



Solving IP Problems

This chapter describes Internet protocols and provides information on how to resolve associated problems. This chapter contains the following sections:

- [IP Overview, page 8-1](#)
- [IP Addressing, page 8-5](#)
- [Troubleshooting TCP/IP, page 8-11](#)

IP Overview

Internet protocols can be used to communicate across any set of interconnected networks. They are suited equally well for LAN and WAN communications.

There are two main types of Internet protocols:

- Transport protocols move packets along the selected path.
- Routing protocols find the best path for packets to travel from one point to another, either within the same network or across networks.

Transport Protocols

Transport protocols move data so that it can be processed by upper-layer protocols. The most common upper-layer protocols include File Transfer Protocol (FTP), Simple Network Management Protocol (SNMP), and Simple Mail Transfer Protocol (SMTP).

The transport layer is implemented by the following two protocols:

- [Transmission Control Protocol](#)
- [User Datagram Protocol](#)

Transmission Control Protocol

Transmission Control Protocol (TCP) provides connection-oriented data transport, including full-duplex, acknowledged, and flow-controlled service to upper-layer protocols.

TCP moves data in a continuous, unstructured byte stream in which bytes are identified by sequence numbers. By using sequence numbers and acknowledgment messages, TCP can provide a sending node with delivery information about packets transmitted to a destination node. When data has been lost in transit from the source to the destination, TCP can retransmit the data until a timeout condition is reached or until successful delivery has been achieved. TCP can also recognize duplicate messages and discard them appropriately. If the sending device is transmitting too fast for the receiving device, TCP can employ flow-control mechanisms to slow data transfer. TCP can also communicate delivery information to the upper-layer protocols and applications that it supports.

User Datagram Protocol

User Datagram Protocol (UDP) provides connectionless data transport. UDP is a much simpler protocol than TCP and is more efficient in situations in which the reliability mechanisms of TCP are not required.

In connectionless network architectures, such as IP, the responsibility for session establishment and signaling resides in the end stations. To successfully emulate voice services across an IP network, enhancements to the signaling stacks are required.

For example, an H.323 agent is added to the router for standards-based support of the audio and signaling streams; the following protocols are all part of the flow:

- The Q.931 protocol is used for call establishment and teardown between H.323 agents, or end station.
- The Real Time Control Protocol (RTCP) is used to establish the audio channels.

- A reliable session-oriented protocol, TCP, is deployed between end stations to carry the signaling channels.
- The Real Time Transport Protocol (RTP), which is built on top of UDP, is used for transport of the real-time audio stream.
- RTP uses UDP as its transport mechanism because UDP has less delay than TCP and because actual voice traffic, unlike data traffic or signaling, tolerates low levels of loss and cannot effectively exploit retransmission

Therefore, UDP is well suited for voice traffic, for the following reasons:

- No retransmission of dropped packets—Unlike TCP, UDP does not attempt to retransmit dropped packets. Since voice traffic is far more sensitive to delays than data traffic, UDP is favored for real-time applications because it minimizes transmission delays by omitting the connection setup process, flow control, and retransmission.
- Less CPU overhead—Because UDP is stateless, it lessens the CPU workload by not requiring the CPU to keep track of the state information that connection-oriented protocols, such as TCP, require.
- Smaller packets—Voice over IP (VoIP) uses small packets that are sent out at consistent intervals, depending on the digital signal processor (DSP) and codec (coder-decoder) used. The small size of the UDP header (8 bytes) provides savings in bandwidth costs.

Internet Protocol

Internet Protocol (IP) is the primary network layer protocol in the Internet protocol suite. In addition to internetwork routing, IP provides error reporting and fragmentation and reassembly of long datagrams (information units) for transmission over networks with different maximum data unit sizes.

Routing Protocols

There are many routing protocols. Interior routing protocols instruct routers how to route packets within an autonomous system, such as a corporate intranet. By comparison, border routing protocols route packets between the autonomous systems that make up the Internet.

The following are among the most common interior routing protocols.

- [Routing Information Protocol](#)
- [Enhanced Interior Gateway Routing Protocol](#)
- [Open Shortest Path First](#)

Routing Information Protocol

Routing Information Protocol (RIP) is commonly used for interior routing. RIP broadcasts its routing table every 30 seconds. RIP allows 25 routes per packet; on large networks, multiple packets are required to send the whole routing table. Bandwidth utilization is a problem on large RIP networks that include low-bandwidth links.

