CHAPTER 4

Network Integration and Interception

This chapter provides an in-depth review of the network integration and interception capabilities of Cisco WAAS. The chapter begins by describing the options for basic connectivity, including link aggregation and NIC teaming. This is followed by a discussion of the interception methods available for redirecting traffic to a WAAS device for optimization. The techniques and methods discussed in this chapter form the foundation of the design and deployment solutions presented in subsequent chapters of this book.

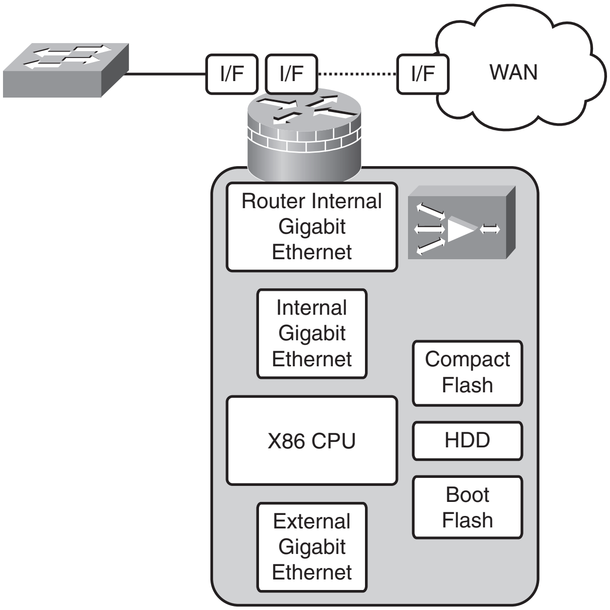
Interface Connectivity

Each Cisco Wide Area Application Engine (WAE) has two 10/100/1000BASE-T Ethernet interfaces. In a typical deployment, each WAE is connected using a single interface to a LAN switch or router. By default, WAE interfaces auto-negotiate their speed and duplex. You can optionally configure the interface speed to 10 or 100 Mbps. In order for the interface speed to run at 1000 Mbps, it must be configured for auto-negotiation. The duplex of the interface is also configurable.

CAUTION Do not configure WAE interfaces for half-duplex operation. The collision and retransmission behavior of half-duplex Ethernet has a negative effect on WAE performance.

The router-integrated network module (NME-WAE) is also equipped with two Ethernet interfaces, but only one interface is accessible externally. The other interface connects directly to the internal router PCI bus at 1 Gbps and is configured in a similar manner as an external interface would be configured on a WAE appliance. Unlike a WAE appliance configuration, the WAE interface IP address and default gateway are configured as part of the Cisco IOS interface configuration where the NME-WAE is installed. Figure 4-1 shows the physical interface layout on the router-integrated NME-WAE.

Figure 4-1 *NME-WAE Physical Interface Connectivity*


I/F
I/F
I/F
WAN
Router Internal Gigabit Ethernet
Internal Gigabit Ethernet
Compact Flash
X86 CPU
HDD
Boot Flash
External Gigabit Ethernet

The WAE interface configuration options are similar to the Cisco IOS configuration options, both in terms of function and CLI commands. Example 4-1 shows the interface configuration options available on a WAE.

Example 4-1 *WAE Interface Configuration Options*

```
WAE-612(config)# interface gigabitEthernet 1/0
WAE-612(config-if)# ?
  autosense      Interface autosense
  bandwidth      Interface bandwidth
  cdp            Cisco Discovery Protocol Interface Config commands
  channel-group  Configure EtherChannel group
  description    Interface specific description
  exit           Exit from this submode
  full-duplex    Interface fullduplex
  half-duplex    Interface halfduplex
  ip             Interface Internet Protocol Config commands
  mtu            Set the interface Maximum Transmission Unit (MTU)
  no             Negate a command or set its defaults
  shutdown       Shutdown the specific interface
  standby        Standby interface config commands
WAE-612(config-if)#
```

One of the interface configuration commands that behaves differently in WAAS versus IOS is the **bandwidth** command. The **bandwidth** interface configuration command in WAAS is used to specify the speed of the interface when auto-negotiation is disabled. The way in which the **standby** interface command is used is another important difference between WAAS and IOS. In IOS, the **standby** interface command is used for configuring the Hot Standby Router Protocol (HSRP) feature, while in WAAS it is used to configure the standby interface feature, described in the next section. You can see from the output in Example 4-1 that the remaining WAAS interface configuration commands are similar to the corresponding IOS interface configuration commands.

You can explicitly configure the interface with an IP address and subnet mask, or the WAE can acquire an IP address using DHCP. Each WAE interface can also be configured with multiple secondary IP addresses. It is also possible for the same interface to acquire an IP address through DHCP, and have multiple secondary IP addresses statically configured. By default, the interfaces on a WAE are administratively disabled, and are automatically enabled when a valid IP address is configured.

Each WAE interface is primarily referenced using the standard Cisco IOS interface naming scheme:

<interface-name> <slot/port>

This is how WAE interfaces are referred to during configuration through the CLI or GUI. The interfaces also have an internal name by which the Linux operating system knows them. Table 4-1 shows the mapping between the internal and external interface names.

Table 4-1 *WAE External and Internal Interface Names*

IOS Name	Internal Name
gigabitEthernet 1/0	eth0
gigabitEthernet 2/0	eth1

Understanding the internal name of an interface is useful for understanding system log messages and using internal operating system tools, such as Ethereal or Tcpdump, which are useful for capturing traffic for offline analysis.

Just like the interface configuration, the outputs of interface **show** commands in WAAS are similar to Cisco IOS. Example 4-2 shows the output from the **show interface** command in WAAS.

Example 4-2 *WAE* **show interface** *Command Output*

```
AST6-CCO-02# show interface gigabitEthernet 1/0
Type:Ethernet
Ethernet address:00:11:25:AB:43:28
Internet address:10.88.81.2
```

continues

Example 4-2 *WAE* **show interface** *Command Output (Continued)*

```
Broadcast address:10.88.81.15
Netmask:255.255.255.240
Maximum Transfer Unit Size:1500
Metric:1
Packets Received: 966044
Input Errors: 0
Input Packets Dropped: 0
Input Packets Overruns: 0
Input Packets Frames: 0
Packet Sent: 1046794
Output Errors: 0
Output Packets Dropped: 0
Output Packets Overruns: 0
Output Packets Carrier: 0
Output Queue Length:1000
Collisions: 0
Base address:0x2000
Flags:UP BROADCAST RUNNING MULTICAST
Mode: autoselect, full-duplex, 100baseTX
AST6-CCO-02#
```

In addition to the normal interface information, such as IP address, Ethernet address, and counters, each interface also has a set of flags. These flags are the same flags that can be seen in the output of the **ifconfig** command in Linux. The two most important flags are UP and RUNNING. The presence of the UP flag indicates that the interface is administratively enabled. The presence of the RUNNING flag indicates that line protocol on the interface is operational.

Link Aggregation Using EtherChannel

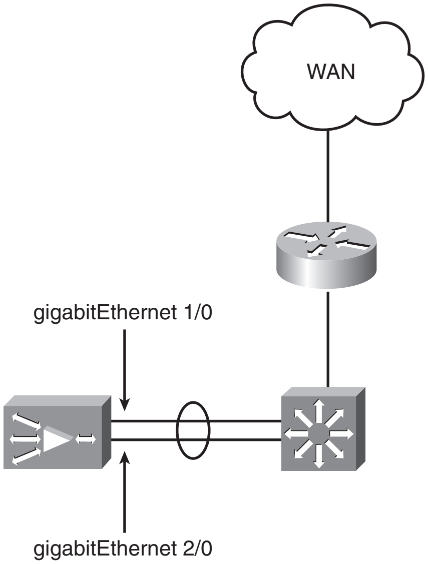
To increase the available interface bandwidth for a WAE, Cisco WAAS supports EtherChannel. EtherChannel allows for the grouping of multiple physical interfaces to create a single "virtual" interface. The virtual interface, which functions as a single interface, has the aggregate bandwidth of the available physical interfaces in the channel group. EtherChannel is useful when the output from a single WAE exceeds the physical limitations of a single interface. For example, some remote sites may only have 100-Mbps LAN connections available, whereas the traffic from a single WAE can easily exceed 100 Mbps. In these situations, using EtherChannel to group both physical WAE interfaces together provides 200 Mbps of usable interface bandwidth.

In addition to increasing the available interface bandwidth, the EtherChannel has automatic failure and recovery detection based on the link state of each individual interface. In the event of a single interface failure, traffic continues to pass over the remaining interface in the channel group. The EtherChannel interface uses the MAC address from one of the physical interfaces in the group. The same MAC address is used persistently for the EtherChannel interface, even if the physical interface associated with that MAC address goes down. The formation of an EtherChannel in WAAS is based purely on device configuration. WAAS does not support Cisco Port Aggregation Protocol (PAgP) or 802.3ad Link Aggregation Control Protocol (LACP). When configuring EtherChannel between a WAE and a LAN switch, the channel mode on the LAN switch should be set to On for the WAE EtherChannel.

By default, the WAE load balances packets across all available interfaces in the channel group using a round-robin algorithm. WAAS also supports load balancing using destination IP address or destination MAC address.

Figure 4-2 shows a WAE connected to a single LAN switch using EtherChannel.

Figure 4-2 *WAE Connected Using EtherChannel Feature*



EtherChannel Configuration

Configuring EtherChannel in WAAS involves the following steps:

Step 1 Create a virtual PortChannel interface.

Step 2 Configure an IP address and subnet mask for the PortChannel interface.

Step 3 Assign the physical interfaces to the PortChannel.

Example 4-3 shows a basic EtherChannel configuration.

Example 4-3 *WAE EtherChannel Configuration*

```
!
interface PortChannel 1
 description ** EtherChannel Link to Switch ABC ***
 ip address 10.10.10.5 255.255.255.0
 exit
!
interface GigabitEthernet 1/0
 channel-group 1
 exit
interface GigabitEthernet 2/0
 channel-group 1
 exit
!
```

You should observe the following limitations when configuring EtherChannel in WAAS:

- Both interfaces in the channel group must run at the same speed.
- Access control lists (ACL) are still applied to each physical interface.

The load-balancing algorithm used for distributing traffic across the EtherChannel is configured using the following command:

```
port-channel load-balance dst-ip | dst-mac | round-robin
```

The command is configured globally and applies to all PortChannels on the WAE.

To check the status of the PortChannel interface, use the **show interface PortChannel** *channel-number* command. Example 4-4 demonstrates the output of this command.

