WCCPv2—IPv6 Support

This feature introduces support for Web Cache Communication Protocol version 2 (WCCPv2) in an IPv6 environment.

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

Multiple routers can use WCCPv2 to service a content engine cluster. In WCCPv1, only one router can redirect content requests to a cluster.

Finding Feature Information

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

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

Prerequisites for WCCPv2—IPv6 Support

- IPv6 must be configured on the interface used for redirection and on the interface facing the content engine.

- The interface connected to the content engine must be a Fast Ethernet or Gigabit Ethernet interface.

Restrictions for WCCPv2—IPv6 Support

WCCPv2

- WCCP bypasses Network Address Translation (NAT) when Cisco Express Forwarding is enabled.
- For routers servicing a multicast cluster, the Time To Live (TTL) value must be set at 15 or lower.
- Service groups can comprise up to 32 content engines and 32 routers.
- All content engines in a cluster must be configured to communicate with all routers servicing the cluster.
- Multicast addresses must be in the range from 224.0.0.0 to 239.255.255.255.

Layer 2 Forwarding and Return

- Layer 2 redirection requires that content engines be directly connected to an interface on each WCCP router. Unless multicast IP addresses are used, WCCP configuration of the content engine must reference the directly connected interface IP address of the WCCP router and not a loopback IP address or any other IP address configured on the WCCP router.

Information About WCCPv2—IPv6 Support

WCCP Overview

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

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

A content engine receiving a request attempts to service it from its own local cache. If the requested information is not present, the content engine issues its own request to the originally targeted server to get the required information. A content engine retrieving the requested information forwards it to the requesting client and caches it to fulfill future requests, thus maximizing download performance and substantially reducing transmission costs.

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

Layer 2 Forwarding Redirection and Return

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

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

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

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

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

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

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

For information about Cisco ACNS commands used to configure Cisco Content Engines, see the Cisco ACNS Software Command Reference.

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

WCCP Mask Assignment

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

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

For information about Cisco ACNS commands used to configure Cisco Content Engines, see the Cisco ACNS Software Command Reference.

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

WCCP Hash Assignment

The Cisco ASR 1000 Series Aggregation Services Routers support hash assignment for IPv6 load balance across different content engines, but does not support mask assignment. However, it supports both hash assignment and mask assignment for IPv4.

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

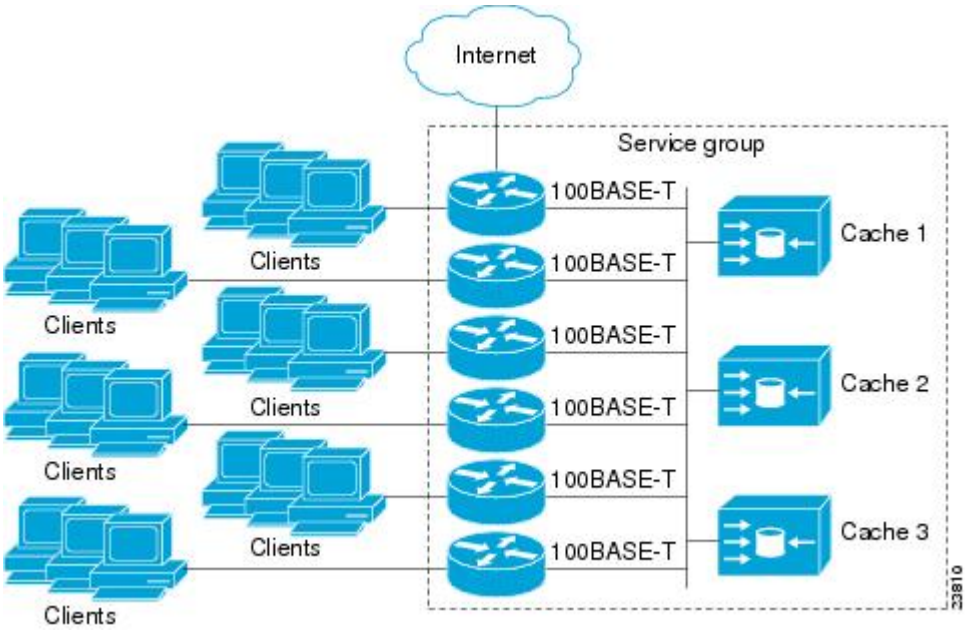
For information about Cisco ACNS commands used to configure Cisco Content Engines, see the Cisco ACNS Software Command Reference.

