics field when the **show ip traffic** command is configured. When the **debug ip tcp packet** command is configured, and a zero-field TCP packet is received, a debug message similar to the following is displayed:

```
Jan 19 21:57:28.487: TCP: Alert! Received a segment with cleared flags 10.4.14.49
```

How to Configure TCP

Configuring TCP Performance Parameters

Both sides of the link must be configured to support window scaling or the default of 65,535 bytes will apply as the maximum window size.

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 queuemax** *packets*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip tcp synwait-time** *seconds* **Example:** `Router(config)# ip tcp synwait-time 60`	(Optional) Sets the amount of time the Cisco IOS XE software will wait to attempt to establish a TCP connection. • The default is 30 seconds.
Step 4	**ip tcp path-mtu-discovery** [**age-timer** {*minutes* | **infinite**}] **Example:** `Router(config)# ip tcp path-mtu-discovery age-timer 11`	(Optional) Enables Path MTU Discovery. • **age-timer**—Time interval, in minutes, TCP reestimates the path MTU with a larger MSS. The default is 10 minutes. The maximum is 30 minutes. • **infinite**—Disables the age timer.
Step 5	**ip tcp selective-ack** **Example:** `Router(config)# ip tcp selective-ack`	(Optional) Enables TCP selective acknowledgment.

	Command or Action	Purpose
Step 6	**ip tcp timestamp** **Example:** `Router(config)# ip tcp timestamp`	(Optional) Enables the TCP time stamp.
Step 7	**ip tcp chunk-size** *characters* **Example:** `Router(config)# ip tcp chunk-size 64000`	(Optional) Sets the TCP maximum read size for Telnet or rlogin. **Note** We do not recommend that you change this value.
Step 8	**ip tcp window-size** *bytes* **Example:** `Router(config)# ip tcp window-size 75000`	(Optional) Sets the TCP window size. The *bytes* argument can be set to an integer from 0 to 1073741823. • To enable window scaling to support LFNs, the TCP window size must be more than 65535. The default window size is 4128 if window scaling is not configured.
Step 9	**ip tcp queuemax** *packets* **Example:** `Router(config)# ip tcp queuemax 10`	(Optional) Sets the TCP outgoing queue size.

Configuring the MSS Value and MTU for Transient TCP SYN Packets

Perform this task to configure the MSS for transient packets that traverse a router, 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:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface GigabitEthernet 1/0/0`	Configures an interface type and enters interface configuration mode.
Step 4	**ip tcp adjust-mss** *max-segment-size* **Example:** `Router(config-if)# ip tcp adjust-mss 1452`	Adjusts the MSS value of TCP SYN packets going through a router. • The *max-segment-size* argument is the maximum segment size, in bytes. The range is from 500 to 1460.
Step 5	**ip mtu** *bytes* **Example:** `Router(config-if)# ip mtu 1492`	Sets the MTU size of IP packets, in bytes, sent on an interface.
Step 6	**end** **Example:** `Router(config-if)# end`	Exits to global configuration mode.

Verifying TCP Performance Parameters

SUMMARY STEPS

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

DETAILED STEPS

Step 1 **show tcp** [*line-number*]
Displays the status of TCP connections. The arguments and keyword are as follows:

- *line-number*—(Optional) Absolute line number of the Telnet connection status.

The following is sample output that displays the status and option flags:

Example:

```
Router# 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, KRTT: 0 ms
minRTT: 0 ms, maxRTT: 300 ms, ACK hold: 200 ms
 Status Flags: active open, retransmission timeout
 Option Flags: vrf id set
 IP Precedence value: 6
```

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 the addresses in a Domain Name System (DNS) hostname format. Without this keyword, endpoints in the LISTEN state are not shown. Use the optional **numeric** keyword to display the status for all endpoints with the addresses in IP format.

