Distributed Routing Software

Routing Protocols User's Guide

Part Number: AA–QL2DC–TE

January 1996

This manual explains how to configure and monitor the Distributed Routing Software routing protocols shipped with your RouteAbout Access router.

Revision/Update Information:	This is a revised manual.
Software Version:	Distributed Routing Software V1.1

This manual was produced by Shared Engineering Services.

Contents

8 Configuring and Monitoring DNA IV

9 Configuring OSI/DNA V

10 Monitoring OSI/DNA V

11 Configuring DVMRP

12 Monitoring DVMRP

13 Configuring IP

14 Monitoring IP

15 Configuring IPX

16 Monitoring IPX

17 Configuring OSPF

18 Monitoring OSPF

19 Configuring SNMP

20 Monitoring SNMP

21 Configuring Bandwidth Reservation

22 Monitoring Bandwidth Reservation

23 Configuring BGP4

Figures

Tables

Preface

Objectives

This *Routing Protocols User's Guide* explains how to configure and monitor the protocol software shipped with your Bridging Router.

Audience

This guide is intended for persons who install and operate computer networks. Although experience with computer networking hardware and software is helpful, you do not need programming experience to use the protocol software.

For further information on each of the protocols discussed in this book, refer to the *Bridging Router Reference Guide*.

Organization

This manual is organized as follows:

- Chapter 1 describes how to access the protocol configuration and monitoring processes.
- Chapter 2 describes how to configure AppleTalk Phase1.
- Chapter 3 describes how to monitor AppleTalk Phase 1.
- Chapter 4 describes how to configure AppleTalk Phase 2.
- Chapter 5 describes how to monitor AppleTalk Phase 2.
- Chapter 6 describes how to configure the Address Resolution Protocol (ARP) and how to use the ARP configuration commands.

- Chapter 7 describes how to monitor ARP protocol activity and how to use the ARP console commands.
- Chapter 8 describes how to access the DNA IV (Digital Network Architecture) protocol configuration and console processes and how to use the DNA IV commands.
- Chapter 9 describes how to access the OSI/DNA V protocol configuration processes and how to use the OSI/DNA V configuration commands.
- Chapter 10 describes how to access the OSI/DNA V console processes and how to use the OSI/DNA V console commands.
- Chapter 11 describes how to configure DVMRP (Distance Vector Multicast Routing Protocol) using the DVMRP configuration commands.
- Chapter 12 describes how to monitor DVMRP protocol activity and how to use the DVMRP console commands.
- Chapter 13 describes how to access the IP (Internet Protocol) configuration process and how to use the IP configuration commands.
- Chapter 14 describes how to access the IP console process and how to use the IP console commands.
- Chapter 15 describes how to access the IPX (Internet Exchange Packet Protocol) configuration process and how to use the IPX configuration commands.
- Chapter 16 describes how to access the IPX console process and how to use the IPX console commands.
- Chapter 17 describes how to access the OSPF (Open Shortest-Path-First Protocol) configuration process and how to use the OSPF configuration commands.
- Chapter 18 describes how to access the OSPF console process and how to use the OSPF console commands.
- Chapter 19 describes how to access the SNMP (Simple Network Management Protocol) configuration process and how to use the SNMP configuration commands.
- Chapter 20 describes how to access the SNMP console process and how to use the SNMP console commands.
- Chapter 21 describes the Bandwidth Reservation configuration commands.
- Chapter 22 describes the Bandwidth Reservation monitoring commands.
- Chapter 23 describes the Border Gateway Protocol configuration commands.

- Chapter 24 describes the Border Gateway Protocol monitoring commands.
- Appendix A lists all the SNMP objects for each router interface.
- Appendix B discusses the sizes of packets for the various networks and protocols that the bridging routers support.
- Appendix C compares some of the well known protocols that the bridging routers support.
- Appendix D describes the MIBs or portions of MIBs contained in the Digital-Router-SNMP-Agent.

Associated Digital Documents

The following documents provide additional information about the router hardware and software:

- *Bridging Configuration Guide*, AA–QL29C–TE
- *Event Logging System Messages Guide*, AA–QL2AC–TE
- *Network Interface Operations Guide*, AA–QL2BC–TE
- *Routing Protocols Reference Guide*, AA–QL2CC–TE
- *System Network Architecture Guide*, AA–QU5SA–TE
- *System Software Guide*, AA–QL2EC–TE

Conventions Used in This Guide

`Special type`	This special type in examples indicates system output or user input.
`Boldface`	Boldface type in examples indicates user input.
lowercase-italics	Lowercase italics in command syntax or examples indicate variables for which either the user or the system supplies a value.
{ }	Braces indicate a choice you must make. Braces enclose values that either are separated by a vertical bar (|) or are listed vertically. Choose either from the values separated by the vertical bar or from the list enclosed by the braces. Do not type the braces in the line of code.
[ ]	Brackets enclose operands or symbols that are either optional or conditional. Specify the operand and value if you want the condition to apply. Do not type the brackets in the line of code.
|	A vertical bar indicates a choice you must make from the values separated by the bar. Do not type the vertical bar in the line of code.
key	Indicates that you press the specified key.
Ctrl/x	Indicates that you should hold the CONTROL key down and press the key specified by the x. The server displays the key combination as ^x.
RET	Indicates that you should press the Return key.

1

Getting Started

This chapter explains how access the processes required to configure and monitor the protocol software shipped with your bridging router.

For further information about the protocols discussed in this book, refer to the *Routing Protocols Reference Guide*.

Accessing Protocol Configuration and Console Processes

All protocols described in this guide have commands that are executed by doing one of the following:

- Accessing the protocol configuration process to initially configure and enable the protocol as well as perform later configuration changes.
- Accessing the protocol console process to monitor information about each protocol or make temporary configuration changes.

The procedures for accessing these processes is basically the same for all protocols. The next sections describe these procedures.

Accessing the Protocol Configuration Process

Each protocol configuration process is accessed through the router's CONFIG process. CONFIG is the second-level process of the router user interface that lets you communicate with third-level processes. Protocol processes are examples of third-level processes.

The CONFIG command interface is made up of levels that are called modes. Protocol configuration command interfaces are modes of the CONFIG interface. Each protocol configuration interface has its own prompt. For example, the prompt for the TCP/IP protocol command interface is `IP config>`.

In general, the procedure for accessing the protocol configuration processes is as follows:

- Enter the CONFIG command process from OPCON and obtain the CONFIG prompt.
- Enter the desired protocol configuration process (with its own prompt) from the CONFIG prompt using the **protocol** command.

The following sections describe these procedures in more detail.

Entering the CONFIG Process

To enter the CONFIG command process from OPCON and obtain the CONFIG prompt:

1. At the OPCON prompt, enter the **status** command to find the pid (process ID) of CONFIG.

```
* status

Pid    Name    Status  TTY    Comments
 1    COpCon    IOW    TTY0
 2    Monitr    DET     --
 3    Tasker    IDL     --
 4    MOSDDT    DET     --
 5    CGWCon    IOW     --
 6    Config    IOW     --
 7    ROpCon    IOW    TTY1    janb
 8    ROpCon    RDY    TTY2
```

2. Enter the OPCON **talk** command and the pid for CONFIG. The pid for CONFIG is 6.

```
* talk 6
```

The console displays the CONFIG prompt (`Config>`). If the prompt does not appear, press **RETURN** again.

Entering the Desired Protocol Configuration Process

To enter the desired protocol configuration process from the CONFIG prompt:

1. At the CONFIG prompt, enter the **list configuration** command to see the numbers and names of the protocols available for the router. For example:

```
Config>list configuration
Hostname: [none]
Maximum packet size: [autoconfigured]
Maximum number of global buffers: [autoconfigured]
Number of Restarts before a Reload/Dump: 64
Logging disposition: detached
Console inactivity timer (minutes): 0
Physical console login: disabled
Modem control: disabled
Contact person for this node: [none]
Location of this node: [none]

Configurable Protocols:
Num Name  Protocol
0   IP    DOD-IP
3   ARP   Address Resolution
4   DN    DNA Phase IV
5   XNS   Xerox Network Systems
7   IPX   NetWare IPX
8   OSI   ISO CLNP/ESIS/ISIS
9   DVM   Distance Vector Multicast Routing Protocol
10  BGP   Border Gateway Protocol
11  SNMP  Simple Network Management Protocol
12  OSPF  Open SPF-Based Routing Protocol
14  APL   AppleTalk
15  DDS   Apollo Domain
22  AP2   AppleTalk Phase 2
23  ASRT  Adaptive Source Routing Transparent Enhanced Bridge
24  HST   TCP/IP Host Services

Configurable Features:
Num Name  Feature
0   WRS   WAN Restoral
1   BRS   Bandwidth Reservation
2   MCF   MAC Filtering
```

2. From the CONFIG prompt, enter the **protocol** command with the number or *short name* of the protocol you want to configure. You can obtain the protocol number and short name from the **list configuration** command display. The following example shows the command for accessing the IP protocol configuration process by the protocol short name:

```
Config> protocol IP
```

   The protocol configuration prompt then displays on the console. This example shows the IP protocol prompt `IP config>`:

```
IP config>
```

You can achieve the same result by entering the **protocol** command followed by the protocol "number." In the following example, the command was entered to access the IP protocol configuration process by the protocol number:

```
Config> protocol 0
```

The protocol configuration prompt then displays on the console.

```
IP config>
```

You can now begin entering that protocol's configuration commands. See the corresponding protocol section of this guide for more information about specific protocol configuration commands.

In summary, the **protocol** command lets you enter the configuration process for the protocol software installed in your router. The **protocol** command enters a protocol's command process. After entering the **protocol** command, the prompt of the specified protocol appears. From the prompt, you can enter commands specific to that protocol.

Exiting the Protocol Configuration Process

After configuring or changing the protocol configuration process, exit the protocol configuration process:

1. Return to the CONFIG process by entering the protocol **exit** command. For example:

   ```
   IP config> exit
   ```

2. Return to the OPCON process by entering the OPCON intercept character (**ctrl-p**). For example:

   ```
   Config> ^p
   ```

Restarting the Router

Changes that you make to the protocol parameters through CONFIG do not take effect until you restart the router or reload the router software. The only exception is for certain DNA IV **NCP set** commands.

Note: The changes you make through CONFIG are retained in a configuration database in non-volatile memory. They are retained during power downs and are recalled when you restart the router.

To restart the router, enter the OPCON **restart** command. For example:

```
* restart
Are you sure you want to restart the router? (Yes or No): yes
```

Accessing the Protocol Console Process

To view information about the protocol or to change parameters at the console, you must access and use the protocol console process. Protocol console command interfaces are modes of the GWCON interface. Within the GWCON mode, each protocol console interface has its own prompt. For example, the prompt for the TCP/IP protocol is `IP>`.

In general, the procedure for accessing the protocol console processes is as follows:

- Enter the GWCON command process from OPCON and obtain the GWCON prompt.
- Enter the desired protocol console process from the GWCON prompt using the **protocol** command.

The next sections describe these procedures in more detail.

Entering the GWCON Command Process

The general process for entering the GWCON process from OPCON and obtaining the GWCON prompt is as follows:

1. Enter the **status** command to find the pid (process ID) of GWCON. For example:

```
* status

Pid    Name    Status  TTY    Comments
 1    COpCon    IOW    TTY0
 2    Monitr    DET     --
 3    Tasker    IDL     --
 4    MOSDDT    DET     --
 5    CGWCon    IOW     --
 6    Config    IOW     --
 7    ROpCon    IOW    TTY1    janb
 8    ROpCon    RDY    TTY2
*
```

2. At the OPCON prompt, enter the OPCON **talk** command and the pid number for GWCON. For example:

```
* talk 5
```

The GWCON prompt (+) then displays on the console. If the prompt does not appear, press **RETURN** again.

Entering a Protocol Console Process

To enter a protocol console process from the GWCON prompt:

1. At the GWCON prompt, enter the **configuration** command to see the protocols and networks configured for the bridge. For example:

```
+configuration

Portable MC68040 C Gateway [not configured] S/N 452
V15.1[]
Boot ROM version 0.4
Watchdog timer enabled
Auto–boot switch enabled
Manufacturing rest enabled
Manufacturing test disabled
Console baud rate: 0

Num Name  Protocol
0   IP    DOD–IP
3   ARP   Address Resolution
7   IPX   NetWare IPX
11  SNMP  Simple Network Management Protocol
14  APL   AppleTalk
22  AP2   AppleTalk Phase 2
23  ASRT  Adaptive Source Routing Transparent Enhanced Bridge

Num Name  Feature
2   MCF   MAC Filtering

7 Networks:
Net Interface  MAC/Data–Link         Hardware                    State
0   FDDI/0     IEEE 802.2/FDDI       WGE200 FDDI                  Up
1   Eth/0      Ethernet/IEEE 802.3                                Up
2   Eth/1      Ethernet/IEEE 802.3                                Up
3   Eth/2      Ethernet/IEEE 802.3                                Up
4   Eth/3      Ethernet/IEEE 802.3                                Down
5   Eth/4      Ethernet/IEEE 802.3                                Down
6   Eth/5      Ethernet/IEEE 802.3                                Down
```

2. Enter the GWCON **protocol** command with the protocol number or short name of the desired protocol displayed in the configuration information. In the following example, the command was entered for accessing the IP protocol console process:

```
+ protocol 0
```

OR

```
+ protocol IP
```

The protocol console prompt then displays on the console. This example shows the IP bridge prompt (`IP>`):

```
IP>
```

You can now begin entering that protocol's console commands. See the corresponding protocol section of this guide for more information about specific protocol console commands.

Exiting the Protocol Console Process

To exit the protocol console process and return to the OPCON process:

1. Return to the GWCON process by entering the protocol **exit** command. For example:

   ```
   IP> exit
   ```

2. Return to the OPCON process by entering the OPCON intercept character (**ctrl-p**). For example:

   ```
   + ^p
   ```

Protocol Names and Numbers

Table 1–1 lists the numbers that you enter along with the **protocol** command when accessing a specific protocol configuration or console process.

Table 1–1 Protocol Numbers and Names

Protocol Number	Protocol Short Name	Accesses the following protocol process
0	IP	Internet Protocol
3	ARP	Address Resolution Protocol
4	DN	DNA – a subset of Network Control Program
7	IPX	Novell NetWare Internetwork Packet Exchange
8	OSI	ISO Open Systems Interconnect Connectionless Network Layer Protocol / ESIS / ISIS
9	DVM	Distance Vector Multicast Routing Protocol
10	BGP	Border Gateway Protocol
11	SNMP	Simple Network Management Protocol
12	OSPF	Open Shortest Path First
14	APL	AppleTalk Phase 1
20	SDLC	SDLC Relay
22	AP2	AppleTalk Phase 2
23	ASRT	Adaptive Source Routing Transparent Bridge
24	HST	TCP/IP Host Services

2

Configuring AppleTalk Phase 1

This chapter describes the AppleTalk configuration commands.

For more information about AppleTalk, refer to the *Routing Protocols Reference Guide.*

AppleTalk Phase 1 and AppleTalk Phase 2

Your router provides separate packet forwarders to support both AppleTalk Phase 1 (which is described in this chapter and is also referred to as AppleTalk or APL) and its enhancement, AppleTalk Phase 2 (AP2). The difference between Phase 1 and Phase 2 is that Phase 2 removes the Phase 1 restriction of a maximum number of 254 concurrently active AppleTalk devices on one network. You can now assign more than one network number to a single AppleTalk network. The size of the range of network numbers assigned to a network determines the maximum number of concurrently active AppleTalk devices that the network can support (253 devices per network number).

To allow Phase 1 hosts to transparently communicate with Phase 2 hosts, you must enter the AppleTalk Phase 2 configuration process on the router running AP2 and then enable the AppleTalk Phase 1/Phase 2 translation process through that router's AP2 **enable translation** configuration command. For more information about the **enable translation** command and AppleTalk Phase 2, refer to Chapter 4.

In addition to providing the Phase 1/Phase 2 translation process function, this router now acts as both a Phase 1 and Phase 2 router on whatever interfaces these protocols are configured. Routing information is passed between Phase 1 and Phase 2 networks through the translation process resulting in a (logically) single internet.

Accessing the AppleTalk Phase 1 Configuration Environment

For information about accessing the AppleTalk Phase 1 configuration environment, see the chapter in this guide titled “Getting Started.”

Basic Configuration Procedures

This section outlines the initial steps required to get the AppleTalk Phase 1 protocol up and running. Information about how to make further configuration changes is covered in the command sections of this chapter. For the new configuration changes to take effect, you must restart the router.

Enabling Router Parameters

When you configure a router to forward AppleTalk Phase 1 packets, you must enable certain parameters regardless of the number or type of interfaces in the router. If you have multiple routers transferring AppleTalk Phase 1 packets, specify these parameters for each router.

- **Globally Enable AppleTalk Phase 1** – To begin, you must globally enable the AppleTalk Phase 1 software using the AppleTalk Phase 1 **enable apl** configuration command. If the router displays an error in this step, there is no AppleTalk Phase 1 software present in your load. Contact your customer service representative.

- **Enable Specific Interfaces** – You must then enable the specific interfaces over which AppleTalk Phase 1 is to send and receive packets. Use the **enable interface** *interface number* command to do this.

Setting Network Parameters

For each network and interface that sends and receives AppleTalk Phase 1 packets, you must specify certain parameters.

After specifying the parameters, use the AppleTalk Phase 1 **list** configuration command to view the results of the configuration.

- **Set the Network Numbers** – AppleTalk Phase 1 network numbers are 16-bit integers, specified in decimal, in the range of 1 to 65535. (Network 0 is illegal.) Each physical network must have a unique network number. Although you may have multiple routers with interfaces on one network, you need only configure the network number on one router. The configured router, called the seed router, dynamically sends the network number to the connected routers through the RTMP routing protocol. If you do not configure the network number on a router, the router can learn it from other seed routers on the network.

  To connect the interface numbers to the network numbers, use the **set net-number** *interface-number APL-network-number* command while in the AppleTalk Phase 1 configuration process.

- **Set the Node Addresses** – AppleTalk Phase 1 node numbers are 1-byte integers, specified in decimal, in the range of 1 to 254. You can configure a node address for each interface on a network that sends and receives AppleTalk Phase 1 packets. This is the node address used for the Link Access Protocol, such as the Ethernet Link Access Protocol. If you do not configure a node address, it is selected automatically. However, for management purposes, it is desirable to use a fixed node address for a router.

  To set the node addresses, use the AppleTalk Phase 1 **set node-number interface-number** *number* configuration command.

- **Set a Zone Name** – You can configure a zone name for each network in the internetwork. In addition, you can configure the zone name for a given network in any router connected to that network. Only the seed router needs to contain the zone name information for a connected network. Attached routers dynamically acquire the zone name from adjacent routers using the ZIP protocol. Apple recommends that for a given network, you choose the same seed router for the network number and the zone name.

  To set the zone name for each network number, use the AppleTalk Phase 1 **set zone interface-number** *node-name* configuration command.

- **Set the Routing Table Size** – The size of the routing table is configurable. The routing table limits the size by setting the amount of available memory. If there are more than 194 active networks, the router sends the Routing Table Maintenance Protocol (RTMP) data as multiple packets.

To set the routing table size, use the AppleTalk Phase 1 **set nnets** *number* configuration command.

- **Set the Network Packet Header Size** – You can specify whether the router uses short or long header DDP packets on each network in certain situations. For example, you can configure the router to generate short headers for packets destined for a host on a directly connected network. If not configured, the router defaults to long DDP headers, as recommended by Apple.

  To set the network header packet size, use the AppleTalk Phase 1 **set ddp-header** *name interface-number* configuration command.

Configuration Restrictions

There are no configuration restrictions on Phase 1 networks. All network numbers, however, must be either Phase 1 only or Phase 2 only. A physical network can contain both Phase 1 and Phase 2 hosts as long as the router is configured with different network numbers.

With Phase 2 networks, when Phase1/Phase 2 translation is enabled, a network can belong to only one zone. The reason for this restriction is that the Phase 1 Zone Information Protocol (ZIP) maps a network number into a single zone. Without the single zone restriction, a Phase 1 ZIP query cannot be properly processed.

AppleTalk Phase 1 Configuration Commands

This section summarizes and then explains all the AppleTalk Phase 1 configuration commands.

The AppleTalk Phase 1 configuration commands allow you to specify network parameters for router interfaces that transmit AppleTalk Phase 1 packets. The information you specify with the configuration commands activates when you restart the router.

Enter the AppleTalk Phase 1 configuration commands at the `APL config>` prompt.

Table 2–1 AppleTalk Phase 1 Configuration Commands Summary

Command	Function
? (Help)	Lists the AppleTalk Phase 1 configuration commands or lists the options associated with specific commands.
Disable	Disables checksum or takedown, disables a specified interface, or globally disables AppleTalk Phase 1.
Enable	Enables checksum or takedown, enables a specified interface, or globally enables AppleTalk Phase 1.
List	Displays the current AppleTalk Phase 1 configuration.
Set	Sets the DDP header, network number, node number, size of the routing table, and zone parameters.
Exit	Exits the AppleTalk Phase 1 configuration process and returns to the CONFIG environment.

? (Help)

List the commands that are available from the current prompt level. You can also enter a **?** after a specific command name to list its options.

Syntax: ?

Example: **?**

Disable

Disable the checksum generation or takedown, to disable a specified interface, or to globally disable the AppleTalk Phase 1 protocol.

Syntax: disable apl
checksum
interface . . .
takedown

apl

Disables the AppleTalk Phase 1 packet forwarder as a whole.

Example: **disable apl**

checksum

Specifies that the router does not compute the checksum in the packets it generates. This is the default. The router verifies checksums on all packets that it forwards.