Example 4-4 *WAAS* **show interface PortChannel** *Output*

```
AST6-CCO-01# show interface PortChannel 1
Interface PortChannel 1 (2 physical interface(s)):
        GigabitEthernet 1/0 (active)
        GigabitEthernet 2/0 (active)
---------------------
Type:Ethernet
Ethernet address:00:11:25:AB:43:32
Internet address:10.88.80.130
Broadcast address:10.88.80.255
Netmask:255.255.255.128
Maximum Transfer Unit Size:1500
Metric:1
Packets Received: 815996
Input Errors: 0
Input Packets Dropped: 0
Input Packets Overruns: 0
Input Packets Frames: 0
```

Example 4-4 *WAAS* **show interface PortChannel** *Output (Continued)*

```
Packet Sent: 321842
Output Errors: 0
Output Packets Dropped: 0
Output Packets Overruns: 0
Output Packets Carrier: 0
Output Queue Length:0
Collisions: 0
Flags:UP BROADCAST RUNNING MASTER MULTICAST
AST6-CCO-01#
```

Each member of the channel group, along with the status of the interface, is shown at the beginning of the output. The MASTER flag in the example output indicates that this is the virtual EtherChannel interface. Also notice the Ethernet address, which is taken from one of the physical interfaces in the channel group. Example 4-5 demonstrates the same command for each physical interface in the channel group.

Example 4-5 *Channel Group Member Interface Output*

```
AST6-CCO-01# show interface gigabitEthernet 1/0
Type:Ethernet
Ethernet address:00:11:25:AB:43:32
Internet address:10.88.80.130
Broadcast address:10.88.80.255
Netmask:255.255.255.128
Maximum Transfer Unit Size:1500
Metric:1
Packets Received: 816176
Input Errors: 0
Input Packets Dropped: 0
Input Packets Overruns: 0
Input Packets Frames: 0
Packet Sent: 321880
Output Errors: 0
Output Packets Dropped: 0
Output Packets Overruns: 0
Output Packets Carrier: 0
Output Queue Length:1000
Collisions: 0
Base address:0x2000
Flags:UP BROADCAST RUNNING SLAVE MULTICAST
Mode: full-duplex, 100baseTX
AST6-CCO-01#
AST6-CCO-01# show interface gigabitEthernet 2/0
Type:Ethernet
Ethernet address:00:11:25:AB:43:32
Internet address:10.88.80.130
Broadcast address:10.88.80.255
Netmask:255.255.255.128
```

continues

Example 4-5 *Channel Group Member Interface Output (Continued)*

```
Maximum Transfer Unit Size:1500
Metric:1
Packets Received: 0
Input Errors: 0
Input Packets Dropped: 0
Input Packets Overruns: 0
Input Packets Frames: 0
Packet Sent: 0
Output Errors: 0
Output Packets Dropped: 0
Output Packets Overruns: 0
Output Packets Carrier: 0
Output Queue Length:1000
Collisions: 0
Base address:0x3400
Flags:UP BROADCAST SLAVE MULTICAST
Mode: autoselect
AST6-CCO-01#
```

The command output for each physical interface is the same as it is without EtherChannel configured, with the following exceptions:

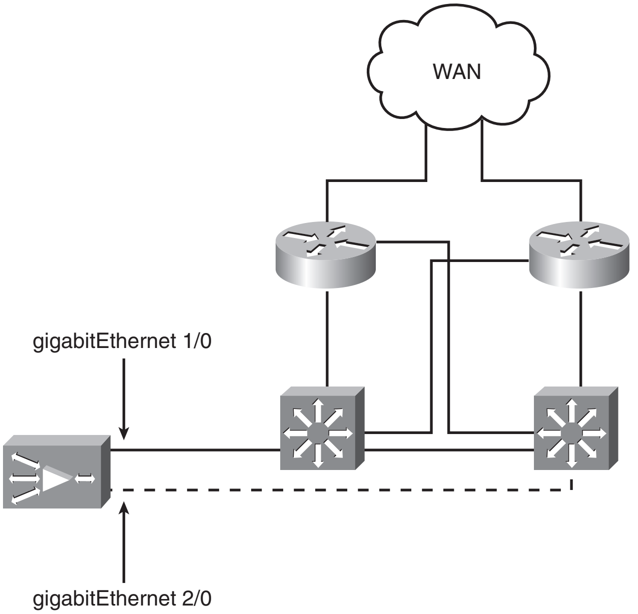
- The SLAVE flag is set, indicating that the interface is part of an EtherChannel group.
- The Ethernet address for each interface is the same, and matches the MAC address used by the virtual EtherChannel interface.

Using the Standby Interface Feature

When you do not require increased interface bandwidth but desire interface redundancy, you can use the standby interface feature. The standby interface feature configures both physical interfaces on the WAE in an active/standby failover pair. At any point in time, only one of the interfaces is active and passing traffic. The second interface, or standby interface, is passively waiting to take over in the event that the active interface fails. When the active interface fails, the standby interface takes over the active role. When the previously active interface recovers, it assumes the standby role. The interface with the highest priority is preferred as the active interface. The priority is configurable.

The standby interface feature has become a popular choice for deployments because the WAE can be physically connected to two different LAN switches. This prevents the failure of a single LAN switch or switchport from disrupting the operation of the WAE. Figure 4-3 shows an example of a WAE connected using the standby interface feature.

Figure 4-3 *WAE Connected Using Standby Interface Feature*



The failure of the active interface in the standby group is detected using three different methods. The first method monitors the link state of the active interface. Line protocol is up when the RUNNING flag is present on the Ethernet interface. If line protocol fails on the active interface, the interface is marked as down.

The second method uses an ICMP ping to check availability of the default gateway configured on the WAE. An ICMP echo request is sent once every 10 seconds to the configured default gateway. If there is a response to the echo request, the interface is considered up. If there is no response to the echo request, five more echo requests are sent. If at least three responses are received, the interface is considered up. Otherwise, the interface is considered failed, and the interface is marked as down.

The final method available for monitoring the health of the active interface uses the interface error count to determine if an unacceptable number of errors have been seen on the interface. The error count is the absolute number of transmit and receive errors on the active interface. This check is disabled by default, but can be enabled using the following command:

```
errors 1-2147483647
```

The interface state and error counts (when configured) are checked once every 10 seconds. If the active link fails or the error count threshold is exceeded, the interface with the next highest priority is activated. When the failed interface recovers, it becomes the standby interface for the group. The standby interface does not have a preempt capability. When a new interface is activated, the WAE generates a gratuitous ARP to update the MAC address for the shared IP on all other devices on the same subnet. This prevents devices from sending traffic to the shared IP address on the WAE to the MAC address of the failed WAE interface.

Standby Interface Configuration

Configuring the standby interface feature in WAAS involves the following steps:

Step 1 Create a virtual standby interface.

Step 2 Configure an IP address and subnet mask for the standby interface.

Step 3 Assign the physical interfaces to the standby group.

Example 4-6 shows a basic standby interface configuration.

Example 4-6 *WAE Standby Interface Configuration*

```
!
interface Standby 1
 ip address 10.88.80.130 255.255.255.128
 exit
!
interface GigabitEthernet 1/0
 standby 1 priority 105
 exit
interface GigabitEthernet 2/0
 standby 1
 exit
!
```

You should observe the following limitations when configuring the standby interface feature within WAAS:

- The physical interfaces in the standby group do not require IP addresses.
- The standby interface feature does not have a preempt capability.

Each physical interface can be assigned a numeric priority between 1 and 2,147,483,647. The default standby priority for an interface is 100. The virtual standby interface uses the MAC address of the active interface. When the active interface fails and the standby interface takes over, the WAE generates a gratuitous ARP request to update the adjacent devices with the new MAC address associated with the WAE IP address.

To check the status of the standby interface, use the **show interface Standby** *standby-interface-number* command. Example 4-7 shows the output of this command.

Example 4-7 *WAAS* **show interface Standby** *Interface Output*

```
AST6-CCO-01# show interface Standby 1
Standby Group: 1
        IP address: 10.88.80.130, netmask: 255.255.255.128
        Member interfaces:
                GigabitEthernet 1/0     priority: 105
                GigabitEthernet 2/0     priority: 100
        Active interface: GigabitEthernet 1/0
AST6-CCO-01#
```

Each member of the standby group, as well as the status of the interface, is shown in the output. The current active interface is also displayed. The output for each physical interface is shown in Example 4-8.

Example 4-8 *Standby Group Member Interface Output*

```
AST6-CCO-01# show interface gigabitEthernet 1/0
Type:Ethernet
Ethernet address:00:11:25:AB:43:32
Internet address (secondary): 10.88.80.130 Netmask: 255.255.255.128
Maximum Transfer Unit Size:1500
Metric:1
Packets Received: 819025
Input Errors: 0
Input Packets Dropped: 0
Input Packets Overruns: 0
Input Packets Frames: 0
Packet Sent: 322492
Output Errors: 0
Output Packets Dropped: 0
Output Packets Overruns: 0
Output Packets Carrier: 0
Output Queue Length:1000
Collisions: 0
Base address:0x2000
Flags:UP BROADCAST RUNNING MULTICAST
Mode: full-duplex, 100baseTX
AST6-CCO-01#
AST6-CCO-01# show interface gigabitEthernet 2/0
Type:Ethernet
Ethernet address:00:11:25:AB:43:33
Maximum Transfer Unit Size:1500
Metric:1
Packets Received: 0
Input Errors: 0
Input Packets Dropped: 0
```

continues

Example 4-8 *Standby Group Member Interface Output (Continued)*

```
Input Packets Overruns: 0
Input Packets Frames: 0
Packet Sent: 0
Output Errors: 0
Output Packets Dropped: 0
Output Packets Overruns: 0
Output Packets Carrier: 0
Output Queue Length:1000
Collisions: 0
Base address:0x3400
Flags:UP BROADCAST MULTICAST
Mode: autoselect
AST6-CCO-01#
```

In this output, the only indication that the interface is a member of a standby group is the secondary IP address, which matches the IP address configured on the virtual standby interface.

Interception Techniques and Protocols

There are two approaches for leveraging the network infrastructure to intercept and redirect traffic to WAAS for optimization. The first method relies on interception protocols or routing configuration used by the networking components (routers and switches) to selectively intercept traffic and redirect it to the WAAS infrastructure. This method is referred to as off-path interception. The most common method for off-path network interception is the Web Cache Communication Protocol, or WCCPv2.

The second method places the WAE physically inline between two network elements, most commonly a router and LAN switch. All traffic between the two network elements is passed through the WAE, which can then selectively intercept traffic for optimization. This method is referred to as *in-path interception*, because the WAE is physically placed in the data path between the clients and servers.

This section discusses both off-path (WCCPv2) and in-path interception in detail. It also discusses other interception options for specific use cases, such as policy-based routing (PBR) and content switching. These additional interception options add to the flexibility with which WAAS can be integrated into existing network infrastructures of all sizes.