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

WCCPv2 Configuration

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

Figure 1: Cisco Content Engine Network Configuration Using WCCPv2



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

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

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

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

The following sequence of events details how WCCPv2 configuration works:

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

WCCPv2 Support for Services Other Than HTTP

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

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

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

WCCPv2 Support for Multiple Routers

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

WCCPv2 MD5 Security

WCCPv2 provides optional authentication that enables you to control which routers and content engines become part of the service group using passwords and the Hashed Message Authentication Code—Message

Digest (HMAC MD5) standard. Shared-secret MD5 one-time authentication (set using the **ip wccp** [**password** [**0** | **7**] *password*] global configuration command) enables messages to be protected against interception, inspection, and replay.

WCCPv2 Web Cache Packet Return

If a content engine is unable to provide a requested object it has cached due to error or overload, the content engine will return the request to the router for onward transmission to the originally specified destination server. WCCPv2 provides a check on packets that determines which requests have been returned from the content engine 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)

WCCPv2 Load Distribution

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

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

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

WCCP VRF Support

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

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

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

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

IPv6 WCCP Tunnel Interface

The use of GRE redirection results in the creation of new tunnel interfaces. You can display these tunnel interfaces by entering the **show ipv6 interface brief | include tunnel** command:

```
Device# show ipv6 interface brief | include tunnel

Tunnel0                2001::DB8:1::1      YES unset  up                    up
Tunnel1                2001::DB8:1::1      YES unset  up                    up
Tunnel2                2001::DB8:1::1      YES unset  up                    up
Tunnel3                2001::DB8:1::1      YES unset  up                    up
Device#
```

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

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

One tunnel is created for each service group that is using GRE redirection. One additional tunnel is created to provide an IP address that allows the other tunnel group interfaces to be unnumbered but still enabled for IPv6.

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

```
Device# show tunnel groups wccp

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

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

```
Device# show tunnel interface t0

Tunnel0
   Mode:multi-GRE/IP, Destination UNKNOWN, Source 2001::DB8:1::2
   Application ID 2: WCCP : service group 0 in "Default", ver v2, assgnmnt: hash-table
   Linestate - current up
   Internal linestate - current up, evaluated up

Device# show tunnel interface t1

Tunnel1
   Mode:multi-GRE/IP, Destination UNKNOWN, Source 2001::DB8:1::1
   Application ID 2: unspecified
   Linestate - current up
   Internal linestate - current up, evaluated up
```

```
Device# show tunnel interface t2

Tunnel2
   Mode:multi-GRE/IP, Destination UNKNOWN, Source 2001::DB8:1::1
   Application ID 2: WCCP : service group 318 in "Default", ver v2, assgnmnt: hash-table
   Linestate - current up
   Internal linestate - current up, evaluated up

Device# show tunnel interface t3

Tunnel3
   Mode:multi-GRE/IP, Destination UNKNOWN, Source 2001::DB8:1::1
   Application ID 2: WCCP : service group 317 in "Default", ver v2, assgnmnt: hash-table
   Linestate - current up
   Internal linestate - current up, evaluated up
Device#
```

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

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

```
Device# show adjacency t0

Protocol Interface                Address
IP       Tunnel0                  2001::DB8:1::1(3)

Device# show adjacency t0 encapsulation

Protocol Interface                Address
IPV6     Tunnel1                  2001:DB8:1::11(2)
  Encap length 48
  6000000000002FFF20010DB801000000
  00000000000000000120010DB800010000
  00000000000000110000883E00000000
  Provider: TUNNEL
IPV6     Tunnel1                  2001:DB8:1::12(2)
  Encap length 48
  6000000000002FFF20010DB801000000
  00000000000000000120010DB800010000
  00000000000000120000883E00000000
  Provider: TUNNEL

Device# show adjacency t0 detail

Protocol Interface                Address
IPV6     Tunnel1                  2001:DB8:1::11(2)
                                  0 packets, 0 bytes
                                  epoch 0
                                  sourced in sev-epoch 22
                                  Encap length 48
                                  6000000000002FFF20010DB801000000
                                  00000000000000000120010DB800010000
                                  00000000000000110000883E00000000
                                  Tun endpt
                                  Next chain element:
                                   punt