Example: **disable checksum**

interface ***interface#***

Disables all AppleTalk Phase 1 functions on the specified interface.

Example: **disable interface 3**

takedown

Prevents ZIP takedown and bringup packets from affecting the routers network numbers and zone names. This is the default for security reasons.

Example: **disable takedown**

Enable

Enable the checksum generation or takedown, to enable a specified interface, or to globally enable the AppleTalk Phase 1 protocol.

Syntax: enable apl
checksum
interface . . .
takedown

apl

Allows the router to send AppleTalk Phase 1 packets over all of the interfaces.

Example: **enable apl**

checksum

Specifies that the router does not compute the checksum in the packets it generates. The router verifies checksums on all packets that it forwards.

Example: **enable checksum**

interface *interface#*

Allows the router to send and receive AppleTalk Phase 1 packets over the specified interface.

Example: **enable interface 3**

takedown

Allows any node on the AppleTalk Phase 1 internetwork to use ZIP takedown and bringup packets to change network numbers and zone names for the router.

Example: **enable takedown**

List

Display the current AppleTalk Phase 1 configuration. In the example, the router is a seed router on networks 13 and 22. It chooses a node number dynamically on net 22 and on interface 2. It also learns the network number and zone name from a seed router on interface 2.

Syntax: list

Example: **list**

```
APL globally  enabled
Checksumming  disabled
Takedown      disabled
Table size    32


List of configured interfaces:

Interface         DFLT DDP hdr      APL address    Zone
   0               long            13/4          ″Jupiter″
   1               short           22/0          ″Neptune″
   2               long             0/0           ″″
```

APL globally	Indicates whether AppleTalk Phase 1 is globally enabled or disabled.
Checksumming	Indicates whether checksum is enabled or disabled.
Takedown	Indicates whether takedown is enabled or disabled.

Table size	Indicates the size of the table.
List of configured interfaces	Lists each interface number and its associated DDP header, APL address, and zone name. The zone name is enclosed in double quotes in case there are imbedded spaces or non-printing characters.

Set

Define specific AppleTalk Phase 1 parameters, including the DDP header, network number, node number, size of the routing table, and zone.

Syntax: set ddp-header long . . .
ddp-header short . . .
net-number . . .
node-number . . .
nnets . . .
zone . . .

ddp-header long *interface#*

Specifies long DDP headers for packets sent on that interface number. This is the default and is recommended by Apple.

Example: **set ddp-header long 2**

ddp-header short *interface#*

Specifies short DDP headers for packets sent on that interface number. Use this only for compatibility with software that does not support long DDP headers.

Example: **set ddp-header short 2**

net-number *interface# AppleTalk Phase 1-net#*

Assigns an AppleTalk Phase 1 network number to the associated directly-connected network. This router is a seed for the network number. If it is not set, it is learned from a seed router. Setting the network number to 0, restores it to the unseeded state.

Example: **set net-number 1 33**

node-number ***interface# node#***

Specifies the number of the interface. This is optional. The default is auto-configure. Setting the node number to 0 restores auto-configuration.

Example: **`set node-number 0 12`**

nnets*#*

Specifies the size of the AppleTalk Phase 1 routing table. This reflects the number of networks in the internet that are running AppleTalk Phase 1. There is no maximum size limit; however, the router can run out of memory.

Example: **`set nnets 4`**

zone ***interface# name***

Specifies the zone name to be seeded on this network. Setting the zone name to an empty string restores auto-configuration.

Example: **`set zone 1 jupiter`**

Exit

Return to the previous prompt level.

Syntax: exit

Example: **`exit`**

3

Monitoring AppleTalk Phase 1

This chapter describes the AppleTalk Phase 1 console commands.

For more information about AppleTalk Phase 1, refer to the *Routing Protocols Reference Guide.*

Accessing the AppleTalk Phase 1 Console Environment

For information about accessing the AppleTalk Phase 1 console environment, see Chapter 1.

AppleTalk Phase 1 Commands

This section summarizes and then explains the AppleTalk Phase 1 console commands. The AppleTalk Phase 1 console commands allow you to view the parameters and statistics of the interfaces and networks that transmit AppleTalk Phase 1 packets. Console commands display configuration values for the physical, frame, and packet levels. You also have the option of viewing the values for all three protocol levels at once.

Enter the AppleTalk Phase 1 console commands at the APL> prompt.

Table 3–1 AppleTalk Phase 1 Console Command Summary

Command	Function
? (Help)	Lists all the AppleTalk Phase 1 console commands or lists the options associated with specific commands.
Counters	Displays the number of packet overflows.
Dump	Displays the contents of the routing table.
Exit	Exits the AppleTalk Phase 1 console process and returns to the GWCON environment.
Interface	Lists the addresses of all the interfaces in the router.

? (Help)

List the commands that are available from the current prompt level. You can also enter a **?** after a specific command name to list its options.

Syntax: ?

Example: **?**

```
COUNTERS
DUMP ROUTING TABLES
EXIT
INTERFACE ADDRESS
```

Counters

Display the number of packet overflows on each network that sends and receives AppleTalk Phase 1 packets. This command displays the number of times the AppleTalk Phase 1 forwarder input queue was full when packets were received from the specified network.

Syntax: counters

Example: **counters**

```
APL input packet overflows
    Net   Count
   Eth/0  4
   TKR/0  0
```

Dump

Obtain routing table information about the interfaces on the router that forwards AppleTalk Phase 1 packets.

Syntax: dump

Example: **dump**

```
Dest Net   Cost       State   Next hop       Source   Zone
    3        0         Dir       3/0         APL     ″Blue″
   72        0         Dir       3/0         AP2     ″Green″
   13        1        Good       3/13        AP2     ″Fuchsia″
   63        2        Good       3/13        APL      NIL
   42        3        Suspct     3/13        APL     ″Orange″
5 entries used out of 32
```

Dest Net Specifies the destination network number in decimal.

Cost Specifies the number of router hops to this destination network.

State Specifies the state of the entry in the routing table. It includes the following:

- **Dir** – Indicates that the router is directly connected to the routing table, the interface is enabled and up, and the network number is known.
- **Good** – Indicates that an RTMP packet containing a good tuple for this network was heard in the last 20 seconds.
- **Suspct** – Indicates that no RTMP tuple was received for this network in the last 20 seconds.
- **Bad** – Indicates that no RTMP tuple was received for this network in the last 40 seconds. RTMP packets with tuples listing this network as unreachable are sent for 20 seconds, then the network is deleted from the RTMP routing table. (For more information, refer to the RTMP chapter in *Inside AppleTalk* by Gursharan S. Sidhu, First Edition.)

Next hop Specifies the next hop for packets going to networks that are not directly connected. For directly-connected networks, the node number is 0.

Source Specifies the originating router type for that routing table entry. APL indicates an AppleTalk Phase 1 router. AP2 indicates AppleTalk Phase 2.

Note: If the Phase 1/Phase 2 translation gateway process was not enabled, you do not see the source column.

Zone Specifies the human-understandable name for that network. The zone name is enclosed in double quotes in case there are embedded spaces or non-printing characters. If the zone name contains characters beyond the 7-bit ASCII character set (they are 8-bit), the zone name that displays depends on the characteristics of your console terminal. If there is no zone name known for this network, the name NIL (without quotes) is displayed.

Note: At the bottom of the display is the number of entries used and the total available. If all the entries are used, it is likely that the routing table is not large enough. Use the AppleTalk Phase 1 **set nnets** configuration command to increase the size.

Exit

Return to the previous prompt level.

Syntax: exit

Example: **exit**

Interface

Display the addresses of all the interfaces in the router on which AppleTalk Phase 1 is enabled. If the interface is present in the router but it is disabled, this command shows that status.

Syntax: interface

Example: **interface**

```
Interface      Addresses
  Eth/0        3/29
  TKR/0        APL not enabled
```

4

Configuring AppleTalk Phase 2

This chapter describes the AppleTalk Phase 2 (AP2) configuration commands.

For more information about AppleTalk Phase 2, refer to the *Routing Protocols Reference Guide.*

AppleTalk Phase 1 and AppleTalk Phase 2

Your router provides separate packet forwarders to support both AppleTalk Phase 1 (APL) and its enhancement, AppleTalk Phase 2 (AP2). The difference between Phase 1 and Phase 2 is that Phase 2 removes the Phase 1 restriction of a maximum number of 254 concurrently active AppleTalk devices on one network. You can now assign more than one network number to a single AppleTalk network. The size of the range of network numbers assigned to a network determines the maximum number of concurrently active AppleTalk devices that can be supported on that network (253 devices per network number).

To allow Phase 1 hosts to transparently communicate with Phase 2 hosts, you must enter the AppleTalk Phase 2 configuration process on the router running AP2 and enable the AppleTalk Phase 1/2 translation process through that router's AP2 **enable translation** configuration command. For more information about the **enable translation** command, see the command section of this chapter.

In addition to providing the Phase 1/Phase 2 translation process function, this router now acts as both a Phase 1 and Phase 2 router on whatever interfaces these protocols are configured. Routing information is passed between Phase 1 and Phase 2 networks through the translation process resulting in a (logically) single internet.

Accessing the AppleTalk Phase 2 Configuration Environment

For information about accessing the AppleTalk Phase 2 configuration environment, see Chapter 1.

Basic Configuration Procedures

This section outlines the initial steps required to get the AppleTalk Phase 2 protocol up and running. Information about how to make further configuration changes is covered in the command sections of this chapter. For the new configuration changes to take effect, the router must be restarted.

Enabling Router Parameters

When you configure a router to forward AppleTalk Phase 2 packets, you must enable certain parameters regardless of the number or type of interfaces in the router. If you have multiple routers transferring AppleTalk Phase 2 packets, specify these parameters for each router.

- **Globally Enable AppleTalk Phase 2** – To begin, you must globally enable the AppleTalk Phase 2 software using the AppleTalk Phase 2 configuration **enable ap2** command. If the router displays an error in this step, there is no AppleTalk Phase 2 software present in your load. If this is the case, contact your customer service representative.

- **Enable Specific Interfaces** – You must then enable the specific interfaces over which AppleTalk Phase 2 is to send the packets. Use the **enable interface** *interface-number* command to do this.

- **Enable Checksumming** – You can then determine whether the router computes DDP checksums of the packets it originates. Checksum software does not work correctly in some AppleTalk Phase 2 implementations, so you may not want to originate packets with checksums for compatibility with these implementations. Any packet forwarded with a checksum has its checksum verified.

- **Enable Phase 1/Phase 2 translation (optional)** – Use the **enable translation** command to allow Phase 2 hosts to transparently communicate with Phase 1 hosts.

Setting Network Parameters

You must also specify certain parameters for each network and interface that sends and receives AppleTalk Phase 2 packets. After you specify the parameters, use the AppleTalk Phase 2 **list** configuration command to view the results of the configuration.

- **Set the Network Range for Seed Routers** – Coordinating network ranges and zone lists for all routers on a network is simplified by having specific routers designated as seed routers. Seed routers are configured with the network range and zone list while all other routers are given null values. Null values indicate that the router queries the network for values from the seed routers. There are usually several seed routers on a network in case one of them fails. Also, a router can be a seed router for some or all of its network interfaces. Use the **set net-range** command to assign the network range in seed routers.

- **Set the Starting Node Number** – Use the **set node-number** command to assign the starting node number for the router. The router AARPs for this node, but if it is already in use, a new **node-number** is chosen.

- **Add a Zone Name** – You can add one or more zone names for each network in the internetwork. You can add a zone name for a given network in any router connected to that network; however, only the seed router needs to contain the zone name information for a connected network. Attached routers dynamically acquire the zone names from adjacent routers using the ZIP protocol. Apple recommends that for a given network, you choose the same seed router for the network range and the zone names. The zone names cannot be configured for a network unless the network range is also configured. To add a zone name to an interface, use the AppleTalk Phase 2 configuration **add zone** *name* command.

Setting Up Zone Filters

Zone filtering lets you filter zones in each direction on each interface. To filter incoming packets, set up an input filter. To filter outgoing packets, set up an output filter. The interface does not readvertise filtered zone information in the direction that you define. To set up a zone filter:

1. Add zone filters to an interface.

To add an input zone filter, use the **add zfilter in** command. To add an output zone filter, use the **add zfilter out** command. The software prompts you for the interface number and the name of the zone that you want to filter.

```
AP2 config>add zfilter in
Interface # [0]? 1
Zone name []? Admin
```

2. Enable the zone filters that you added.

   To enable an input zone filter, enter **enable zfilter in**. To enable an output zone filter, enter **enable zfilter out**. The software prompts you for the interface number and for whether or not the filter is inclusive or exclusive. Inclusive filters forward only the zone information in a filter. Exclusive filters block only the zone information in a filter.

   ```
   AP2 config>enable zfilter in
   Interface # [0]? 1
   INCLUSIVE/EXCLUSIVE [INCLUSIVE]? exc
   ```

Setting Up Network Filters

Network filters are similar to zone filters, except they let you filter an entire network. To set up a network filter, follow these steps:

1. Add a network filter. Use the **add nfilter in** command to add an input network filter to an interface. Use the **add nfilter out** command to add an output network filter to an interface. For example:

   ```
   AP2 config>add nfilter out
   Interface # [0]? 0
   First Network range number (decimal)[0]? 11
   Last Network range number (decimal) [0]? 15
   ```

   The network range you enter here must match the range that you assigned to that network.

2. Enable the network filter that you added and make it either inclusive or exclusive. Inclusive filters forward only network information in a filter. Exclusive filters block only network information in a filter, and they allow all other network information to be forwarded.

   ```
   AP2 config>enable nfilter in
   Interface # [0]? 0
   INCLUSIVE/EXCLUSIVE [INCLUSIVE]? exc
   ```

Disabling Checksumming

As the default, the router computes DDP checksums of packets it originates. Checksum software does not work correctly in some AppleTalk implementations, so you may not want to originate packets with checksums. In this case, disable checksum using the **disable checksum** command.

Configuration Restrictions

There are no configuration restrictions on Phase 2 networks. All network numbers must be either Phase 1 only or Phase 2 only. A physical network can contain both Phase 1 and Phase 2 hosts as long as the router is configured with different network numbers.

With Phase 1 networks, a network can belong to only one zone. The reason for this restriction is that the Phase 1 Zone Information Protocol (ZIP) maps a network number into a single zone. Without the single zone restriction, a Phase 1 ZIP query cannot be properly processed.

AppleTalk Phase 2 Configuration Commands

This section explains the AppleTalk Phase 2 configuration commands.

The AppleTalk Phase 2 configuration commands allow you to specify network parameters for router interfaces that transmit AppleTalk Phase 2 packets. The information you specify with the configuration commands is activated when you restart the router.

Enter the AppleTalk Phase 2 configuration commands at the `AP2 config>` prompt.

Table 4–1 AppleTalk Phase 2 Configuration Commands Summary

Command	Function
? (Help)	Lists the AppleTalk Phase 2 configuration commands or lists the options associated with specific commands.
Add	Adds the interface default zone name to the interface zone list, an IP tunnel address, a network filter or a zone filter, or a zone name to the interface zone list.
Delete	Deletes an interface definition, an IP tunnel address, a network filter or a zone filter, or a zone name from the interface zone list.
Disable	Disables AppleTalk Phase 2 globally, checksum generation, a specified interface, an IP tunnel, Phase 1/2 translation, or a network filter or zone filter.
Enable	Enables AppleTalk Phase 2 globally, checksum generation, a specified interface, an IP tunnel, Phase 1/2 translation, or a network filter or a zone filter.
List	Displays the current AppleTalk Phase 2 configuration.
Set	Sets the net range or node number for an interface.
Exit	Exits the AppleTalk Phase 2 configuration process and returns to the CONFIG environment.

? (Help)

List the commands that are available from the current prompt level. You can also enter a **?** after a specific command name to list its options.

Syntax: ?

Example: **?**

Add

Add a zone name to the interface zone list as the default for the interface, an IP tunnel address, a network filter or zone filter, or a zone name to the interface zone list.

Syntax: add defaultzone . . .
ip-tunnel-address . . .
nfilter . . .
zfilter . . .
zone . . .

defaultzone *interface# zonename*

Adds the zone name to the interface zone list as the default for the interface. If two defaults are defined, the last one overrides the first. If no default is defined, the first zone name is the default.

Example: **add defaultzone 2 newyork**

ip-tunnel-address *address* DEC

Adds an IP tunnel endpoint. The value *address* is an 32-bit IP address is in the form *n.n.n.n*, where each *n* is a decimal integer representing one octet of the address. DEC is the encapsulation type and is currently the only one supported.

Example: **add ip-tunnel-address 1.2.3.4 DEC**

nfilter <in or out> *interface# first-range# last-range#*

Adds a network filter to let you filter an entire network. The arguments **in** and **out** let you define the direction of flow on which the filter applies. The values of *first-range#* and *last-range#* represent the network numbers that defined the network range when the network was created.

Example: **add nfilter in**

```
Interface # [0]? 1
First Network range number (decimal)[0]? 11
Last Network range number (decimal) [0]? 15
```

zfilter <in or out> *interface# zonename*

Adds a zone filter to let you filter zones on input or output. The arguments **in** and **out** let you define the direction of flow on which the filter applies. The values of *interface#* and *zonename* represent the interface and the name of the zone that you are filtering.

```
Example: add zfilter in

interface # [0]? 1
Zone name []? Admin
```

zone *interface# zonename*

Adds the zone name to the interface zone list. If no default zone name is defined, the first zone name for the interface is the default. If there are no zone names for an interface, the default is "*".

```
Example: add zone 2 newyork
```

Delete

Delete an interface definition, an IP tunnel address, a network filter or a zone filter, or a zone name from the interface zone list.

Syntax: delete interface . . .
ip-tunnel-address . . .
nfilter . . .
zfilter . . .
zone . . .

interface *interface#*

Deletes all AP2 information for the specified interface. Sometimes this is the only way to delete zone names that have non-printing characters.

```
Example: delete interface 2
```

ip-tunnel-address *address*

Deletes the IP tunnel endpoint whose address is *address*. *address* is an 32–bit IP address in the form n.n.n.n, where each n is a decimal integer representing one octet of the address.

```
Example: delete ip-tunnel-address 1.2.3.4
```

nfilter <in or out> ***interface# first-network# last-network#***

Deletes a network filter from the input or output of the interface. Enter the same network range you set using the **add nfilter in** command.

Example: **delete nfilter in**

```
Interface # [0]? 0
First Network range number (decimal) [0]? 1
Last Network range number (decimal) [0]? 12
```

zfilter <in or out> ***interface# zonename***

Deletes a zone name filter from the input or the output of the interface.

Example: **delete zfilter out**

```
Interface # [0]? 1
Zone name []? Marketing
```

zone ***interface# zonename***

Deletes the specified zone name from the interface zone list.

Example: **delete zone 2 newyork**

Disable

Disable AppleTalk Phase 2 globally, checksumming, a specified interface, an IP tunnel, AppleTalk Phase 1/2 gateway functionality, network filters, or zone filters.

Syntax: disable ap2
checksum
interface . . .
ip-tunnel
nfilter . . .
translation
zfilter . . .

ap2

Disables the AppleTalk Phase 2 packet forwarder for all interfaces.

Example: **disable ap2**

checksum

Specifies that the router does not compute the checksum in packets that it generates. This is the default. The router verifies checksums on all packets that it forwards.

Example: **disable checksum**

interface *interface#*

Disables all AP2 functions on the specified network interface. The network continues to remain available for all other protocols.

Example: **disable interface**

```
Interface #  [0]? 2
```

ip-tunnel

Globally disables the IP tunnel.

Example: **disable IP-tunnel**

nfilter <in or out> *interface#*

Disables, but does not delete, the input or output network filters on this interface.

Example: **disable nfilter in**

```
Interface # [0]? 2
```

translation

Disables the translation process that allows Phase 2 hosts to transparently communicate with Phase 1 hosts.

Example: **disable translation**

zfilter <in or out> *interface#*

Disables, but does not delete, the input or output zone filters on this interface.

Example: **disable zfilter out**

```
Interface # [0]? 1
```

Enable

Enable the AppleTalk Phase 2 protocol globally, the checksum function, a specified interface, an IP tunnel, the AppleTalk Phase 1/2 gateway functionality, a network filter, or a zone filter.

Syntax: enable ap2
checksum
interface . . .
ip-tunnel
nfilter . . .
translation
zfilter . . .

ap2

Enables the AppleTalk Phase 2 packet forwarder over all of the interfaces.

Example: **enable ap2**

checksum

Specifies that the router computes the checksum in packets that it generates. The router checksums all AP2 packets that it forwards.

Example: **enable checksum**

interface *interface#*

Enables the router to send and receive AppleTalk Phase 2 packets over the specified interface.

Example: **enable interface**

```
Interface #  [0]? 1
```

ip-tunnel

Globally enables the IP tunnel.

Example: **enable ip-tunnel**

nfilter <in or out> ***interface#***

Enables network input or output filters and controls how the filter is applied to the interface. Inclusive forwards matches. Exclusive drops matches.

Example: **enable nfilter in**

```
Interface # [0]? 1
INCLUSIVE/EXCLUSIVE [INCLUSIVE]? inc
```

translation

Enables the translation process that allows Phase 2 hosts to transparently communicate with Phase 1 hosts. Besides providing the translation function, this router now acts as both a Phase 1 and Phase 2 router on whatever interfaces these protocols are configured. Routing information passes between Phase 1 and Phase 2 networks through the translation process, resulting in a (logically) single internet.