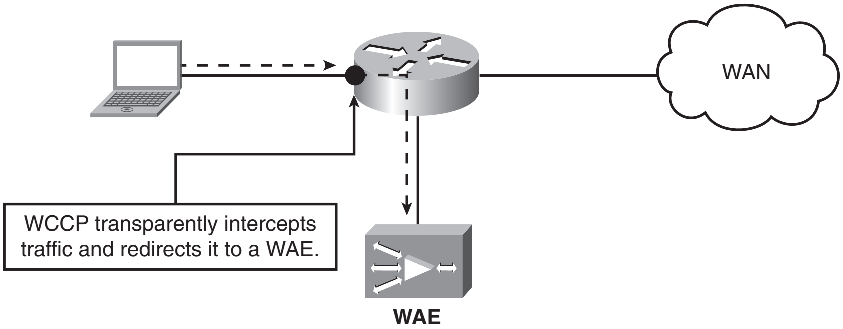
Web Cache Communication Protocol

This section does not provide an exhaustive reference for the WCCPv2 protocol. Rather, it provides enough information about the protocol background and concepts to enable you to understand the WCCPv2 implementation in Cisco WAAS. For an in-depth understanding of the WCCPv2 protocol, you are encouraged to read the WCCPv2 protocol draft. The full WCCPv2 IETF draft is available online at http://www.wrec.org/Drafts/draft-wilson-wrec-wccp-v2-00.txt.

WCCP Overview

WCCP is a transparent interception protocol first developed by Cisco Systems, Inc. in 1997. WCCP is a control plane protocol that runs between devices running Cisco IOS and WCCP "clients" such as WAAS. The protocol enables the network infrastructure to selectively intercept traffic based on IP protocol and port numbers, and redirect that traffic to a WCCP client. WCCP is considered transparent, because it allows for local interception and redirection of traffic without any configuration changes to the clients or servers. WCCP has built-in load-balancing, scalability, fault-tolerance, and service assurance (fail open) mechanisms. Figure 4-4 shows the basic functions of WCCP.

Figure 4-4 *Basic WCCP Functionality*


WAN
WCCP transparently intercepts traffic and redirects it to a WAE.
WAE

The current version, WCCPv2, is used by Cisco WAAS to transparently intercept and redirect all TCP traffic, regardless of port. The following section describes the basic WCCPv2 concepts and how they are specifically used by Cisco WAAS.

Service Groups

The routers and WAEs participating in the same service constitute a service group. A service group defines a set of characteristics about what types of traffic should be intercepted, as well as how the intercepted traffic should be handled. There are two types of service groups:

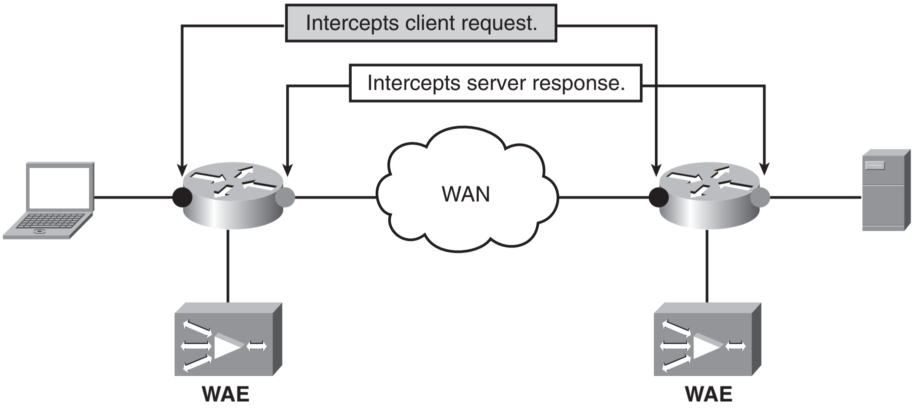
- Well-known services
- Dynamic services

Well-known services, also referred to as static services, have a fixed set of characteristics that are known by both IOS and WCCPv2 client devices. There is currently a single well-known service called web-cache. This service redirects all TCP traffic with a destination port of 80. The characteristics of a dynamic service are initially only known to the WCCPv2 clients within the service group. The characteristics of the service group are communicated to the IOS devices by the first WCCPv2 client device to join the service group.

A unique service ID identifies service groups, which is a number from 0 to 255. Service IDs 0 to 50 are reserved for well-known services.

The WCCPv2 implementation in WAAS supports a single dynamic WCCPv2 service, the tcp-promiscuous service. Although referred to in WAAS as a single service, the tcp-promiscuous service is in fact two different services. The two service IDs enabled with the tcp-promiscuous service are 61 and 62. These are the two service group IDs that are configured in IOS when using WCCPv2 with WAAS. Two different service groups are used because both directions (client-to-server and server-to-client) of a TCP connection must be transparently intercepted. To optimize a connection, WAAS must see both directions of the connection on the same WAE. Not only does WAAS intercept the connection in both directions, but it also intercepts the connection on both sides of the WAN link. Because the packet Layer 3 and Layer 4 headers are preserved, transparent interception is used on both sides of the WAN in both directions to redirect connections to the WAAS infrastructure for optimization. Figure 4-5 shows a basic topology with WCCPv2 interception configured for WAAS.

Figure 4-5 *Basic Network Topology with WCCP*



What is the difference between services 61 and 62? You can view the service attributes using CLI commands in both WAAS and IOS. Example 4-9 shows the attributes of services 61 and 62 using the IOS CLI.

Example 4-9 *WCCP Service Group Attributes*

```
AST6-RTR-02# show ip wccp 61 service
WCCP service information definition:
        Type:          Dynamic
        Id:            61
        Priority:      34
        Protocol:      6
        Options:       0x00000501
        --------
            Hash:      SrcIP
            Alt Hash:  SrcIP SrcPort
            Ports:     -none-

AST6-RTR-02#
AST6-RTR-02# show ip wccp 62 service
WCCP service information definition:
        Type:          Dynamic
        Id:            62
        Priority:      34
        Protocol:      6
        Options:       0x00000502
        --------
            Hash:      DstIP
            Alt Hash:  SrcIP SrcPort
            Ports:     -none-

AST6-RTR-02#
```

A description of each value is provided in Table 4-2.

Table 4-2 *WCCP Service Group Attributes*

Value	Description
Type	Well-known or dynamic service.
Id	The numeric service ID for the group.
Priority	The priority for the service group. When multiple service groups are configured on the same interface in the same direction, they are evaluated in descending priority order.
Protocol	The IP protocol number defined by the service group.
Options	Flags field indicating further service characteristics.
Hash	The value(s) in the redirected packet used as the hash key.
Alternate Hash	The value(s) in the redirected packet used as the alternate hash key.
Ports	The Layer 4 port numbers defined by the service group.

The command output shows that the only difference between services 61 and 62 is the value from the packet used as the hash key. By default, service group 61 hashes on the source IP address and service group 62 hashes on the destination IP address. Later, this chapter discusses the significance of the hash key used in each service group. By default, the **spoof-client-ip** feature is enabled for both services. This is the WCCPv2 feature that allows WAAS to handle optimized traffic transparently. Traffic forwarded to the WAE uses the same source and destination IP addresses and TCP ports as when it entered the WAE.

The tcp-promiscuous services define TCP as the protocol and do not define any ports. By not defining any ports as part of the service groups, this causes interception and redirection of all TCP traffic. When traffic passes through an interface in the IOS device with WCCPv2 redirection configured, it is evaluated against the protocol and port combination defined by the service to determine whether or not the packet should be redirected. By default this is the only criteria that is used to determine whether or not a packet is redirected. It is important to note that the IOS WCCPv2 implementation is not stateful. This means that IOS WCCPv2 is only dealing with redirected traffic on a packet-by-packet basis. It does not keep track of TCP connection state for redirected traffic. On the other hand, the WCCPv2 implementation in WAAS is stateful. WAAS tracks each connection as a flow throughout the life of the connection.

Forwarding and Return Methods

WCCPv2 supports different methods for forwarding redirected traffic to a WAE, and for the WAE to return traffic to the router for forwarding. These methods are referred to as the *forwarding* and *return* methods and are negotiated between IOS and the WAE when a WAE joins the service group.

The forwarding method defines how traffic that is being redirected from IOS to the WAE is transmitted across the network. The first method, GRE forwarding, encapsulates the original packet in an IP GRE header with the destination IP address set to the target WAE and the source IP address set to the WCCPv2 router ID of the redirecting router. When the WAE receives the GRE-encapsulated packet, the GRE header is removed, and the packet is processed. Figure 4-6 shows an example of GRE forwarding.

The second forwarding method, L2 forwarding, simply rewrites the destination MAC address of the packet being redirected to equal the MAC address of the target WAE. This forwarding method assumes that the WAE is Layer 2 adjacent to the redirecting router. L2 forwarding was originally developed for the WCCPv2 implementation on hardware-based platforms, such as the Catalyst 6500. Figure 4-7 shows an example of L2 forwarding.

Figure 4-6 *WCCP Redirection Using GRE Forwarding*

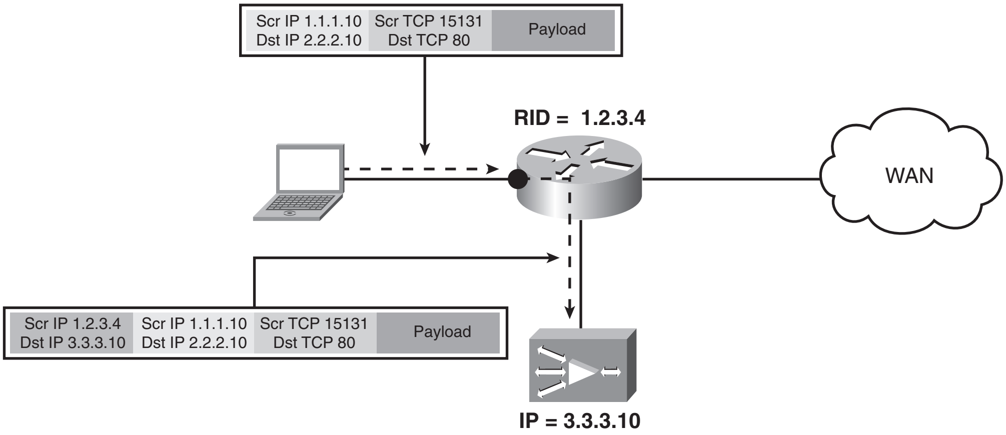
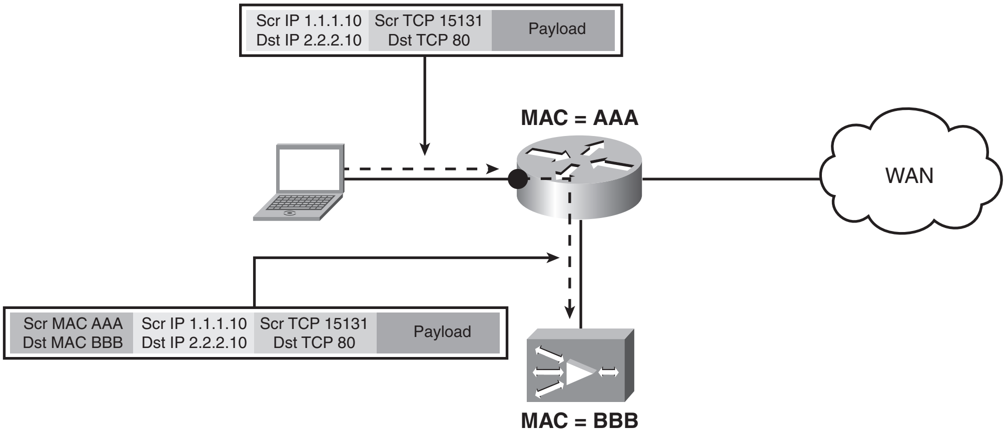


Figure 4-7 *WCCP Redirection Using L2 Forwarding*



One of the benefits of L2 forwarding is that it allows for the WCCPv2 redirection to occur in hardware on Cisco Catalyst Series switches. In fact, on the Catalyst 3560/3750 and 4500/4948 series switches, the only forwarding method supported by WCCPv2 is L2 forwarding. Additional information about the configuration requirements for deploying WCCPv2 on Cisco Catalyst switches is provided in the "WCCP Configuration" section.