Device# show adjacency t0 internal
```

```
Protocol Interface                 Address
IPV6     Tunnel1                   2001:DB8:1::11(2)
                                   0 packets, 0 bytes
                                   epoch 0
                                   sourced in sev-epoch 22
                                   Encap length 48
                                   6000000000002FFF20010DB801000000
                                   0000000000000000120010DB800010000
                                   0000000000000000110000883E00000000
                                   Tun endpt
                                   Next chain element:
                                    punt
                                    parent oce 0x68C55B00
                                    frame originated locally (Null0)
                                   L3 mtu 0
                                   Flags (0x2808C6)
                                   Fixup disabled
                                   HWIDB/IDB pointers 0x200900DC/0x20090D98
                                   IP redirect disabled
                                   Switching vector: IPv6 midchain adjacency oce
                                   Next-hop cannot be inferred
                                   IP Tunnel stack to 2001:DB8:1::11 in Default (0x0)

Device#
```

WCCP Bypass Packets

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

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

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

WCCP Closed Services and Open Services

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

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

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

WCCP Outbound ACL Check

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

WCCP Service Groups

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

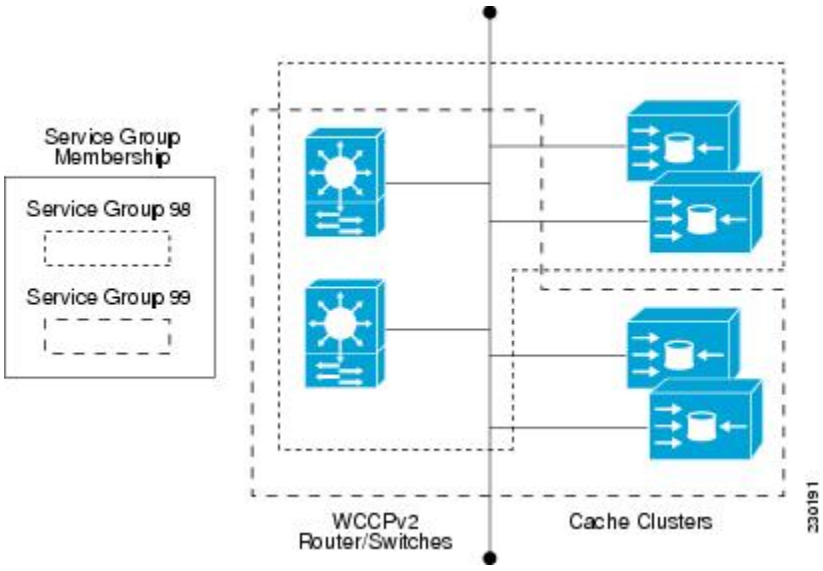
The nature of the selected traffic for redirection is defined by service groups (see figure below) specified on content engines and communicated to routers by using WCCP. The maximum number of service groups allowed across all VRFs is 256.

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

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

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

Figure 2: WCCP Service Groups



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

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

WCCP—Check All Services

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

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

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

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

will continue to attempt to match the packet against any remaining lower priority services configured on the interface.

WCCP Interoperability with NAT

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

WCCP—Configurable Router ID Overview

WCCP uses a router ID in its control messages that a WCCP client can use to uniquely identify a particular WCCP server. The router ID is an IP address and is used as the source address of any WCCP-generated Generic Routing Encapsulation (GRE) frames. Prior to the WCCP—Configurable Router ID feature, WCCP selected a router ID using an automatic mechanism; the highest reachable IP address on the system (or the highest loopback IP address, if there is one) was used as the WCCP router ID. The highest IP address on the system is not always the best choice as the router ID or as the source address of GRE frames. A change in addressing information on the system may cause the WCCP router ID to change unexpectedly. During this changeover period, WCCP clients briefly advertise the existence of two routers (the old router ID and the new router ID) and GRE frames are sourced from a different address.

The WCCP—Configurable Router ID feature enables you to define a WCCP source interface from which the router ID will be obtained. The IP address of this configured source interface is then used as the preferred WCCP router ID and WCCP GRE source address. When a WCCP router ID is manually configured, the router ID does not change when another IP address is added to the system. The router ID changes only when a new router ID is manually configured using the **ip wccp source- interface** or the **ipv6 wccp source- interface** command, or when the address on the manually configured interface is no longer valid.

WCCP Troubleshooting Tips

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

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

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

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

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

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

How to Configure WCCPv2—IPv6 Support

Configuring a General WCCPv2—IPv6 Session

Perform this task to configure a general IPv6 WCCPv2 session.