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

Example:

```
Router# show tcp brief

TCB       Local Address           Foreign Address        (state)
609789AC  Router.cisco.com.23     cider.cisco.com.3733   ESTAB
```

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

Example:

```
Router# 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. This command is particularly useful for debugging a performance problem on a TCP/IP network that you have isolated above the data-link layer.

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

Example:

```
Router# 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 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 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):

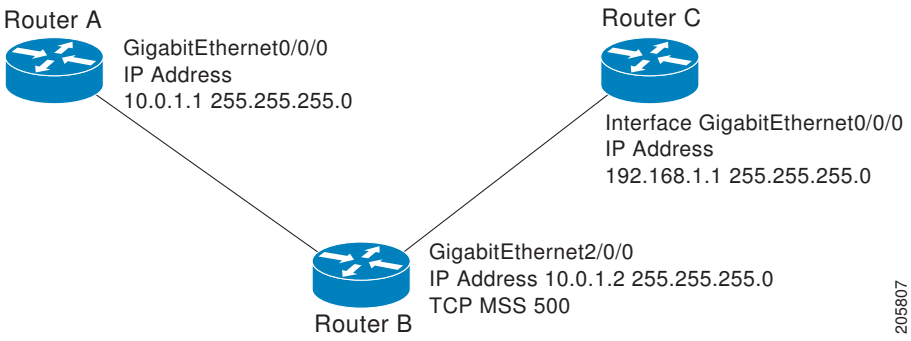
Example:

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

Configuration Examples for TCP

Example Configuring the TCP MSS Adjustment

Figure 3 Example Topology for TCP MSS Adjustment



The following example shows how to configure and verify the interface adjustment value for the example topology displayed in the figure above. 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 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 PPPoE client with the MSS value set to 1452:

```
Router(config)# vpdn enable
Router(config)# no vpdn logging
Router(config)# vpdn-group 1
Router(config-vpdn)# request-dialin
Router(config-vpdn-req-in)# protocol pppoe
Router(config-vpdn-req-in)# exit
Router(config-vpdn)# exit
Router(config)# interface GigabitEthernet0/0/0
Router(config-if)# ip address 192.168.100.1.255.255.255.0
Router(config-if)# ip tcp adjust-mss 1452
Router(config-if)# ip nat inside
Router(config-if)# exit
Router(config)# interface ATM0
Router(config-if)# no ip address
Router(config-if)# no atm ilmi-keepalive
Router(config-if)# pvc 8/35
Router(config-if)# pppoe client dial-pool-number 1
Router(config-if)# dsl equipment-type CPE
Router(config-if)# dsl operating-mode GSHDSL symmetric annex B
Router(config-if)# dsl linerate AUTO
```

```
Router(config-if)# exit
Router(config)# interface Dialer1
Router(config-if)3 ip address negotiated
Router(config-if)# ip mtu 1492
Router(config-if)# ip nat outside
Router(config-if)# encapsulation ppp
Router(config-if)# dialer pool 1
Router(config-if)# dialer-group 1
Router(config-if)# ppp authentication pap callin
Router(config-if)# ppp pap sent-username sohodyn password 7 141B1309000528
Router(config-if)# ip nat inside source list 101 Dialer1 overload
Router(config-if)# exit
Router(config)# ip route 0.0.0.0.0.0.0.0 Dialer1
Router(config)# access-list permit ip 192.168.100.0.0.0.0.255 any
```

Example: Configuring the TCP Application Flags Enhancement

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

```
Router# 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, KRTT: 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:

```
Router# 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
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IP application services configuration tasks	*Cisco IOS XE IP Application Services Configuration Guide*

Related Topic	Document Title
IP application services commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS XE IP Application Services Command Reference*
Path MTU Discovery	Configuring IP Services module

Standards

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

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

RFCs

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 3782	The NewReno Modification to TCP's Fast Recovery Algorithm
RFC 4022	Management Information Base for the Transmission Control Protocol (TCP)

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 5 Feature Information for TCP

Feature Name	Releases	Feature Information
TCP Application Flags Enhancement	Cisco IOS XE Release 2.1	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; for example, retransmission timeouts, application closed, and synchronized (SYNC) handshakes for listen. The additional flags indicate the state of set options; for example, whether a VRF identification 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 MIB for RFC 4022 Support	Cisco IOS XE Release 2.1	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. There are no new or modified command for this feature.
TCP MSS Adjust	Cisco IOS XE Release 2.1	The TCP MSS Adjust feature enables the configuration of the maximum segment size (MSS) for transient packets that traverse a router, specifically TCP segments in the SYN bit set. The following command was introduced by this feature: **ip tcp adjust-mss**.
TCP Show Extension	Cisco IOS XE Release 2.1	The TCP Show Extension feature introduces the capability to display addresses in IP format instead of hostname format and to display the virtual private network VRF table associated with the connection. The following command was modified by this feature: **show tcp brief**.
TCP Window Scaling	Cisco IOS XE Release 2.1	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. The following commands were introduced or modified by this feature: **ip tcp window-size**.