Example: **enable translation**

zfilter <in or out> ***interface#***

Enables and controls how the zone input or output filter is applied to the interface. Inclusive forwards matches. Exclusive drops matches.

Example: **enable zfilter out**

```
Interface # [0]? 0
INCLUSIVE/EXCLUSIVE [INCLUSIVE]? inc
```

List

Display the current AppleTalk Phase 2 configuration. In the example, the router is a seed router on networks 13 and 22. It chooses a node number dynamically on net 22, and on interface 2. It also learns the network number and zone name from a seed router on interface 2.

Syntax: list

Example: **list**

```
AP2 globally        enabled
AP2 IP tunnel       enabled
Checksumming        disabled
Translation Gateway enabled

List of configured interfaces:
  Interface      netrange / node   Zones
       0       1000–1000  / 1    ″SerialLine″(Def)
 Input ZFilters disabled
 Input NFilters disabled
 Output ZFilters disabled
 Output NFilters disabled

       1         10–19    / 52   ″Sales″(Def),″EtherTalk″
 Input ZFilters disabled
 Input NFilters disabled
 Output ZFilters disabled
 Output NFilters disabled

  IP Tunnel Endpoint  Encapsulation
     1.2.3.4          DEC
```

AP2 globally	Indicates whether AppleTalk Phase 2 is globally enabled or disabled.
AP2 IP tunnel	Indicates whether the AppleTalk Phase 2 IP tunnel is globally enabled or disabled.
Checksumming	Indicates whether checksum is enabled or disabled.
Table size	Indicates the size of the table.
Translation Gateway	Indicates whether the AppleTalk Phase 1/Phase 2 translation is globally enabled or disabled.
List of configured interfaces	Lists each interface number and its associated net range, node number, zones (including the default zone), and input and output network filters and zone filters. The zone name is enclosed in double quotes in case there are imbedded spaces or non-printing characters.
Input/Output Zfilters	Indicates whether Input or Output Zfilters are enabled or disabled for each interface.

Input/Output Nfilters	Indicates whether Input or Output Nfilters are enabled or disabled for each interface.
IP Tunnel Endpoint	Indicates the IP address of the endpoint.
Encapsulation	Indicates the type of IP encapsulation. DEC is the only encapsulation currently supported.

Set

Define the network range or node number for an interface.

Syntax: set net-range . . .
node . . .

netrange *interface# start# end#*

Assigns the network range in seed routers using the following:

- **interface#** – Designates the router interface to operate on.
- **start#** – Assigns the lowest number of the network range. Legal values are 1 to 65279 (FEFF hexadecimal).
- **end#** – Sets the highest number of the network range. Legal values are *start#* to 65279.

A single numbered network has the same start and end values. A start value of zero deletes the netrange for the interface and turns the seeded interface into an unseeded interface. *Start#* and *end#* are inclusive in the network range.

Example: **set net-range 2 43 45**

node *interface# node#*

Assigns the starting node number for the router interface. The router will AARP for this node but if it is already in use, a new node-number is chosen. The following explains each argument that is entered after this command:

- **interface#** – Designates the router interface to operate on.

- **node#** – Designates the first attempted node number. Legal values are 1 to 253. A node# value of zero deletes the node number for the interface and force the router to choose one at random.

Example: **set node 2 2**

Exit

Return to the previous prompt level.

Syntax: exit

Example: **exit**

5

Monitoring AppleTalk Phase 2

This chapter describes the AppleTalk Phase 2 (AP2) console commands.

For more information about AppleTalk Phase 2, refer to the *Routing Protocols Reference Guide.*

Accessing the AppleTalk Phase 2 Console Environment

For information about accessing the AppleTalk Phase 2 console environment, see Chapter 1.

AppleTalk Phase 2 Commands

This section summarizes and then explains the AppleTalk Phase 2 console commands. The AppleTalk Phase 2 console commands allow you to view the parameters and statistics of the interfaces and networks that transmit AppleTalk Phase 2 packets.

Enter the AppleTalk Phase 2 console commands at the `AP2>` prompt.

Table 5–1 AppleTalk Phase 2 Console Command Summary

Command	Function
? (Help)	Lists all the AppleTalk Phase 2 console commands or lists the options associated with specific commands.
Counters	Displays the input overflow count of AP2 packets for each interface.
Dump	Displays the current state of the routing table for all networks in the internet and their associated zone names.
Interface	Displays the current addresses of the interfaces.
Exit	Exits the AppleTalk Phase 2 console process and returns to the GWCON environment.

? (Help)

List the commands that are available from the current prompt level. You can also enter a **?** after a specific command name to list its options.

Syntax: ?

Example: **?**

```
COUNTERS
DUMP ROUTING TABLES
INTERFACE ADDRESSES
EXIT
```

Counters

Display the number of input packet overflows on each network that sends and receives AppleTalk Phase 2 packets. This command displays the number of times the AppleTalk Phase 2 forwarder input queue was full when packets were received from the specified interface.

Syntax: counters

Example: **counters**

```
AP2 input packet overflows
     Net   Count
    Eth/0  4
    TKR/0  0
```

Dump

Obtain routing table information about the interfaces on the router that forwards AppleTalk Phase 2 packets.

Syntax: dump

Example: **dump**

```
Dest Net   Cost    State   Next hop         ZoneList
  10–19      0      Dir    0/0        “Ethertalk”, “Sales”
  40–49      1     Good    10/13      “Marketing”,“CustomerSer”,
  20–29      2     Sspct   10/13      “Fuchsia”, “Backbone”,
                                      “Engineering”, “MKTING”
3 entries
```

Dest Net	Specifies the destination network number range, in decimal.
Cost	Specifies the number of router hops to this destination network.
State	Specifies the state of the entry in the routing table. It includes the following: • **Dir** – Indicates that the router is directly connected to the destination network, the interface is enabled and up, and the network number is known. • **Good** – Indicates that an RTMP packet containing a good tuple for this network was heard in the last 20 seconds. • **Suspct** – Indicates that no RTMP tuple was received for this network in the last 20 seconds.

- **Bad** – Indicates that no RTMP tuple was received for this network in the last 40 seconds. RTMP packets with tuples listing this network as unreachable are sent for 20 seconds, then the network is deleted from the RTMP routing table.

Next hop Specifies the next hop for packets going to networks that are not directly connected. For directly-connected networks, this field is blank.

Zone(s) Specifies the zone list for that network. The zone names are enclosed in quotation marks in case there are embedded spaces or non-printing characters. If a zone name contains characters beyond the 7-bit ASCII character set (they are 8-bit), the zone name that displays depends on the characteristics of your console terminal.

Note: If the Phase 1/Phase 2 translation gateway process was enabled you are also shown another column (between next hop and zone). This column is labelled **source** and lists the originating network type for that table entry.

Interface

Display the addresses of all the interfaces in the router on which AppleTalk Phase 2 is enabled. If the interface is present in the router but is disabled, this command shows that status.

Syntax: interface

Example: **interface**

```
Interface      Addresses
  SL/0         0/1 on net 1000–1000  default zone “SerialLine”
  Eth/0        10/52 on net 10–19    default zone  “Sales”
  SL/1         0/0 in startup range
```

Exit

Return to the previous prompt level.

Syntax: exit

Example: **exit**

6

Configuring ARP

This chapter describes how to configure the Address Resolution Protocol (ARP) and how to use the ARP configuration commands.

For more information about ARP, refer to the *Routing Protocols Reference Guide.*

Accessing the ARP Configuration Environment

You can access the ARP configuration commands by typing **protocol ARP** at the `config>` prompt:

```
config> protocol ARP
```

For more information about accessing the ARP configuration environment, see Chapter 1.

ARP Configuration Commands

This section explains all the ARP configuration commands. Table 6–1 lists the ARP configuration commands.

Table 6–1 ARP Configuration Commands Summary

Command	Function
? (Help*)*	Lists the ARP configuration commands or lists the options associated with specific commands.
Add Entry	Adds a MAC address translation entry.
Change Entry	Changes a MAC address translation entry.
Delete Entry	Deletes a MAC address translation entry.
Disable Auto-refresh	Disables ARP auto-refresh.
Enable Auto-refresh	Enables ARP auto-refresh.
List	Lists ARP configuration data in SRAM.
Set	Sets the usage and refreshes timeout values.
Exit	Exits the ARP configuration process.

? (Help)

List the commands that are available from the current prompt level. You can also enter a **?** after a specific command name to list its options.

Syntax: ?

Example: **?**

Add Entry

Add a MAC address translation entry.

Syntax: add entry *ifc# prot-type prot-addr MAC-addr*

Example: **add entry**

```
Interface Number [0]?
Protocol [IP]?
IP Address [0.0.0.0]?
Mac Address []?
```

Change Entry

Change a MAC address translation entry. The hardware address parameter (*MAC-addr*) is the address of the node being changed.

Syntax: change entry *ifc# prot-type prot-addr MAC-addr*

Example: **change entry**

```
Interface Number [0]?
Protocol [IP]?
IP Address [0.0.0.0]?
Mac Address []?
```

Delete Entry

Delete a MAC address translation entry.

Syntax: delete entry *ifc# prot-type prot-addr*

Example: **delete entry**

```
Interface Number [0]?
Protocol [IP]?
IP Address [0.0.0.0]?
```

Disable Auto-Refresh

Disable the auto-refresh function. The auto-refresh function is the router's capability to send another ARP request based on the entry in the translation cache before the refresh timer expires. The request is sent directly to the hardware address in the current translation instead of a broadcast. If auto-refresh is disabled, no additional ARP request is made, and the refresh timer is allowed to expire.

Syntax: disable auto- refresh

Example: **disable auto-refresh**

Enable Auto-Refresh

Enable the auto-refresh function. The auto-refresh function is the router's capability to send another ARP request based on the entry in the translation cache before the refresh timer expires. The request is sent directly to the hardware address in the current translation instead of a broadcast. If auto-refresh is enabled, an additional ARP request is made in this manner before the refresh timer is allowed to expire.

Syntax: enable auto-refresh

Example: **enable auto-refresh**

List

Display the contents of the router's ARP configuration as stored in SRAM. The **list** command displays the current timeout settings for the refresh and usage timer.

Syntax: list all
config
entry

all

Lists the ARP configuration followed by all of the ARP entries.

Example: **list all**

```
ARP configuration:

Refresh Timeout: 5 minutes
Auto Refresh: disabled

Mac address translation configuration

IF #           Prot #         Protocol --> Mac Address
```

config

Lists the configuration for the different ARP timers.

Example: **list config**

```
ARP configuration:

Refresh Timeout: 5 minutes
Auto refresh: disabled
```

entry

Lists the ARP entries in nonvolatile memory.

Example: **list entry**

```
Mac address translation configuration

IF #           Prot #         Protocol --> Mac Address
```

Set

Set an ARP configuration parameter.

Syntax: set refresh-timer

refresh-timer

Changes the timeout value for the refresh timer. To change the timeout value for the refresh timer, enter the timeout value in minutes. A setting of zero (0) turns off (disables) the refresh timer.

Example: **set refresh-timer 3**

Exit

Return to the previous prompt level.

Syntax: exit

Example: **exit**

7

Monitoring ARP

This chapter describes how to monitor ARP protocol activity and how to use the ARP console commands.

For more information about ARP, refer to the *Routing Protocols Reference Guide*.

Accessing the ARP Console Environment

You can access ARP console commands by entering **protocol arp** at the + prompt. For example:

```
+ protocol arp
```

For more information about accessing the ARP console environment, see Chapter 1.

ARP Console Commands

This section explains the ARP console commands entered from the ARP> prompt.

Table 7–1 ARP Console Command Summary

Command	Function
? (Help)	Lists the ARP console commands or lists the options associated with specific commands.
Clear	Clears the cache for a specified interface.
Dump	Displays the cache for a specified interface.
Hardware	Lists each ARP-configured network.
Protocol	Lists each ARP-configured protocol.
Statistics	Displays ARP information.
Exit	Exits the ARP console process.

? (Help)

List the commands that are available from the current prompt level. You can also enter a **?** after a specific command name to list its options.

Syntax: ?

Example: **?**

Clear

Flush the ARP cache for a given network interface. The **clear** command can be used to force the deletion of bad transactions.

To clear a particular interface, enter the interface or network number as part of the command. To obtain the interface number, use the CONFIG **list devices** command.

Syntax: clear *interface#*

Example: **clear 1**

Dump

Display the ARP cache for a given network/protocol combination. To display the ARP cache for a particular interface, enter the interface or network number as part of the command. To obtain the interface number, use the CONFIG **list devices** command.

If there is more than one protocol on that network, the protocol number must also be given. This causes the console to display the hardware address-to-protocol mappings stored in that database. If ARP is in use by only one protocol on the specified interface, then the protocol number is optional. To obtain the protocol number, use the CONFIG **protocol** command.

Syntax: dump *interface# protocol#*

Example: **dump 2**

```
Hardware Address    IP Address      Refresh
02-07-01-00-00-01   192.9.1.2         5
```

Hardware

Display the networks registered with ARP. The **hardware** command lists each ARP-registered network, and displays each network's hardware address space (Hardware AS) and local hardware address. Hardware addresses are displayed according to hardware type (decimal for token ring, hexadecimal for Ethernet).

Syntax: hardware

Example: **hardware**

```
Network       Hardware AS    Hardware Address

2 Eth/0       1              02-07-01-00-00-01
```

Protocol

Display (by network) the protocols with addresses registered with ARP. This command displays the network, protocol name, protocol number, protocol address space (in hexadecimal), and local protocol addresses.

Syntax: protocol

Example: **protocol**

```
Network  Protocol (num)  AS    Protocol Address(es)
2 Eth/0  IP        (0)   0800  192.9.1.1 18.124.0.11
```

Statistics

Display a variety of statistics about the operation of the ARP module.

Syntax: statistics

Example: **statistics**

```
ARP input packet overflows
   Net     Count
   Eth /0    0
   ARPA/0    0

ARP cache meters
Net Prot Max Cur Cnt Alloc Rfrsh:Tot Fail TMOs:Rfrsh TMOs:Use
  0   4    1   1   1    1          0    0          0        0
  1   0    2   2  12   12          0    0          0        0
  2   4    1   1   1    1          0    0          0        0
```

ARP input packet overflows	Displays counters that represent the number of ARP packets discarded on input because the ARP layer was too busy. The counts shown are per network interface.
ARP cache meters	Consists of a variety of meters on the operation of the ARP cache. The counts shown are all per protocol, per interface.
Net	Displays the interface numbers.
Prot	Displays the protocol numbers.
Max	Displays the all-time maximum length hash chain.
Cur	Displays the current maximum length hash chain.
Cnt	Displays the count of entries currently active.
Alloc	Displays the count of entries created.
Rfrsh:Tot	Displays the number of refresh requests sent for this network interface and protocol.

Fail	Displays the number of auto-refresh attempt failures due to unavailability of internal resources. This count is not related to whether or not an entry was refreshed.
TMOs:Rfrsh	Displays the count of entries deleted due to a timeout of the refresh timer.
TMOs:Use	Displays the count of entries deleted due to a timeout of the usage timer.

Exit

Return to the previous prompt level.

Syntax: exit

Example: **exit**

8

Configuring and Monitoring DNA IV

This chapter describes how to configure and monitor the Digital Network Architecture Phase IV (DNA IV) protocol using the Network Control Program (NCP) and how to use the NCP configuration and console commands.

Note: When operating DNA IV networks together with DNA V networks, all DNA IV monitoring must be done from the process described in this chapter. For more information about DNA IV and DNA V compatibility, refer to the *Routing Protocols Reference Guide.*

NCP is the user interface for this router's implementation of DECnet Phase IV. This NCP supports a limited subset of the DECnet-VAX NCP commands.

For more information about DNA and NCP, refer to the *Routing Protocols Reference Guide.*

Accessing the NCP Environment

Both NCP configuration and console commands can be accessed from either the CONFIG (configuration) or GWCON (console) environments. For information about accessing the NCP environment, see Chapter 1.

NCP Command Syntax

The command syntax has 3 parts: a command, a component, and an argument.

A command indicates the action to perform. A component indicates the subsystem to which the command applies. An argument is either an attribute and its value or a keyword representing a group of attributes.

Table 8–1 shows several commands, their components, and their arguments.

Table 8–1 Example NCP commands

Command	Component	Argument
set	circuit eth/0	router priority 100
define	all circuits	state off
show	executor	characteristics

NCP Configuration and Console Commands

This section explains the NCP configuration and console commands. Enter the commands at the `NCP>` prompt.

Table 8–2 summarizes the NCP commands.

Table 8–2 NCP Configuration and Console Command Summary

Command	Function
? (Help)	Lists all the NCP commands or lists the options associated with a specific command.
Define	Sets or modifies configuration information in the permanent database.
Purge	Removes configuration information from the permanent database.
List	Displays configuration information in the permanent database.
Set	Sets or modifies information in the volatile database.
Show	Displays information in the volatile database.
Zero	Clears counters in the volatile database.
Exit	Exits NCP.

The **define**, **purge**, and **list** commands act on the configuration information stored in the router. This configuration information is referred to as the permanent database, and survives restarts, software loads, and power cycles. The router uses it each time it starts.

The **set**, **show**, and **zero** commands act on the information currently used by the running router. This information is referred to as the volatile database. It is initialized at startup from the permanent database but may change due to console commands or normal operation of the DNA IV protocol. Changes to the volatile database do not remain in effect when the router is restarted.

? (Help)

List the commands that are available from the current prompt level. You can also enter a **?** after a specific command name to list its options.

Syntax: ?

Example: **?**

```
DEFINE
LIST
PURGE
SET
SHOW
ZERO
EXIT
```

Define

Define access control lists and routing filters.

Syntax: define module access-control . . .
module routing-filter . . .

Set/Define

Use the set command to set or modify information in the volatile database. Use the define command to set or modify configuration information in the permanent database. The command syntax for set and define is identical, except as noted below.

Syntax:	set	circuit-specifier . . .
		executor . . .
		node . . .
Syntax:	define	circuit-specifier . . .
		executor . . .
		module . . .
		node . . .

circuit-specifier *argument*

Sets or changes circuit arguments in the volatile database of DNA when the **set** command is used. The **define** command sets or changes circuit arguments in the permanent database. The circuit(s) must be in the off state to modify numeric arguments in the volatile database.

The *circuit-specifier* options include the following:

active circuits	Specifies all circuits who are up and whose state is on (**set** only).
all circuits	Specifies all circuits on the router.
circuit [*name*]	Specifies the named (for example, Eth/0, TKR/0) circuit.
known circuits	Specifies all circuits on the router.

The arguments include the following:

cost [*number*]	Sets the cost to receive a packet on this circuit. This is used by the routing algorithm to determine the cost of a circuit in choosing routes (cost is not the same as an IP metric). Range*:* 1 to 25. Default: 4.

The following values are suggested starting points:

Circuit type	Cost
Ethernet	4
Sync 56 Kb	6
Sync T1	5
X.25	25
FDDI	1

hello timer [*range*]

Specifies how often (in seconds) router hellos are sent on this circuit. Range*:* 1 to 8191 seconds. Default: 15 seconds (recommended).

maximum routers [*range*]

Specifies how many other routers there are on this circuit. Range*:* 1 to 33. Default: 16. Valid only with **define**.

If this is a level 1 router, only routers on this circuit in the same area count. If this is a level 2 router, also count all level 2 routers on this circuit. The local router does not count against the limit.

Do not set this argument to less than the actual number of routers on the circuit. This can result in anomalies in routing.

Note: For a point-to-point (synchronous line) circuit, set this argument to 1. The result is *significant* memory savings on a router with multiple point-to-point lines.

router priority [*number*]

Specifies the router's priority in bidding to become the designated router for the endnodes on this circuit. Range*:* 1 to 127, where 127 is the highest priority. Default: 64.

If two routers have the same priority, the one with the higher node address wins. The router priority has no effect on area routing decisions, or in reaching the closest "attached" level 2 router.

router type standard

Specifies that the router is using conventional phase IV addressing where the MAC address is built from the area and node number. The router defaults to this type.

router type ama

Specifies that the router can route packets that use phase IV addressing where the MAC address is arbitrary and learned from the data link layer. Valid only on token-ring circuits.

router type bilingual	Specifies that the router can route packets that use both conventional and phase IV with AMA addressing. Valid only on token ring circuits.
state on	Specifies that the circuit is enabled for use by DNA.
state off	Specifies that the circuit is disabled for use by DNA. This is the default.

Example: **set circuit eth/0 cost 4**

Example: **define circuit eth/0 cost 4**

executor *argument*

Sets or modifies arguments global to DNA in the volatile database when the **set** command is used. The **define** command does the same in the permanent database.

The executor must be in the off state to modify numeric arguments or the type in the volatile database.

address [*area.node*]	The DNA IV address of this router. Area range: 1 to 63. Node range: 1 to 1023. Area must not be greater than executor maximum area, and node must not be greater than executor maximum address. The default 0.0 is illegal. **Note:** DNA does not go **on** if the executor address is not set to a legal value.
area maximum cost [*number*]	Maximum cost that is allowed between this level 2 router and any other area. If the best route to an area is more expensive than this, that area is considered unreachable. Maximum: 1022. Default: 1022. This argument does not apply to level 1 routers. Make this value greater than the maximum legal cost to the most distant area. A suggested value is 25 times **area maximum hops.**

area maximum hops [*number*]

Maximum number of hops allowed between this level 2 router and any other area. If the best route to an area requires more than the maximum number of hops, that area is considered unreachable. Maximum: 30. Default: 30. This argument does not apply to level 1 routers. Make this value about twice the longest path length (in hops) that is expected.