The return method defines how traffic should be returned from the WAE to the redirecting router for normal forwarding. Like the forwarding method, there are two different return methods:

- **GRE return:** Egress traffic from the WAE using GRE return are encapsulated using IP GRE, with a destination IP address of the WCCPv2 router ID and a source IP address of the WAE itself. When the WCCPv2-enabled router receives the returned packet, the IP GRE header is removed and the packet is forwarded normally. WCCPv2 in IOS knows not to re-intercept traffic returned to it using GRE return.
- **L2 return:** The L2 return method returns traffic to the WCCPv2-enabled router by rewriting the destination MAC address of the packet to equal the MAC address of the WCCPv2-enabled router.

Load Distribution

When multiple WAEs exist in a service group, WCCPv2 automatically distributes redirected traffic across all WAEs in the service group. When traffic passes through an IOS device with WCCPv2 redirection configured, the IOS device assigns traffic for that connection to a bucket. Each bucket is assigned to a specific WAE. The method that determines to which bucket traffic is assigned, which determines how traffic is distributed across multiple WAEs within a service group, is called the assignment method. The bucket assignments are communicated from the lead WAE to all of the IOS devices in the service group. The assignment method can use either a hashing or masking scheme, and is negotiated between IOS and WAE during the formation of the service group.

Hash assignment, which is the default assignment method, performs a bitwise hash on a key identified as part of the service group. In WAAS, the hash key used for service group 61 is the source IP address, while the hash key used for service group 62 is the destination IP address. The hash is not configurable, and is deterministic in nature. This means that all of the routers within the same service group will make the same load-balancing decision given the same hash key. This deterministic behavior is what allows WCCPv2 to support asymmetric traffic flows, so long as both directions of the flow pass through WCCPv2-enabled IOS devices in the same service group. Hash assignment uses 256 buckets. Figure 4-8 shows an example of the hash assignment method and bucket-based distribution model used by WCCPv2.

The second assignment method is called mask assignment. With mask assignment, the source IP address, destination IP address, source port, and destination port are concatenated and ANDed with a 96-bit mask to yield a value. The resulting 96-bit value is compared to a list of mask/value pairs. Each mask/value pair is associated with a bucket, and each bucket is in turn assigned to a WAE. Unlike hash assignment, the number of buckets used with mask assignment depends on the number of bits used in the mask. By default, WAAS uses a mask of 0x1741. This results in 2^6 buckets that can be assigned across the WAEs in a service group. With current Catalyst WCCPv2 implementations, up to 7 bits can be defined for the mask. Figure 4-9 shows an example of the mask assignment method and bucket-based distribution model used by WCCPv2.

Figure 4-8 *WCCP Redirection Using Hash Assignment*

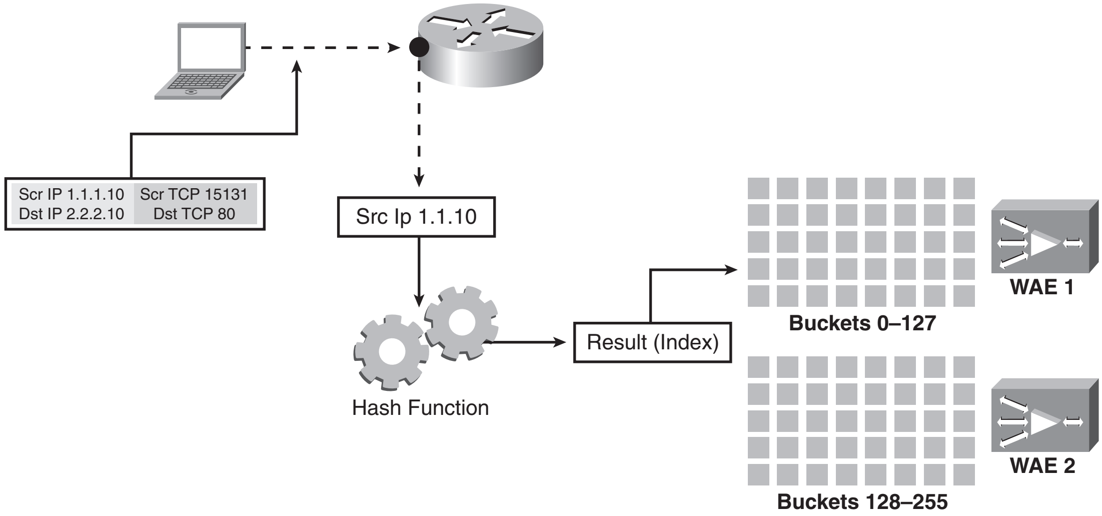
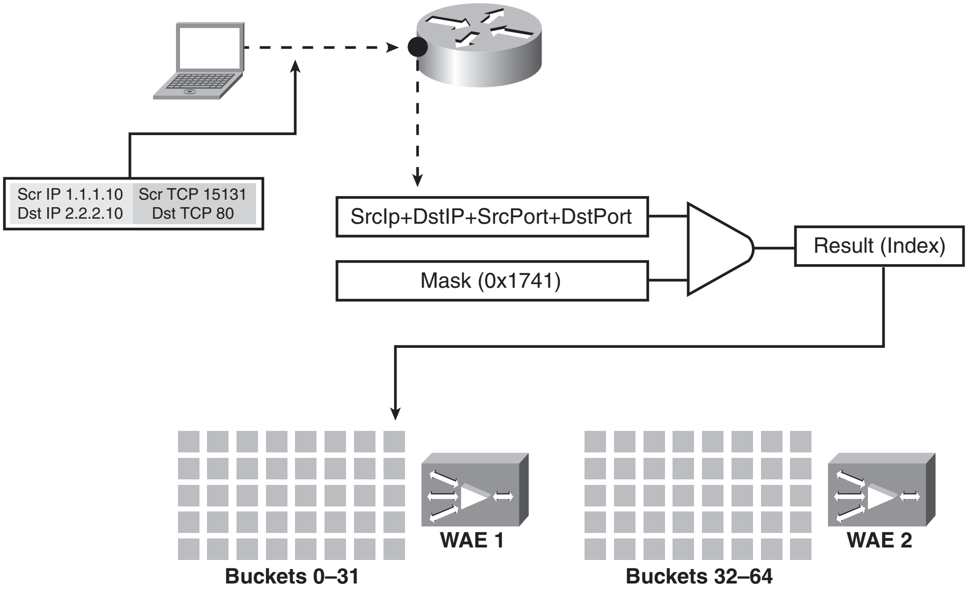


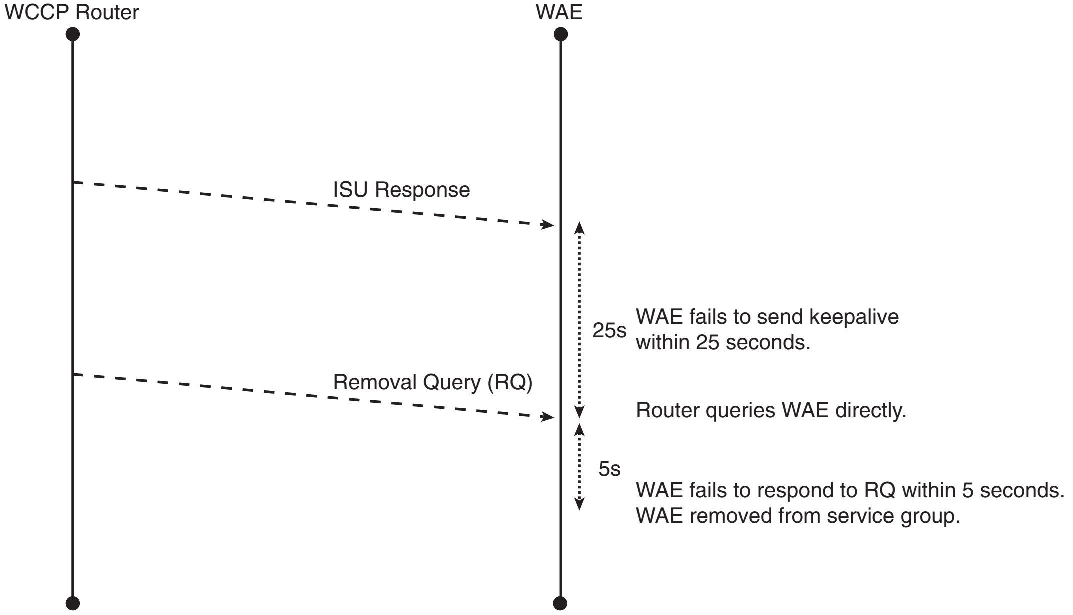
Figure 4-9 *WCCP Redirection Using Mask Assignment*



Failure Detection

Once a WAE has successfully joined a service group, a periodic keepalive packet is sent every 10 seconds from the WAE to each router in the service group. The keepalive mechanism occurs independently for each configured service group. If a router in the service group has not received a keepalive packet from the WAE in 25 seconds, the router unicasts a Removal Query message to that WAE requesting that it immediately respond. If no response is received within 5 seconds, for a total of 30 seconds since the last keepalive message from the WAE, the WAE is considered offline and is removed from the service group. Figure 4-10 illustrates this behavior.

Figure 4-10 *WCCP Keepalive Timeout*


WCCP Router
WAE
ISU Response
25s
WAE fails to send keepalive within 25 seconds.
Removal Query (RQ)
Router queries WAE directly.
5s
WAE fails to respond to RQ within 5 seconds. WAE removed from service group.

When the WAE is removed from the service group, it is reflected in the Router View advertised from each router in the service group. When the lead WAE determines that a WAE has been removed from the service group, it generates a Redirect Assignment message to each router in the service group. The Redirect Assignment message instructs the routers how to reallocate the buckets across the remaining WAEs in the service group. The length of time required to calculate the new assignments might very depending upon when the group of WAEs becomes stable. The WAE waits a minimum of 9 seconds. The maximum length of time depends on when the IOS device sends an update message without any changes indicated, typically between 19 and 39 seconds.