RIP uses a single routing metric (hop count) to measure the distance to a destination network. This means that if multiple paths to a destination exist, RIP maintains only the path with the fewest hops, even if other paths have a higher aggregate bandwidth, lower aggregate delay, less congestion, and so on.

Enhanced Interior Gateway Routing Protocol

Enhanced Interior Gateway Routing Protocol (EIGRP) uses significantly less bandwidth than Interior Gateway Routing Protocol (IGRP) or other distance-vector protocols. A router using this algorithm develops its routing table by using the concept of a feasible successor. A feasible successor is a neighboring router that has the lowest-cost path to a destination.

When a router detects that a link has failed, if a feasible successor has an alternate route, the router switches to the alternate route immediately, without causing any network traffic. If there is no successor, the router sends a query to neighbors. The query propagates across the network until a new route is found.

An EIGRP router develops a topology table that contains all destinations advertised by neighboring routers. Each entry in the table contains a destination and a list of neighbors that have advertised the destination. For each neighbor, the entry includes the metric that the neighbor advertised for that destination. A router computes its own metric for the destination by using each neighbor's metric in combination with the local metric that the router uses to reach the neighbor. The router compares metrics and determines the lowest-cost path to a destination and a feasible successor to use if the lowest-cost path fails.

Open Shortest Path First

The Open Shortest Path First (OSPF) routing protocol propagates only changes, to minimize bandwidth utilization. Other network traffic is limited to database-synchronization traffic that occurs infrequently (every 30 minutes) and hello packets that establish neighbor adjacencies and that are used to elect a designated router on LANs.

An OSPF router multicasts link-state advertisements (LSAs) to all other routers within the same hierarchical area. The LSA gives the status of attached interfaces and the cost of sending a data packet on the interface. (Cost is the OSPF metric.)

OSPF routers accumulate link-state information to calculate the shortest path to a destination network. The result of the calculation is a database of the topology, called the *link-state database*. Each router in an area has an identical database.

All routers run the same algorithm, in parallel. From the link-state database, each router constructs a tree of shortest paths, with itself as the root of the tree. The shortest-path tree provides the route to each destination. Externally derived routing information appears on the tree as leaves. When there are several equal-cost routes to a destination, the traffic is distributed equally among them.

IP Addressing

IP addresses are globally unique, 32-bit numbers assigned by the Network Information Center. Globally unique addresses permit IP networks anywhere in the world to communicate with each other.

An IP address consists of four octets (1 octet = 8 bits), or 32 bits, written in dotted decimal format; for example, 34.0.0.1. The value in each octet ranges from 0 to 255 (decimal), or 00000000-11111111 (binary).

The following is an example of how binary octets convert to decimal:

```
1  1  1  1  1  1  1  1
128 64 32 16 8 4 2 1 (128+64+32+16+8+4+2+1=255)
```

The following is an example of an octet conversion:

```
0  1  0  0  0  0  0  1
0 64 0 0 0 0 0 1 (0+64+0+0+0+0+0+1=65)
```

The following is an example of an IP address in dotted-decimal format (4 octets):

```
10.      1.      23.      19 (decimal)
00001010.00000001.00010111.00010011 (binary)
```

An IP address is divided into three parts; these three parts span the 4 octets. The first part of an IP address designates the network address, the second part designates the subnet address, and the third part designates the host address. Subnet addresses are present only if the administrator has divided the network into subnetworks. (See the “[Dividing Networks with Subnets](#)” section on [page 8-7](#).)

The octets in an IP address are broken down to provide an addressing scheme that can accommodate large and small networks. There are five different classes of networks, A through E. Classes D and E are reserved but are not generally used.

To determine the class of an address, look at the first octet of the dotted-decimal address.

- Class A: 1–127
- Class B: 128–191
- Class C: 192–223

The lengths of the network, subnet, and host fields all vary, depending on the class.