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. 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. (1110R)

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.

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 XE Release 2.2 supports only 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

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 document.

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

- WCCP works only with IPv4 networks.
- WCCP does not redirect IP multicast packets.
- WCCP packet redirection on outbound interfaces is not supported in Cisco IOS XE releases prior to Cisco IOS XE Release 3.1S.
- There is no SNMP support and no MIB has been implemented for WCCPv2.
- Cisco ASR 1000 Series Routers do not support WCCPv1.
- Cisco ASR 1000 Series Routers do not support inter-VRF redirection.
- Service groups can comprise up to 32 content engines and 32 routers.
- WCCP does not support InService Software Upgrade (ISSU), stateful switchover (SSO) or nonstop forwarding (NSF).
- Transiting packets are lost in the event of a forwarding processor (FP) failover on a 6-rack-unit (6RU) and 13RU chassis.
- All content engines in a cluster must be configured to communicate with all routers servicing the cluster.
- Hash assignment as a load-balancing method for a WCCP service is not supported. As of Cisco IOS XE Release 3.1S, clients that send hash assignment will not be allowed by the router to come online. On Cisco ASR 1000 Series Routers, the **show ip wccp 61 detail** command displays that hash is an incompatible assignment method.
- For routers servicing a multicast cluster, the Time To Live (TTL) value must be set at 15 or lower.
- The **show ip wccp** command displays information about software-based (process, fast and Cisco Express Forwarding [CEF]) forwarding of WCCP packets. The Cisco ASR 1000 Series Routers implement WCCP in hardware, rather than in the CEF or process-switching paths. Implementing WCCP in hardware results in a packet count of 0 when the **show ip wccp** command is entered. Use the **show platform software wccp interface counters** or **show platform software wccp counters** commands to display global statistics related to WCCP on the Cisco ASR 1000 Series Routers.
- When the Cisco ASR 1000 Series Router and WCCP are used to redirect traffic to a WAE that is using WCCP GRE return as the egress method and the HTTP accelerator is enabled, the ASR router's handling of proxied HTTP connections may cause HTTP slowness. To work around this issue, on the ASR router, use the **ip wccp** [**vrf** *vrf-name*] **web-cache** command to create a web cache service in the same VRF as that of the 61/62 service.
- In Cisco IOS XE Release 3.1S, the **show ip wccp** command displays redirected WCCP packets.
- When the IP address of an interface that is being used as the router ID (highest IP address of the interfaces) is removed when a WCCP cache engine is connected via Generic Routing Encapsulation (GRE) adjacency, the source-IP address of the outer IP packet (of GRE) will continue to use the removed IP address. The traffic will continue to get redirected to the cache engine. This symptom is not visible, because the Cisco IOS XE software updates the router ID in the protocol messages to the cache engine, and the cache engine uses the new router ID when it returns packets to the router. Configure a loopback address and assign an IP address to the loopback address so that the assigned loopback address is used as the router ID. Removal of such a loopback IP address is unlikely, but when the loopback address is removed, the source IP address of the GRE packet from the router to the cache engine will carry the removed IP address. Enter the **shutdown** command, followed by the **no shutdown** command on the cache engine interface that has the GRE redirect method configured to stop the interface from using the removed IP address.
- The following limitation applies to WCCP Layer 2 Forwarding and Return feature:
  Layer 2 redirection requires that content engines be directly connected to an interface on each WCCP router. 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 WCCP

WCCP Overview

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

WCCP enables Cisco IOS XE routing platforms to transparently redirect content requests. The main benefit of transparent redirection is that users do not need to configure 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.

When a content engine receives a request, it 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. When the content engine retrieves the requested information, it 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 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. On Cisco ASR 1000 Series Routers, both the GRE and L2 forward/return methods use the hardware, so there is not any significant performance degradation between them.

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.

For more information on Cisco ACNS commands used to configure Cisco Content Engines, see the *Cisco ACNS Software Command Reference*, Release 5.5.13.

For more information on WAAS commands used to configure Cisco Content Engines, see the *Cisco Wide Area Application Services Command Reference (Software Versions 4.2.1)*.

WCCP Mask Assignment

The WCCP Mask Assignment feature enables mask assignment as the load-balancing method for a WCCP service.