Until you configure a WCCP service using the **ipv6 wccp** {**web-cache** | *service-number*} global configuration command, WCCP is disabled on the router. The first use of a form of the **ipv6 wccp** command enables WCCP. By default WCCPv2 is used for services.

Using the **ipv6 wccp web-cache password** command, you can set a password for a router and the content engines in a service group. MD5 password security requires that each router and content engine that wants to join a service group be configured with the service group password. The password can be up to eight characters in length. Each content engine or router in the service group will authenticate the security component in a received WCCP packet immediately after validating the WCCP message header. Packets failing authentication will be discarded.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 wccp** [**vrf** *vrf-name*] **source-interface** *source-interface*
4. **ipv6 wccp** [**vrf** *vrf-name*] { **web-cache** | *service-number*} [**group-address** *group-address*] [**redirect-list** *access-list*] [**group-list** *access-list*] [**password** *password* [**0** | **7**]]
5. **interface** *type number*
6. **ipv6 wccp** [**vrf** *vrf-name*] {**web-cache** | *service-number*} **redirect** {**out** | **in**}
7. **exit**
8. **interface** *type number*
9. **ipv6 wccp redirect exclude in**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 wccp** [**vrf** *vrf-name*] **source-interface** *source-interface* **Example:**	Configures a preferred WCCP router ID.

	Command or Action	Purpose
	`Device(config)# ipv6 wccp source-interface GigabitEthernet 0/0/0`	
Step 4	**ipv6 wccp** [**vrf** *vrf-name*] { **web-cache** | *service-number*} [**group-address** *group-address*] [**redirect-list** *access-list*] [**group-list** *access-list*] [**password** *password* [**0** | **7**]] **Example:** `Device(config)# ipv6 wccp web-cache password password1`	Specifies a web-cache or dynamic service to enable on the router, specifies a VRF-name to associate with the service group, specifies the IP multicast address used by the service group, specifies any access lists to use, specifies whether to use MD5 authentication, and enables the WCCP service.
Step 5	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Targets an interface number for which the web cache service will run, and enters interface configuration mode.
Step 6	**ipv6 wccp** [**vrf** *vrf-name*] {**web-cache** | *service-number*} **redirect** {**out** | **in**} **Example:** `Device(config-if)# ipv6 wccp web-cache redirect in`	Enables packet redirection on an outbound or inbound interface using WCCP. • As indicated by the **out** and **in** keyword options, redirection can be specified for outbound interfaces or inbound interfaces.
Step 7	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode.
Step 8	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/2/0`	Targets an interface number on which to exclude traffic for redirection, and enters interface configuration mode.
Step 9	**ipv6 wccp redirect exclude in** **Example:** î `Device(config-if)# ipv6 wccp redirect exclude in`	(Optional) Excludes traffic on the specified interface from redirection.

Configuring Services for WCCPv2—IPv6

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

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. Enter one of the following commands:
   - **ipv6 wccp** [**vrf** *vrf-name*] *service-number* [**service-list** *service-access-list* **mode** {**open** | **closed**}]
   - **ipv6 wccp** [**vrf** *vrf-name*] **web-cache mode** {**open** | **closed**}
4. **ipv6 wccp check services all**
5. **ipv6 wccp** [**vrf** *vrf-name*] {**web-cache** | *service-number*}
6. **exit**

DETAILED STEPS

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

Step 3	Enter one of the following commands:
	• **ipv6 wccp** [**vrf** *vrf-name*] *service-number* [**service-list** *service-access-list* **mode** {**open**
		closed}]
		• **ipv6 wccp** [**vrf** *vrf-name*] **web-cache mode** {**open**
			closed}
			Example:
			`Device(config)# ipv6 wccp 90 service-list 120 mode closed`
			or
			`Device(config)# ipv6 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	**ipv6 wccp check services all** **Example:** `Device(config)# ipv6 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 **ipv6 wccp check services all** command is a global WCCP command that applies to all services and is not associated with a single service.

	Command or Action	Purpose
Step 5	**ipv6 wccp** [**vrf** *vrf-name*] {**web-cache** | *service-number*} **Example:** `Device(config)# ipv6 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:** `Device(config)# exit`	Exits to privileged EXEC mode.

Registering a Router to a Multicast Address for WCCPv2— IPv6

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

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

- Enable IP multicast routing using the **ipv6 multicast-routing** global configuration command.