The hop count is used by routing only to speed the decay of routes to unreachable areas. This argument may be reduced to cause unreachable areas to become unreachable more quickly.

broadcast routing timer [*number*]

Specifies how often level 1 (and 2 in a level 2 router) routing messages are sent (in seconds). This is how often they are sent in the absence of any cost or adjacency changes. This protects the routing database from corruption. Routing updates are sent immediately if any cost or adjacency changes. Range: 1 to 65535. Default: 180. Lower values increase the overhead for this and all adjacent routers. Larger values increase the time required to correct the routing database if a routing update message is lost.

maximum address [*number*]

The highest node address within the area to which routes are kept. The routing database does not include routes to any nodes in the area with a higher node address. Range: 1 to 1023. Default 32. Valid only with **define**.

Make this value greater than or equal to the highest node address in the router's area.

maximum area [*number*]	The highest area to which routes are kept, if this is a level 2 router. The routing database does not include routes to any higher-numbered areas. Range: 1 to 63. Default: 63. Valid only with **define**. Make this value greater than or equal to the highest area number in the network.
maximum broadcast nonrouters [*number*]	Maximum number of endnodes that can be adjacent to (one hop away from) this router on broadcast circuits. Range: 1 to 1023. Default: 64. Valid only with **define**. Make this value greater than or equal to the total number of endnodes on all broadcast circuits. If this value is too small, some endnodes will not be reachable by this router, causing unpredictable routing problems.
maximum broadcast routers [*number*]	Maximum number of routers that can be adjacent to (one hop away from) this router on broadcast circuits. Range: 1 to 33 times the number of broadcast circuits. Default: 32. Valid only with **define**. Make this value greater than or equal to the total number of routers on all broadcast circuits. If this value is too small, routes will not be accepted from some routers, causing unpredictable routing problems.
maximum cost [*number*]	Maximum cost that is allowed between this router and any other node in the area. If the best route to a node is more expensive than this, that node is considered unreachable. Maximum: 1022. Default: 1022. A suggested value is 25 times *maximum hops*.

maximum hops [*number*]	Maximum number of hops that are allowed between this router and any node in the area. If the best route to a node requires more than the maximum number of hops, that node is considered unreachable. Maximum: 30. Default: 30. It is about twice the longest path length (in hops) that is expected. The hop count is used by routing only to speed the decay of routes to unreachable nodes. This argument may be reduced to cause unreachable nodes to become unreachable more quickly.
maximum visits [*number*]	Maximum number of times a packet can be forwarded. The router will drop any packet it receives that has already been forwarded maximum visits times. Range: 1 to 63. Default: 63. This is used to detect packets that are in routing loops, which can occur temporarily as routes change. It should be set to at least twice the value of *maximum hops* or *area maximum hops*, whichever is larger.
state on	Enables DNA IV. Not valid with **set** if the router was started in *state off* or without a valid address, or if DNA IV initialization failed due to insufficient memory.
state off	Disables DNA IV. The default state is *off*.
type area	Causes the router to act as a level 2 router. It accepts adjacencies with routers in other areas, and keeps routes to all areas. If it can reach other areas, it also advertises itself as a route to other areas to level 1 routers.
type routing-IV	Causes the router to act as a level 1 router, which is the default. Adjacencies are accepted only to routers in the same area.

```
Example: set executor state on
         set executor maximum broadcast routers 10

Example: define executor state on
         define executor maximum broadcast routers 10
```

module access-control *circuit-specifier argument*

Defines access control lists, which are used to restrict the forwarding of packets between certain origins and destinations. Each access list is associated with one circuit, and applies to DECnet Long Format Data Packets received on that circuit. Access control does not apply to any routing or hello packets. Valid only with **define**.

The arguments for the circuit-specifiers include the following:

all circuits	Specifies all circuits on the router.
circuit name	Specifies the named circuit.
known circuits	Specifies all circuits on the router.

The following items are the arguments you select from after you enter the **define module access-control** command and the circuit-specifier:

state on	Enables the access control list on this circuit.
state off	Disables the access control list on this circuit.
type exclusive	Specifies that any packets matching one or more of the filters in the access control list for this interface are dropped.
type inclusive	Specifies that only packets matching one or more of the filters in the access control list for this interface are forwarded.
filter [*source-result source-mask dest-result dest-mask*]	Adds a filter to the list for the specified circuit. The filter is added to the end of the existing list. The source address is masked with the source-mask, and compared to the source result. The same is done with the dest-mask and dest-result. The action depends on what type of access control is in use on the circuit.

The following items are the options you select from after you enter the **define module access-control** command and the ***filter*** circuit-specifier:

source-result	Address that the source address is compared to after masking.
source-mask	Mask used for the source address.
dest-result	Address that the destination address is compared to after masking.
dest-mask	Mask used for the destination address.

Example: **define module access-control circuit eth/0 state on**

module routing-filter *circuit-specifier argument*

Defines routing filters, which are used to restrict the sending of Area routes by level 2 (Executor Type Area) routers. Valid only with **define**.

all circuits	Specifies all circuits on the router.
circuit name	Specifies the named circuit.
known circuits	Specifies all circuits on the router.

The following items are the direction options you select from after you enter the **define module routing-filter** command and the circuit-specifier:

incoming	Affects the filter on routing information received on this circuit.
outgoing	Affects the filter on routing information sent on this circuit.

The following items are the arguments you select from after you enter the **define module routing-filter** command and the circuit-specifier:

area [*area-list*]	Specifies that the filter allows routing information to pass for the set of areas in the *area-list*. The *area-list* is a comma-separated list of areas or ranges of areas. A range is specified by two area numbers separated by a dash. The value for *area-list* can also be *none*, specifying that information is passed for no areas. The following are area-list examples: 1,4,9,60 — Areas 1, 4, 9, and 60 1-7,9-13,23 — Areas 1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12, 13, and 23
state on	Specifies that the filter is active.
state off	Specifies that the filter is disabled, but continues to be stored in the permanent database. The only way to remove the filter is by using the **purge** command.

Example: **define module routing-filter circuit eth/0 state on**

node *address argument*

Identical to executor, when used with the router's address. No other addresses are valid. See the **set/define executor** command description for more information.

Show/List

Use the **show** command to display information in the volatile database. Use the **list** command to display configuration information in the permanent database. The command syntax for show and list is identical, except as noted below.

Syntax: show area-specifier . . .
circuit-specifier . . .
executor . . .
module . . .
node-specifier . . .

Syntax: list circuit-specifier . . .
executor . . .
module . . .
node-specifier . . .

area-specifier *argument*

Examines the status of the volatile area routing database. This lets you find out what areas are reachable, and what the routes are to various areas. Valid only with **show**.

The options for the area-specifiers include the following:

active areas	Provides information about those areas that are currently reachable.
all areas	Provides information about all areas (up to the executor maximum area).
area [*area*]	Provides information about the specified area. If the area is not provided, you are prompted for it.
known areas	Provides information about those areas that are currently reachable.

The arguments are the following:

characteristics	Shows the current state of the specified area. (The same as **summary**.)
status	Provides detailed information about the specified areas, including cost and hops.
summary	Shows the current state of the specified areas. This is the default.

The following area items are displayed by these commands:

area	Indicates the area for this line of the display.
circuit	Indicates which circuit the next hop to this node goes over. No circuit is given for the router's own area.
cost	Indicates the cost to this area.
hops	Indicates the hops to this area.
next node	Indicates the router that is the next hop (intermediate destination) to the specified area.
state	Indicates that this is reachable or unreachable.

Example: **show active areas**

```
Active Area Volatile Summary
Area State       Circuit Next
                         Node
1    reachable    Eth /0  1.22
2    reachable            2.26
```

Example: **show active areas status**

```
Active Area Volatile Status
Area State       Cost Hops Circuit Next
                                   Node
1    reachable   3    1     Eth /0  1.22
2    reachable   0    0             2.26
```

circuit-specifier *argument*

Use the **show circuit-specifier** command to retrieve information about the current state of the specified circuit(s) from the volatile database. The **list circuit** command retrieves the data that is stored in the permanent data base for circuits.

The circuit-specifier options are the following:

active circuits	Specifies all circuits that are currently on. Valid only with **show**.
all circuits	Specifies all circuits on the router.
circuit [*name*]	Specifies the named circuit.
known circuits	Specifies all circuits on the router. Valid only with **show**.

The arguments are the following:

characteristics	Provides detailed information about all of the argument settings for the circuit.
counters	Shows counters for the circuit. Valid only with **show**.
s*tatus*	Shows detailed information about the circuit from the volatile database.
s*ummary*	Shows summary information about the circuit from the volatile database. This is the default if no argument is supplied.

The following circuit items are displayed by these commands:

adjacent node	Node ID of a node that has an adjacency with this node on the circuit being displayed. While adjacencies with endnodes automatically make that node reachable, a router adjacency does not automatically make that node reachable. A router is not considered reachable unless a routing message was received over an active adjacency from that router. Nodes may show as adjacent in the circuit database, but are not in the reachable nodes database (**show active nodes**).
block size	Maximum data block size that the associated adjacent node is willing to receive. This is typically 1498 bytes, which is the standard 1500 bytes of an Ethernet packet, less the 2 byte length field used with DECnet.
circuit	Circuit(s) to which this data applies.
designated router	Displays what this node believes to be the designated router for this area on this circuit. (There may be some temporary disagreements when a new router starts up.) This normally is the same for all routers on the circuit. Endnodes send all packets for destinations not on the local circuit to their designated router.
hello timer	Hello timer for this circuit. Router hello messages are sent this often on the circuit.
listen timer	Amount of time designating how often router or endnode hellos must be received from this adjacency on this circuit. It is three times the hello timer set for this circuit on the adjacent machine.
router priority	Router priority for this circuit, used in vying for designated router status.
router type	Router type for this circuit – standard, phase IV with AMA, or Bilingual.
maximum routers	Maximum number of routers allowed on this circuit.

state

Either ON or OFF. In the volatile database, the state is ON if the circuit is enabled, and is passing self-test. If the circuit has failed self-test, or the device is not present, the state is OFF.

In the permanent database, this tells whether DNA tries to enable the circuit.

Example: **show all circuits**

```
Circuit Volatile Summary

Circuit State        Adjacent
                       Node

 Eth /0   on            1.22
 Eth /0                 2.14
 Eth /0                 1.13
 Eth /1   off
```

Example: **list circuit eth/0 characteristics**

```
Circuit Permanent Characteristics

Circuit            = Eth /0

State              = On
Cost               = 4
Router priority    = 64
Hello timer        = 15
Maximum routers    = 16
Router type        = Standard
```

Example: **show active circuits status**

```
Active Circuit Volatile Status

Circuit State       Adjacent   Block
                      Node     Size

 Eth /0  on            1.22     1498
 Eth /0                2.14     1498
 Eth /0                1.13     1498
```

Example: **show all circuits characteristics**

This example shows the current characteristics of the circuits on this machine. This includes all of the configuration arguments, as well as the current adjacencies, and the Listen timer (three times the adjacency's hello timer).

```
Circuit Volatile Characteristics

Circuit            = Eth /0

State              = on
Designated router  = 2.26
Cost               = 4
Router priority    = 64
Hello timer        = 15
Maximum routers    = 16
Adjacent node      = 1.22
  Listen timer     = 45
Adjacent node      = 2.14
  Listen timer     = 45
Adjacent node      = 2.39
  Listen timer     = 90
```

Example: **show circuit eth/0 counters**

This example shows the counters that are kept for the circuits. Note that some counters kept by DECnet-VAX are not kept here, but are instead read through the **network** command of GWCON.

```
Circuit Volatile Counters

Circuit = Eth /0

    525249  Seconds since last zeroed
         0  Terminating packets received
         0  Originating packets sent
      3693  Transit packets received
      4723  Transit packets sent
         0  Transit congestion loss
         0  Circuit down
         0  Initialization failure
         0  Packet corruption loss
```

executor *argument*

Retrieves information about the current state of the volatile database for DNA with the **show executor** command. The **list executor** command retrieves the data that is stored in the permanent data base for DNA.

The arguments are the following:

characteristics	Gives the detailed information about the settings of all of the adjustable arguments of the routing database.
counters	Gives the global event and error counters for DNA. Valid only with **show**.
status	Gives key information about the state of DNA.
summary	Gives a brief summary on the state of DNA. This is the default.

The following executor items are displayed by these commands:

area maximum cost	Specifies the maximum allowed cost to an area.
area maximum hops	Specifies the maximum allowed hops to an area.
broadcast routing timer	Specifies the frequency of sending routing messages in the absence of any changes.
buffer size	Specifies the buffer size for the router.
executor node	Specifies the node address and node name. The node name is the name set by the CONFIG **set hostname** command.
identification	Specifies the the identification of the router software, as sent in MOP System ID messages.
maximum address	Specifies the highest node number in the router's area to which routes are kept.
maximum area	Specifies the highest area to which routes are kept.
maximum broadcast nonrouters	Specifies the maximum number of endnodes adjacent to this router.
maximum broadcast routers	Specifies the maximum number of routers adjacent to this router.
maximum buffers	Specifies the number of packet buffers in the router.

maximum cost	Specifies the maximum allowed cost to a node in the router's area.
maximum hops	Specifies the maximum allowed hops to a node in the router's area.
maximum visits	Specifies the maximum number of routers through which a packet may be routed between source and destination.
physical address	Specifies the physical Ethernet address set on all Ethernet circuits when DNA starts. Derived from the node address.
routing version	Version is always Version 2.0.0.
state	The state of DNA, on or off.
type	Either ROUTING IV or AREA, corresponding to level 1 and level 2.

Example: **show executor**

```
Node Volatile Summary
Executor node            = 2.26 (gato)
State                    = on
Identification           = DECnet-MC68020 V9.0
```

Example: **show executor characteristics**

```
Node Volatile Characteristics
Executor node            = 2.26 (gato)
State                    = on
Identification           = DECnet-MC68020 V9.0
Physical address         = AA-00-04-00-1A-08
Type                     = area
Routing version          = V2.0.0
Broadcast routing timer  = 180
Maximum address          = 64
Maximum cost             = 1022
Maximum hops             = 30
Maximum visits           = 63
Maximum area             = 63
Max broadcast nonrouters = 64
Max broadcast routers    = 32
Area maximum cost        = 1022
Area maximum hops        = 30
Maximum buffers          = 103
Buffer size              = 2038
```

Example: **list executor status**

```
Node Permanent Status
Executor node            = 2.26 (gato)
State                    = on
Type                     = area
```

Example: **show executor counters**

```
Node Volatile Counters
Executor node            = 2.26 (gato)
525948  Seconds since last zeroed
     0  Aged packet loss
     0  Node unreachable packet loss
     0  Node out-of-range packet loss
     0  Oversized packet loss
     0  Packet format error
     0  Partial routing update loss
     0  Verification reject
```

module access-control *circuit-specifier argument*

The **show module access-control** command displays access control list information from the volatile database. The **list module access-control** command displays access control list configuration information from the permanent database.

The options for the circuit-specifiers include the following:

all circuits	Specifies all circuits on the router.
circuit [*name*]	Specifies the named circuit.
known circuits	Specifies all circuits on the router.

The arguments are the following:

counters	Gives counters on the use of the access control lists. Valid only with **show**.
status	Shows detailed information about the access control lists, including the filters in the access control list.
summary	Shows summary information about the state of the access control lists. This is the default.

Example: **show module access-control circuit eth/0 counters**

```
Module Access-Control Volatile Counters

Circuit = Eth /0

6337           Seconds since last zeroed
   0           Packets processed
   0           Packets rejected
   0           Access control loop iterations
```

module routing-filter circuit-specifier *argument*

The **show module routing-filter** command displays area routing filter information from the volatile database. The **list module routing-filter** command displays area routing filter configuration information from the permanent database.

Lists the DECnet area routing filters that were defined in the permanent database for the router.

all circuits	Specifies all circuits on the router.
circuit [*name*]	Specifies the named circuit.
known circuits	Specifies all circuits on the router.

The arguments are the following:

characteristics	Shows detailed information about the routing filters, including the area list.
status	Shows detailed information about the routing filters, including the area list.
summary	Shows summary information about the state of the routing filters. This is the default.

Example: **show module routing-filter circuit eth/0 status**

Example: **list module routing-filter circuit eth/0 status**

node-specifier argument

The **show node** command displays the contents of the volatile level 1 routing database. This indicates which nodes in the area are reachable and the next hop to each.

The permanent database only contains node information for the router itself. The **list node [*executor-address*]** command is equivalent to the **list executor** command.

The node-specifiers can be any of the following:

active nodes	Provides information about all nodes that are currently reachable. Valid only with **show**.
adjacent nodes	Provides information about nodes that are adjacent to (one hop away from) the router.
all nodes	Provides information about all nodes in the area (up to the *executor maximum address*). Valid only with **show**.
node [*node*]	Provides information about the specified node. If the node is not provided, you are prompted. Nodes other than the router itself are valid only with **show**.
known nodes	Provides information about those nodes that are currently reachable. Valid only with **show**.

The arguments include the following:

characteristics	Shows the current state of the specified nodes, including type and specified nodes.
status	Provides detailed information about the specified nodes, including type, cost, and hops.
summary	Shows the current state of the specified nodes. This is the default.

Example: **show active nodes**

This example shows the reachable nodes.

```
Active Node Volatile Summary
Executor node            = 2.26 (gato)
State                    = on
Identification           = DECnet-MC68020 V9.0

Node    State       Circuit Next
Address                  Node
 2.14   reachable    Eth /0  2.14
 1.22   reachable    Eth /0  1.22
```

Example: **show adjacent nodes status**

This example shows the detailed routing information about all adjacent nodes. Only nodes with one hop are shown. Note that the node type is only known and displayed for adjacent nodes, as this information is only contained in hello messages.

```
Adjacent Node Volatile Status

Executor node            = 2.26 (gato)
State                    = on
Physical address         = AA-00-04-00-1A-08
Type                     = area
Node    State       Type       Cost  Hops Circuit  Next
Addr                                             Node
 2.14  reachable  routing IV     3    1    Eth /0   2.14
 1.22  reachable  area           3    1    Eth /0   1.22
```

Purge

Remove access control lists and routing filters from the permanent database.

Syntax: purge module access-control . . .
module routing-filter . . .

module access-control *circuit-specifier*

Removes access control lists from the permanent database. You can delete an entire access control list; you cannot delete one filter.

all circuits	Specifies all circuits on the router.
circuit name	Specifies the named circuit.

Example: **purge module access-control all circuits**

module routing-filter *circuit-specifier*

Removes routing filters from the permanent database. You can purge a specified filter or you can purge all filters.

The options for the circuit-specifiers include the following:

all	Specifies all routing filters in the configuration memory.
circuit name	Specifies the routing filter for the named circuit.

Example: **purge module routing-filter all**

Zero

Clear circuit counters in the volatile database, global counters in the volatile database, and counters in the access control list module.

Syntax: zero circuit-specifier
executor
module access-control circuit-specifier

circuit-specifier

all circuits	Specifies all circuits on the router.
circuit [*name*]	Specifies the named circuit.
known circuits	Specifies all circuits on the router.

Example: **zero all circuits**

executor

Sets all global counters in the volatile database to a zero value. There are no options.

Example: **zero executor**

module access-control circuit-specifier

all circuits	Specifies all circuits on the router.
circuit [*name*]	Specifies the named circuit.

Example: **zero module access-control all circuits**

Exit

Return to the previous prompt level.

Syntax: exit

Example: **exit**

9

Configuring OSI/DNA V

This chapter describes the OSI configuration commands.

Note: When operating DNA IV networks together with DNA V networks, all DNA IV configuring and monitoring must be done from the DNA IV NCP> configuration process. For information about configuring DNA IV, refer to the chapter entitled "Configuring DNA IV." For more information about DNA IV and DNA V compatibility, refer to the *Routing Protocols Reference Guide.* From this point on, the use of the term "OSI" refers to both the OSI and DNA V environments unless otherwise indicated.

For more information about OSI and DNA V, refer to the *Routing Protocols Reference Guide.*

Accessing the OSI Configuration Environment

For information about accessing the OSI configuration environment, see Chapter 1.

Basic Configuration Procedure

This section outlines the minimum configuration steps that you are required to perform to get the OSI/DNA V protocol up and running over a LAN (Ethernet, token ring, and FDDI), serial lines, X.25 packet-switching networks, and Frame Relay. Before beginning any configuration procedure, use the **list device** command from the **config** process to list the interface numbers of the different devices. If you desire any further configuration command explanations, refer to the configuration commands described in this chapter.

Note: You must restart the router for new configuration changes to take effect.

Do the following basic configuration procedure before beginning the specialized procedures described in the following sections.

- **Setting the Network Entity Title (NET)**. You must set the router's NET using the **set network-entity-title** command. The NET consists of the router's system ID and its area address. Use the **list globals** command to verify that the NET is configured correctly.

- **Globally enabling OSI**. Enable the OSI software to run on the router using the **enable OSI** command. Use the **list globals** command to verify that the OSI protocol is enabled.

Configuring OSI Over an Ethernet, Token Ring, or FDDI LAN

To configure the OSI protocol to run over an Ethernet, token ring, or FDDI LAN, use the **set subnet** command. There is a one-to-one correspondence between subnetworks and interfaces. Use the **set subnet** command to configure all LAN subnets (Ethernet, token ring, and FDDI). Use the default multicast addresses for Ethernet and FDDI. When configuring a token ring subnet, use the addresses listed in Table 9–1. Use the **list subnet detailed** or **list subnet summary** commands to verify that you have configured the subnets correctly.