For content engines running Application and Content Networking System (ACNS) software, use the **wccp web-cache** command with the **mask-assign**keywords 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 more information on Cisco ACNS commands used to configure Cisco Content Engines, see the *Cisco ACNS Software Command Reference*, Release 5.5.13.

For more information on WAAS commands used to configure Cisco Content Engines, see the Cisco Wide Area Application Services Command Reference (Software Versions 4.2.1).

Hardware Acceleration

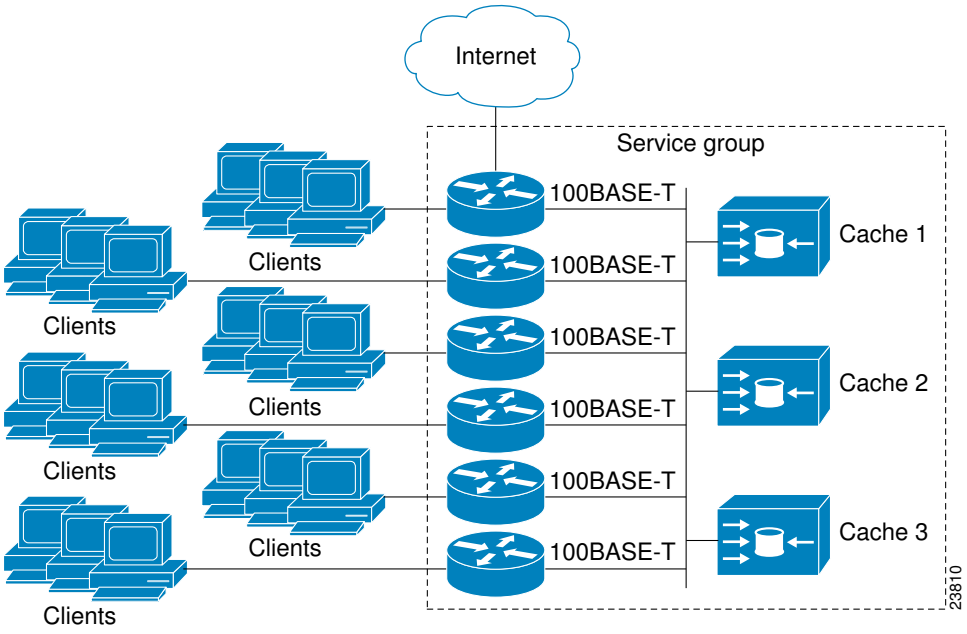
WCCP implementation on the Cisco ASR 1000 Series Routers is hardware accelerated by default.

You do not need to configure the **ip wccp web-cache accelerated** command on Cisco ASR routers to enable hardware acceleration.

WCCPv2 Configuration

Multiple routers can use WCCPv2 to service a content engine cluster. The figure below illustrates a sample configuration using multiple routers.

Figure 4 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.

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 the following method:

- 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** 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 introduces 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 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 unserviced. 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)

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 packet flows are intercepted and redirected by a Cisco IOS router to external WCCP client devices, it may be necessary to block the packet flows 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, WCCP discards packets that do not have a WCCP client registered to receive the redirected traffic.

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** command can only be used for closed-mode services. Use the **service-list** keyword and *service-access-list* argument to register an application protocol type or port number.

When there is a mismatch between the service-list ACL and the definition received from a cache engine, the service is not allowed to start.

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. This 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 XE 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.

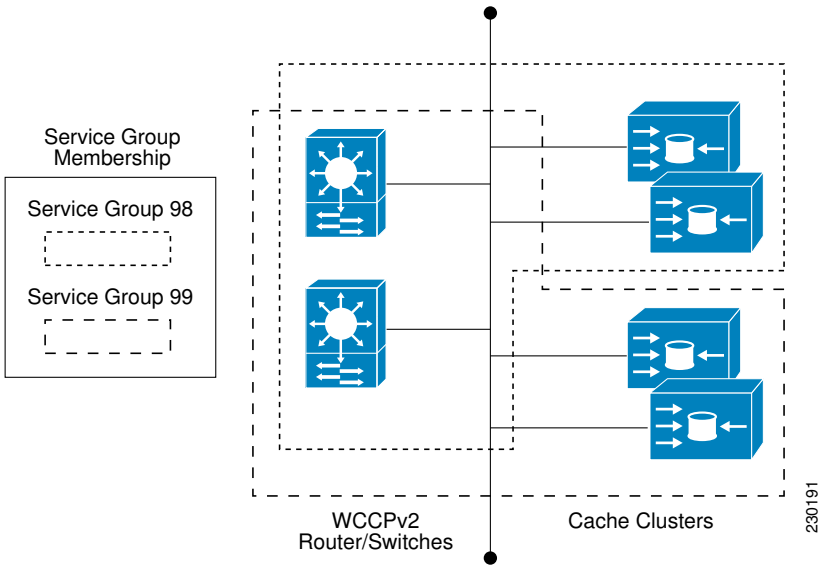
The nature of the selected traffic for redirection is defined by service groups specified on content engines and communicated to routers by using WCCP. The current implementation of WCCP in Cisco IOS XE software allows for a maximum of 256 service groups across all VRFs.

