# Configuring IPv4 Broadcast Packet Handling

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

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


- Finding Feature Information, on page 1
- Information About IPv4 Broadcast Packet Handling, on page 1
- How to Configure IP Broadcast Packet Handling, on page 12
- Configuration Examples for IP Broadcast Packet Handling, on page 22
- Additional References for WCCP—Configurable Router ID, on page 23
- Feature Information for IP Broadcast Packet Handling, on page 24


# Finding Feature Information

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

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

# Information About IPv4 Broadcast Packet Handling

## IP Unicast Address

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

# IP Broadcast Address

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

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

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

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

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

# IP Directed Broadcast Address

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

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

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

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

# IP Directed Broadcasts

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

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

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

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

# IP Multicast Addresses

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

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

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

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

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

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

# Early IP Implementations

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

# DHCP and IPv4 Broadcast Packets

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

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

# UDP Broadcast Packet Forwarding

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

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

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

# UDP Broadcast Packet Flooding

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

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

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

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

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

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

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

# IP Broadcast Flooding Acceleration

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

# Default UDP Port Numbers

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

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

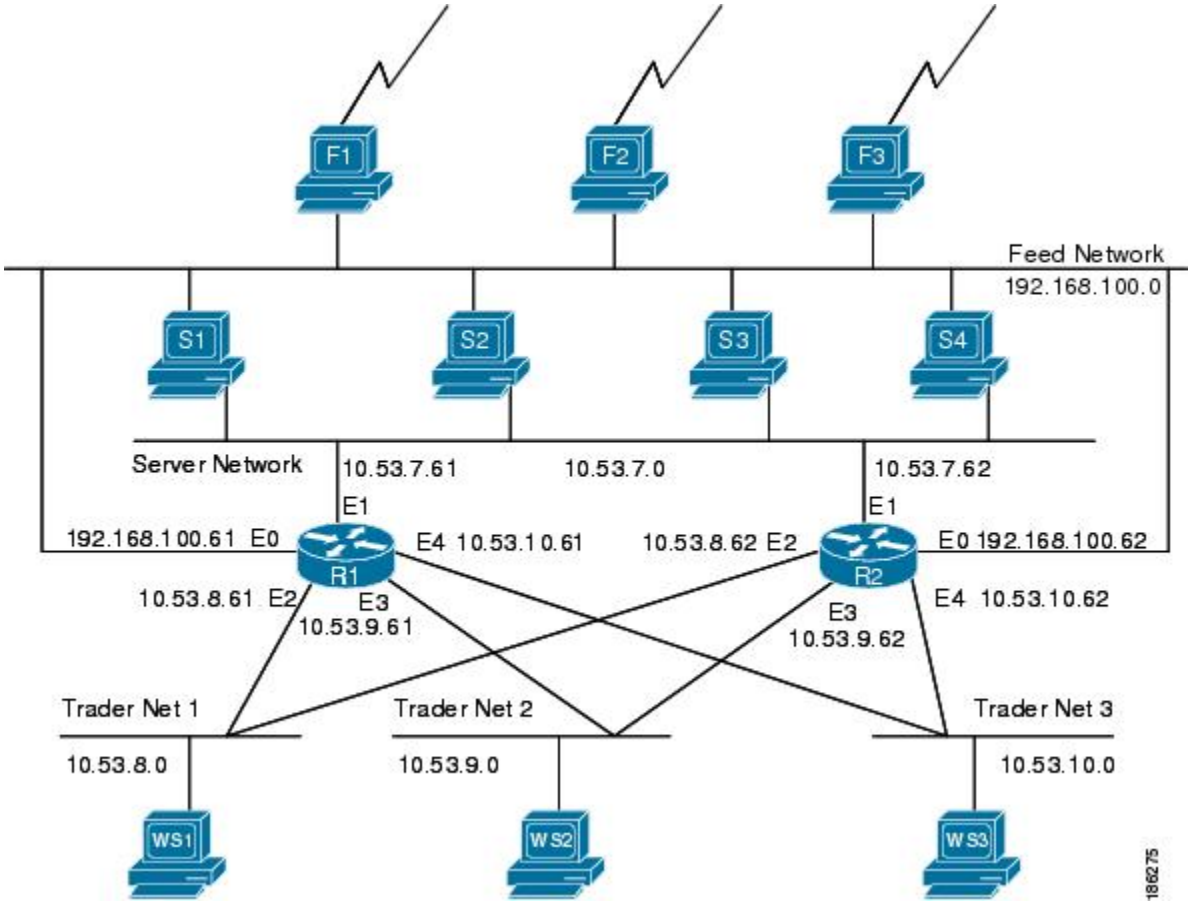
# Default IP Broadcast Address

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

# UDP Broadcast Packet Case Study

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

*Figure 1: Topology that Requires UDP Broadcast Forwarding*

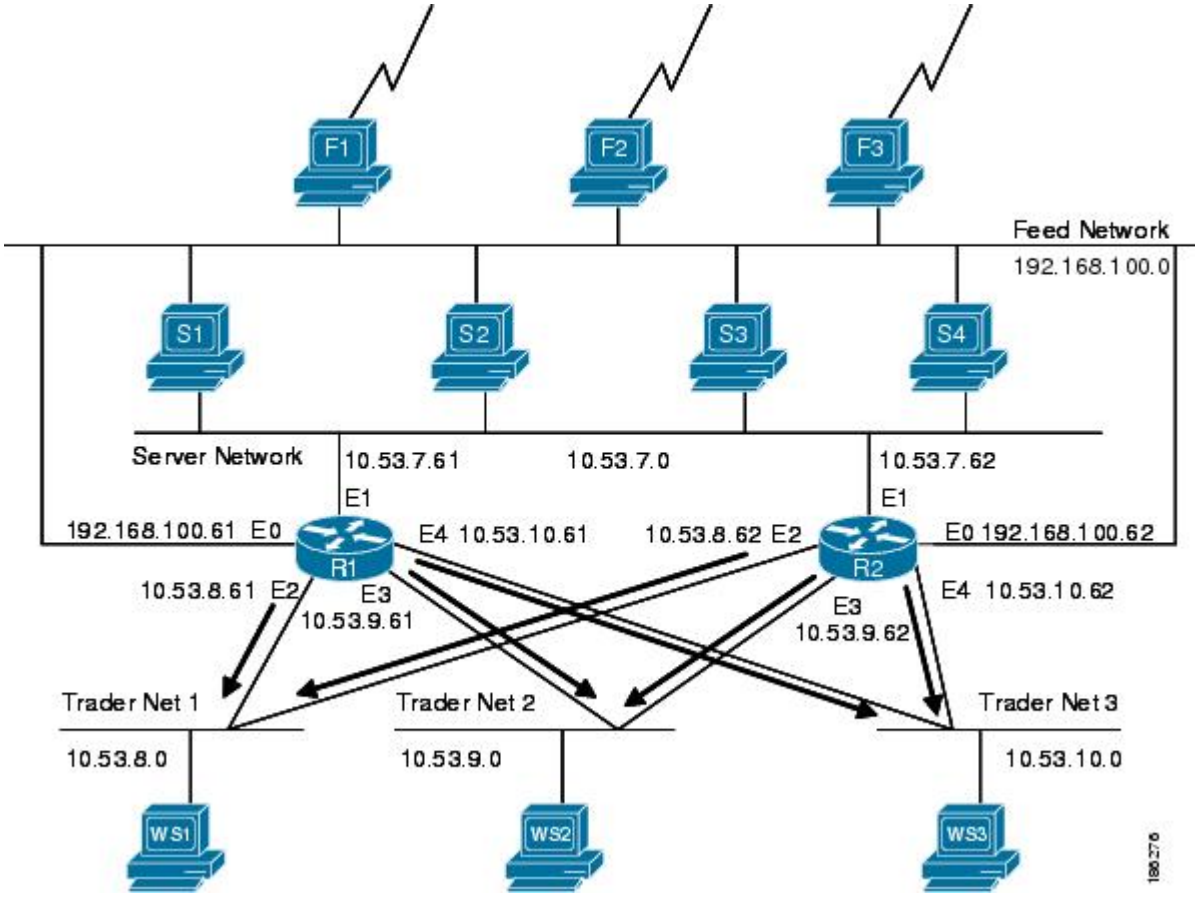



The following sections explain the possible solutions for this application:

## UDP Broadcast Packet Forwarding

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

*Figure 2: Flow of UDP Packets*




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

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

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

**Note** Bidirectional traffic in this topology significantly impacts router performance.

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

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

## UDP Broadcast Packet Flooding

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

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

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

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

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

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

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

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

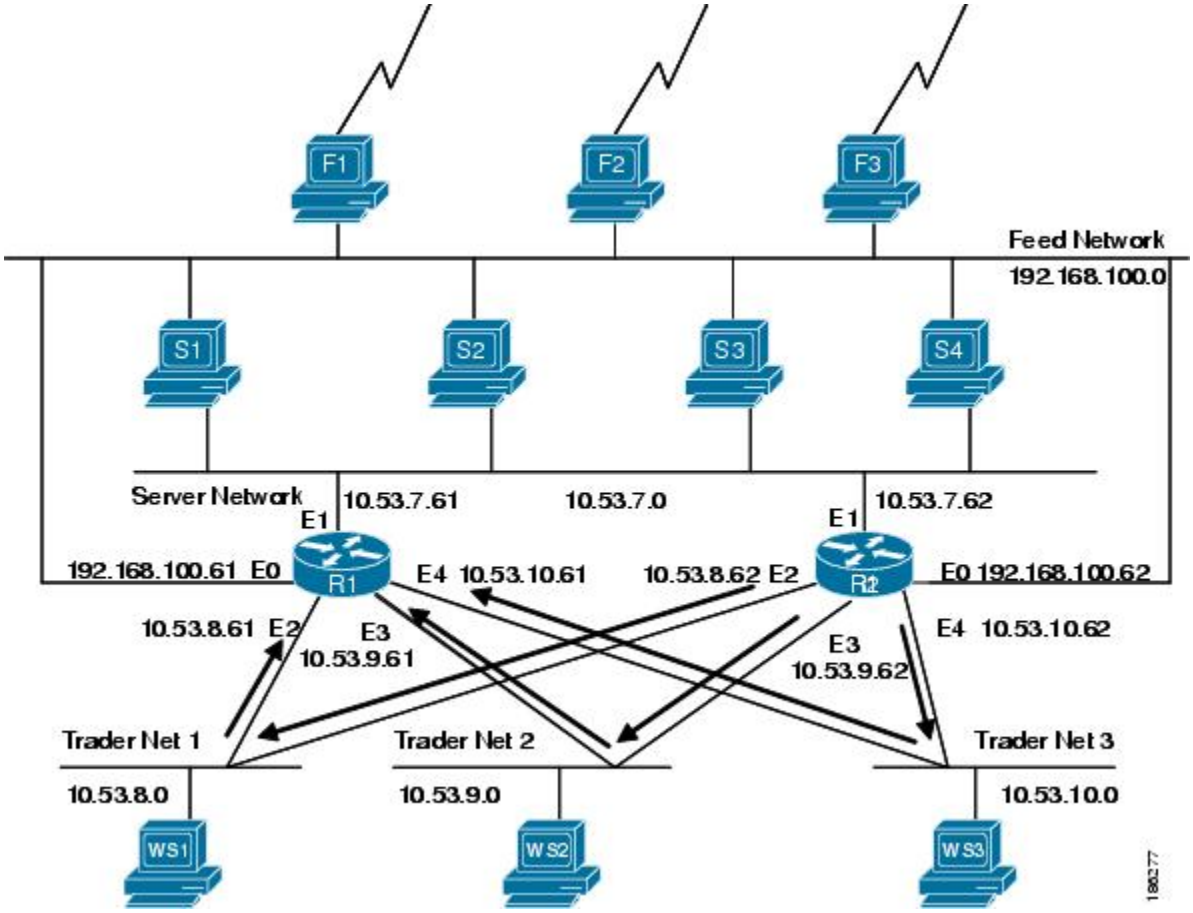
On the routers, the **holdtime**, **maxadvertinterval**, and **minadvertinterval** keywords of the **ip irdp** command reduce the advertising interval from the default so that the IRDP daemons running on the hosts expect to see advertisements more frequently. With the advertising interval reduced, the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

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

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

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

*Figure 3: Data Flow with UDP Flooding and IRDP*




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

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

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

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

# How to Configure IP Broadcast Packet Handling

## Enabling IP Directed Broadcasts Without an Access List

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

### SUMMARY STEPS

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

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **interface** *type number*<br>**Example:**<br>`Device(config)# interface fastethernet 0/1` | Specifies an interface and enters interface configuration mode. |
| **Step 4** | **ip address** *address mask*<br>**Example:**<br>`Device(config-if)# ip address 172.16.10.1 255.255.255.0` | Assigns an IP address to the interface. |
| **Step 5** | **ip directed-broadcast**<br>**Example:**<br>`Device(config-if)# ip directed-broadcast` | Enables IP directed broadcasts on the interface.<br>• Configure this command on the interface that is connected to the IP network address of the directed broadcast packets.<br>• In this example the directed broadcast packets are addressed to 172.16.10.255. |

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

# Enabling IP Directed Broadcasts with an Access List

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

### SUMMARY STEPS

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

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **access-list** *100-199* **permit ip** *source-address mask destination-address mask*<br>**Example:**<br>`Device(config)# access-list 100 permit ip 10.4.9.167 0.0.0.0 172.16.10.0 0.0.0.255` | Creates an access list to limit the IP directed broadcasts that are forwarded.<br>• In this example the IP directed broadcasts are sent by the host with the IP address of 10.4.9.167 to the IP directed broadcast address 172.16.10.255. |
| **Step 4** | **interface** *type number*<br>**Example:**<br>`Device(config)# interface fastethernet 0/0` | Specifies an interface and enters interface configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 5** | **ip address** *address mask*<br>**Example:**<br>`Device(config-if)# ip address 172.16.10.1 255.255.255.0` | Assigns an IP address to the interface. |
| **Step 6** | **ip directed-broadcast** *access-list*<br>**Example:**<br>`Device(config-if)# ip directed-broadcast 100` | Enables IP directed broadcasts on the interface for broadcast packets that are allowed by the access list you assigned. Configure this command on the interface that is connected to the IP network address of the directed broadcast packets.<br>• In this example the directed broadcast packets are addressed to 172.16.10.255. |
| **Step 7** | **end**<br>**Example:**<br>`Device(config-if)# end` | Exits the current configuration mode and returns to privileged EXEC mode. |