- Enable the interfaces to which the cache engines will connect to receive multicast transmissions using the **ipv6 wccp group-listen** interface configuration command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 multicast-routing** [**vrf** *vrf-name*] [**distributed**]
4. **ipv6 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. **ipv6 wccp** [**vrf** *vrf-name*] {**web-cache** | *service-number*} **group-listen**

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 multicast-routing** [**vrf** *vrf-name*] [**distributed**] **Example:** `Device(config)# ipv6 multicast-routing`	Enables IP multicast routing.
Step 4	**ipv6 wccp** [**vrf** *vrf-name*] {**web-cache** | *service-number*} **group-address** *multicast-address* **Example:** `Device(config)# ipv6 wccp 99 group-address FF15::8000:1`	Specifies the multicast address for the service group.
Step 5	**interface** *type number* **Example:** `Device(config)# interface ethernet 0/0`	Enables the interfaces to which the content engines will connect to receive multicast transmissions for which the web cache service will run, and enters interface configuration mode.
Step 6	**ip pim** {**sparse-mode** | **sparse-dense-mode** | **dense-mode** [**proxy-register** {**list** *access-list* | **route-map** *map-name*}]} **Example:** `Device(config-if)# ip pim dense-mode`	(Optional) Enables Protocol Independent Multicast (PIM) on an interface. **Note** To ensure correct operation of the **ipv6 wccp group-listen** command, you must enter the **ip pim** command in addition to the **ipv6 wccp group-listen** command.
Step 7	**ipv6 wccp** [**vrf** *vrf-name*] {**web-cache** | *service-number*} **group-listen** **Example:** `Device(config-if)# ipv6 wccp 99 group-listen`	Configures an interface to enable or disable the reception of IP multicast packets for WCCP.

Using Access Lists for WCCPv2—IPv6 Service Group

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

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list** *access-list-number* **remark** *remark*
4. **access-list** *access-list-number* **permit** {*source* [*source-wildcard*] | **any**} [**log**]

5. **access-list** *access-list-number* **remark** *remark*
6. **access-list** *access-list-number* **deny** {*source* [*source-wildcard*] | **any**} [**log**]
7. Repeat some combination of Steps 3 through 6 until you have specified the sources on which you want to base your access list.
8. **ipv6 wccp** [**vrf** *vrf-name*] **web-cache group-list** *access-list*
9. **ipv6 wccp** [**vrf** *vrf-name*] **web-cache redirect-list** *access-list*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**access-list** *access-list-number* **remark** *remark* **Example:** `Device(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 in length can precede or follow an access list entry.
Step 4	**access-list** *access-list-number* **permit** {*source* [*source-wildcard*]	**any**} [**log**] **Example:** `Device(config)# access-list 1 permit 172.16.5.22 0.0.0.0`	Creates an access list that enables or disables traffic redirection to the cache engine and permits the specified source based on a source address and wildcard mask. • 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* string is omitted, a wildcard mask of 0.0.0.0 is assumed, meaning match on all bits of the source address. • Optionally use the keyword **any** as a substitute for the *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:**	(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
	`Device(config)# access-list 1 remark Give access to user1`	
Step 6	**access-list** *access-list-number* **deny** {*source* [*source-wildcard*] | **any**} [**log**] **Example:** `Device(config)# access-list 1 deny 172.16.7.34 0.0.0.0`	Denies the specified source based on a source address and wildcard mask. • If the *source-wildcard* string is omitted, a wildcard mask of 0.0.0.0 is assumed, meaning match on all bits of the source address. • Optionally use the abbreviation any as a substitute for the *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	**ipv6 wccp** [**vrf** *vrf-name*] **web-cache group-list** *access-list* **Example:** `Device(config) ipv6 wccp web-cache group-list 1`	Indicates to the router from which IP addresses of content engines to accept packets.
Step 9	**ipv6 wccp** [**vrf** *vrf-name*] **web-cache redirect-list** *access-list* **Example:** `Router(config)# ipv6 wccp web-cache redirect-list 1`	(Optional) Disables caching for certain clients.

Enabling the WCCP—IPv6 Outbound ACL Check

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

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 wccp** [**vrf** *vrf-name*] {**web-cache** | *service-number*} [**group-address** *multicast-address*] [**redirect-list** *access-list*] [**group-list** *access-list*] [**password** *password*]
4. **ipv6 wccp check acl outbound**