Table 9–1 Functional Addresses for Token Ring

Parameter	Functional Address
	802.5
All ESs [09002B000004]	C00000004000
All ISs [09002B000005]	C00000008000
All L1 ISs [0180C2000014]	C00000008000
All L2 ISs [0180C2000015]	C00000008000

Configuring OSI Over X.25 or Frame Relay

To configure the OSI protocol to run over the X.25 or Frame Relay interface, do the following:

- Set the subnet. Use the **set subnet** command to set the interface to X.25 or FRL (Frame Relay). Use the defaults for all the required information. Use the **list subnet detailed** or **list subnet summary** commands to verify that you configured the subnets correctly.

- Set the virtual circuit. Use the **set virtual-circuit** command to establish the a virtual circuit between the router and X.25 PSN (Packet Switching Node) or the frame relay switch.

Note: The router prompts you for a DTE address. For frame relay, enter the DLCI (Data Link Control Identifier) number. For X.25, enter the PSN's DTE address.

Configuring OSI Over a Serial Line

To configure the OSI protocol to run over a serial line, use the **set subnet** command to set the interface to SL (serial line). There is a one-to-one correspondence between subnetworks and interfaces. Use the defaults for all the required information. Use the **list subnet detailed** or **the list subnet summary** commands to verify that you configured the subnets correctly.

Configuring a DNA V Router for a DNA IV Environment

When configuring a DNA V router, it may be necessary to configure an interface to run in a DNA IV environment. For example, the router is attaching to both a DNA V and DNA IV network, or a DNA IV ES is attached to a DNA V router.

Before beginning the steps below, use the appropriate preceding section to configure OSI over a LAN, X.25, Serial Line, or Frame Relay.

- Enter the DN configuration process. Exit `OSI config>` and enter `NCP>`. Use the **protocol DN** command.

- Define the global DNA address. Use the **define executor address** command to configure the DNA node and area number of the router.
- Globally enable DNA. Use the **define executor state** command to enable the DNA protocol to run on the router.
- Enable inter-area routing. If the L2 routing algorithm is distance vector at level 2, use the **define executor type area** command to ensure that this router can exchange DNA IV level 2 routing information.
- Enable the DNA IV circuit. Enable the circuit that the router uses to exchange the routing information. Use the **define circuit** *type* **state on** command.

DNA IV and DNA V Algorithm Considerations

DNA IV uses a distance-vector routing algorithm. DNA V can use either a distance-vector or a link-state routing algorithm. The algorithm is selected according to what is enabled and disabled, and combinations that can result from these two protocols.

- **DNA IV disabled. OSI/DNA V enabled.** This combination is considered a pure OSI/DNA V environment and the algorithm is automatically set to link-state at both levels 1 and 2 regardless of how the **set algorithm** command is configured.
- **DNA IV enabled. OSI/DNA V disabled.** This combination is considered a pure DNA IV environment and the algorithm is set automatically to distance-vector regardless of how the **set algorithm** command is configured.
- **DNA IV enabled. OSI/DNA V enabled.** This a mixed environment, and the algorithm information is configured and read out of SRAM. Use the **set algorithm** command to configure this information into SRAM.

OSI Configuration Commands

This section summarizes and then explains the OSI configuration commands. The OSI configuration commands allow you to create or modify an OSI configuration. Enter all the OSI configuration commands following the `OSI Config>` prompt. Defaults for any command and its parameters are enclosed in brackets immediately following the prompt.

Table 9–2 OSI Configuration Commands Summary

Command	Function
? (Help)	Lists the configuration commands or lists any parameters associated with that command.
Add	Adds areas this node supports; receive-passwords for authentication purposes; prefix addresses for other domains; and aliases.
Change	Modifies a prefix address.
Clear	Clears a receive password, transmit password, or SRAM.
Delete	Deletes areas, virtual circuits, prefix-addresses, adjacencies, aliases, and subnets.
Disable	Disables a subnet or the OSI protocol.
Enable	Enables a subnet or the OSI protocol.
List	Displays the current configuration of adjacencies, aliases, virtual circuits, prefix-addresses, subnets, algorithm, phaseivpfx, or global information.
Set	Configures the properties associated with OSI parameters (switches, globals, NETs, timers, subnets, transmit-password, prefix-addresses, adjacencies, virtual circuits, algorithm, and phaseivpfx).
Exit	Exits the OSI configuration and returns to the CONFIG environment.

? (Help)

List the commands that are available from the current prompt level. You can also enter a **?** after a specific command name to list its options.

Syntax: ?

Example: **?**

```
LIST
ADD
CHANGE
DELETE
ENABLE
DISABLE
SET
CLEAR
EXIT
```

Example: **list ?**

```
GLOBALS
SUBNETS
VIRTUAL-CIRCUIT
ADJACENCIES
PREFIX-ADDRESSES
ALIAS
TIMERS
ALGORITHM
PHASEIVPFX
INTEGRATED-ISIS
SUMMARY-ADDRESSES
```

Add

Configure area and prefix addresses, receive passwords, and address aliases.

Syntax: add alias
area . . .
prefix-address
receive-password

alias

Adds an ASCII string that designates a particular area address or system ID. The ASCII string can be a-z, A-Z, -9, a few other characters including the hyphen (–), comma (,), and underscore (_). Do not use escape characters.

The offset indicates the position, in semi-octets (nibbles), where the ASCII string begins within the address (aliases used for system IDs have an offset of 1). The string must be the same size or longer than the segment it is designating or you receive an `invalid segment length` message. The maximum allowable alias is 20 bytes.

Note: When using an alias input, you must surround it with brackets. For example: **`l1_update 47[newname]99999000012341234`**.

Example: **add alias**

```
Alias []:
Segment []:
Offset [1]:
```

Alias	The character string you want to use.
Segment	The NSAP segment that the alias is replacing.
Offset	The location of the alias (in 4-bit, semi-octets) within the NSAP. The offset is determined from the beginning (left) of the NSAP as it is displayed on the console.

area *area-addr*

Adds additional area addresses (18 byte maximum) that the node supports. An L1 node that supports other areas considers those synonymous areas. One area address is the area portion of the configured NET. If you try to add a duplicate area address, the router displays an error message.

Example: **add area 47000580999999000012341234**

Note: When adding synonymous areas to a L1 node, use the **set globals** command to configure the maximum number synonymous areas allowed for this node. All routers within an area must use the same maximum number of synonymous areas. Adjacencies cannot be established if they are different.

prefix-address

Adds static routes to destinations external of the IS-IS domain. This parameter prompts you for different information depending on the type of subnet (X.25, SL, LAN, or FRL) that was configured using the **set subnet** command.

Note: If no Address Prefix is entered, the default prefix is assumed.

Example: **add prefix-address**

LAN Subnet:

```
Interface Number [0]:
Address Prefix []:
MAC Address []:
Default Metric [20]:
Metric Type [Internal]:
State [ON]?
```

X.25 Subnet:

```
Interface Number [0]:
Address Prefix []:
Mapping Type [Manual]:
DTE Address []:
Default Metric [20]:
Metric Type [Internal]:
State [ON]?
```

Serial Line Subnet:

```
Interface Number [0]:
Address Prefix []:
Default Metric [20]:
Metric Type [Internal]:
State [ON]?
```

Frame Relay Subnet:

```
Interface Number [0]:
Address Prefix []:
DTE Address []:
Default Metric [20]:
Metric Type [Internal]:
State [ON]?
```

Note: You receive the error message `Subnet does not exist – cannot define a reachable address` if the subnet does not exist.

`Interface Number` Defines the interface over which the address is reached.

`Address Prefix` Defines the NSAP prefix (20 bytes maximum).

`MAC Address` Defines the destination MAC address. You must specify this address if the interface corresponds to a LAN subnet. This prompt only appears if the interface is connected to a LAN subnet.

`Mapping Type` Defines how the destination physical address is determined, manual or X.121.

- If manual, the protocol prompts for the DTE address.
- If X.121, the protocol does not prompt you for the DTE address. The DTE address in this instance is extracted from the NSAP.

DTE Address	Defines the destination DTE address. You must specify this address if the interface is X.25 and the mapping type is manual. This prompt only appears if the interface is configured for X.25 and the mapping type is manual.
Default Metric	Defines the cost of the address.
Metric Type	Defines whether the metric cost is used for external (E) routing or internal (I) routing.
State	When set to ON, this prefix-address is advertised to other L2 routers. When set to OFF, this is a non-functional prefix-address.

receive-password

Adds an ASCII character string (16 character maximum) that authenticates all incoming packets. An incoming packet, whose password matches one of the set of receive-passwords, is processed through the IS. Any incoming packets whose passwords do not match are dropped.

Example: **add receive-password**

```
Password type [Domain]:
Password []:
Reenter password:
```

Note: To use IS–IS authentication for the receive password, you must enable IS-IS authentication with the **set switches on** command. You receive an error message if you use an invalid password type.

Password type	Designates one of the two types of passwords: Domain or Area. • Domain passwords are used with L2 LSPs (Level 2, Link State Packets) and SNPs (Sequence Number PDU). • Area passwords are used with L1 LSPs and SNPs.
Password	Designates the character string that you are using for authentication. Maximum allowable string is 16 characters.

Change

Modify a reachable address prefix. This command prompts you for different information, depending on the type of subnet (X.25, SL , LAN, or FRL) that you are changing.

Syntax: change prefix-address

Example: **change prefix-address**

LAN Subnet:

```
Interface Number [0]:
Address Prefix []:
MAC Address []:
Default Metric [20]:
Metric Type [Internal]:
State [ON]?
```

X.25 Subnet:

```
Interface Number [0]:
Address Prefix []:
Mapping Type [Manual]:
DTE Address []:
Default Metric [20]:
Metric Type [Internal]:
State [ON]?
```

Serial Line Subnet:

```
Interface Number [0]:
Address Prefix []:
Mapping Type [Manual]:
Default Metric [20]:
Metric Type (Internal or external)[Internal]:
State [ON]?
```

Frame Relay Subnet:

```
Interface Number [0]:
Address Prefix []:
DTE Address []:
Default Metric [20]:
Metric Type [Internal]:
State [ON]?
```

Interface Number	Indicates the interface over which the address is reached.
Address Prefix	Indicates the destination NSAP prefix (20 bytes maximum).

MAC Address	Indicates the destination MAC address. You must specify this address if the interface corresponds to a LAN subnet. This prompt only appears if the interface is connected to a LAN subnet.
Mapping Type	Indicates how the destination physical address is determined, manual or X.121. • If manual, the protocol prompts you for the DTE address. • If X.121, the protocol does not prompt you for the DTE address. The DTE address in this instance is extracted from the NSAP.
DTE Address	Defines the destination DTE address. You must specify this address if the interface is X.25 and the mapping type is manual. This prompt only appears if the interface is configured for X.25 and the mapping type is manual.
Default Metric	Indicates the cost of the address.
Metric Type	Indicates whether the metric cost is used for external (E) routing or internal (I) routing.
State	When set to ON, this address receives packets. When set to OFF, this is a non-functional address.

Clear

Erase SRAM or to remove the receive or transmit password.

Syntax: clear receive-password
sram
transmit-password

receive-password

Removes all of the receive-passwords previously configured using the **add receive-password** command.

Note: You receive an error message if you use an invalid password type.

Example: **clear receive**

```
Password Type [Domain]:
```

Password Type — Specifies the type of password being used: Domain or Area. Refer to the **add receive-password** command for description of these passwords.

SRAM

Warning: Using this parameter erases the OSI configuration from SRAM. Use this command only if you intend to erase the configuration.

Example: **clear sram**

```
Warning:  All OSI SRAM Information is erased .
Do you want to continue? (Y/N) [N]?
```

Transmit-password

Removes the transmit-password previously configured using the **set transmit-password** command. The output for this parameter is the same as that of the receive-password parameter.

Note: You receive an error message if you use an invalid password type.

Example: **clear password transmit**

```
Password Type [Domain]:
```

Delete

Remove parameters previously configured using the **set** or **add** commands.

Syntax: delete adjacency
alias
area . . .
prefix-address
subnet
virtual-circuit

adjacency

Removes a statically configured ES adjacency previously configured with the **set adjacency** command..

Example: **delete adjacency**

```
Interface Number [0]?
Area Address []?
System ID []?
```

Interface number	Indicates the interface where the adjacency is located.
Area address	Indicates the area address of the adjacency.
System ID	Indicates the portion of the NET that identifies the adjacency within the area.

alias

Removes the ASCII string that designates a portion of an area address or system ID.

Example: **delete alias**

```
ALIAS []?
```

area *address*

Removes the area address (*address*) previously configured with the **add area** command.

Example: **delete area 47000580999999000012341234**

prefix-address

Removes the prefix-address previously configured with the **set prefix-address** command.

Example: **delete prefix-address**

```
Interface Number [0]?
Address Prefix []?
```

Interface Number	Indicates the interface number over which the prefix-address is configured.
Address Prefix	Indicates the destination NSAP prefix.

subnet *intfc#*

Removes a subnet that was previously configured with the **set subnet** command. *Intfc#* indicates the interface number of the configured subnet.

Example: **delete subnet 1**

virtual-circuit

Removes an X. 25, SVC, or a Frame Relay virtual circuit that was previously configured with the **set virtual-circuit** command.

Example: **delete virtual-circuit**

```
Interface number [0]?
DTE address[]?
```

Interface number	Indicates the interface number over which the virtual circuit is configured.
DTE address	Indicates the DTE address of the X.25 network to which you are connecting or the DLCI of Frame Relay network to which you are connecting.

Disable

Disable those features previously enabled using the **enable** command.

Syntax: disable osi
subnet . . .
integrated–isis . . .

osi

Disables the OSI protocol on the router.

Example: **disable osi**

subnet *interface#*

Disables the OSI protocol on the specified subnet (*interface#*).

Example: **disable subnet 0**

integrated–isis

Disables the integrated ISIS protocol on the router.

Example: **disable integrated–isis**

Enable

Enable the OSI protocol, an OSI subnet, or the integrated ISIS protocol.

Syntax: enable osi. . .
subnet
integrated–isis

osi

Enables the OSI protocol on the router.

Example: **Enable osi**

subnet *interface#*

Enables the OSI protocol on the specified subnet (*interface#*).

Example: **Enable subnet 0**

integrated–isis

Enables the integrated ISIS protocol on the router.

Example: **enable integrated–isis**

List

Display the current configuration of the OSI protocol.

Syntax: list adjacencies
algorithm
alias
globals
phaseivpfx
prefix-addresses
subnets
timers
virtual-circuit

adjacency

Displays all statically configured ES adjacencies.

Example: **list adjacency**

```
Ifc           Area Address   System ID      MAC Address
 0                           0001-0203-0405 0001-0203-0405
 1                           0002-4000-0000 0000-0019-3004
```

Ifc	Indicates the interface number that connects to the adjacency.
Area Address	Indicates the area address of this ES adjacency.

System ID	Indicates the portion of the NET that identifies the adjacency.
MAC Address	Indicates the MAC address (SNPA) of the adjacency.

algorithm

Displays the routing algorithm that is configured in SRAM for the DNA V protocol. If you are running the OSI protocol only, this parameter is unsupported.

Example: **list algorithm**

```
Level 1 algorithm - LINK_STATE
Level 2 algorithm - DISTANCE_VECTOR
```

Level 1 Algorithm	Indicates the current configuration of the routing algorithm for level 1, Link State (default) or Distance Vector.
Level 2 Algorithm	Indicates the current configuration of the routing algorithm for level 2, Link State or Distance Vector (default).

Note: Depending on whether or not DNA IV is enabled or disabled, the routing algorithm displayed here may be different from what is actually running on the router.

alias

Displays the configured aliases and their corresponding address segments.

Example: **list aliases**

```
Alias          Segment                       Offset

joplin         AA0004000104                     1
moon           0000931004F0                     1
trane          000093E0107A                     1
```

globals

Displays the router's current NET, area addresses, switch settings, global parameters, and timer configuration.

Example: **list globals**

```
OSI State: Enabled*            Network Entity Title:
4700050001:0000931004F0
DNAV State: Enabled*

Area Addresses:
1. 4700050001    2. 7700050011

Switches:
ESIS Checksum = On            ES-IS Init Option = Off
Authentication = Off
```

```
Globals:
IS Type = L2                    System ID Length = 6
L1 LSP Size = 1492 bytes        L2 LSP Size = 1492 bytes
Max IS Adjs = 50                Max ES Adjs = 200
Max Areas = 50                  Max ESs per Area = 50
Max Ifc Prefix Adds = 100       Max Ext Prefix Adds = 100
Max Synonymous Areas = 3
```

OSI State or *DNAV State*	Indicates if the OSI or DNA V protocol is running on the router.
Network Entity Title:	Indicates the area address and system ID that make up the router's NET.
Area Addresses:	Indicates the areas that the router operates within. The first area address reflects the router's configured NET area address. Additional area addresses were added with the **add area** command.
Globals:	Indicates the currently configured global parameters:
IS Type	The router's designation in the OSI environment L1 or L2.
Domain ID Length	The size (in bytes) of the system ID portion of the NET. **Note:** All routers throughout the domain must agree on the length of the domain ID.
L1 LSP Size /L2 LSP Size	Displays the L1 and L2 maximum LSP buffer size.
Max IS Adjacencies/ Max ES Adjacencies	Displays the maximum number of ES and IS adjacencies this allowed for all circuits.
Max Areas	Displays the maximum number of areas in the routing domain.
Max ESs per Area	Displays the maximum number ESs allowed in one area.
Max Int Prefix Adds	Displays the maximum number of internal prefix addresses.

Max Ext Prefix Adds	Displays the maximum number of external prefix addresses.
Max Synonymous Areas	Displays the maximum number level 1 areas serviced by this router.

phaseivpfx

Displays the configured DNA phase IV address-prefix that the OSI protocol is using to route packets to a connected DNA IV network.

Example: **list phaseivpfx**

```
Local Phase IV Prefix: 49
```

prefix-address

Displays all the SNPAs for statically configured routes.

Example: **list prefix-addresses**

```
Ifc  Type  Metric  State  Address Prefix  Dest Phys Address
 0    INT    20     On        470006        302198112233
 1    EXT    50     OFF       470006        302198223344
```

Ifc	Indicates the interface number where the address can be reached.
Type	Indicates the type of metric, internal (INT) or external (EXT).
Metric	Indicates the cost of the reachable address.
Address prefix	Indicates the destination NSAP prefix. This prefix may be 20 bytes long.
Dest Phys Address	Indicates the destination DTE address if this interface is X.25 and the configured mapping is manual.

Subnet *subnet.reprt intfc#*

Displays subnet information.

- *Subnet.reprt* has two options, Summary and Detailed.
  - The summary option displays information for all configured subnets.
  - The detailed option displays information for LAN subnets only.
- Intfc# is the interface that connects to the subnet.

Example: **list subnet summary**

```
Ifc  State  Type  ESIS  ISIS  L2 Only  Ext Dom  Metric  EIH (sec) IIH(sec)
 0    On    LAN   Enb   Enb   False    False     20        10       3
 1    OFF   SL    Dis   Dis   True     True      50        10       3
 2    On    X25
 3    On    Frl
```

Ifc	Indicates the interface number of the subnet.
State	Indicates the state of the interface, ON or OFF.
Type	Indicates the type of subnet: LAN, X25, and Serial Line (SL).
ESIS	Indicates the state of the ES-IS protocol, enabled (Enb) or disabled (Dis).
ISIS	Indicates the state of the IS-IS protocol, enabled (Enb) or disabled (Dis).
L2 Only	Indicates if the router is operating at level 2 only: yes (true) or no (false).
Ext Dom	Indicates if the router is operating outside the IS-IS routing domain (external domain).
Metric	Indicates the cost of using this subnet.
EIH	Indicates the interval that ES hello messages are sent out over the subnet.
IIH	Indicates the interval that IS hello message are sent out over the subnet.

Example: **list subnet detailed**

```
Interface Number [0]?

Detailed information for subnet 0:
   ISIS Level 1 Multicast: 018002B000014
   ISIS Level 2 Multicast: 018002B000015
   All ISs Multicast:      009002B000005
   All ESs Multicast:      009002B000004
   Level 1 Priority: 64
   Level 2 Priority: 64
```

ISIS Level 1 Multicast	Indicates the multicast address to use when transmitting and receiving L1 IS-IS PDUs.
ISIS Level 2 Multicast	Indicates the multicast address to use when transmitting and receiving L2 IS-IS PDUs.
All ISs Multicast	Indicates the multicast address to use when receiving ES hellos.
All ESs Multicast	Indicates the multicast address to use when transmitting IS hellos.
Level 1 Priority/Level 2 Priority	Indicates the router's priority for becoming the designated router on the LAN.

timers

Displays the OSI/DNA V timer configuration.

```
Timers:*
Complete SNP (sec) = 10       Partial SNP (sec) = 2
Min LSP Gen (sec) = 30        Max LSP Gen (sec) = 900
Min LSP Xmt (sec) = 30        Min Br LSP Xmt (msec) = 33
Waiting Time (sec) = 60       DR ISIS Hello (sec) = 1
ES Config Timer (sec) = 10
```

* This output reflects what is actually running on the router: OSI or DNA V.

Timers:	Indicates the configuration of the OSI timers excluding any per circuit timers.
Complete SNP	The interval between generation of complete SNPs.
Partial SNP	The minimum interval between sending partial SNPs.

Min LSP Generation/ Max LSP Generation	The minimum and maximum intervals between generations of LSPs.
Min LSP Transmission	The minimum interval between LSP retransmissions.
Min Broadcast LSP Transmission	The minimum interval between LSP retransmissions on a broadcast circuit.
Waiting Time	The time the update process must delay before entering the ON state.
DR ISIS Hello	The interval between generations of IS-IS hello PDUs if this router is a designated router.
ES Config Timer	The minimum interval between that an ES must send a hello packet each time an interface comes up.

virtual-circuit

Displays all the configured X.25 SVCs or all the Frame Relay configured virtual circuits.