# Enabling Forwarding of UDP Broadcast Packets to a Specific Host

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

### SUMMARY STEPS

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

### DETAILED STEPS

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

| | Command or Action | Purpose |
|---|---|---|
| **Step 3** | **ip forward-protocol udp**<br>**Example:**<br>`Device(config)# ip forward-protocol udp` | Enables forwarding of UDP broadcast packets. |
| **Step 4** | **interface** *type number*<br>**Example:**<br>`Device(config)# interface fastethernet 0/1` | Specifies an interface and enters interface configuration mode. |
| **Step 5** | **ip address** *address mask*<br>**Example:**<br>`Device(config-if)# ip address 172.16.10.1 255.255.255.0` | Assigns an IP address to the interface. |
| **Step 6** | **ip helper-address** *address*<br>**Example:**<br>`Device(config-if)# ip helper-address 172.16.10.2` | Enables an IP helper address for the interface that is receiving the UDP broadcast packets.<br>• In this example the IP destination address of the IP UDP broadcast packets is rewritten to 172.16.10.2. |
| **Step 7** | **end**<br>**Example:**<br>`Device(config-if)# end` | Exits the current configuration mode and returns to privileged EXEC mode. |

# Enabling Forwarding of UDP Broadcast Packets to a Range of Hosts

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

### SUMMARY STEPS

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

**DETAILED STEPS**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **ip forward-protocol udp**<br>**Example:**<br>`Device(config)# ip forward-protocol udp` | Enables forwarding of UDP broadcast packets. |
| **Step 4** | **interface** *type number*<br>**Example:**<br>`Device(config)# interface fastethernet 0/0` | Specifies an interface and enters interface configuration mode. |
| **Step 5** | **ip address** *address mask*<br>**Example:**<br>`Device(config-if)# ip address 192.168.10.1 255.255.255.0` | Assigns an IP address to the interface. |
| **Step 6** | **ip helper-address** *address*<br>**Example:**<br>`Device(config-if)# ip helper-address 172.16.10.255` | Enables an IP helper address for the interface that is receiving the UDP broadcast packets.<br>• In this example an IP directed broadcast address is used. The IP destination address of the IP UDP broadcast packets is rewritten to 172.16.10.255.<br>• All of the hosts on the 172.16.10.0/24 network that support the application or service that the UDP broadcast packets are intended for will respond to the UDP broadcast packets.<br>**Note** This often results in the source of the UDP broadcast packets receiving responses from two or more hosts. In most circumstances the source of the UDP broadcast packets accepts the first response and ignores any subsequent responses. In some situations the source of the UDP broadcast packets cannot handle duplicate responses and reacts by reloading, or other unexpected behavior. |

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

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

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

***Table 1: Configuration Register Settings for Broadcast Address Destination***

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

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

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

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

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

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

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

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

### SUMMARY STEPS

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

### DETAILED STEPS

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

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

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

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

## SUMMARY STEPS

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

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **interface** *type number*<br>**Example:**<br>`Device(config)# interface fastethernet 0/1` | Specifies an interface and enters interface configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 4** | **ip address** *address mask*<br>**Example:**<br>`Device(config-if)# ip address 172.16.10.1 255.255.255.0` | Assigns an IP address to the interface. |
| **Step 5** | **ip broadcast-address** *address*<br>**Example:**<br>`Device(config-if)# ip broadcast-address 172.16.10.255` | Specifies the IP broadcast address<br>• In this example IP broadcasts are sent to 172.16.10.255. |
| **Step 6** | **end**<br>**Example:**<br>`Device(config-if)# end` | Exits the current configuration mode and returns to privileged EXEC mode. |

# Configuring UDP Broadcast Packet Flooding

### Before you begin

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

### SUMMARY STEPS

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

### DETAILED STEPS

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

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

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

# Configuration Examples for IP Broadcast Packet Handling

## Example: Enabling IP Directed Broadcasts with an Access List

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

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

## Example: Configuring UDP Broadcast Packet Flooding

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

# Additional References for WCCP—Configurable Router ID

### Related Documents

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

### Standards

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

### MIBs

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

### RFCs

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

### Technical Assistance

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

# Feature Information for IP Broadcast Packet Handling

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

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

***Table 2: Feature Information for IP Broadcast Packet Handling***

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