Flow Protection

When a WAE (re)joins the service group, a new Redirect Assignment message is generated by the lead WAE. When the new WAE begins receiving redirected traffic from the routers in the service group, it does one of two things, depending on whether or not the redirected traffic is for a new connection or part of an existing connection. Traffic associated with newly established connections is evaluated against the Application Traffic Policy (ATP) and processed normally by the WAE. Traffic associated with existing connections is forwarded directly to the WAE that previously owned the bucket for that connection. This WCCPv2 mechanism is called flow protection and is enabled by default. Flow protection allows for existing connections to continue to be optimized even when the traffic assignments for the WAEs in a service group change.

Graceful Shutdown

After the **no wccp ver 2** command is issued, WCCPv2 checks whether any connections are being served by the WAE. If zero connections are being served, the shutdown is immediately carried out. If there are more than zero connections being served, WCCPv2 waits for the user-configured **wccp shutdown max-wait** *XX* time.

During this time, if the connection count goes down to zero, shutdown is immediately done. At the end of the max-wait time, if the connection count has decreased but is still non-zero, the shutdown count waits another 60 seconds, in the hope that if the connection count has decreased other connections may complete too. At the end of the max-wait time, if the connection count has not decreased, shutdown is immediately done. During the 60-second incremental wait, if the connection count becomes zero, shutdown is done. At the end of the 60-second incremental wait, if the connection count has not reduced, the shutdown is done. At the end of the 60-second incremental wait, if the count has further reduced but is still non-zero, another 60-second incremental wait is done.

Unless the user interrupts the wait period, the code waits first for the configured length of time. If it thinks that connections are reducing, it waits a little longer in the hope that more connections can be completed. However, if it realizes that the connection count has not decreased, it discontinues waiting and shuts down.

Scalability

With WCCPv2, each service group can support up to 32 routers and 32 WAEs. This means that a single service group can support $N \times 32$ concurrent optimized TCP connections, where N is the number of concurrent optimized TCP connections supported by the largest WAE model. Each WAE in the service group is manually configured with the IP address of each router in the service group. The WAE then uses unicast packets to exchange WCCPv2 messages with each router. It is not required that the routers in the service are manually configured with the IP address of each WAE in the service group. Each router listens passively for WCCPv2 messages from the WAEs in the service group and responds only as a result of receiving those messages.

The WAE in the service group with the lowest IP address is elected as the “lead” WAE. The lead WAE is responsible for communicating the list, or view, of the routers in the service group to the service group routers. The lead WAE is also responsible for informing the routers how traffic should be distributed across WAEs in the service group. Upon receiving the view of the routers in the service group from the lead WAE, each router responds individually with a Router View. The Router View contains a list of each WAE that the router is currently communicating with. What is implied is that the routers in the service group do not communicate directly with each other; they learn about each other through the Router View advertised by the WAE. Likewise, the WAEs in a service group do not communicate directly with each; they learn about each other from the WAE View advertised by the routers.

Redirect Lists

For deployments where you may want to limit redirection to specific types of traffic, you can use a WCCPv2 redirect list. A WCCPv2 redirect list is a standard or extended IOS access list that is associated with a WCCPv2 service. Traffic passing through an interface on the router with WCCPv2 redirection configured must match not only the protocol/port specified as part of the service group, but also a permit entry in the redirect list. Packets that match the service group protocol/port criteria but do not match a permit entry in the redirect list are forwarded normally. Example 4-10 demonstrates the use of a WCCPv2 redirect list.

Example 4-10 *WCCP Redirection Using a Redirect List*

```
!
access-list 100 permit ip 10.10.10.0 0.0.0.255 any
access-list 100 permit ip any 10.0.0.0 0.0.0.255
access-list 100 deny ip any any
!
ip wccp 61 redirect-list 100
ip wccp 62 redirect-list 100
!
```

In this example, TCP traffic sourced from or destined to subnet 10.10.10.0/24 will be intercepted and redirected by WCCPv2. Another option is to use a redirect list to specify a subset of TCP ports for redirection. Example 4-11 shows how this could be done.

Example 4-11 *WCCP Redirect List Using Application Ports*

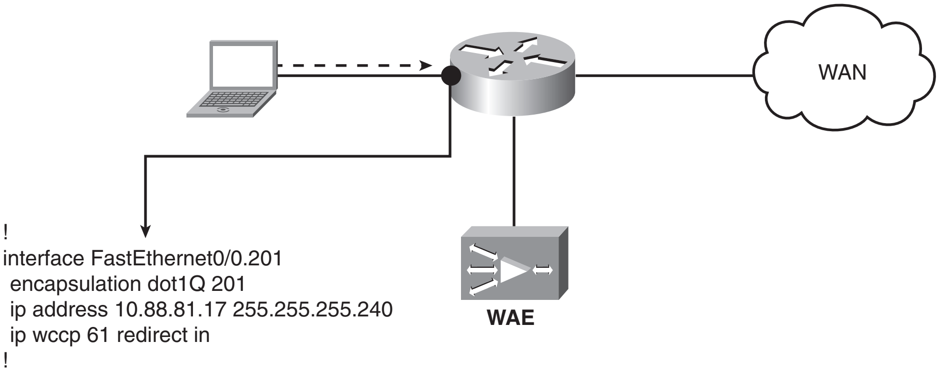
```
!
access-list 101 permit tcp any any eq 25
access-list 101 permit tcp any eq 25 any
access-list 101 deny ip any any
!
ip wccp 61 redirect-list 101
ip wccp 62 redirect-list 101
!
```

This example uses a redirect list to allow WCCPv2 to intercept and redirect SMTP traffic only on port 25.

Service Group Placement

The placement of service groups 61 and 62 should not be overlooked in your deployment. The placement refers to which IOS interfaces are configured with service group 61 and which interfaces are configured with service group 62. In most environments, service group 61 should be configured on the client-facing interfaces. For example, when deploying WCCPv2 on a remote-office WAN router, service group 61 is configured to intercept a client request. Configuring group 61 inbound on the router's LAN interface or outbound on the router's WAN interface accomplishes this. Figure 4-11 shows an example of configuring service group 61 inbound on the router's LAN interface.

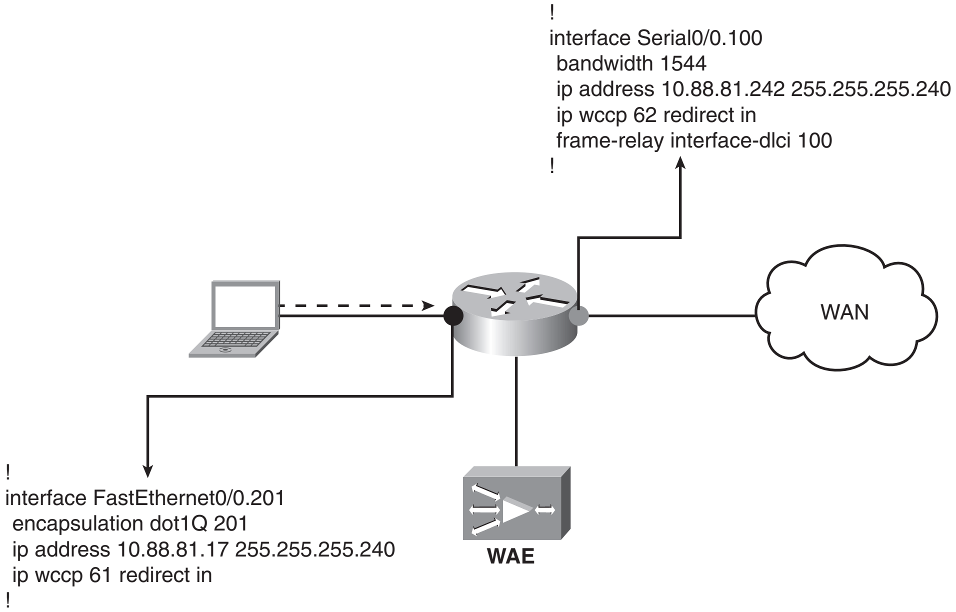
Figure 4-11 *WCCP Service Group 61 Placement*



```
!
interface FastEthernet0/0.201
 encapsulation dot1Q 201
 ip address 10.88.81.17 255.255.255.240
 ip wccp 61 redirect in
!
```

For the reverse direction of the connection, service group 62 is used. Service group 62 will be configured in the opposite direction of service group 61. Using the same example shown in Figure 4-11, Figure 4-12 shows service group 62 configured inbound on the router's WAN interface. The following figure shows the complete placement and configuration using both service groups.

Figure 4-12 *WCCP Service Group 61 and 62 Placement*



```
!
interface Serial0/0.100
 bandwidth 1544
 ip address 10.88.81.242 255.255.255.240
 ip wccp 62 redirect in
 frame-relay interface-dlci 100
!
```

```
!
interface FastEthernet0/0.201
 encapsulation dot1Q 201
 ip address 10.88.81.17 255.255.255.240
 ip wccp 61 redirect in
!
```

WCCP Configuration

This section provides a basic overview of configuring WCCPv2 within both IOS and WAAS. Detailed WCCPv2 configurations specific to various design options are presented in Chapters 5 and 6.

There are three primary steps involved when configuring WCCPv2 in WAAS. First, you must define which routers the WAE will establish WCCPv2 communication with. WCCPv2 can be configured to use either unicast or multicast for communication. Unicast is the most commonly deployed configuration. For unicast communication, you must define the IP address of each router in the service group that the WAE will communicate with. This is done using a router list. A router list is configured using the following syntax:

```
wccp router-list 1-4 ip_addr...
```

Example 4-12 shows a basic WCCP router list configuration.

Example 4-12 *WAAS WCCP Router List Configuration*

```
wccp router-list 1 10.10.10.1
```

Up to six IP addresses may be defined per line. For deployments where there are more than six routers in the service group, additional router IP addresses can be defined by configuring a second line using the same router list number. Example 4-13 shows a WCCPv2 router list configured with ten IP addresses.

Example 4-13 *WCCP Router List Using Multiple IP Addresses*

```
wccp router-list 1 10.10.10.1 10.10.10.2 10.10.10.3 10.10.10.4 10.10.10.5
10.10.10.10.6
wccp router-list 1 10.10.10.7 10.10.10.8 10.10.10.9 10.10.10.10
```

CAUTION Do not use virtual IP (VIP) addresses, such as an HSRP virtual IP address, in the WCCPv2 router list. The router list should contain only interface IP addresses. When the WAE is L2 adjacent to the WCCP-enabled router(s), the IP address(es) used in the WCCP router list should be the directly connected interface IP addresses. In cases where the WAE is not L2 adjacent to the WCCP-enabled router(s) (that is, the WAE is multiple L3 hops away from the WCCP-enabled router or routers), a loopback interface IP address should be used in the router list configuration. Using a loopback interface IP address improves the reliability of the WCCP service group, because the loopback interface IP address is not tied to the availability of any single physical interface.

For the second step, the WCCPv2 tcp-promiscuous service is configured and associated with the router list created in the first step. The following command syntax is used:

```
wccp tcp-promiscuous router-list-num 1
```

The final configuration step is to enable WCCPv2 using the command **wccp version 2**. This command starts the WCCPv2 negotiation with any IOS devices configured in the router list. Example 4-14 shows a complete WCCPv2 configuration in Cisco WAAS.