Example: **list virtual-circuit**

```
Ifc State ISIS  L2 Only  Ext Dom  Metric IIH   Dest DTE
 0   On    Ena   False    False     20    3   1238765742
```

Ifc	Indicates the interface number over which the configured virtual circuit runs.
State	When set to ON, OSI can operate over this circuit.
ISIS	Indicates if the IS-IS protocol is running (enabled) over the interface.
L2 Only	Indicates if the circuit is operating at level 2 only: yes (true) or no (false).
Ext Dom	Indicates if the circuit is operating outside the IS-IS routing domain (external domain).
Metric	Indicates the cost of the virtual circuit.

IIH	Indicates the interval at which IS-IS hellos are sent out.
Dest DTE	Indicates the DTE address of the X.25 network.

Set

Configure the router to run the OSI protocol.

Syntax: set
- adjacency
- algorithm
- globals
- network-entity-title
- phaseivpfx
- subnet
- switches
- timers
- transmit-password
- virtual-circuit

adjacency

Adds or changes an ES adjacency. Add an ES adjacency for all LAN ESs that do not run the ES-IS protocol.

Example: **set adjacency**

```
Interface Number [0]:
Area Address []:
System ID []:
MAC Address []:
```

Interface Number	Indicates the interface number that connects to the adjacency.
Area Address	Indicates the area where the adjacency is located.
System ID	Indicates system ID portion of the NET that is used to identify the adjacency.
MAC Address	Indicates the MAC address (SNPA) of the adjacency.

algorithm

Note: This is a DNA phase V command. This command only works if the DNA phase V protocol is included in the software load.

This allows you to select the type of routing algorithm that you are using for the DNA routing protocol, link state (DNA V) or distance vector (DNA IV).

Example: **set algorithm**

```
Level 1 Algorithm [link_state]?
Level 2 Algorithm [distance_vector]?
```

Level 1 Algorithm	Selects the type of routing algorithm, link_state (for DNA V networks) or distance_vector (for DNA IV networks).
Level 2 Algorithm	Selects the type of routing algorithm, link_state (for DNA V networks) or distance_vector (for DNA IV networks).

globals

Configures the global parameters required by the OSI protocol.

Example: **set globals**

```
IS Type [L2]:
Domain ID Length [6 bytes]:
Max Synonymous Areas [3]:
L1 LSP Buffer Size [1492 bytes]:
L2 LSP Buffer Size [1492 bytes]:
Max IS Adjacencies [50]:
Max ES Adjacencies [200]:
Max Areas [50]:
Max ESs per Area [500]:
Max Internal Prefix Addresses [100]:
Max External Prefix Addresses [100]:
Max Link State Updates [100]?
```

IS Type (L1 or L2)	Selects the level of the router, level 1 or level 2.
Domain ID Length	Selects the length of the domain ID portion of the NET. This length must be the same for all routers in the same domain.

Max Synonymous Areas	Selects the maximum number of level 1 areas that are serviced by this router.
L1 LSP Buffer Size	Selects the buffer size of the level 1 LSPs and SNPs originated by the router. Range is 512 to 1492. If the interface packet size is less than what you configured here, OSI does not run, and the router generates the ELS message ISIS.053.
L2 LSP Buffer	Selects the buffer size of the level 2 LSPs and SNPs originated by the router. Range is 512 to 1492. If the interface packet size is less than what you configured here, OSI does not run, and the router generates the ELS message ISIS.053.
Max IS Adjacencies	Selects the total number of IS adjacencies allowed for all circuits. This number is used to size the IS adjacency free pool.
Max ES Adjacencies	Selects the total number of ES adjacencies allowed for all circuits. This number is used to size the ES adjacency free pool.
Max Areas	Selects the total number of areas in the routing domain. This number is used to size the L2 routing table.
Max ESs per Area	Selects the total number ESs in any one area. This number is used to size the L1 routing table.
Max Internal Reachable Addresses	Selects the number you are using to size the internal metric routing table.
Max External Reachable Addresses	Selects the number you are using to size the external metric routing table.
Max Link State Updates	Selects the number you are using to size the link state database.

network-entity-title

Configures the router's NET. The NET consists of the router's system ID and area address.

Example: **set network-entity-title**

```
Area-address []:
System-ID []:
```

Area-address	Indicates one of area address portion of the router's NET. It is included as the first address in the router's set of manual area addresses. Each area address may be a maximum of 19 bytes.
System-ID	Defines the portion of the NSAP that identifies this specific router. The system ID can be a maximum of 19 bytes, but the length must agree with the domain ID length that you configured with the **set globals** command.

phaseivpfx

Configures the prefix-address to allow the OSI protocol to route packets to the attached DNA IV network. The default is 49 (hexadecimal).

Example: **set phaseivpfx**

```
Local Phase IV prefix [49]?
```

subnet

Adds or changes a subnet. This parameter prompts you for different information depending on the type of subnet that your configuring: X.25, serial line, or LAN.

Example: **set subnet**

X.25 subnet:

```
Interface number [0]:
Interface Type [LAN]:
```

Serial Line subnet:

```
Interface number [0]:
Interface Type [LAN]:
Enable IS-IS [N]?
Level 2 Only [N]?
External Domain [N]?
Default Metric [20]:
ISIS Hello Timer [3 sec]:
Modify Transmit password [No]?
Modify the set of receive passwords [No]?
```

LAN subnet:

```
Interface number [0]:
Interface Type [LAN]:
Enable ES-IS [N}?
Enable IS-IS [N]?
Level 2 Only [N]?
External Domain [N]?
Default Metric [20]:
ESIS IS Hello Timer [10 sec]:
ISIS Hello Timer [3 sec]:
Modify Transmit password [No]?
Modify the set of receive passwords [No]?
L1 Priority [64]:
L2 Priority [64]:
All ESs [0x09002B000004]:
All ISs [0x09002B000005]:
All L1 ISs [0x0180C2000014]:
All L2 ISs [0x0180C2000015]:
```

Frame Relay subnet:

```
Interface number [0]:
Interface Type [LAN]:
```

Interface number	Binds the subnet to the specified interface.
Enable ES-IS	Indicates whether the ES-IS protocol is going to run over the interface: yes (Y) or no (N).
Enable IS-IS	Indicates whether the IS-IS protocol is going to run over the interface: yes (Y) or no (N).
Interface Type	Indicates the type of subnet: LAN, Serial Line (SL), X.25, and Frame Relay (FRL). LAN includes Ethernet, token ring, and FDDI.

Level 2 Only	Indicates whether the subnet runs at level 2 only: yes (Y) or no (N). A no designation allows the router to route over that subnet at both level 1 and level 2.
External Domain	Indicates whether the circuit is operating outside the IS-IS routing domain.
Default Metric	Indicates the cost of the subnet. Cost range 20-63.
IS Hello Timer	Indicates the period between transmissions of IS hello PDUs.
ISIS Hello Timer	Indicates the period between transmissions of L1 and L2 IS-IS hello PDUs.
Modify Transmit password	Removes or changes a circuit transmit password. When you select yes, this option prompts you with the following message: `Delete or change the transmit password [change]?`
Modify the set of receive passwords	Removes all or adds one circuit receive-password. When you select yes, this option prompts you with the following message: `Delete all or add 1 receive password [add]?`
L1 Priority/L2 Priority	Indicates the router priority for becoming the designated router on the LAN.
All ESs	Indicates the multicast address to use when transmitting IS hellos. The default address reflects the ethernet/802.3 multicast address. If you are connecting to a 802.5 LAN, use C00000004000.
All ISs	Indicates the multicast address to use when receiving ES hellos. The default address reflects the ethernet/802.3 multicast address. If you are connecting to a 802.5 LAN, use C00000008000.

All L1 ISs	Indicates the multicast address to use when transmitting and receiving L1 IS-IS PDUs. The default address reflects the ethernet/802.3 multicast address. If you are connecting to a 802.5 LAN, use C00000008000.
All L2 ISs	Indicates the multicast address to use when transmitting and receiving L2 IS-IS PDUs. The default address reflects the ethernet/802.3 multicast address. If you are connecting to a 802.5 LAN, use C00000008000.

switches

Turns the OSI options on or off.

Example: **set switches**

```
ES-IS Checksum Option [OFF]?
ES-IS Init Option [OFF]?
Authentication [OFF]?
```

ES-IS Checksum Option	When switched on, the router generates checksums for all sourced ES-IS packets.
ES-IS Init Option	When switched on, the router sends a directed IS Hello to a new ES neighbor.
IS-IS Authentication	If switched on, each IS-IS packet includes the transmit password configured for the domain, area, and circuits. No checking of receive passwords is done.

timers

Configures the OSI timers, excluding any circuit timers.

Example: **set timers**

```
Complete SNP [10 sec]:
Partial SNP [2 sec]:
Min LSP Generation [30 sec]:
Max LSP Generation [900 sec]:
Min LSP Transmission [5 sec]:
Min Broadcast LSP Transmission [33 msec]:
Waiting Time [60 sec]:
Designated Router ISIS Hello [1 sec]:
Suggested ES Configuration Timer (sec) [10]:
```

Complete SNP	Selects the interval between the generation of complete sequence number PDUs (SNP) by the designated router on a broadcast circuit.
Partial SNP	Selects the minimum interval between sending partial sequence number PDUs (SNP).
Min LSP Generation	Selects the minimum interval between successive generations of Link State Packets (LSPs) with the same LSP ID generated by the router.
Max LSP Generation	Selects the maximum interval between LSPs generated by the router.
Min LSP Transmission	Selects the minimum interval between retransmissions of a LSP.
Min Broadcast LSP Transmission	Selects the minimum transmission, in milliseconds, between transmission of LSPs on a broadcast circuit.
Waiting Time	Selects the number of seconds the update process delays in the waiting state before entering the ON state.
Designated Router ISIS Hello	Selects the interval between the generation of IS-IS hello PDUs by the router if the router is the designated router on a LAN.
Suggested ES Configuration Timer	Sets the option field of the IS hello message that instructs the ES to change the rate at which it sends ES hellos.

transmit-password

Sets or changes a transmit password.

Example: **set transmit-password**

```
Password type [Domain]:
Password []:
Renter password:
```

Password type — Selects the type of password: domain or area.

- Domain passwords are used with L2 LSPs and SNPs.
- Area passwords are used with L1 LSPs and SNPs.

Password — Indicates the character string that your using for authentication. Maximum allowable string can be 16 characters.

virtual-circuit

Configures a X.25 SVC or a Frame Relay virtual circuit.

Example: **set virtual-circuit**

```
Interface Number [0]:
DTE Address []:
Enable IS-IS (Y or N) [Y]?
L2 only (Y or N) [N]?
External Domain (Y or N) [N]?
Default Metric [20]:
ISIS Hello Timer [3 sec]?
Modify transmit password (y or n) [N]?
Modify the set of receive passwords [No]?
```

Interface Number — Indicates the X.25 or Frame Relay interface over which the virtual circuit is configured.

DTE Address — Indicates the destination DTE address for X.25 or the DLCI (Data Link Control Identifier) for Frame Relay. This address must be the same as the one defined for the virtual circuit in the X.25 configuration or the Frame Relay configuration.

Default Metric — Indicates the cost of the circuit.

Enable IS-IS — Indicates whether the IS-IS protocol is going to run over the interface: yes (Y) or no (N).

L2 only	Indicates whether the circuit runs at level 2 only: yes (Y) or no (N). A no designation allows the router to route at both level 1 and level 2.
External Domain	Indicates whether the circuit is operating outside the IS-IS routing domain.

Exit

Return to the previous prompt level.

Syntax: exit

Example: **exit**

10

Monitoring OSI/DNA V

This chapter describes the OSI console commands you can use to monitor OSI/DNA V.

If you need more information on OSI and DNA V, refer to the *Routing Protocols Reference Guide*.

Accessing the OSI Console Environment

For information about accessing the OSI console environment, see Chapter 1.

OSI Console Commands

This section summarizes and then explains the OSI Console commands. Use these commands to gather information from the database.

Table 10–1 OSI Console Commands Summary

Command	Function
? (Help)	Displays all the OSI console commands or any options associated with a specific command.
Addresses	Displays the router's NET and area addresses.
Change Metric	Modifies the cost of a circuit.
CLNP-Stats	Displays OSI CNLP statistics.
Designated-router	Displays the designated router for the LAN.
DNAIV-Info	Displays the routing algorithm currently running on the router.
ES-adjacencies	Displays all the ES adjacencies in the adjacency database.
ES-IS-Stats	Displays statistics associated with the ES-IS protocol.
IS-adjacencies	Displays all the IS adjacencies in the adjacency database.
IS-IS-Stats	Displays statistics associated with the IS-IS protocol.
L1-routes	Displays all the L1 routes in the Level 1 database.
L2-routes	Displays all the L2 routes in the Level 2 database.
L1-summary	Displays a summary of the level 1 link state database.
L2-summary	Displays a summary of the level 2 link state database.
L1-update	Displays the information contained in L1 link state update packet.
L2-update	Displays the information contained in L2 link state update packet.
Route	Displays the route a packet takes to a specified destination.
Send echo packet	Encodes an echo request message in the CLNP packet.
Subnets	Displays the state of all operational subnets.
Toggle	Enables or disables the NSAP alias substitution function.
Traceroute	Displays the route a packet travels to its destination.
Exit	Exits the OSI console command process.

? (Help)

List the commands that are available from the current prompt level. You can also enter a **?** after a specific command name to list its options.

Syntax: ?

Example: **?**

```
ADDRESSES
SUBNETS
DESIGNATED-ROUTER
ES-ADJACENCIES
IS-ADJACENCIES
L1-ROUTES
L2-ROUTES
L1-SUMMARY
L2-SUMMARY
L1-UPDATE
L2-UPDATE
CLNP-STATS
ES-IS-STATS
IS-IS-STATS
CHANGE METRIC
SEND ECHO PACKET
TRACEROUTE ADDRESS
ROUTE NSAP
TOGGLE ALIAS/NO-ALIAS
DNAV–INFO
```

Addresses

List the router's NET and the area addresses configured for this router.

Syntax: addresses

Example: **addresses**

```
Network Entity Title:
4700-0500-01  000-9310-04F0
Area Addresses:
4700-0500-01
4900–02
```

Network Entity Title	Identifies the router. The NET consists of an area address and a system ID.
Area Address	Indicates addresses within the routing domain. The router can have a maximum of three area addresses configured at any one time.

Change Metric

Modify the cost of the circuit.

Syntax: change metric

Example: **change metric**

```
Circuit [0]?
New Cost [0]?
```

Circuit	Indicates the circuit number that you want to change.
New Cost	Indicates the new cost of the circuit. Range: 1 – 63.

CLNP-Stats

Display the OSI Connectionless Layer Network Protocol (CLNP) information.

Syntax: clnp-statistics

Example: **clnp-statistics**

```
Received incomplete packet                     0
Received packet with bad NSAP length           0
Received packet with bad checksum              0
Received packet with bad version number        0
Received packet with bad type                  0
Received packet with expired lifetime          0
Received packet with bad option                0
Received packet with unknown destination       0
Received packet with no segmentation permitted 0
Received data packet cannot be forwarded       0
No buffer available to send error packet       0
No route to send error packet                  0
Received OK CLNP packet                        0
Cannot forward error packet                    0
IS0 unknown initial protocol ID                0
Received error packet                          0
Received local data packet                     0
Sent error packet                              0
```

Received incomplete packet	Indicates that a data packet fragment recognized as an ISO CLNP data packet was received.
Received packet with bad NSAP length	Indicates that an ISO CLNP data packet was received with an illegal NSAP length.
Received packet with bad checksum	Indicates that an ISO CLNP data packet was received with a bad checksum.
Received packet with bad version number	Indicates that an ISO CLNP data packet was received with an incorrect or unsupported version number.
Received packet with bad type	Indicates that an ISO CLNP data packet was received with an incorrect or unsupported type field.
Received packet with expired lifetime	Indicates that an ISO CLNP data packet was received with an expired lifetime.
Received packet with bad option	Indicates that an ISO CLNP data packet was received with a bad optional parameter.
Received packet with unknown destination	Indicates that an ISO CLNP data packet was received but was not routed. The routing table contains no entry for the destination.
Received packet with no segmentation permitted	Indicates that an ISO CLNP data packet was received that needed segmentation. The segmentation permitted flag was not set.
Received data packet cannot be forwarded	Indicates that an ISO CLNP data packet was received but was not routed because of a handler error.
No buffer available to send error packet	An attempt to send an ISO CLNP error packet failed because of a lack of system I/O buffers.
No route to send error packet	An attempt to send an ISO CLNP error packet failed because it was not routed.

Received OK CLNP packet	Indicates that an ISO CLNP data packet was received and passed error checking.
Cannot forward error packet	Indicates that an ISO CLNP error packet was not routed because of a handler error.
ISO unknown initial protocol ID	Indicates that an ISO CLNP packet was received with an unknown or unsupported initial protocol identifier.
Received error packet	Indicates that an ISO CLNP error packet was received for this router.
Received local data packet	Indicates that an ISO CLNP data packet was received with the destination NSAP indicating one of the router's NSAPs.
Sent error packet	Indicates that ISO CLNP error packet was sent on receipt of a bad packet.

Designated-router

Display the designated router for the LAN subnets that are physically attached to this router and actively running IS-IS.

Syntax: designated-router

Example: **designated-router**

```
Designated Router Information:
Hdw    Int#     Circ    L1DR                    L2DR
Eth/3     4      2      0000-0000-0025-02       0000-0000-0025-02
Eth/1     2      1      0000-0000-0025-01       0000-0000-0025-01
```

Hdw	Indicates the type and instance of LAN attached to this router.
Int#	Indicates the interface number of this router that attaches to the LAN.
Circ	Indicates the circuit number assigned by the router.

L1DR — Indicates the LAN ID of the designated router. If the use of alias is enabled, this command displays the alias of the particular segment. The LAN ID is the designated router's system ID concatenated with a 1 byte locally assigned circuit ID.

L2DR — Description is the same as L1DR described above.

Note: If the designated router was not elected yet, `Not Elected` is displayed instead of a LAN ID.

DNAV-info

Display the routing algorithm that is currently running on the router.

Syntax: dnav-info

Example: **`dnav-info`**

```
DNA V Level 1 Routing Algorithm: Distance-vector
DNA V Level 2 Routing algorithm: Distance-vector
```

Note: Depending on whether or not DNA IV is enabled or disabled, the routing algorithm displayed here may differ from what is configured in SRAM using the `OSI config>` **set algorithm** command.

If DNA IV is enabled, the routing algorithm is the one configured in SRAM. If DNA IV is disabled, the routing algorithm is set to link state and may differ from that set in SRAM.

ES-Adjacencies

Display all the End System (ES) adjacencies that are either configured or were learned through the ES-IS protocol.

Syntax: es-adjacencies

Example: **es-adjacencies**

```
End System Adjacencies
System ID        MAC Address     Interface    Lifetime   Type
6666-6666-6666   1234-FEAA-041C      0        50         DNAIV
0000-9310-0040   4221-FEAA-03B2      1        static     MNUAL
AA00-0400-0C04   AA00-0400-0C04      1        128        OSI
```

System ID	The system ID of the ES adjacency.
MAC Address	Indicates the MAC address of the ES on the subnet.
Interface	Indicates the router's interface number where the ES adjacency was learned.
Lifetime	Indicates the amount of time (in seconds), that the router has left before the information received in the last ES Hello message is discarded. In the case of static or a manually configured ES-Adjacency, this field reads "Static."
Type	Indicates the type of ES adjacency, OSI, DNAIV, DNAIV', and MANUAL for statically configured adjacencies.

ES-IS-Stats

Display the statistics for the ES-IS protocol.

Syntax: es-is-stats

Example: **es-is-stats**

```
ESIS input queue overflow                    0
Received incomplete packet                   0
Received packet with bad checksum            0
Received packet with bad version             0
Received packet with bad type                0
No iob available to send hello               0
Cannot send hello due to packet handler error  0
Sent hello                                   3672
```

```
Received packet with bad header                0
Received hello with bad nsap                   0
Received hello packet with bad option          0
Received hello                                 0
Received hello with unsupported domain source  0
No resources to install route                  0
Received hello with conflicting route          0
Timed out route reactivated                    0
No resources to send redirect                  0
Redirect not sent – handler error              0
Sent redirect                                  0
Timed out route                                0
```

ESIS input queue overflow	The ES-IS packet was dropped because a task input queue overflowed.
Received incomplete packet	A packet fragment recognized as an ES-IS packet was received.
Received packet with bad checksum	An ES-IS packet with a bad checksum was received.
Received packet with bad version	An ES-IS packet with a bad or unsupported version was received.
Received packet with bad type	An ES-IS packet with a bad or unsupported type field was received.
No iob available to send hello	An attempt to send an ES-IS hello failed because of a lack of system I/O buffers.
Cannot send hello due to packet handler error	An ES-IS hello was not sent because of a handler error.
Sent hello	An ES-IS hello was sent through an interface.
Received packet with bad header	An ES-IS hello packet with a bad holding time or received field was received.
Received hello with nsap	An ES-IS hello packet with a bad NSAP or one that overflowed the field was received.
Received hello packet with bad option	An ES-IS CLNP data packet was received with a bad option parameter.