5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 wccp** [**vrf** *vrf-name*] {**web-cache** | *service-number*} [**group-address** *multicast-address*] [**redirect-list** *access-list*] [**group-list** *access-list*] [**password** *password*] **Example:** `Device(config)# ipv6 wccp web-cache`	Enables support for a Cisco content engine service group or any content engine service group and configures a redirect ACL list or group ACL.
Step 4	**ipv6 wccp check acl outbound** **Example:** `Device(config)# ipv6 wccp check acl outbound`	Checks the access control list (ACL) for egress interfaces for packets redirected by WCCP.
Step 5	**exit** **Example:** `Device(config)# exit`	Exits global configuration.

Enabling WCCPv2—IPv6 Interoperability with NAT

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 nat inside**
5. **ipv6 wccp** *service-number* **redirect in**
6. **exit**
7. **interface** *type number*
8. **ipv6 nat outside**
9. **ipv6 wccp** *service-number* **redirect in**
10. **exit**

11. **interface** *type number*
12. **ipv6 nat inside**
13. **ipv6 wccp redirect exclude in**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface ethernet 1`	Specifies an interface on which to enable NAT and enters interface configuration mode. • This is the LAN-facing interface.
Step 4	**ipv6 nat inside** **Example:** `Device(config-if)# ipv6 nat inside`	Designates that traffic originating from or destined for the interface is subject to NAT and indicates that the interface is connected to the inside network (the network subject to NAT translation).
Step 5	**ipv6 wccp** *service-number* **redirect in** **Example:** `Device(config-if)# ipv6 wccp 61 redirect in`	Enables packet redirection on an inbound interface using WCCP.
Step 6	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 7	**interface** *type number* **Example:** `Device(config)# interface ethernet 2`	Specifies an interface on which to enable NAT and enters interface configuration mode. • This is the WAN-facing interface.
Step 8	**ipv6 nat outside** **Example:** `Device(config-if)# ipv6 nat outside`	Designates that traffic originating from or destined for the interface is subject to NAT and indicates that the interface is connected to the outside network.

	Command or Action	Purpose
Step 9	**ipv6 wccp** *service-number* **redirect in** **Example:** `Device(config-if)# ipv6 wccp 62 redirect in`	Enables packet redirection on an inbound interface using WCCP.
Step 10	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 11	**interface** *type number* **Example:** `Device(config)# interface ethernet 3`	Specifies an interface on which to enable NAT and enters interface configuration mode. • This is the WAAS-facing interface.
Step 12	**ipv6 nat inside** **Example:** `Device(config-if)# ipv6 nat inside`	Designates that traffic originating from or destined for the interface is subject to NAT and indicates that the interface is connected to the inside network (the network subject to NAT translation).
Step 13	**ipv6 wccp redirect exclude in** **Example:** `Device(config-if)# ipv6 wccp redirect exclude in`	Configures an interface to exclude packets received on an interface from being checked for redirection.

Verifying and Monitoring WCCPv2—IPv6 Configuration Settings

SUMMARY STEPS

1. **enable**
2. **show ipv6 wccp** [**vrf** *vrf-name*] [*service-number* | **web-cache**] [**detail** | **view**]
3. **show ipv6 interface**
4. **more system:running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ipv6 wccp** [**vrf** *vrf-name*] [*service-number*	**web-cache**] [**detail**	**view**] **Example:**	(Optional) Displays global information related to WCCP, including the protocol version currently running, the number of content engines in the router service group, which content

	Command or Action	Purpose
	`Device# show ipv6 wccp 24 detail`	engine group is allowed to connect to the router, and which access list is being used. The argument and keywords are as follows: • *service-number*—(Optional) Dynamic number of the web-cache service group being controlled by the content engine. The range is from 0 to 99. For web caches that use Cisco Content Engines, the reverse proxy service is indicated by a value of 99. • **web-cache**—(Optional) Statistics for the web-cache service. • **detail**—(Optional) Other members of a particular service group or web cache that have or have not been detected. • **view**—(Optional) Information about a router or all web caches.
Step 3	**show ipv6 interface** **Example:** `Device# show ipv6 interface`	(Optional) Displays status about whether any **ip wccp redirection** commands are configured on an interface; for example, "Web Cache Redirect is enabled / disabled."
Step 4	**more system:running-config** **Example:** `Device# more system:running-config`	(Optional) Displays contents of the currently running configuration file (equivalent to the **show running-config** command).

Configuration Examples for WCCPv2—IPv6 Support

Example: Configuring a General WCCPv2—IPv6 Session