Example 4-14 *Complete WAAS WCCP Configuration*

```
!
wccp router-list 1 10.10.20.1
wccp tcp-promiscuous router-list-num 1
wccp version 2
!
```

The IOS WCCPv2 configuration involves two steps. First, the WCCPv2 services are configured in global configuration mode. The WCCPv2 services in IOS are configured using the numeric service ID, as opposed to the service name used on the WAAS configuration. Example 4-15 shows the tcp-promiscuous services configured in IOS.

Example 4-15 *Cisco IOS WCCP Global Configuration*

```
!
ip wccp 61
ip wccp 62
!
```

The second step involves configuring WCCPv2 redirection on each interface through which client and server data passes. Unless you are using the WCCPv2 negotiated return egress method discussed later in this chapter, WCCPv2 redirection should never be configured on the interface connecting to the WAE. Interception is configured in either the inbound or outbound direction. When using outbound redirection, the **ip wccp redirect exclude in** command must be configured in the interface connecting to the WAE. This prevents traffic coming into the WCCPv2 server (router) from being re-intercepted, which would cause a redirection loop. Example 4-16 demonstrates a complete IOS WCCPv2 configuration, including the use of the **ip wccp redirect exclude in** command.

Example 4-16 *Complete Cisco IOS WCCP Configuration*

```
!
ip wccp 61
ip wccp 62
!
ip cef
!
interface Serial0/0
 bandwidth 1536
 no ip address
 encapsulation frame-relay
!
interface Serial0/0.100
 ip add 10.88.80.18 255.255.255.252
 ip wccp 61 redirect out
 ip wccp 62 redirect in
 frame-relay interface-dlci 100
!
interface GigabitEthernet0/0
 no ip address
 duplex auto
 speed auto
!
interface GigabitEthernet0/0.1
 description ** Branch Client VLAN **
 encapsulation dot1q 10
```

Example 4-16 *Complete Cisco IOS WCCP Configuration (Continued)*

```
 ip address 10.10.10.1 255.255.255.0
!
interface GigabitEthernet0/0.20
 description ** Branch WAE VLAN **
 ip address 10.10.20.1 255.255.255.0
 ip wccp redirect exclude in
!
end
```

Note that the **ip wccp redirect exclude in** command is configured on the subinterface connecting to the WAE. This is required because outbound redirection is used on the serial interface connecting to the WAN. An alternative configuration is shown in Example 4-17.

Example 4-17 *Cisco IOS WCCP Configuration Using Inbound Redirection*

```
!
ip wccp 61
ip wccp 62
!
ip cef
!
interface Serial0/0
 bandwidth 1536
 no ip address
 encapsulation frame-relay
!
interface Serial0/0.100
 ip add 10.88.80.18 255.255.255.252
 ip wccp 62 redirect in
 frame-relay interface-dlci 100
!
interface GigabitEthernet0/0
 no ip address
 duplex auto
 speed auto
!
interface GigabitEthernet0/0.1
 description ** Branch Client VLAN **
 encapsulation dot1q 10
 ip address 10.10.10.1 255.255.255.0
 ip wccp 61 redirect in
!
interface GigabitEthernet0/0.20
 description ** Branch WAE VLAN **
 ip address 10.10.20.1 255.255.255.0
!
end
```

This example uses inbound redirection on the interface connecting to the client subnet and the serial interface connecting to the WAN. Because outbound redirection is not used, the **ip wccp redirect exclude in** command is not required on the interface connecting to the WAE.

Hardware-Based Platforms

In addition to running WCCPv2 on software-based IOS platforms such as the Cisco Integrated Services Router (ISR), WCCPv2 is supported on Cisco Catalyst Series switches. At the time of this writing, the Cisco Catalyst Series switches listed in Table 4-3 support WCCPv2 for use with Cisco WAAS.

Table 4-3 *Cisco Catalyst Platforms Supporting WCCPv2 with WAAS*

Catalyst 3560/3750
Catalyst 4500/4900
Catalyst 6500, Sup2
Catalyst 6500, Sup32
Catalyst 6500, Sup720

With the exception of the Catalyst 6500 with a Sup720, the hardware-based platforms require L2 forwarding and mask assignment for all of the redirection to happen in hardware. The Sup720 is capable of performing GRE forwarding in hardware, but still requires mask assignment for hardware acceleration. In addition to the requirements for forwarding and assignment methods, only inbound WCCPv2 redirection should be used on hardware-based platforms. In fact, the Catalyst 3560/3750 and 4500/4900 only support inbound redirection. While it is possible to configure outbound redirection on the Catalyst 6500 platform, it is not recommended because it causes the first packet for every redirected connection to be processed in software by the MSFC. Likewise, using the **ip wccp redirect exclude in** command on a Catalyst 6500 causes the first packet for every flow entering the interface to be processed by the MSFC and switched in software. However, because inbound redirection is the recommendation for hardware-based platforms, this command is not required.

The following configuration guidelines should be followed to ensure WCCPv2 redirection on hardware-based platforms is handled completely in hardware:

- Use L2 forwarding instead of GRE forwarding.
- Always use mask assignment.
- Only use inbound redirection.
- Do not use the **ip wccp redirect exclude in** command.

The L2 forwarding and mask assignment options are configured as part of the service definition in WAAS. These capabilities are advertised to the WCCPv2-enabled IOS devices when a WAE first joins the service group. Example 4-18 demonstrates the WAAS WCCPv2 configuration with the L2 forwarding and mask assignment options.

Example 4-18 *WCCP Configuration Using L2 Forwarding and Mask Assignment*

```
!
wccp router-list 1 10.10.20.1
wccp tcp-promiscuous router-list-num 1 l2-redirect mask-assign
wccp version 2
!
```

Unlike the hash algorithm used with hash assignment, the mask used for mask assignment is configurable. As mentioned previously in this chapter, the default mask used by WAAS is 0x1741. The default mask is applied to the source IP address for service group 61 and is applied to the destination IP address for service group 62. Depending on the IP addressing used in your environment, you may want to change the default mask to provide for better load distribution among the WAEs in a service group. The default mask is changed on the WAE using the following command syntax:

```
wccp tcp-promiscuous mask src-ip-mask 0-4261412864
```

The configured mask is applied to service group 61. Service group 62 mirrors the configuration and cannot be configured separately. Example 4-19 shows using a non-default mask with WCCPv2.

Example 4-19 *Custom WCCP Mask*

```
!
wccp router-list 1 10.10.20.1
wccp tcp-promiscuous mask src-ip-mask 0xf
wccp tcp-promiscuous router-list-num 1 l2-redirect mask-assign
wccp version 2
!
```

Policy-Based Routing

Policy-based routing (PBR) provides another alternative for transparent interception with WAAS, although it is less commonly deployed than WCCPv2 and inline interception. PBR can be used in situations where customers are unable to run WCCPv2 or inline interception. PBR can also be used in conjunction with a content switch, such as the Cisco Application Control Engine (ACE), to provide transparent interception and load balancing for large-scale data center deployments. Deployment examples using PBR for transparent interception are provided in Chapters 5 and 6.

PBR functions in a similar manner to WCCPv2, in that a router/switch running Cisco IOS is configured to intercept interesting traffic and redirect it to a WAE. Unlike WCCPv2, no configuration is required on the WAE to support interception using PBR. The following configuration steps are required for a basic PBR configuration:

Step 1 Create an access list to define interesting traffic for redirection.

Step 2 Create a route map that matches the ACL created in Step 1 and sets an IP next-hop address of the target WAE.

Step 3 Apply the route map to interfaces through which client and server traffic traverses.

Example 4-20 demonstrates a basic PBR configuration used for redirecting all TCP traffic to a single WAE.

Example 4-20 *PBR Configuration*

```
!
ip cef
!
access-list 199 permit tcp any any
!
route-map WAAS-INTERCEPT 10
 match ip address 199
 set ip next-hop 10.10.20.5
!
interface Serial0/0
 bandwidth 1536
 no ip address
 encapsulation frame-relay
!
interface Serial0/0.100
 ip add 10.88.80.18 255.255.255.252
 ip policy route-map WAAS-INTERCEPT
 frame-relay interface-dlci 100
!
interface GigabitEthernet0/0
 no ip address
 duplex auto
 speed auto
!
interface GigabitEthernet0/0.1
 description ** Branch Client VLAN **
 encapsulation dot1q 10
 ip address 10.10.10.1 255.255.255.0
 ip policy route-map WAAS-INTERCEPT
!
interface GigabitEthernet0/0.20
 description ** Branch WAE VLAN **
```

Example 4-20 *PBR Configuration (Continued)*

```
 ip address 10.10.20.1 255.255.255.0
!
end
```

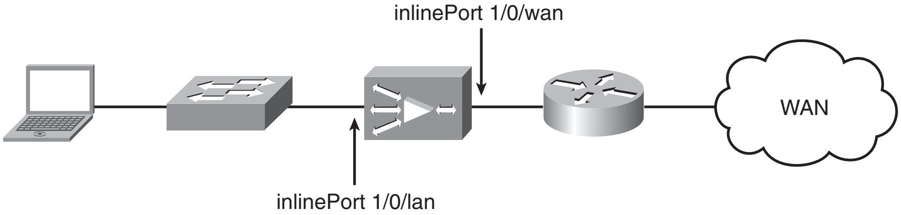
Because PBR evaluates only traffic entering an interface, the route map entries are configured on both the client and server-facing interfaces. This is the equivalent of using only inbound redirection with WCCPv2. The **set ip next-hop** command in the route map is configured with the IP address of the WAE. By default, PBR does not validate to availability of the IP address specified as the next-hop address. As long as the next-hop address exists in the routing table, the route map entry will be applied. On software-based platforms (ISR, and so forth), Cisco Service Assurance Agent (SAA) can be used to track the availability of the next-hop IP address. If the next-hop address becomes unreachable, traffic matching the route map entry is forwarded normally using the routing table. However, this capability does not currently exist on hardware-based platforms.

Other difference between WCCPv2 and PBR is that PBR does not perform automatic load distribution and failover when multiple WAEs exist. The first next hop IP address configured in the route map is used until it becomes unavailable. Only at that point is traffic redirected to a secondary next hop IP address in the route map. Chapters 5 and 6 provide examples of PBR deployments that include next hop availability tracking using SAA and load distribution among multiple WAEs.

Inline Interception

An alternative to the various off-path interception mechanisms is to place the WAE physically inline between two network elements, such as a WAN access router and local-area network (LAN) switch. Figure 4-13 shows a basic topology with the WAE deployed physically inline.