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** 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 5 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. The configuration information in this document deals with enabling general services on the Cisco ASR 1000 Series Routers.

WCCP Check Services All

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** 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 as well as by the service priority.

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** command is configured. When the **ip 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 Configurable Router ID

WCCP uses a router ID in its control messages and the router ID serves as a means by which a WCCP client can identify a particular WCCP server. The router ID is treated as an IPv4 address and may also be used as the source address of any WCCP-generated GRE frames. Prior to the WCCP Configurable Router ID feature, WCCP selects a router ID using an automatic mechanism; the highest reachable IPv4 address on the system is used as the WCCP router ID. The highest IPv4 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. 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 not 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. The router ID changes only when a new router ID is manually configured using the **ip wccp source-address** command.

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** command on the WAAS interface. If you are not able to configure the **ip 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

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 CLI to investigate further. The counters allow you to determine the percent of traffic being bypassed.

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 router. The first use of a form of the **ip wccp** command enables WCCP. By default WCCPv2 is used for services.

Using the**ip 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 consist of up to eight characters. 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.

Note WCCPv1 is not supported on the Cisco ASR 1000 Series Routers.

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip wccp** [**vrf** *vrf-name*] {**web-cache** | *service-number*} | [**group-address** *group-address*] [**redirect-list** *access-list*] [**group-list** *access-list*] [**password** *password*] **Example:** `Router(config)# ip 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 4	**ip wccp** [**vrf** *vrf-name*] **source-interface** *source-interface* **Example:** `Router (config)# ip wccp source-interface GigabitEthernet 0/0/0`	(Optional) Configures a preferred WCCP router ID.
Step 5	**interface** *type number* **Example:** `Router(config)# interface GigabitEthernet0/1/0`	Targets an interface number for which the web cache service will run, and enters interface configuration mode.
Step 6	**ip wccp** [**vrf** *vrf-name*] {**web-cache** | *service-number*} **redirect** {**out** | **in**} **Example:** `Router(config-if)# ip wccp web-cache redirect in`	Enables packet redirection on an inbound or outbound interface using WCCP.

	Command or Action	Purpose
Step 7	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 8	**interface** *type number* **Example:** `Router(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:** `Router(config-if)# ip wccp redirect exclude in`	(Optional) Excludes traffic on the specified interface from redirection. You can use this command in conjunction with the **ip wccp redirect out** command.

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:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

Step 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**}
	
	Example:
	`Router(config)# ip wccp 90 service-list 120 mode closed`
	or
	`Router(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:** `Router(config)# ip wccp check services all`	(Optional) Enables a check of all WCCP services. • 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. **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.
Step 5	**ip wccp** [**vrf** *vrf-name*] {**web-cache** | *service-number*} **Example:** `Router(config)# ip wccp 201`	Specifies the WCCP service identifier. • 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:** `Router(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

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip multicast-routing** [**vrf** *vrf-name*] [**distributed**] **Example:** `Router(config)# ip multicast-routing`	Enables IP multicast routing.