```
Device# configure terminal
Device(config)# ipv6 wccp web-cache password password1
Device(config)# ipv6 wccp source-interface GigabitEthernet 0/1/0
Device(config)# ipv6 wccp check services all
 Configures a check of all WCCP services.
Device(config)# interface GigabitEthernet 0/1/0
Device(config-if)# ipv6 wccp web-cache redirect in
Device(config-if)# exit
Device(config)# interface GigabitEthernet 0/2/0
Device(config-if)# ipv6 wccp redirect exclude in
Device(config-if)# exit
```

Example: WCCPv2—IPv6—Setting a Password for a Router and Content Engines

```
Device# configure terminal
Device(config)# ipv6 wccp web-cache password password1
```

Example: WCCPv2—IPv6—Configuring a Web Cache Service

```
Device# configure terminal
Device(config)# ipv6 wccp web-cache
Device(config)# interface GigabitEthernet 0/1/0
Device(config-if)# ipv6 wccp web-cache redirect in
Device(config-if)# exit
Device# copy running-config startup-config
```

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

```
Device# configure terminal
Device(config)# interface GigabitEthernet 0/1/0
Device(config-if)# ipv6 wccp web-cache redirect in
Device(config-if)# exit
Device# show ip interface GigabitEthernet 0/1/0
.
.
.
WCCP Redirect inbound is enabled
WCCP Redirect exclude is disabled
.
.
.
```

Example: WCCPv2—IPv6—Running a Reverse Proxy Service

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

```
Device# configure terminal
Device(config)# ipv6 wccp 99
Device(config)# interface GigabitEthernet 0/1/0
Device(config-if)# ipv6 wccp 99 redirect out
```

Example: WCCPv2—IPv6—Registering a Router to a Multicast Address

```
Device# configure terminal
Device(config)# ipv6 wccp web-cache group-address 224.1.1.100
Device(config)# interface gigabitethernet 0/1/0
Device(config-if)# ipv6 wccp web cache group-listen
```

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

```
Device# configure terminal
Device(config)# ipv6 wccp 99 group-address 224.1.1.1
Device(config)# interface gigabitethernet 0/1/0
Device(config-if)# ipv6 wccp 99 redirect out
```

Example: WCCPv2—IPv6—Using Access Lists for a WCCPv2 IPv6 Service Group

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

```
Device(config)# access-list 10 permit host 10.1.1.1
Device(config)# access-list 10 permit host 10.1.1.2
Device(config)# access-list 10 permit host 10.1.1.3
Device(config)# ipv6 wccp web-cache group-list 10
```

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

```
Device(config)# ipv6 wccp web-cache redirect-list 120
Device(config)# access-list 120 deny tcp host 10.1.1.1 any
Device(config)# access-list 120 deny tcp any host 10.3.1.1
Device(config)# access-list 120 permit ip any any
```

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

```
Device(config)# access-list 100 deny ip any host 209.165.200.224
Device(config)# access-list 100 permit ip any any
Device(config)# ipv6 wccp web-cache redirect-list 100
Device(config)# interface gigabitethernet 0/1/0
Device(config-if)# ipv6 wccp web-cache redirect in
```

Example: WCCPv2—IPv6—Configuring Outbound ACL Check

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

```
Device(config)# ipv6 wccp web-cache
Device(config)# ipv6 wccp check acl outbound
Device(config)# interface gigabitethernet 0/1/0
Device(config-if)# ip access-group 10 out
Device(config-if)# exit
Device(config)# ipv6 wccp web-cache redirect-list redirect-out
Device(config)# access-list 10 deny 10.0.0.0 0.255.255.255
Device(config)# access-list 10 permit any
```

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

Example: WCCPv2—IPv6—Enabling WCCP Interoperability with NAT

```
Device(config)# interface ethernet1 ! This is the LAN-facing interface
Device(config-if)# ipv6 nat inside
Device(config-if)# ipv6 wccp 61 redirect in
Device(config-if)# exit
Device(config)# interface ethernet2 ! This is the WAN-facing interface
Device(config-if)# ipv6 nat outside
Device(config-if)# ipv6 wccp 62 redirect in
Device(config-if)# exit
Device(config)# interface ethernet3 ! This is the WAAS-facing interface
Device(config-if)# ipv6 nat inside
Device(config-if)# ipv6 wccp redirect exclude in
```

Example: WCCPv2—IPv6—Verifying WCCP Settings

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