Received hello	An ES-IS hello packet was received on the interface.
Received hello with unsupported domain source	An ES-IS hello packet was received from an unspecified domain source.
No resources to install route	An ES-IS hello packet was received, but there were no resources to install the route.
Received hello with conflicting route	An ES-IS hello packet was received but was not entered into the database. A previously defined static or dynamic route in the database was conflicting with the route in the hello.
Timed out route reactivated	An ES-IS hello packet with a previously timed out route was received.
No resources to send redirect	An ES-IS redirect packet was not sent because of a lack of resources.
Redirect not sent - handler error	An ES-IS redirect packet was not sent because of a handler error.
Sent redirect	An ES-IS redirect packet was sent out the interface.
Timed out route	An ES-IS hello route has timed out.

IS-Adjacencies

List all the IS adjacencies that are learned through the IS-IS protocol.

Syntax: is-adjacencies

Example: **is-adjacencies**

```
End System Adjacencies
System ID       MAC Address    Int  Level  Usage   State   Life   Type
0000-9310-04C8  AA00-0400-EF04  0     L1   L1/L2   DOWN           OSI
0000-9310-04C8  AA00-0400-EF04  0     L2   L1/L2   DOWN         DNAIV
AA00-0400-0504  AA00-0400-0504  1     L2     L2     UP     5390    OSI
```

System ID	The system ID of the IS adjacency.
MAC Address	Indicates the MAC Address of the IS adjacency.

Int	Indicates the router's interface number that connects to the IS adjacency.
Level	For LANs this indicates the neighbor system level from type of hello message, L1 or L2. For point-to-point, this indicates the neighbor system type: L1 only, otherwise L2.
Usage	Indicates from the hello packet circuit type: L1 only, L2 only, or L1 and L2.
State	Indicates the operational state of the IS adjacency: up or down.
Life	Indicates the amount of time (in seconds), before discarding the last IS Hello message.
Type	Indicates the routing protocol type of the IS adjacency: OSI or DNA IV.

IS-IS-Stats

Display information associated with the IS-IS protocol.

Syntax: is-is-stats

Example: **is-is-stats**

```
Link State Database Information

no. of level 1 LSPs      1       no. of level 2 LSPs        0
no. of L1 Dijkstra runs  21      no. of L2 Dijkstra runs    0
no. of L1 LSPs deleted   0       no. of L2 LSPs deleted     0
no. of routing table entries allocated                      6
Packet Information

level 1 lan hellos rcvd  0       level 1 lan hellos sent  10967
level 2 lan hellos rcvd  0       level 2 lan hellos sent  10967
pnt to pnt  hellos rcvd  0       pnt to pnt  hellos sent  0
level 1 LSPs rcvd        0       level 1 LSPs sent        40
level 2 LSPs rcvd        0       level 2 LSPs sent        0
level 1 CSNPs rcvd       0       level 1 CSNPs sent       0
level 2 CSNPs rcvd       0       level 2 CSNPs sent       0
level 1 PSNPs rcvd       0       level 1 PSNPs sent       0
level 2 PSNPs rcvd       0       level 2 PSNPs sent       0
```

no. of level 1/level 2 LSPs	Indicates the number of L1 and L2 link state packets that are in the database.
no. of L1/L2 Dijkstra runs	Indicates the number of times the router computed the L1 and L2 routing tables.
no. of L1/L2 LSPs deleted	Indicates the number of L1 and L2 link state packets that were deleted from the database.
no. of routing table entries allocated	Indicates the number of entries the routing able currently holds.
level 1/level 2 lan hellos rcvd	Indicates the number of LAN hellos the router received.
level 1/level 2 hellos sent	Indicates the number of LAN hellos the router sent.
pnt to pnt hellos rcvd	Indicates the number of point to point hellos that the router received.
pnt to pnt hellos sent	Indicates the number of point to point hellos that the router sent.
level 1/level 2 LSPs rcvd	Indicates the number of L1 and L2 link state packets (LSPs) that the router received.
level 1/level 2 LSPs sent	Indicates the number of L1 and L2 LSPs that the router sent.
level 1/level 2 CSNPs rcvd	Indicates the number of L1 and L2 complete sequence number PDUs (CSNPs) that the router received.
level 1/level 2 CSNPs sent	Indicates the number of L1 and L2 CSNPs that the router sent.
level 1/level 2 PSNPs rcvd	Indicates the number of L1 and L2 partial sequence number PDUs (PSNPs) that the router received.
level 1/level 2 PSNPs sent	Indicates the number of L1 and L2 PSNPs that the router sent.

L1-Routes

Display all the level 1 routes that are in the L1 routing database.

Syntax: l1-routes

Example: **l1-routes**

```
Level 1 Routes
Destination System ID  Cost   Source     Next Hop
0000-9300-0047          0     LOC-Area      *
AA00-0400-080C          1     ES-IS      AA00-0400-0C04, Ifc 7
7777-7777-7777          0     IS-IS      3455-6537-2215
```

Destination System ID	Indicates the system ID of the destination host.
Cost	Indicates the cost of this route.
Source	Indicates the one of three sources where the router learned of the route: LOC-AREA, ES-IS, or IS-IS.
Next Hop	Indicates the next hop a packet takes on its route. An asterisk (*) designation refers to the router itself as the packet's destination. An address with an interface number is either the MAC address of a directly connected ES, or the DTE address, if the next hop is an X.25 switch, or a DLCI if the next hop is Frame Relay switch. A system ID (*345565372215)* refers to the next hop to destination.

L2-Routes

Display all the level 2 routes in the L2 database.

Syntax: l2-routes

Example: **l2-routes**

```
Level 2 Routes
Destination    Cost          Type          Next Hop
4700-0500–01    0            LOC-AREA          *
4900-02         20           AREA          0000-9310-04C9
```

Destination	Indicates the system ID of the destination area or reachable address.
Cost	Indicates the cost of this route.
Type	Indicates the four types of routes: LOC-area (local), LOC-prefix, area, prefix/I, and prefix/E. LOC-area is a directly connected area; a LOC-prefix is a prefix that this router advertises; prefix/I and prefix/E are routes that require another hop to reach their destination.
Next Hop	Indicates the next hop a packet would take on its route. An * designation or a "direct" designation refers to a directly connected host off the router. A system ID refers to the next router the packet must pass through to reach its destination.

L1-Summary

Display a summary of the level 1 link state database.

Syntax: l1-summary

Example: **l1-summary**

```
Link State Database Summary - Level One

LSP ID              Lifetime   Sequence #  Checksum   Flags  Cost

0000-9300-40B0-0000   0           0           0        0    1024
0000-93E0-107A-0000   384         CE          3CC9     1       0
AA00-0400-0504-0000   298         8E          40F1     B      20
AA00-0400-0504-0100   4           B8          A812     3      20

Total Checksum 25CC
```

LSP ID	This represents the system ID of the source of the link state PDU, plus two additional bytes. The first additional byte designates the type of update. 0 represents a non-pseudonode update. 1-*FF* represents a pseudonode update for that circuit number. The second byte represents the LSP number. This number is attached to the packet when the data is contained in more than one packet.
Lifetime	Indicates the amount of time (in seconds), that the router maintains the LSP.
Sequence #	Indicates the sequence number of the LSP.
Checksum	Indicates the checksum value of the LSP.

Flags Indicates a one octet value that reflects the flag field of the LSP. The eight bits are broken down as follows:

- **Bit 8** – Indicates the *P* flag. When set (1) the issuing IS supports the optional Partition Repair function.
- **Bits 7-4** – Indicate the *ATT* flag. When set (1) the issuing IS is attached to other areas using one of the following: the Default Metric (bit 4), the Delay Metric (bit 5), the Expense Metric (bit 6), or the Error Metric (bit 7).
- **Bit 3** – Indicates the *LSPDBOL* flag. When set (1) a LSP database overload has occurred. An LSP with this bit set is not used by the decision process to calculate routes to another I through the originating system.
- **Bits 2-1** – Indicate the *IS Type* flag. When set to the following values designates the type of IS router, level 1 or level 2.

Value	Description
0	Unused.
1	Bit 1 set. Level 1 IS.
2	Unused.
3	Bits 1 and 2 set. Level 2 IS.

Cost Indicates the cost of routing to that neighbor.

L2-Summary

Display a summary of the level 2 link state database.

Syntax: l2-summary

Example: **l2-summary**

```
Link State Database Summary - Level Two

LSP ID               Lifetime   Sequence #  Checksum    Flags   Cost
0000-9310-04F0-0000     33E     12            EF19        3       0
0000-5000-FB06-0000     455      4            2BB1        3      20
0000-5000-FB06-0100     469     12            DE32        3      20

Total Checksum 0
```

Description of the **L2-summary** output is the same as the **l1-summary** command listed on the previous page.

L1-Update

Display a link state update for the specified level 1 IS.

Syntax: l1-update

Example: **l1-update**

```
LSP ID []? 000931004F0000

Link State Update For ID 0000931004F00000

Area Addresses

470005001

Intermediate System Neighbors Metric         Two Way

0000931004F002                  20             N
0000931004F001                  20             Y

End System Neighbors          Metric

00009310004F0                   *
```

LSP ID	Indicates the system ID of the source of the link state PDU, plus two additional bytes. The first byte designates the type of update. 0 represents a non-pseudonode update. 1-*FF* represents a pseudonode update. The second byte represents the LSP number. This number is attached to the packet when the data is contained in more than one packet.
Area Addresses	Indicates the area addresses in which this router is configured to route packets.

Intermediate System Neighbors	Indicates adjacent neighbor ISs.
Metric	Indicates the cost to the neighbor IS.
Two Way	Indicates whether the router is receiving updates from its neighbor.
End System Neighbors	Indicates any directly connected ESs.

L2-Update

Display the link state update for the specified level 2 IS.

Syntax: l2-update

Example: **l2-update**

```
LSP ID []? 000931004F0000

Link State Update For ID 0000931004F00000

INTERMEDIATE SYSTEM NEIGHBORS METRIC   TWO WAY
   0000931004F002                          20        N
   0000931004F001                          20        N
   55002000182000                          20        N
```

Intermediate System Neighbors	Indicates other directly connected ISs.
Metric	Indicates the cost to the IS.
Two Way	Indicates whether the router is receiving updates from its neighbor.

Route

Display the next hop a packet takes to a specified destination (*dest-nsap*).

Syntax: route *dest-nsap*

Example: **route 490002aa0004000e08**

```
Destination System: 0000-9310-04C9
Destination MAC Address: AA00-0400-1408
Interface: 0
```

Destination System	Indicates the system ID of the next hop IS. For a directly connected ES, this is blank.
Destination MAC Address	Indicates the MAC address of the next hop IS or the directly connected ES.
Interface	Indicates the interface that a packet goes out over to reach the the next hop IS or the directly connected ES.

Send (Echo Packet)

Encode an echo request message in the CLNP packet to the specified destination nsap. During this command, the system does not interact with the OSI console. To verify that the echo request was sent and that an echo reply was received, check the ELS (Event Logging System).

Note: You cannot send an echo packet to yourself. If you try, you receive a CLNP.004 ELS message.

Syntax: send

Example: **send**

```
Destination NSAP: []?
```

Subnets

Display information on all operational subnets. Subnets that are down or disabled are not listed.

Syntax: subnets

Example: **subnets**

```
                    L2
 Hdw    Int #  Circ  Only  ES-IS IS-IS  L1DR  LPri  L2DR  L2pri  Cost  Ext
 SL /2   2       3     N     N     Y
 Eth /0  0       1     N     Y     Y      Y     64    N     64     20    N
FDDI/1   1       2     N     Y     Y      N     64    N     64     20    N
```

Hdw	Indicates the type and instance of the network that connects to the subnet.
Int #	Indicates the router's interface number that connects to the subnet.
Circ	Indicates the circuit assigned ID for the IS-IS protocol.
L2 only	Indicates whether this router is a level 2 router only: Y (yes) or N (no).
ES-IS	Indicates if ES-IS protocol is enabled on the subnet: Y or N.
IS-IS	Indicates if the IS-IS protocol is enabled on the subnet: Y or N.
L1DR	Indicates if this router is the level 1 designated router for this subnet: Y or N.
LPri	Indicates the subnet's level 1 priority for becoming the designated router.
L2DR	Indicates if this router is the level 2 designated router for this subnet: Y or N.
LPri	Indicates the LAN subnet's level 2 priority for becoming the designated router.
Cost	Indicates the cost of the circuit.
Ext	Indicates whether the subnet is operating outside the IS-IS routing domain (external).

Toggle (Alias /No Alias)

Enable or disable the NSAP alias display function for the OSI protocol.

Syntax: toggle

Example: **toggle**

```
Alias substitution is ON
```

Traceroute

Track the path an OSI packet takes to a destination.

Note: You cannot do a traceroute to yourself or else you receive the following error message:

```
Sorry, can't traceroute to this router.
```

Syntax: traceroute *address*

Example: **traceroute 490002aa0004000e08**

```
Successful trace:

TRACEROUTE 470007: 56 databytes

1          490002aa0004000e08     32ms     5 ms      5ms
Destination unreachable response:

Destination unreachable
No response:

1 * * *
2 * * *
```

TRACEROUTE	Displays the destination area address and the size of the packet being sent to that address.
1	The first trace showing the destination's NSAP and the amount of time it took the packet to arrive at the destination. The packet is traced three times.
Destination unreachable	Indicates that no route to the destination is available.
*1 * * * 2 * * *	Indicates that the router is expecting some form of response from the destination, but the destination is not responding. The router waits 32 hops before timing out. Go to the ELS and turn on OSI CLNP messages to determine why the host is not responding.

Exit

Return to the previous prompt level.

Syntax: exit

Example: **exit**

11

Configuring DVMRP

This chapter describes how to configure DVMRP (Distance Vector Multicast Routing Protocol) using the DVMRP configuration commands.

For additional information about DVMRP, refer to the *Routing Protocols Reference Guide*.

Accessing the DVMRP Configuration Environment

For information about accessing the DVMRP configuration environment, see Chapter 1.

DVMRP Configuration Commands

This section explains the DVMRP configuration commands. The commands are entered at the `DVMRP Config>` prompt. To activate the commands, you must restart the router.

Table 11–1 DVMRP Configuration Commands Summary

Command	Function
? (Help)	Lists all of the DVMRP configuration commands or lists the options associated with specific commands.
DVMRP	Enables or disables DVMRP.
MOSPF	Sets the metric and threshold for the DVMRP interface running over MOSPF. This command also disables the MOSPF VIF.
Phyint	Sets the metric and threshold for LAN interfaces associated with DVMRP. This command also deletes LAN interfaces associated with DVMRP.
Tunnel	Adds or deletes tunnels in a MOSPF/DVMRP configuration.
List	Displays the current DVMRP configuration.
Exit	Exits the DVMRP configuration process and returns to the CONFIG environment.

? (Help)

List the available commands from the current prompt level. You can also enter a **?** after a specific command name to list its options.

Syntax: ?

Example: **?**

DVMRP

Enable or disable DVMRP on the bridging router.

Syntax: dvmrp on
off

on

Enables DVMRP on the bridging router. When enabled, DVMRP interfaces are automatically assigned to all LAN interfaces that are NOT running MOSPF.

Example: **dvmrp on**

off

Disables DVMRP on the router.

Example: **dvmrp on**

List

Display the current DVMRP configuration. The output displays the current DVMRP state (disabled or enabled), tunnel configuration information, and MOSPF configuration information.

Syntax: list

Example: **list**

```
DVMRP enabled
tunnel 0.0.0.0 0.0.0.0 1 1
MOSPF 1 1
```

MOSPF

Set the metric and threshold for the DVMRP interface running over MOSPF. This command also disables the MOSPF VIF.

Syntax: mospf *metric threshold*
delete

metric threshold

Sets the metric and threshold for the MOSPF VIF. Default values for the metric and threshold parameters are 1.

When using a MOSPF domain to join DVMRP tunnels, DVMRP is actually run over MOSPF. When this occurs, a DVMRP interface named "MOSPF VIF" (VIF or virtual interface) is automatically created. DVMRP tries to run over MOSPF automatically using the MOSPF VIF.

Example: **mospf 1 1**

delete

Disables the MOSPF VIF. When MOSPF is enabled, DVMRP tries to run over MOSPF automatically using the MOSPF VIF.

Example: **mospf delete**

Phyint

Set the metric and threshold for LAN interfaces associated with DVMRP. This command also deletes LAN interfaces associated with DVMRP.

Syntax: phyint *intrfc_address metric threshold*
intrfc_address delete

intfc_address metric threshold

Sets the metric and threshold for LAN interfaces (specified by the i*ntrfc_address* parameter) associated with DVMRP. Default values for the metric and threshold parameters are 1.

Example: **phyint XXXXX 1 1**

off

Deletes LAN interfaces associated with DVMRP.

Example: **phyint XXXXX delete**

Tunnel

Add tunnels or delete tunnels in a MOSPF/DVMRP configuration.

Syntax: tunnel *source–adr destination–addr metric threshold*
source–addr destination–addr delete

source–adr destination–addr metric threshold

Adds a tunnel to a MOSPF/DVMRP configuration.

Example: **tunnel XXX XXX 1 1**

source–addr destination–addr **delete**

Deletes a tunnel from a MOSPF/DVMRP configuration.

Example: **tunnel XXX XXX delete**

Exit

Return to the previous prompt level.

Syntax: exit

Example: **exit**

12

Monitoring DVMRP

This chapter describes how to monitor DVMRP protocol activity and how to use the DVMRP console commands.

For additional information about the DVMRP protocol, refer to the *Routing Protocols Reference Guide*.

Accessing the DVMRP Console Environment

For information about accessing the DVMRP console environment, see Chapter 1.

DVMRP Console Commands

The DVMRP console commands allow you to view the parameters and statistics of networks with enabled DVMRP.

Enter the DVMRP console commands at the `DVMRP>` prompt.

Table 12–1 DVMRP Console Command Summary

Command	Function
? (Help)	Lists all the DVMRP console commands or lists the options associated with specific commands.
Dump routing tables	Displays the OSPF routes contained in the routing table.
Interface summary	Displays OSPF interface statistics and parameters.
Join	Configures the router to belong to one or more multicast groups.
Leave	Removes the router from membership in multicast groups.
Mcache	Displays a list of currently active multicast forwarding cache entries.
Mgroups	Displays the group membership of the router's attached interfaces.
Mstats	Displays various multicast routing statistics.
Exit	Exits the DVMRP console process and returns to the GWCON environment.

? (Help)

List the commands that are available from the current prompt level. You can also enter a **?** after a specific command name to list its options.

Syntax: ?

Example: **?**

```
DUMP routing tables
INTERFACE summary
JOIN
LEAVE
MCACHE
MGROUPS
MSTATS
EXIT
```

Dump Routing Tables

Display the set of known DVMRP multicast sources. Each source is listed together with the DVMRP router it was learned from, an associated cost, and the number of seconds since the routing table entry was refreshed.

Syntax: dump

Example: **dump**

```
                Multicast Routing Table
Type   Origin-Subnet   From-Gateway    Metric  Age  In  Out-Vifs
DVMRP  18.26.0.0       192.35.82.97      10     30   1    0  2*
DVMRP  18.58.0.0       192.35.82.97       4     30   1    0  2*
DVMRP  18.85.0.0       192.35.82.97       4     30   1    0  2*
DVMRP  18.180.0.0      192.35.82.97       3     30   1    0  2*
DVMRP  36.8.0.0        192.35.82.97       9     30   1    0  2*
DVMRP  36.56.0.0       192.35.82.97       7     30   1    0  2*
DVMRP  36.103.0.0      192.35.82.97       9     30   1    0  2*
DVMRP  128.61.0.0      192.35.82.97       8     30   1    0  2*
DVMRP  128.89.0.0      192.35.82.97      10     30   1    0  2*
DVMRP  128.109.0.0     192.35.82.97       4     30   1    0  2*
DVMRP  128.119.0.0     192.35.82.97       4     30   1    0  2*
DVMRP  128.150.0.0     192.35.82.97       6     30   1    0  2*
```

Type	Displays the type of multicast sources (DVMRP)
Origin-Subnet	Displays the IP address of the originating subnet.
From-Gateway	Displays the IP address of the gateway from which the entry came.
Metric	Displays the associated cost of that route.
Age	Displays the age of routing table entry as the number of seconds since the routing table entry was refreshed.
In	Displays the DVMRP VIF that multicast datagram from the source must be received on.
Out-Vifs	Displays those VIFs that send the multicast datagrams. VIFs marked with an asterisk indicate that a datagram is only forwarded if there are group members on the attached network.

Interface Summary

Display current list of DVMRP interfaces (or VIFs).

Syntax: interface interface-ip-address

Example: **interface**

```
Virtual Interface Table
 Vif  Local-Address                          Metric  Thresh  Flags
  0   10.1.153.22      subnet: 10.1.153.0        1       1    querier
  1   10.1.154.22      subnet: 10.1.154.0        1       1    down
```

Vif	Displays the number assigned to DVMRP interfaces (or VIFs) command. Each VIF is assigned a number, which is used to identify the VIF in other commands
Local Address	Displays the local IP address of the DVMRP interface.
Flags	Displays whether the VIF is down or that the router is the querier (sender of IGMP Host Membership Queries) on the interface.

Join

Establish the router as a member of a multicast group.

This command is similar to the **join** command in the OSPF configuration console with two differences:

- The effect on group membership is immediate when the commands are given from the OSPF monitor (a restart/reload is not required).
- The command keeps track of the number of times a particular group is ‘‘joined.”

When the router is the member of a multicast group, it responds to pings and SNMP queries sent to the group address.

Syntax: join *multicast-group-address*

Example: **join 128.185.00.00**

Leave

Remove a router's membership in a multicast group. This prevents the router from responding to pings and SNMP queries sent to the group address.