- Class A address—The first octet constitutes the network portion of the IP address. The leading bit is always a *0*. Octets 2, 3, and 4 (the next 24 bits) are for the network administrator to divide into subnets and hosts, as needed. Class A networks are intended mainly for use with a few very large networks that have more than 65,536 hosts.
- Class B address—The first two octets constitute the network portion of the IP address. The leading bits are always *10*. Octets 3 and 4 (16 bits) are used for local subnets and hosts. Class B addresses are used for networks that have between 256 and 65,534 hosts.
- Class C address—The first three octets constitute the network portion of the IP address. The leading bits are always *110*. Octet 4 (8 bits) is used for local subnets and hosts, making Class C networks perfect for networks with less than 256 hosts.

[Table 8-1](#) shows the breakdown of Class A, B, and C addresses. Although the IP address range of a Class A network is from 1 through 127, 127 is reserved for loopback testing.

Table 8-1 Class A, B, and C Network Addresses

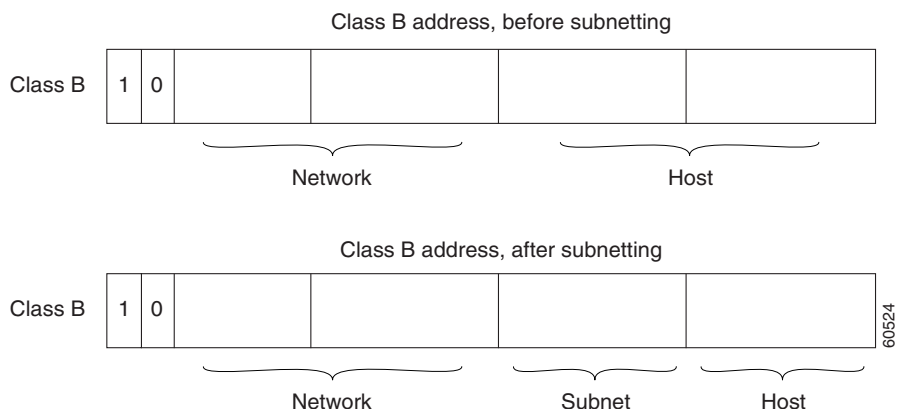
Class	Network.Host	Leading Bits	IP Address Range	Maximum Number of Networks / Maximum Number of Hosts
A	N.H.H.H	0	1–127	127 / 16,777,214
B	N.N.H.H	10	128–191	16,384 / 65,534
C	N.N.N.H	110	192–223	2,097,152 / 254

Dividing Networks with Subnets

IP networks can be divided into smaller units, called *subnets*. Subnets provide extra flexibility for administrators by creating multiple logical networks within a single Class A, B, or C network. The use of subnetting logically expands the size of the network.

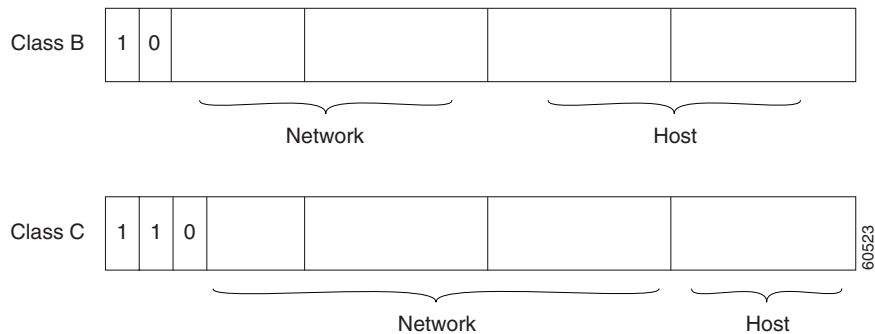
Each data link on a network must be a unique subnet, with every node on that link being a member of the same subnet. For serial interfaces (standard high-level data link control [HDLC]), you need to configure one subnet for the circuit, or “wire” (both ends of the serial connection will be in the same subnet).

Figure 8-1 shows a Class B address before and after subnetting.

Figure 8-1 Class B Address Formats

The dotted decimal representation of a Class B network, for example, could be 172.10.0.0. (All zeros in the host field of an address signifies the entire network.) To logically extend the network, the administrator can subdivide the network, using subnetting. This is done by borrowing bits from the host portion of the address and using them in the subnet field, as shown in [Figure 8-2](#).