	Command or Action	Purpose
Step 4	**ip wccp** [**vrf** *vrf-name*] {**web-cache** | *service-number*} **group-address** *multicast-address* **Example:** `Router(config)# ip wccp 99 group-address 239.1.1.1`	Specifies the multicast address for the service group.
Step 5	**interface** *type number* **Example:** `Router(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:** `Router(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 Catalyst 6500 series switches and 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:** `Router(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 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. **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	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**access-list** *access-list-number* **remark** *remark* **Example:** `Router(config)# access-list 1 remark Give access to user1`	(Optional) Adds a user-friendly comment about an access list entry. • A remark of up to 100 characters can precede or follow an access list entry.
Step 4	**access-list** *access-list-number* **permit** {*source* [*source-wildcard*] | **any**} [**log**] **Example:** `Router(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. • 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 *source-wildcard* 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 *source source-wildcard* 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** *access-list-number* **remark** *remark* **Example:** `Router(config)# access-list 1 remark Give access to user1`	(Optional) Adds a user-friendly comment about an access list entry. • A remark of up to 100 characters can precede or follow an access list entry.

	Command or Action	Purpose
Step 6	**access-list** *access-list-number* **deny** {*source* [*source-wildcard*] | **any**} | [**log**] **Example:** `Router(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. • If the *source-wildcard* 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 *source source-wildcard* 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** [**vrf** *vrf-name*] **web-cache group-list** *access-list* **Example:** `Router(config) ip wccp web-cache group-list 1`	Indicates to the router from which IP addresses of content engines to accept packets.
Step 9	**ip wccp** [**vrf** *vrf-name*] **web-cache redirect-list** *access-list* **Example:** `Router(config)# ip wccp web-cache redirect-list 1`	(Optional) Disables caching for certain clients.

Enabling the WCCP Outbound ACL Check

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:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip wccp** [**vrf** *vrf-name*] {**web-cache** | *service-number*} [**group-address** *multicast-address*] [**redirect-list** *access-list*] [**group-list***access-list*] [**password** *password*] **Example:** `Router(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.
Step 4	**ip wccp check acl outbound** **Example:** `Router(config)# ip wccp check acl outbound`	Enables the ACL outbound check on the originating interface. **Note** The **ip wccp outbound-check-acl** command can also be configured.
Step 5	**exit** **Example:** `Router(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. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface ethernet 1`	Specifies an interface on which to enable NAT and enters interface configuration mode. • 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).

	Command or Action	Purpose
Step 5	**ip wccp** *service-number* **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** *type number* **Example:** `Router(config)# interface ethernet 2`	Specifies an interface on which to enable NAT and enters interface configuration mode. • This is the WAN-facing interface.
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** *service-number* **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** *type number* **Example:** `Router(config)# interface ethernet 3`	Specifies an interface on which to enable NAT and enters interface configuration mode. • This is the WAAS-facing interface.

	Command or Action	Purpose
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

The **show ip wccp** command displays information about software-based (process, fast, and Cisco Express Forwarding [CEF]) forwarding of WCCP packets. The Cisco ASR 1000 Series Routers implement WCCP in hardware, rather than in the CEF or process-switching paths. Implementing WCCP in hardware results in a packet count of 0 when the **show ip wccp** command is entered in Cisco IOS XE releases prior to Cisco IOS XE Release 3.1S. To display global statistics related to WCCP in Cisco ASR 1000, use the **show platform software wccp** command. As of Cisco IOS XE Release 3.1S, the **show ip wccp** command displays redirected WCCP packets.

Use the following commands in privileged EXEC mode to verify and monitor the configuration settings for WCCP.