```
Device# more system:running-config

 Building configuration...
 Current configuration:
 !
 version 12.0
 service timestamps debug uptime
 service timestamps log uptime
 no service password-encryption
 service udp-small-servers
 service tcp-small-servers
 !
 hostname router4
 !
 enable secret 5 $1$nSVy$faliJsVQXVPW.KuCxZNTh1
 enable password password1
 !
 ip subnet-zero
 ipv6 wccp web-cache
 ipv6 wccp 99
 ip domain-name cisco.com
 ip name-server 10.1.1.1
 ip name-server 10.1.1.2
 ip name-server 10.1.1.3
 !
 !
 !
 interface GigabitEthernet0/1/1
 ip address 10.3.1.2 255.255.255.0
 no ip directed-broadcast
 ipv6 wccp web-cache redirect in
 ipv6 wccp 99 redirect in
 no ip route-cache
 no ip mroute-cache
 !
 interface GigabitEthernet0/1/0
 ip address 10.4.1.1 255.255.255.0
 no ip directed-broadcast
```

```
ipv6 wccp 99 redirect in
no ip route-cache
no ip mroute-cache
!
interface Serial0
no ip address
no ip directed-broadcast
no ip route-cache
no ip mroute-cache
shutdown
!
interface Serial1
no ip address
no ip directed-broadcast
no ip route-cache
no ip mroute-cache
shutdown
!
ip default-gateway 10.3.1.1
ip classless
ip route 0.0.0.0 0.0.0.0 10.3.1.1
no ip http server
!
!
!
line con 0
transport input none
line aux 0
transport input all
line vty 0 4
password password1
login
!
end
```

The following example shows how to display global statistics related to WCCP:

```
Device# show ipv6 wccp web-cache detail

WCCP Client information:
WCCP Client ID:      10.1.1.2
Protocol Version:    2.0
State:               Usable
Redirection:         L2
Packet Return:       L2
Packets Redirected:  0
Connect Time:        00:20:34
Assignment:          MASK
Mask  SrcAddr    DstAddr    SrcPort DstPort
----  -------    -------    ------- -------
0000: 0x00000000 0x00001741 0x0000  0x0000
Value SrcAddr    DstAddr   SrcPort DstPort CE-IP
----- -------    -------   ------- ------- -----
0000: 0x00000000 0x00000000 0x0000 0x0000 0x3C010102 (10.1.1.2)
0001: 0x00000000 0x00000001 0x0000 0x0000 0x3C010102 (10.1.1.2)
0002: 0x00000000 0x00000040 0x0000 0x0000 0x3C010102 (10.1.1.2)
0003: 0x00000000 0x00000041 0x0000 0x0000 0x3C010102 (10.1.1.2)
0004: 0x00000000 0x00000100 0x0000 0x0000 0x3C010102 (10.1.1.2)
0005: 0x00000000 0x00000101 0x0000 0x0000 0x3C010102 (10.1.1.2)
0006: 0x00000000 0x00000140 0x0000 0x0000 0x3C010102 (10.1.1.2)
```

For more information about the **show ip wccp web-cache** command, see the *Cisco IOS IP Application Services Command Reference* document.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Commands List, All Releases*
IP addressing and services commands and configuration tasks	• *Cisco IOS 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*

Technical Assistance

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

Feature Information for WCCPv2—IPv6 Support

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

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

Table 1: Feature Information for WCCPv2 —IPv6 Support

Feature Name	Releases	Feature Information
WCCPv2—IPv6 Support	15.1(1)SY1 15.2(3)T	This feature introduces support for Web Cache Communication Protocol version 2 (WCCPv2) in an IPv6 environment. WCCP is a Cisco-developed content-routing technology that intercepts IP packets and redirects those packets to a destination other than that specified in the IP packet. Multiple routers can use WCCPv2 to service a content engine cluster. In WCCPv1, only one router could redirect content requests to a cluster. The following commands were added: **clear ipv6 wccp**, **clear wccp**, **debug ipv6 wccp**, **debug wccp**, **ipv6 wccp**, **ipv6 wccp check acl outbound**, **ipv6 wccp check services all**, **ipv6 wccp group-listen**, **ipv6 wccp redirect**, **ipv6 wccp redirect exclude in ipv6 wccp source-interface**, **show ipv6 wccp**, **show ipv6 wccp global counters**, **show wccp**, **show wccp global counters**, **show platform software wccp** *service-number* **ipv6 counters**, **show platform software wccp rp active** *service-number* **ipv6** , **show platform software wccp fp active** *service-number* **ipv6** , **show platform hardware qfp active feature wccp service id** *service-number* **ipv6** .