Figure 8-2 Subnet Addresses



The number of bits borrowed for the subnet address can vary. To specify how many bits are used, IP provides the subnet mask. Subnet masks use the same format and address representation as IP addresses, with a subnet mask defined for each IP address. The subnet mask identifies which portion of the 4 octets is used to identify the network, with the remaining bits identifying the host. (Subnet masks have ones in all bits except those bits that specify the host field).

The following are the defined default masks for Class A, B, and C networks (255=1-bits, and 0=wildcard):

- Class A: 255.0.0.0
- Class B: 255.255.0.0
- Class C: 255.255.255.0

For example, the IP addresses 171.68.3.3 and 171.68.2.3, with a subnet mask of 255.255.255.0, indicate that the first 24 bits are masked. (These addresses can also be written as 171.68.3.3/24 and 171.68.2.3/24 to signify that 24 bits are used in the mask). A 24-bit mask uses 24 1-bits, which would look as follows in binary format:

```
11111111.11111111.11111111.00000000
```

When data is routed over a network, the router looks at the subnet mask to identify the data link on which an IP address resides. In the above example, using IP addresses 171.68.3.3 and 171.68.2.3, the router compares the first 3 octets of the IP addresses. Since the masked bits are not the same, the router knows that these addresses belong to different subnets.

If the IP addresses in this example were configured with a subnet mask of 255.255.0.0, the first 16 bits would be masked. In this case, the router would compare the first 2 octets of the IP addresses. The masked bits would be the same, indicating to the router that these addresses belong to the same subnet.

There is flexibility with subnetting, depending on your network needs. See the [“Guidelines for Determining Subnet Requirements” section on page 8-10](#). For example, if a network administrator has chosen to use 8 bits of subnetting, the third octet of a Class B IP address provides the subnet number. To illustrate, using IP address 172.10.1.0 as an example, 172.10 refers to the network portion of the address, and 1 refers to the subnet portion of the address. Using IP address 172.10.2.0 as an example, 172.10 refers to network 172.10, and 2 refers to subnet 2; and so on.

An example of subnet masking is shown in [Figure 8-3](#). This example shows that the subnet mask that specifies 16 bits of subnetting for Class A address 34.0.0.0 is 255.255.0.0. The subnet mask that specifies 24 bits of subnetting for Class A address 34.0.0.0 is 255.255.255.0.

Figure 8-3 Sample Subnet Masks

Class A address	0	0	1	0	0	0	1	0	. 0 .	. 0 .	. 0 .	34.0.0.0	
Subnet mask, 8 subnet bits									. 1 .	. 1 .	. 0 .	. 0 .	255.255.0.0
Class A address	0	0	1	0	0	0	1	0	. 0 .	. 0 .	. 0 .	34.0.0.0	
Subnet mask, 16 subnet bits									. 1 .	. 1 .	. 1 .	. 0 .	255.255.255.0

60525

Guidelines for Determining Subnet Requirements

Follow these guidelines to determine your subnet requirements:

- Determine the number of required network IDs
 - One for each subnet
 - One for each WAN connection
- Determine the number required host IDs per subnet
 - One for each TCP/IP host
 - One for each router interface
- Based upon the number of required hosts, create:
 - One subnet mask for the entire network
 - A unique subnet ID for each physical segment
 - A range of host IDs for each subnet

**Note**

For additional information on IP addressing formats and subnet masks, refer to *Internet Protocols*. For information specific to the number of allowable subnets, refer to the *Host/Subnet Quantities Table*.

As IP subnets have grown, administrators have looked for ways to use their address space more efficiently. One of the techniques that has resulted is called the *variable length subnet mask* (VLSM). With VLSM, an administrator can use a long mask on networks with few hosts and can use a short mask on subnets with many hosts. However, this technique is more complex than assigning the same subnet mask, and addresses must be assigned carefully. Both OSPF and EIGRP support VLSM.

Troubleshooting TCP/IP

This section presents protocol-related troubleshooting information for TCP/IP connectivity and performance problems. This section focuses on general TCP/IP problems and on routing problems related to RIP, EIGRP, and OSPF.

Each of the following sections describes a specific symptom, the problems that are likely to cause each symptom, and the solutions to those problems.