SUMMARY STEPS

1. **enable**
2. **debug ip wccp** {**default** | **vrf** *vrf-name* {**events** | **packets** [**control**]} | **events** | **packets** [**bypass** | **control** | **redirect**] | **platform** | **subblocks**}
3. **debug platform hardware qfp active feature wccp** {{**client** | **lib-client** {**all** | **error** | **info** | **trace** | **warning**}} | **datapath all**}
4. **debug platform software wccp** {**configuration** | **counters** | **detail** | **messages**}
5. **show platform software wccp** [*service-number* **counters** | [*slot* [*service-number* [**access-list**] | **cache-info** | **interface** | **statistics** | **web-cache** [**access-list**]] | **vrf** *vrf-identifier* {*service-number* [**access-list**] | **web-cache** [**access-list**]}]] | **interface counters** | **statistics** | [**vrf** *vrf-identifier* {*service-number* **counters** | **web-cache counters**}] | **web-cache counters**]
6. **show platform hardware qfp active feature wccp** [**vrf** *vrf-id*] **service id** *service-id*
7. **show ip wccp global** [**counters**]
8. **show ip interface**
9. **more system:running-config**
10. **configure terminal**
11. **platform trace runtime slot** *slot* **bay** *bay* **process forwarding-manager module wccp level** {*level*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug ip wccp** {**default** | **vrf** *vrf-name* {**events** | **packets** [**control**]} | **events** | **packets** [**bypass** | **control** | **redirect**] | **platform** | **subblocks**} **Example:** `Router# debug ip wccp events`	Display information about WCCP services.
Step 3	**debug platform hardware qfp active feature wccp** {{**client** | **lib-client** {**all** | **error** | **info** | **trace** | **warning**}} | **datapath all**} **Example:** `Router# debug platform hardware qfp active feature wccp client all`	Enables debug logging for the WCCP client in the Cisco Quantum Flow Processor (QFP).
Step 4	**debug platform software wccp** {**configuration** | **counters** | **detail** | **messages**} **Example:** `Router# debug platform software wccp configuration`	Enables WCCP platform debug messages.
Step 5	**show platform software wccp** [*service-number* **counters** | [*slot* [*service-number* [**access-list**] | **cache-info** | **interface** | **statistics** | **web-cache** [**access-list**]] | **vrf** *vrf-identifier* {*service-number* [**access-list**] | **web-cache** [**access-list**]}]] | **interface counters** | **statistics** | [**vrf** *vrf-identifier* {*service-number* **counters** | **web-cache counters**}] | **web-cache counters**] **Example:** `Router# show platform software wccp 61 counters`	Displays global statistics related to WCCP on the Cisco ASR 1000 Series Routers.
Step 6	**show platform hardware qfp active feature wccp** [**vrf** *vrf-id*] **service id** *service-id* **Example:** `Router# show platform hardware qfp active feature wccp [vrf vrf-id] service id 1`	Displays WCCP service group information in the active QFP.

	Command or Action	Purpose
Step 7	**show ip wccp global** [**counters**] **Example:** `Router# show ip wccp global counters`	Displays global, nonservice WCCP information.
Step 8	**show ip interface** **Example:** `Router# 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 9	**more system:running-config** **Example:** `Router# more system:running-config`	(Optional) Displays contents of the currently running configuration file (equivalent to the **show running-config** command.)
Step 10	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 11	**platform trace runtime slot** *slot* **bay** *bay* **process forwarding-manager module wccp level** {*level*} **Example:** `Router(config)# platform trace runtime slot 1 bay 0 process forwarding-manager module wccp level debug`	Enables Forwarding Manager route processor and Embedded Service Processor trace messages for the WCCP process.

Configuration Examples for WCCP

Example: Configuring a General WCCPv2 Session

```
Router# configure terminal
Router(config)# ip wccp web-cache group-address 224.1.1.100 password password1
Router(config)# ip wccp source-interface GigabitEthernet 0/1/0
Router(config)# ip wccp check services all !
 Configures a check of all WCCP services.
Router(config)# interface GigabitEthernet 0/1/0
Router(config-if)# ip wccp web-cache redirect in
Router(config-if)# exit
Router(config)# interface GigabitEthernet 0/2/0
Router(config-if)# ip wccp redirect exclude in
Router(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

```
Router# configure terminal
Router(config)# ip wccp web-cache
Router(config)# interface GigabitEthernet 0/1/0
Router(config-if)# ip wccp web-cache redirect in
Router(config-if)# exit
Router# 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:

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

```
Router# configure terminal
```

```
Router(config)# ip wccp web-cache group-address 224.1.1.100
Router(config)# interface gigabitethernet 0/1/0
Router(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:

```
Router# configure terminal
Router(config)# ip wccp 99 group-address 224.1.1.1
Router(config)# interface gigabitethernet 0/1/0
Router(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 router which IP addresses are valid addresses for a content engine attempting to register with the current router. The following example shows a standard access list configuration session where the access list number is 10 for some sample hosts:

```
Router(config)# access-list 10 permit host 10.1.1.1
Router(config)# access-list 10 permit host 10.1.1.2
Router(config)# access-list 10 permit host 10.1.1.3
Router(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:

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

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

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

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

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 router:

```
Router# 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:

```
Router# 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*.

Additional References

Related Documents

Related Topic	Document Title
Cisco ACNS software configuration information	• Cisco ACNS Software Caching Configuration Guide, Release 4.2 • http://www.cisco.com/en/US/products/sw/conntsw/ps491/products_installation_and_configuration_guides_list.html • Cisco ACNS Software listing page on Cisco.com
Deploying and Troubleshooting WCCP on Cisco ASR 1000 Series Routers	Deploying and Troubleshooting Web Cache Control Protocol Version 2 on Cisco ASR 1000 Series Aggregation Services Routers
IP Access List overview, configuration tasks, and commands	• *Cisco IOS XE Security Configuration Guide: Securing the Data Plane* • *Cisco IOS Security Command Reference*
IP addressing and services commands and configuration tasks	• *Cisco IOS XE IP Addressing Services Configuration Guide* • *Cisco IOS IP Addressing Services Command Reference*
WCCP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Application Services Command Reference*

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
None	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 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 WCCP

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 WCCP

Feature Name	Releases	Feature Information
WCCP Bypass Counters	Cisco IOS XE Release 2.2	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 following commands were modified or introduced by this feature: **show ip wccp**, **show platform software wccp**.

Feature Name	Releases	Feature Information
WCCP: Check Services All	Cisco IOS XE Release 3.1S	The WCCP: Check Services All feature enables you to configure WCCP to search all service groups and redirect ACLs in priority order for a match. The following command was modified by this feature: **ip wccp check services all**
WCCP Closed Services	Cisco IOS XE Release 3.1S	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 AONS (Application-Oriented Network Services) 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 behavior 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--Configurable Router ID	Cisco IOS XE Release 3.1S	The WCCP--Configurable Router ID feature permits the router ID which WCCP uses to be configurable, rather than relying on the router's selection mechanism. The **ip wccp source-interface** commands was introduced by this feature.

Feature Name	Releases	Feature Information
WCCP Egress Redirection Support	Cisco IOS XE Release 3.1S	The WCCP Egress Redirection Support feature enables WCCP based redirection applied to the outbound traffic on the outbound interface. The **ip wccp redirect** command was modified by this feature.
WCCP Exclude Interface	Cisco IOS XE Release 3.1S	The WCCP Exclude Interface feature enables you to configure an interface to exclude packets received on an interface from being checked for redirection by configuring the **ip wccp redirect exclude in**command in interface configuration mode. The following command was introduced by this feature: **ip wccp redirect exclude in**
WCCP Group List	Cisco IOS XE Release 3.1S	The WCCP Group List feature enables you to configure the IP addresses of cache engines from which a router accepts packets. Configuring a group list is used to validate the protocol packets received from the cache engine. Packets matching the address in a configured group-list are processed, others are discarded. The **ip wccp** command was introduced or modified by this feature.

Feature Name	Releases	Feature Information
WCCP--Group Listen and Multicast Service Support	Cisco IOS XE Release 3.1S	The WCCP--Group Listen and Multicast Service Support feature adds the ability to configure a multicast address per service group for sending and receiving protocol messages. In the multicast address method, the cache engine sends a single-address notification that provides coverage for all routers in the service group. • WCCPv2 Configuration, page 71 • Registering a Router to a Multicast Address, page 80 • Example: Registering a Router to a Multicast Address, page 90 • The **ip wccp group-listen**command was modified by this feature.
WCCP Increased Services	Cisco IOS XE Release 3.1S	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**.

Feature Name	Releases	Feature Information
WCCP Layer 2 Redirection / Forwarding	Cisco IOS XE Release 2.2	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	Cisco IOS XE Release 2.2	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.
WCCP Mask Assignment	Cisco IOS XE Release 2.2	The WCCP Mask Assignment feature introduces support for ACNS/WAAS devices using mask assignment as a cache engine assignment method. There are no new or modified commands associated with this feature.

Feature Name	Releases	Feature Information
WCCP Outbound ACL Check	Cisco IOS XE Release 3.1S	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. The following commands were introduced or modified by this feature: **ip wccp**, **ip wccp check acl outbound**.
WCCP Redirection on Inbound Interfaces	Cisco IOS XE Release 2.2	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. The following commands were introduced or modified by this feature: **ip wccp redirect**.

Feature Name	Releases	Feature Information
WCCP Version 2	Cisco IOS XE Release 2.2	The WCCP Version 2 feature provides several enhancements and features to the WCCP protocol, including: • 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 unserviced. • 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. 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**.
WCCP VRF Support	Cisco IOS XE Release 3.1S	The WCCP VRF Support feature provides enhancements to the existing WCCPv2 protocol, which supports VRF awareness. 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**

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. 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. (1110R)

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.