Figure 4-13 *WAE Physical In-Path Deployment*

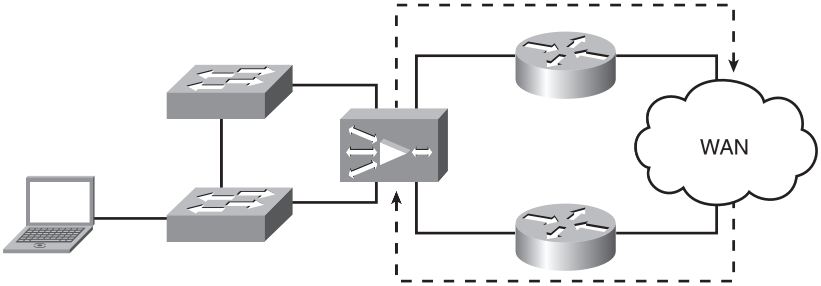


Physical inline interception is an attractive option for situations where it is not possible or ideal to run WCCPv2. It is also possible that the networking equipment at a site is provided and managed by a managed service provider (MSP). The MSP may not be able to configure or support a WCCPv2 solution on the managed devices.

To support physical inline interception, the WAE requires a separate inline module. The inline module is a 4-port, fail-to-wire NIC that supports two separate inline groups. Each inline group has a synchronous pair of inline ports that interconnect two network elements. Traffic entering one inline port is optimized by WAAS (when applicable) and switched out the opposite inline port in the same group. The inline group functions like a transparent Layer 2 bridge.

By providing two inline groups on a single module, the WAE can support designs where multiple paths out of a site exist for redundancy and load sharing. Each unique path is connected to the WAE through a separate inline group. Figure 4-14 shows a sample remote site topology with multiple WAN routers and a single WAE deployed with inline interception.

Figure 4-14 *Physical In-Path Deployment Using Multiple Routers*



As the arrows in Figure 4-14 indicate, traffic can enter or leave the site through either router. Even though the same flow enters the site through one inline group and exits the site through another inline group, the connection will still be optimized. The optimized connection state is not tied to a physical interface, but is tracked for the WAE as a whole independent of the interfaces traversed by the traffic.

Each inline group functions in one of two operating modes:

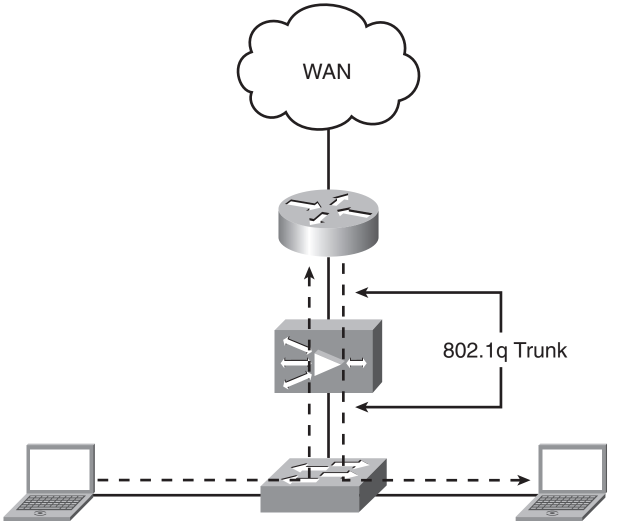
- **Intercept operating mode:** Traffic entering the inline group is evaluated against the Application Traffic Policy (ATP) for optimization.
- **Bypass operating mode:** All traffic entering the inline group is bridged without any optimization.

The bypass operating mode is designed to allow the WAE to continue passing traffic if the WAE looses power. A keepalive mechanism between the network drivers and the inline module is used to determine if the WAE is functioning properly and can optimize connections.

The keepalive frequency is configurable between 1 and 10 seconds. The default failover timer is set to 3 seconds. The transition between intercept operating mode and bypass operating mode does cause a momentary loss of line protocol. If one or more of the inline ports are connected to a LAN switch, this transition in interface state can cause the Spanning Tree Protocol (STP) recalculation. To prevent the STP calculation from interrupting traffic forwarding, the switchport connected to the inline module on the WAE should have the STP PortFast feature enabled. Failure of a single inline port in the group is propagated to the other port in the group. For example, if the LAN0 port in InlineGroup 1/0 goes down, the WAE will take down line protocol on the WAN0 port in the same inline group. This propagation of interface state between the ports in the same inline group prevents situations where adjacent devices connected to an operational InlinePort believe the network path to be online and usable, when in reality the connection on the other side of the WAE is unavailable.

When a WAE is deployed physically inline, all traffic between the two network elements will be seen by the WAE. Non-TCP traffic is bridged through the inline module without modification. In addition, packets associated with a connection that was first seen on the opposite inline port in a group are bridged. This type of traffic flow is common when a WAE is deployed inline on a trunk between a router and LAN switch. If the router is providing routing for traffic going between VLANs locally, it is possible for traffic to traverse the inline module twice. Figure 4-15 shows an example of this type of traffic flow.

Figure 4-15 *Physical In-Path Deployment with One-Armed Routing*



The inline module also supports 802.1Q trunk connections between the two network elements. An added benefit to using the inline module is the ability to define which VLANs are evaluated for interception. Traffic that is received by the inline module tagged with a

VLAN ID that is excluded from interception will be bridged without any optimization. This capability is supported only for tagged VLANs. Traffic received by the inline module on untagged VLANs will be intercepted and evaluated against the ATP for optimization and acceleration. By default, TCP traffic received on all VLANs is intercepted and evaluated against the ATP. VLANs can be excluded or included for interception using the following commands:

```
no inline vlan all
inline vlan 100
```

Example 4-21 shows the resulting InlineGroup configuration.

Example 4-21 *WAE InlineGroup Configuration*

```
!
interface InlineGroup 1/0
 inline vlan all
 no inline vlan native,0-99,101-4095
 exit
!
```

There are different sequences of the inline CLI command that will result in the same VLAN filter being applied. For example,

```
inline vlan all
no inline vlan 100
```

results in all VLANs except for 100 being intercepted. But so does the following:

```
inline vlan native
inline vlan 0-4095
no inline vlan 100-110
inline vlan 101-200
```

In terms of VLAN assignment, the most permissive command takes precedence. If the inline group is already configured with **inline vlan all**, then you need to selectively remove VLANs from interception or remove all VLANs and selectively add individual VLANs back for interception.

When an inline group is in bypass operating mode, a physical cross-connect is enabled between the two ports in the inline group. This behavior essentially creates a crossover cable between the two network elements. In cases where the two network elements are unable to communicate using a crossover cable, line protocol will not be restored when the inline group is in bypass operating mode. This is generally a nonissue when the switchport that the LAN InlinePort is connected to supports automatic medium-dependent interface crossover (MDIX). MDIX allows the switchport to automatically detect the pinouts of the cables used to connect two devices. In cases where the switchport does not support this capability, the cabling guidelines outlined in Table 4-4 should be followed.

Table 4-4 *WAE Inline Module Cabling Guidelines*

Connection	Required Cable
Switch to switch (no WAE)	Crossover
Switch to router (no WAE)	Straight-through
Router to router (no WAE)	Crossover
Switch to WAE	Straight-through
WAE to switch	Crossover
Switch to WAE	Straight-through
WAE to switch	Straight-through
Router to WAE	Straight-through
WAE to router	Straight-through
WAE to WAE	Crossover

Content Switching

Content switching is the final interception mechanism discussed in this chapter. Content switches have traditionally provided load-balancing services for servers, firewalls, and content caches. Within the context of WAAS, content switching provides dedicated hardware for intercepting and load balancing connections across a farm of WAEs. Using content switches for transparent interception with WAAS is useful for large data center deployments, complex topologies, and integration with other advanced features such as application protocol optimization and SSL-offload. In addition, customers with existing content switching deployments can leverage their experience and investments in content switches for transparent interception with WAAS. The Application Control Engine is the Cisco content switch that will be discussed in this section. Deployment and configuration examples for integrating ACE with Cisco WAAS are provided in Chapter 6.

Application Control Engine

The Cisco Application Control Engine (ACE) module is a service module for the Cisco Catalyst 6500 series switches and Catalyst 7600 series routers. ACE provides intelligent load balancing and security services for enterprise applications and network devices. ACE can be used in a large-scale data center environment to transparently intercept and load balance connections for WAAS. The following are some of the key performance characteristics of Cisco ACE:

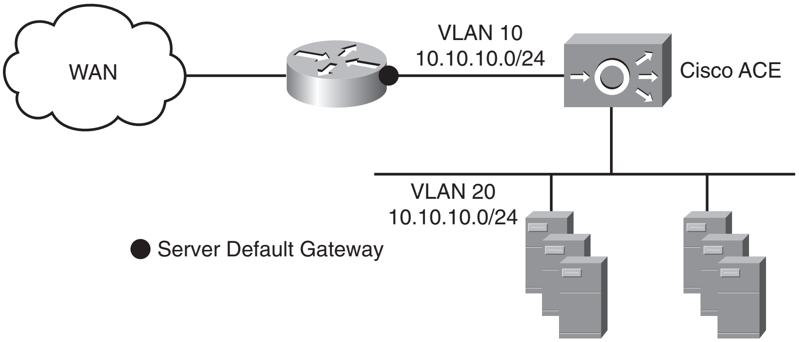
- Up to 16 Gbps of throughput and 345,000 connections per second per module
- Up to 4 million concurrent connections

- Support for up to 250 virtual partitions, allowing customers to create virtual ACE modules using a single hardware module
- Up to 16,000 real servers, which when used with Cisco WAAS provides nearly infinite scalability
- High availability and scalability by using up to four ACE modules in the same Catalyst 6500 chassis or across multiple chassis

There are two common deployment models for integrating ACE into the network infrastructure: bridge mode and routed mode.

In bridge mode, ACE is used to merge two VLANs together. In order for traffic to pass between the two VLANs, it must pass through the ACE module. As traffic passes through the ACE module, it is evaluated against the configured service policies to determine whether or not it should be acted upon. The IP subnet used on the bridged VLAN is the same. Figure 4-16 shows an ACE module deployed using bridge mode.

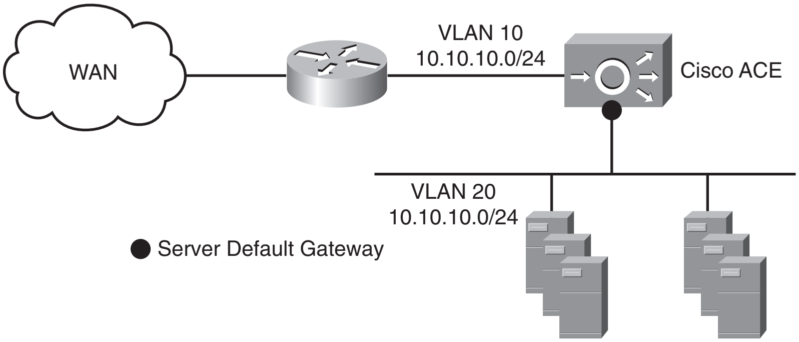
Figure 4-16 *ACE Deployed Using Bridge Mode*



The WAN-facing VLAN in Figure 4-16 is referred to as the client-side VLAN. The VLAN facing the data center resources is referred to as the server-side VLAN. As traffic enters the client-side VLAN, it is evaluated against the configured service policy. Traffic matching the service policy is redirected to a WAE, which has a dedicated VLAN interface configured on the ACE module. Traffic egressing the WAE comes back into the ACE module, where it is switched out the server-side VLAN toward the origin server.