- [Routing Not Functioning Properly on New Interface, page 8-11](#)
- [Host Connections Fail, Using Certain Applications, page 8-14](#)
- [Problems Forwarding BOOTP and Other UDP Broadcasts, page 8-17](#)

Routing Not Functioning Properly on New Interface

Symptom A new interface is added to a router, but when routing is configured, it does not function properly on the new interface.

[Table 8-2](#) outlines the problems that might cause this symptom and describes solutions to those problems.

Table 8-2 Routing Not Functioning Properly on New Interface

Possible Problem	Solution
Interface or LAN protocol is down.	<p>Step 1 Enter the show interface command in privileged EXEC mode to determine whether the interface is “administratively down”:</p> <pre>C7750#show interface serial 0 Serial0 is administratively down, line protocol is down Hardware is HD64570 Internet address is 10.1.1.5 255.255.255.252 [...]</pre> <p>Step 2 If the interface is administratively down, bring up the interface, using the no shutdown configuration command. Following is an example of the no shutdown command:</p> <pre>C7750(config)#int serial0 C7750(config-if)#no shutdown C7750(config-if)#</pre>
	<p>Step 3 Enter the show interface command again to see whether the interface is now up.</p> <p>Step 4 If the interface is still down, there might be a hardware or media problem. For hardware problems, see Chapter 3, “Solving Hardware Problems.” For serial problems, see Chapter 6, “Solving Serial Connection Problems.” For Ethernet problems, see Chapter 7, “Solving Ethernet Problems.”</p>

Table 8-2 Routing Not Functioning Properly on New Interface (continued)

Possible Problem	Solution
<p>Misconfigured or missing ASI, MRP, or router configuration command.</p>	<p>Step 1 Enter the show running-config command in privileged EXEC mode to view the device configuration.</p> <p>Step 2 Ensure that there is a configuration command specified for the network to which the interface belongs.</p> <p>For example, if you assign the IP address 192.168.52.42 to the new interface, enter the following commands to enable OSPF on the interface:</p> <pre>c7750(config)#router OSPF c7750(config-router)#network 192.168.52.0</pre> <p>Also ensure that process IDs, addresses, and other variables are properly specified for the routing protocol that you are using. For more information, refer to the <i>Cisco IOS IP Command Reference, Volume 2 of 3: Routing Protocols, Release 12.2</i>.</p>
<p>No active interfaces are configured with an IP address (OSPF only).</p>	<p>Step 1 OSPF uses an IP address on the device as its router ID. Therefore, to configure the OSPF protocol on a device, you need at least one active interface configured with an IP address. If there is no active interface with an IP address, the device will return the following error:</p> <pre>C7750(config)#router ospf 100 C7750(config)# OSPF: Could not allocate router id</pre>

Table 8-2 Routing Not Functioning Properly on New Interface (continued)

Possible Problem	Solution
	<p>Step 2 Enter the show ip interface command in privileged EXEC mode on the device to ensure that an interface is up and that it is configured with an IP address.</p> <p>Step 3 If there is no active interface with an IP address, configure an interface with the ip address interface configuration command. If necessary, use the no shutdown interface configuration command to bring up an interface. The following example shows the commands to enter configuration mode, assign an IP address to serial 0, and perform a no shutdown command on the interface:</p> <pre>C7750#conf t Enter configuration commands, one per line. End with CNTL/Z. C7750(config)#interface serial 0 C7750(config-if)#ip address 10.1.1.5 255.255.255.252 C7750(config-if)#no shutdown C7750(config-if)#</pre>

Host Connections Fail, Using Certain Applications

Symptom Connection attempts using some applications are successful, but attempts using other applications fail. For instance, you might be able to ping a host successfully, but Telnet connections fail.

[Table 8-3](#) outlines the problems that might cause this symptom and describes solutions to those problems.

Table 8-3 Host Connections Fail, Using Certain Applications

Possible Problem	Solution
Misconfigured access control lists (ACLs) or other filters.	<p>Step 1 Enter the show running-config command to check each ASI, MRP, and router in the path. Determine whether there are IP access control lists (ACLs) configured on each device.</p> <p>Step 2 If IP ACLs are enabled on a device, disable them by entering the appropriate commands. An ACL may be filtering traffic from a TCP or UDP port. For example, to disable input ACL 80, enter the following command:</p> <pre>C7750(config-if)#no ip access-group 80 in</pre> <p>Step 3 After disabling all the ACLs on the device, determine whether the application operates normally.</p> <p>If the application operates normally, an ACL was probably blocking traffic.</p> <p>Step 4 To isolate the problematic ACL, enable ACLs one at a time until the application no longer functions. Check the misconfigured ACL to determine whether it is filtering traffic from any TCP or UDP ports.</p> <p>Step 5 If the misconfigured ACL denies specific TCP or UDP ports, make sure that it does not deny the port used by the application in question (such as TCP port 23 for Telnet).</p> <p>Enter explicit permit statements for those ports used by applications you want to have functional. The following commands allow DNS¹ and NTP² requests and replies:</p> <pre>access-list 101 permit udp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 eq 53 access-list 101 permit udp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 eq 123</pre>

Table 8-3 Host Connections Fail, Using Certain Applications (continued)

Possible Problem	Solution
	<p>Step 6 If you altered an ACL, enable it to determine whether the application can still operate normally.</p> <p>Step 7 If the application operates normally, perform the preceding steps to isolate any other misconfigured ACLs until the application operates correctly with all ACLs enabled.</p> <p>For more information about misconfigured ACLs, see the “Example of Misconfigured Access Control List” section on page 8-16.</p>

1. DNS = Domain Name System
2. NTP = Network Time Protocol

Example of Misconfigured Access Control List

Misconfigured ACLs can cause connectivity and performance problems. Suppose that there are three interconnected routers (X, Y, and Z). When an administrator attempts to trace the route by using the **tracert** command, the connection fails, despite the fact that **telnet** and **ping** commands successfully reach Router Z from Router X.

When examining the configuration of Router Y (the router in the middle between Router X and Router Z), the administrator finds the following extended ACL configured on Router Y:

```
RouterY#show ip access-lists
Extended IP access list 101
    permit tcp any any eq telnet
    permit icmp any any
RouterY#show running-config
[...]
interface Serial0
    ip address 192.168.54.92 255.255.255.0
    ip access-group 101 out
[...]
```

The permit statements in the ACL show that only Internet Control Message Protocol (ICMP) traffic (used by the ping application) and TCP traffic (used by the Telnet application) are permitted to pass serial interface 0 on Router Y. Any traffic sent to UDP ports, including the default ports used by the trace application (UDP ports 33434 and higher), is implicitly denied.

To allow trace traffic to pass through Router Y, the administrator makes the following change to the ACL:

```
RouterY#configure terminal
RouterY(config)#access-list 101 permit udp any any gt 33433
RouterY(config)#^Z
RouterY#
%SYS-5-CONFIG_I: Configured from console by console
RouterY#show ip access-lists
Extended IP access list 101
    permit tcp any any eq telnet
    permit icmp any any
    permit udp any any gt 33433
RouterY#
```

Problems Forwarding BOOTP and Other UDP Broadcasts

Symptom Problems occur when forwarding BOOTP or other UDP broadcast packets. UDP broadcasts sent from network hosts are not forwarded by ASIs, MRPs, or routers. Diskless workstations cannot boot.

[Table 8-4](#) outlines the problems that might cause this symptom and describes solutions to those problems.

Table 8-4 Problems Forwarding BOOTP and Other UDP Broadcasts

Possible Problem	Solution
Missing or misconfigured ip helper-address specification.	<p>Step 1 Enter the debug ip udp command in privileged EXEC mode on the device that should be receiving packets from the host. Check the output of the command to see whether packets are being received from the host.</p> <p>This debug command can use considerable CPU cycles on the device. Do not enable it if your network is heavily congested. If your network is congested, you can attach a network analyzer to see whether UDP broadcasts are being received from the host. (For more information about network analyzers, see the “Using Third-Party Troubleshooting Tools” section on page 2-22.)</p>

Table 8-4 Problems Forwarding BOOTP and Other UDP Broadcasts (continued)