In contrast to bridge mode, deploying ACE in routed mode allows for traffic to be routed between two different IP subnets. Using this deployment model, the client and server-side VLANs are on different IP subnets. Because the ACE module is a Layer 3 hop, traffic must be directed to the ACE module through the routing configuration of the hosts or network infrastructure. Figure 4-17 shows an ACE module deployed using routed mode.

Figure 4-17 *ACE Deployed Using Routed Mode*



ACE is typically deployed in conjunction with WAAS using transparent, or directed, mode. This means that the ACE module does not perform any Network Address Translation (NAT) of traffic passing through it.

Egress Methods for Intercepted Connections

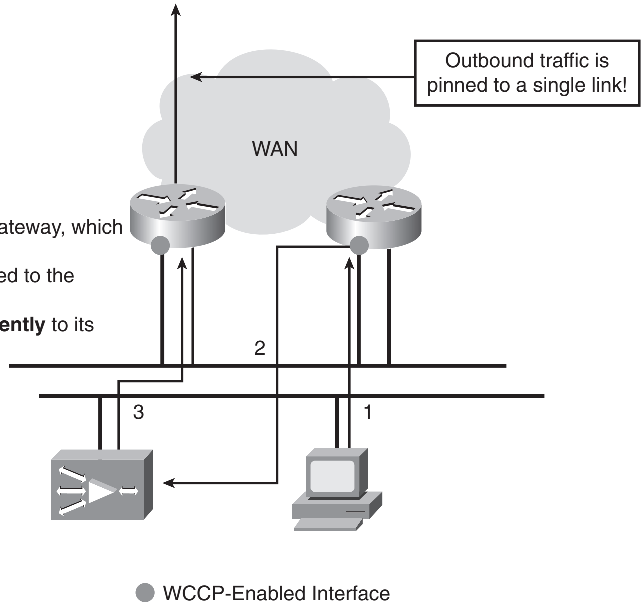
Cisco WAAS provides several options for handling egress traffic received on intercepted connections. These options allow for flexibility when determining where to integrate WAAS into the existing network infrastructure, and help preserve the original path selection for traffic flows. These deployment options, referred to as the egress methods for intercepted connections (EMIC), are discussed in detail in this section.

The first EMIC available in Cisco WAAS is IP forwarding. Egress traffic received on intercepted connections is forwarded based on the configuration of the local WAE routing table, which typically means that traffic is forwarded to the configured default gateway. In addition to supporting a single default gateway, WAAS supports up to 1024 static routes. Static routes are configured with a next hop IP address of a directly connected interface; recursive next hop IP addresses are not supported. Although it is possible to configure multiple static routes for the same destination, there is no support for equal-cost multipath (ECMP). Only a single route will be installed in the routing table at a time. It should be noted that traffic originating from the WAE itself will also use IP forwarding, regardless of the EMIC configuration. The IP forwarding EMIC is suited for very basic topologies where only a single egress path for traffic exists, or in situations where other EMICs are not supported.

For more complex topologies, the IP forwarding EMIC can lead to undesirable forwarding of traffic for intercepted connections. Take for example the topology shown in Figure 4-18. This example shows a remote office with multiple WAN routers connecting to diverse circuits. Traffic can enter or leave the site through either router. When multiple paths exist for traffic leaving a site, it is common for either HSRP or the Gateway Load Balancing

Protocol (GLBP) to be used for default gateway redundancy. HSRP provides an active/standby configuration based on a virtual IP (VIP) address. At any given point in time, a single VIP address is "active" on one of the routers. Hosts are configured with the HSRP VIP address as their default gateway, causing all traffic from those hosts to be forwarded to one of the two routers. In the case of GLBP, either router can be selected as the outbound path for a host, depending on the specific GLBP configuration. Because GLBP operates based on MAC addresses, a WAE running Cisco WAAS appears as a single host. This means that traffic egressing a WAE will also select one of the two routers to forward outbound traffic to. For deployments that use GLBP for default-gateway redundancy, the issue with IP forwarding is the most pronounced.

Figure 4-18 *Branch Topology with Multiple Entry and Exit Points*



1 Client sends traffic to its configured default gateway, which is currently active on the right router (GLBP).
2 Traffic is **intercepted** by WCCP and redirected to the WCCP.
3 Egress traffic from the WAE is sent **transparently** to its default gateway.

You can see in the previous example how all egress traffic from the WAE is "pinned" to a single router. This can defeat the purpose of deploying GLBP in the first place, which is to distribute outbound traffic across both routers.

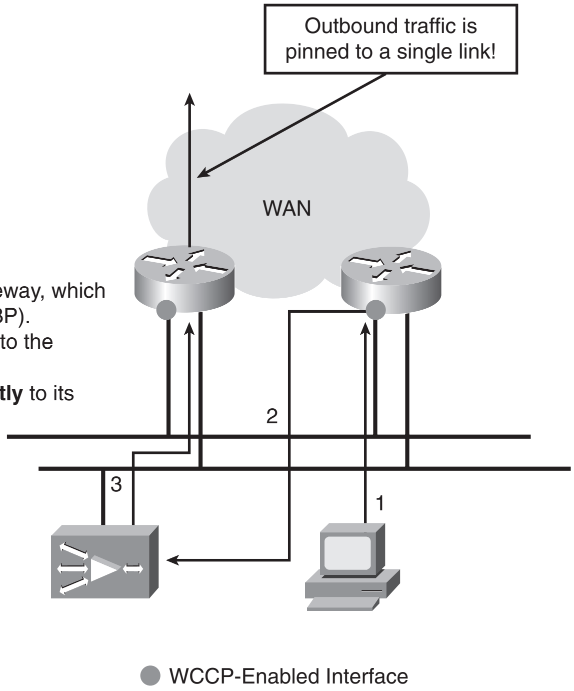
There are several options with WAAS for preserving the network path affinity originally chosen by the host system or network elements. The first two options leverage the WCCPv2 return mechanism. You'll recall from earlier in this chapter that the WCCPv2 return mechanism is used by WCCPv2 clients to handle bypass traffic by sending it back to the WCCPv2-enabled router that redirected it. Cisco WAAS has the ability to leverage the return method negotiated between the router and WAE for forwarding egress traffic from the WAE. The following CLI command changes the default EMIC from IP forwarding to negotiated return:

```
egress-method negotiated-return intercept-method wccp
```

If GRE is the return method negotiated between the WAE and IOS, traffic received on optimized connections is encapsulated in a GRE header with a destination IP address of the WCCPv2 router ID and a source IP address of the WAE. When the WCCPv2-enabled IOS device receives the GRE-encapsulated packet, it removes the GRE header and forwards the packet normally. Because the GRE header uses a source IP address of the WAE, the IOS WCCPv2 process knows not to re-intercept the packet. This capability to return traffic to the IOS device that redirected it allows for the preservation of the original path selection made by the host or network infrastructure.

Another benefit of the GRE return method is that the WAE can reside on the same IP subnet with clients or servers that it optimizes connections for. This greatly simplifies branch deployments by removing the requirement for a separate subnet dedicated to the WAE. Figure 4-19 shows the same topology as Figure 4-18, except using the negotiated return EMIC instead of IP forwarding.

Figure 4-19 *Branch Topology Using GRE Return EMIC*



Outbound traffic is pinned to a single link!

- Client sends traffic to its configured default gateway, which is currently active on the right-hand router (GLBP).
- Traffic is **intercepted** by WCCP and redirected to the WCCP.
- Egress traffic from the WAE is sent **transparently** to its default gateway.

Network Integration Best Practices

The following network integration best practices are recommended for most WAAS deployments:

- **Leave the physical WAE interfaces set to auto-sense:** Because it is possible that some of your WAEs will be able to run at 1-Gbps speed, leaving all of the WAEs deployed set to auto-sense simplifies the configuration and deployment. In addition, an alarm will be raised in the Central Manager if an interface negotiates to half duplex.
- **Use EtherChannel for interface redundancy when both physical WAE interfaces connect to the same LAN switch:** Improve performance by providing 2X the available LAN bandwidth.
- **Use a standby interface for interface redundancy when both physical WAE interfaces connect to different LAN switches:** Increase WAE availability in the event of a problem with the primary interface or connected LAN switch.
- **Always configure a MD5 password for WCCP service groups:** Protect the integrity of the service group members by making sure that only authorized devices can join the service group.
- **Stick to inbound WCCP redirection:** Even on software-based platforms, inbound redirection is more efficient.
- **On hardware-based platforms, configure WCCP using the following guidelines:**
  - — Use L2 forwarding instead of GRE forwarding.
  - — Always use mask assignment.
  - — Only use inbound redirection.
  - — Do not use the **ip wccp redirect exclude in** command.
- **Only use the GRE return EMIC on software-based platforms (i.e. ISR routers):** Hardware-based platforms process GRE return traffic completely in software, which causes serious performance issues.
- **Run a recommended version of IOS for WCCP:** Tables 4-5 and 4-6 list the minimum recommended IOS versions when running WCCP with Cisco WAAS.

Table 4-5 *Minimum IOS Recommendations: Software-Based Platforms*

Major Version	M Train	T Train
12.1	12.1(14)	12.1(3)T
12.2	12.2(26)	12.2(14)T
12.3	12.3(13)	12.3(14)T5
12.4	12.4(10)	12.4(9)T1

Table 4-6 *Minimum IOS Recommendations: Hardware-based Platforms*

Platform	Version
Catalyst 3560/3750	12.2(37)SE
Catalyst 4500/4900	12.2(31)SG
Catalyst 6500, Sup2	12.2(18)SXF13
Catalyst 6500, Sup32	CatOS 8.5/12.2(18)SXF13
Catalyst 6500, Sup720 (Native)	12.2(18)SXF13
Catalyst 6500, Sup720 (Hybrid)	CatOS 8.5/12.2(18)SXF13

Summary

This chapter provided a detailed examination of the various methods for integrating WAAS into the network infrastructure. The chapter reviewed the various techniques for physical connectivity, including options for increased interface bandwidth and high availability. The chapter also previewed the network interception techniques that are used to transparently redirect traffic to the WAAS infrastructure for optimization. Particular focus was given to WCCPv2 and inline interception, which are the two most common interception methods. The interception method you choose is a site-local decision. For example, you can use WCCPv2 at some locations and inline at other locations. Finally, the chapter discussed the different egress methods available in WAAS, which provide control over how traffic on intercepted connections is reinserted into the network after redirection to a WAE. You should now have a good feel for the flexibility of the WAAS solution when it comes to network integration. The techniques available allow Cisco WAAS to integrate into network infrastructures of any size and complexity. The next chapter begins to put these various techniques to use, as you look at specific deployment models for the branch office environment.