Possible Problem	Solution
	<p>Step 2 If the device does not receive packets from the host, there is a problem with the host or the application. Consult the documentation for the host or application.</p> <p>If the device receives packets from the host, enter the show running-config command in privileged EXEC mode to check the configuration of the device interface that first receives the packet from the host.</p> <p>Step 3 Look for an ip helper-address <i>address</i> interface configuration command entry for that interface. Make sure that the specified address is correct (it should be the IP address of a server application such as a BOOTP server). If there is no command entry, then no helper address is configured.</p> <p>Step 4 If there is no IP helper address configured, or if the wrong address is specified, add or change the helper address by entering the ip helper-address <i>address</i> interface configuration command.</p> <p>For example, to configure the IP address 192.168.192.6 as the helper address on Ethernet interface 0, enter the following commands:</p> <pre>C7750(config)#interface e0 C7750(config-if)#ip helper-address 192.168.192.6</pre>
UDP broadcasts are being forwarded out of nondefault ports.	<p>Step 1 Specifying an IP helper address ensures that broadcasts from only a certain default set of UDP ports are forwarded. UDP broadcasts forwarded out of other ports require further configuration.</p> <p>Enter an ip forward-protocol udp <i>port</i> global configuration command on the device for each applicable port. For example, to forward UDP broadcasts from port 200, enter the following command:</p> <pre>C7750(config)#ip forward-protocol udp 200</pre> <p>Step 2 To allow forwarding of all UDP broadcasts, enter the following command:</p> <pre>C7750(config)#ip forward-protocol udp</pre>

Table 8-4 Problems Forwarding BOOTP and Other UDP Broadcasts (continued)

Possible Problem	Solution
UDP broadcast forwarding is disabled on UDP ports.	<p>Step 1 Use the show running-config command in privileged EXEC mode on the device, and look for any no ip forward-protocol udp global configuration command entries. Such entries disable the forwarding of UDP traffic out specific ports.</p> <p>For example, entering the no ip forward-protocol udp 53 global configuration command disables the forwarding of all UDP traffic out of port 53, which is the default port for DNS broadcasts. The following entry is shown in the configuration:</p> <pre>no ip forward-protocol udp domain</pre> <p>Step 2 If UDP broadcasts are disabled at specific UDP ports, enter the ip forward-protocol udp port global configuration command (you can also specify a keyword, such as domain, rather than the port number).</p> <p>For example, to reenab DNS broadcasts, enter the following command:</p> <pre>C7750(config)#ip forward-protocol udp domain</pre> <p>To allow forwarding of BOOTP broadcasts, enter the following command:</p> <pre>C7750(config)#ip forward-protocol udp bootp</pre> <p>To allow forwarding of all UDP broadcasts, enter the following command:</p> <pre>C7750(config)#ip forward-protocol udp</pre>
ACLs or other lists are misconfigured.	<p>Step 1 Enter the show running-config command to check the configuration of each ASI, MRP, or router in the path. See whether there are ACLs configured on the device.</p> <p>Step 2 If ACLs are enabled on the device, disable them by using the appropriate commands. For example, to disable input ACL 10, enter the following command:</p> <pre>C7750(config-if)#no ip access-group 10 in</pre>

Table 8-4 Problems Forwarding BOOTP and Other UDP Broadcasts (continued)

Possible Problem	Solution
	<p>Step 3 After disabling all ACLs, determine whether BOOTP or other UDP broadcasts are forwarded normally. If broadcasts are forwarded normally, a misconfigured ACL was probably blocking traffic.</p> <p>Step 4 To isolate the misconfigured ACL, enable ACLs one at a time until broadcasts are no longer forwarded.</p> <p>Step 5 Check the misconfigured ACL to determine whether it is filtering traffic from any UDP ports. If an ACL denies specific UDP ports, make sure that it does not deny ports used to forward the broadcast traffic in question (such as port 67 for BOOTP transmissions, or port 68 for BOOTP replies).</p> <p>Enter explicit permit statements for those ports used to forward broadcasts that you want to have forwarded.</p> <p>The following is an example of using a permit statement in an ACL:</p> <pre data-bbox="561 857 932 906">C7750(config)#access-list 101 permit udp any any eq</pre> <p>For more information about misconfigured access lists, see the “Example of Misconfigured Access Control List” section on page 8-16.</p> <p>Step 6 If you altered an ACL, enable it to see whether broadcasts are still forwarded normally.</p> <p>Step 7 If problems persist, perform Step 1 through Step 6 on ASIs, MRPs, and routers in the path until broadcast traffic is forwarded correctly.</p>