This command is similar to the **leave** command in the OSPF configuration console with two differences:

- The effect on group membership is immediate when the commands are given from the OSPF monitor (a restart/reload is not required).
- The command does not delete group membership until the "leaves" executed equals the number of "joins" previously executed.

Syntax: leave *multicast-group-address*

Example: **leave 128.185.00.00**

Mcache

Display a list of currently active multicast cache entries. Multicast cache entries are built on demand, whenever the first matching multicast datagram is received. There is a separate cache entry (and therefore a separate route) for each datagram source network and destination group combination.

Cache entries are cleared on topology changes (for example, a point-to-point line in the MOSPF system going up or down), and on group membership changes.

Note: The numbers displayed in the legend at the top of the output do NOT refer directly to VIFs, but instead refer to physical interfaces (which may be running either DVMRP or MOSPF) and tunnels.

Syntax: mcache

Example: **mcache**

```
          0: Eth /0           1: Internal
          2: 128.185.246.17

     Source          Destination     Count   Upst    Downstream
     128.185.146.0   239.0.0.1       1       0       2,4
     128.119.0.0     224.2.199.198   9       4       3
     128.9.160.0     224.2.127.255   1       4       3
     13.2.116.0      224.2.0.1       27      4       3
     140.173.8.0     224.2.0.1       31      4       3
     128.165.114.0   224.2.0.1       25      4       3
     132.160.3.0     224.2.158.99    11      4       3
     132.160.3.0     224.2.170.143   56      4       3
     128.167.254.0   224.2.199.198   27      4       3
     129.240.200.0   224.2.0.1       21      4       3
     131.188.34.0    224.2.0.1       28      4       3
     131.188.34.0    224.2.199.198   28      4       3
```

Source	Source network/subnet of matching datagrams.
Destination	Destination group of matching datagrams.
Count	Displays the number of entries processed for that multicast group.
Upstream	Displays the neighboring network/router from which the datagram must be received in order to be forwarded. When this reads as “none,” the datagram is never forwarded.
Downstream	Displays the total number of downstream interfaces/neighbors to which the datagram is forwarded. When this is 0, the datagram is not forwarded.

There is more information in a multicast forwarding cache entry. A cache entry can be displayed in detail by providing the source and destination of a matching datagram on the command line. If a matching cache entry is not found, one is built. A sample of this command is shown below:

Example: **mcache 128.185.182.9 224.0.1.2**

```
source Net:     128.185.182.0
Destination:    224.0.1.2
Use Count:      472
Upstream Type:  Transit Net
Upstream ID:    128.185.184.114
Downstream:     128.185.177.11 (TTL = 2)
```

In addition to the information shown in the short form of the **mcache** command, the following fields are displayed:

Upstream Type	Indicates the type of node from which the datagram must be received to be forwarded. Possible values for this field are “none” (indicating that the datagram is not forwarded), “router” (indicating that the datagram must be received over a point-to-point connection), “transit network,” “stub network,” and “external” (indicating that the datagram is expected to be received from another Autonomous System).
Downstream	Prints a separate line for each interface or neighbor to which the datagram is sent. A TTL value is also given, indicating that datagrams forwarded out of or to this interface must have at least the specified TTL value in their IP header. When the router is itself a member of the multicast group, a line specifying “internal Application” appears as one of the downstream interfaces/neighbors.

Mgroups

Display the group membership of the router’s attached interfaces. Displays only the group membership for those interfaces on which the router is either the designated router or the backup designated router.

Syntax: mgroups

Example: **mgroups**

```
               Local Group Database

 Group             Interface                Lifetime (secs)
224.0.1.1        128.185.184.11 (Eth /1)       176
224.0.1.2        128.185.184.11 (Eth /1)       170
224.1.1.1        Internal                       1
```

Group	Displays the group address as it was reported (through IGMP) on a particular interface.
Interface	Displays the interface address to which the group address was reported (through IGMP).

	The router's internal group membership is indicated by a value of "internal." For these entries, the lifetime field (see below) indicates the number of applications that have requested membership in the particular group.
Lifetime	Displays the number of seconds that the entry persists if Membership Reports cease to be heard on the interface for the given group.

Mstat

Display various multicast routing statistics. The command indicates whether multicast routing is enabled and whether the router is an inter-area and/or inter-AS multicast forwarder.

Syntax: mstats

Example: **mstats**

```
Multicast forwarding:   Enabled
Inter–area forwarding:  Enabled
Inter–AS forwarding:    Enabled

 Datagrams received:      164612  Datagrams (ext source):        0
 Datagrams fwd (multicast):98807  Datagrams fwd (unicast):       0
 Locally delivered:            0  No matching rcv interface:     0
 Unreachable source:           0  Unallocated cache entries:     0
 Off multicast tree:       77230  Unexpected DL multicast:       0
 Buffer alloc failure:         0  TTL scoping:                   0

 # fwd cache alloc:          649  # fwd cache freed:           648
 # fwd cache GC:               0  # local group DB alloc:        2
 # local group DB free:        2
```

Multicast forwarding	Displays whether the router forwards IP multicast datagrams.
Inter-area multicast	Displays whether the router forwards IP multicast datagrams between areas.

Inter-AS multicast	Displays whether the router forwards IP multicast datagrams between Autonomous Systems.
Datagrams received	Displays the number of multicast datagrams received by the router (datagrams whose destination group lies in the range 224.0.0.1 – 224.0.0.255 are not included in this total).
Datagrams (ext source)	Displays the number of datagrams that were received whose source is outside the AS.
Datagrams fwd (multicast)	Displays the number of datagrams that were forwarded as data-link multicasts. (This includes packet replications, so this count can be greater than the number received.)
Datagrams fwd (unicast)	Displays the number of datagrams that were forwarded as data-link unicasts.
Locally delivered	Displays the number of datagrams that were forwarded to internal applications.
No matching rcv interface	Displays the count of those datagrams that were received by a non-inter-AS multicast forwarder on a non-MOSPF interface.
Unreachable source	Displays a count of those datagrams whose source address was unreachable.
Unallocated cache entries	Displays a count of those datagrams whose cache entries were not created due to resource shortages.
Off multicast tree	Displays a count of those datagrams that were not forwarded, either because there was no upstream neighbor or no downstream interfaces/neighbors in the matching cache entry.
Unexpected DL multicast	Displays a count of those datagrams that were received as data-link multicasts on those interfaces that were configured for data-link unicast.
Buffer alloc failure	Displays a count of those datagrams that were not replicated because of buffer shortages.

TTL scoping	Indicates those datagrams that were not forwarded because their TTL indicated that they were unable to reach a group member.
# fwd cache alloc	Indicates the number of cache entries allocated. The current forwarding cache size is the number of entries allocated (“# fwd cache alloc”) minus the number of cache entries freed (“# fwd cache freed”).
# fwd cache freed	Indicates the number of cache entries freed. The current forwarding cache size is the number of entries allocated (“# fwd cache alloc”) minus the number of cache entries freed (“# fwd cache freed”).
# fwd cache GC	Indicates the number of cache entries were cleared because they were not recently used and the cache overflowed.
# local group DB alloc	Indicates the number of local group database entries allocated. The number allocated (“# local group DB alloc”) minus the number freed (“# local group DB free”) equals the current size of the local group database.
# local group DB free	Indicates the number of local group database entries freed. The number allocated (“# local group DB alloc”) minus the number freed (”# local group DB free”) equals the current size of the local group database.

The number of cache hits can be calculated as the number of datagrams received (“Datagrams received”) minus the total of datagrams discarded due to “No matching rcv interface,” “Unreachable source” and “Unallocated cache entries,” and minus “# local group DB alloc.” The number of cache misses is simply “# local group DB alloc.”

Exit

Return to the previous prompt level.

Syntax: exit

Example: **exit**

13

Configuring IP

This chapter describes how to configure the IP protocol and how to use the IP configuration commands.

For more information about IP, refer to the *Routing Protocols Reference Guide*.

Accessing the IP Configuration Environment

For information about accessing the IP configuration environment, see Chapter 1.

Basic Configuration Procedures

This section outlines the initial steps required to get the IP protocol up and running. Details on making further configuration changes are covered in the command sections of this chapter. The following list outlines the initial configuration tasks to bring up IP on the router. After completing these tasks, you must restart the router for the new configuration to take effect.

1. Access the IP configuration environment. (See Chapter 1.)
2. Assign IP addresses to hardware interfaces.
3. Enable dynamic routing.
4. Add static routing information (if necessary).
5. Enable ARP subnet routing (if necessary).
6. Set up IP access control.

7. Exit the IP configuration process.
8. Restart the router to activate the configuration changes.

The following sections discuss each configuration task in more detail.

Assigning IP Addresses to Network Interfaces

Use the IP configuration **add address** command to assign IP addresses to the network hardware interfaces. The arguments for this command include the hardware interface number (obtained from the `Config>` **list devices** command), and the IP address and its associated address mask.

In the following example, network interface 2 was assigned the address 128.185.123.22 with the associated address mask 255.255.255.0 (using the third byte for subnetting).

```
IP Config> add address 2 128.185.123.22 255.255.255.0
```

IP automatically becomes enabled whenever you assign at least one IP address to any of the router's hardware interfaces. A hardware interface does not accept or send IP packets unless it has at least one IP address.

IP allows you to use a serial line interface for IP traffic without assigning an IP address to the line. However, you must still assign each serial line a label. Use the **add address** command to assign the serial line an address of the form 0.0.0.n, where *n* is the hardware interface number (again obtained from the `Config>` **list devices** command). This address format tells the router that the interface in question is an *unnumbered serial line.* Refer to Chapter 2 of the *Routing Protocols Reference Guide* for information about the limitations on unnumbered serial lines.

To enable IP on serial-line interface number 2 to the router without assigning the interface an IP address, use the following command:

```
IP Config> add address 2 0.0.0.2
```

Enabling Dynamic Routing

Use the following procedures to enable dynamic routing on the router. The routers support OSPF and RIP for Interior Gateway Protocols as well as EGP (Exterior Gateway Protocol).

All three routing protocols can run simultaneously. However, most routers run only a single routing protocol (one of the IGPs). The OSPF protocol is recommended because it is robust and supports additional IP features (such as equal-cost multipath and variable-length subnets).

Enabling the OSPF Protocol

The OSPF routing protocol is enabled on an interface-by-interface basis. Each OSPF interface is assigned a cost. Also, an estimate of the OSPF database's size must be given, and the interaction between OSPF and the other two routing protocols (RIP and EGP) defined. Use the following procedures to initially configure OSPF.

OSPF configuration is done through its own configuration console (entered through the `Config>` **protocol ospf** command). To enable OSPF, use the following command:

```
OSPF Config> enable OSPF
```

After enabling the OSPF protocol, you are prompted for size estimates for the OSPF link state database. This gives the router some idea how much memory must be reserved for OSPF. You must supply the following two values that are used to estimate the size of the OSPF link state database:

- Total number of AS external routes imported into the OSPF routing domain. A single destination may lead to multiple AS external routes when imported by separate AS boundary routers. For example, if the OSPF routing domain has two AS boundary routers, both importing routes to the same 100 destinations, the number of AS external routes is set to 200.

- Total number of OSPF routers in the routing domain.

Enter these values at the following prompts (sample values are provided):

```
OSPF Config> enable ospf
Estimated # external routes[0]? 200
Estimated # OSPF routers [0]? 60
```

Next, configure each IP-interface that is to participate in OSPF routing. To configure an IP interface for OSPF, use the following command:

```
OSPF Config> set interface
```

You are prompted to enter a series of operating parameters. Each interface is assigned a cost as well as a list of OSPF operating parameters.

When running other IP routing protocols besides OSPF, you may want to enable the exchange of routes between OSPF and the other protocols. To do this, use the following command:

```
OSPF Config> enable AS-boundary-routing
```

For more information about the OSPF configuration process, see Chapter 17.

Enabling the RIP Protocol

This section describes how to initially configure the RIP protocol. When configuring the RIP protocol, you can specify which set of routes the router advertises and/or accepts on each IP interface. You can also specify how RIP information effects static routing and the interaction between RIP and EGP. Since RIP uses broadcast messages for its routing updates, the format of the IP broadcast address must also be specified when using the RIP protocol.

First, enable the RIP protocol with the following command:

```
IP Config> enable RIP
```

By default, RIP advertises all network and subnet routes out all interfaces of the router. Once RIP is enabled, you can configure what it listens to and what it advertises by setting the various RIP flags. For detailed information about the RIP flags consult the RIP description in the IP section of the *Routing Protocols Reference Guide*. These flags are configured on a per-IP-interface basis. The following commands can enable or disable the various flags.

```
IP Config> enable/disable sending net-routes
IP Config> enable/disable sending subnet-routes
IP Config> enable/disable sending static-routes
IP Config> enable/disable sending default-routes
IP Config> enable/disable receiving rip
IP Config> enable/disable receiving dynamic nets
IP Config> enable/disable receiving dynamic subnets
IP Config> enable/disable override default
IP Config> enable/disable override static-routes
```

The RIP protocol uses IP broadcast when sending its routing updates. Since there are different formats of IP broadcast in use, you must specify which broadcast format to use. The IP broadcast format is specified on a per-interface basis by using the following command:

```
IP Config> set broadcast-address IP-interface-address
Use a NET or LOCAL-WIRE style address [NETWORK]?
Fill pattern for wildcard part of address (0 or 1) [0]?
```

From the prompts, choose either the LOCAL-WIRE or NETWORK broadcast format and then select whether you want the rest of the broadcast address filled with either ones or zeroes. For more information about the RIP protocol see the RIP description in the IP section of the *Routing Protocols Reference Guide*.

Enabling the EGP Protocol

This section describes how to initially configure the EGP protocol. Your router may need to run the EGP protocol if it is exchanging reachability information with routers belonging to other autonomous systems. For example, this may be the case if you have a MILnet/NSF backbone network connection.

Only routers that lie on the boundary of the Autonomous System can run the EGP protocol. To enable EGP on your router, configure the following:

- An autonomous system number for the router running EGP.
- A list of initial EGP neighbors.
- The exchange of routing information between EGP and the IGPs.

To enable the EGP protocol and configure the autonomous system number for your router, use the command shown in the following example:

```
IP config> enable EGP
EGP autonomous system number [0]? 47
```

You must assign the same AS number to all routers belonging to the same Autonomous System.

Next, configure the list of initial routers with which you want to exchange EGP information. These routers are called EGP neighbors and belong to different Autonomous Systems. Use the **add EGP-neighbor** command to configure each EGP neighbor. With this command you specify the IP address of the neighbor as well as the Autonomous System to which it belongs. For example:

```
IP Config> add EGP-neighbor 192.9.1.1
AS id [1]? 32
```

In this example, the EGP neighbor's IP address is 192.9.1.1 and the neighbor belongs to Autonomous System number 32.

After configuring your EGP neighbors, configure the set of routes that you want to exchange with these neighbors. The route exchange is defined in two directions. In the **out** direction, you specify which routes you want to advertise through EGP. In the **in** direction, you specify which received EGP routes you want to readvertise through your IGP(s) (OSPF and/or RIP).

The EGP routing exchange can be defined on a per-neighboring-AS basis. If two of your EGP neighbors belong to separate ASs, then you can exchange separate sets of routes with each neighbor. To describe the set of routes to exchange with a neighboring Autonomous System, use the **add-EGP-AS-info** command (followed by the neighbor's AS number). After entering the command, you are prompted to select the direction you want the exchange of routes to follow.

In the following example, the interchange direction (flag) is **in.** This means that all routes received from the neighboring AS (here shown as AS number 32) are readvertised by OSPF and RIP. In this case, the user is also prompted for the metric (shown as 10) that is used (by OSPF and RIP) when readvertising the routes:

```
IP config> add EGP-AS-info 32
Interchange Flag (IN/OUT/OFF)- [OFF]? in
Default metric for IGP (-1 use EGP) [-1]? 10
```

If routes are not to be exchanged freely in one (or both) directions, those directions are table driven. When the interchange flag is not set to **out**, the Output Exchange Table lists all those routes that are advertised through EGP. Similarly, when the interchange flag is not set to **in**, the Input Exchange Table lists the received routes that you readvertise through OSPF and/or RIP.

You can configure both the Output and Input Exchange Tables on a per-neighboring-AS basis. Both tables consist of lists of IP networks: to add or delete an IP network to or from one of the tables, use one of the following commands (followed by the desired AS number):

```
IP config> add/delete input-interchange
IP config> add/delete output-interchange
```

Each network is added to an Input/Output Exchange Table together with the route cost that is advertised. In the following example, the command specifies that if network 18.0.0.0 is received by EGP from Autonomous System 32, it readvertises the IGPs with a cost of 2.

```
IP config> add input-interchange 32
Destination network [0.0.0.0]? 18.0.0.0
Metric to advertise (-1 use EGP) [-1]? 2
```

Entries in the Output Exchange Table can specify that a route is to be advertised through EGP only if it was originally received from a particular Autonomous System. In the next example, the command specifies that a route to network 10.0.0.0 is advertised (through EGP) to AS 32, but only if the route was originally received from AS number 50. In this case, the route is advertised by EGP with a cost of 3.

```
IP config> add output-interchange 32
Source AS id (0 for don't care) [0]? 50
Destination network (0.0.0.0 for all) [0.0.0.0]? 10.0.0.0
Metric to advertise (-1 use IGP) [-1]? 3
```

More detailed information about the EGP route exchange can be found in the configuration command section of this chapter.

Using EGP Routers as Defaults

In EGP environments, the EGP router is usually the authoritative router since it has the knowledge of the routers and networks in other Autonomous Systems.

You can configure a router running EGP to advertise itself as the default router through its IGPs (OSPF and RIP). This is called originating default. When this feature is enabled, the router advertises itself as the default only if it has EGP-derived routers in its IP routing table.

Use the following command to enable this feature within OSPF:

```
OSPF Config> enable/disable AS-boundary-routing
```

For more details, see the configuration commands in Chapter 17.

To enable this feature within RIP, use the following commands:

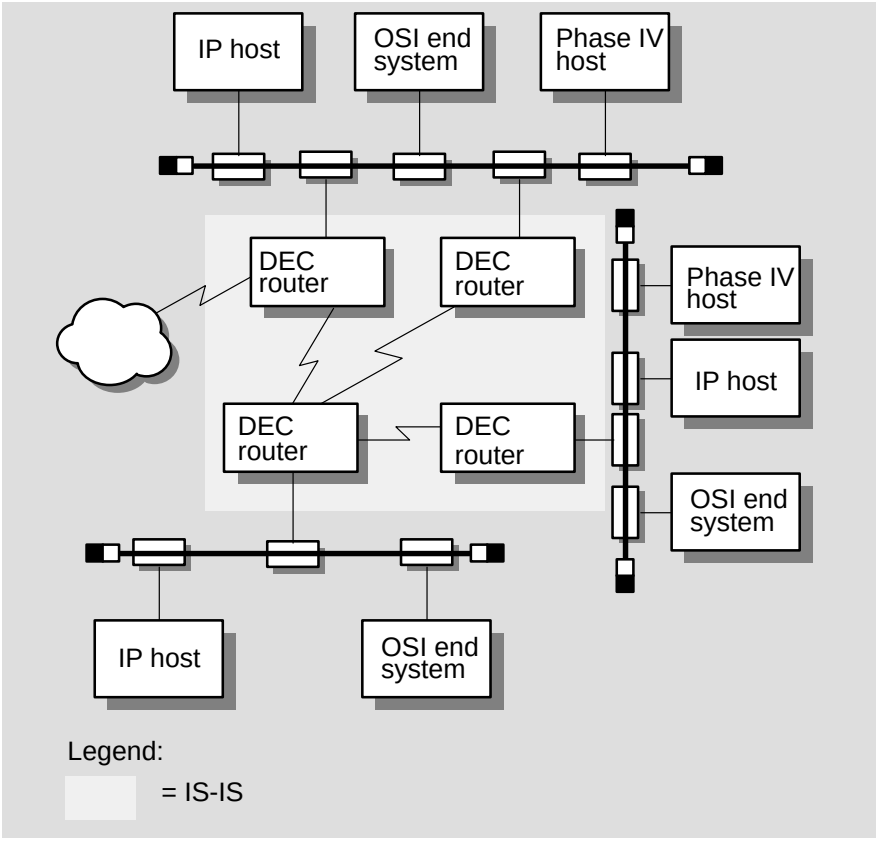
```
IP Config> enable/disable originate-default
IP Config> set advertised default-metric
IP Config> enable/disable sending default
```

For more details, see the configuration command section of this chapter.

Using the IS-IS Protocol in a Combined DECnet and IP Network

Figure 13–1 shows an example configuration with IP routing. In this case, there is no need to accommodate either the RIP protocol or static IP routes. Therefore, use the IS-IS protocol. Note that you cannot use the IS-IS protocol on any level 1-only routing circuit if the ROUTING MANUAL L1 ALGORITHM is ROUTING VECTOR.

Figure 13–1 Using IS-IS in an IP Configuration



LKG-5454-91I

To add the IS-IS protocol to exchange IP routing information as shown in Figure 13–1:

1. Run link state routing (DECnet Phase V) at level 2 and level 1. A router that runs link state routing at level 1 and/or level 2 uses the IS-IS protocol at that level.

2. Assign IP addresses and masks to the interfaces indicated in the diagram.

   For a point-to-point link, do not assign addresses. However, assign IP addresses over all the circuits which you expect to carry IP packets because this helps you diagnose problems.

3. At the OSI config> prompt, enable the integrated IS–IS protocol by entering the command **enable integrated–isis**, and answer the questions about route configuration.

Adding Static Routing Information

This procedure is necessary only if you cannot gain routing information from any of the above dynamic routing protocols. Static routing persists over power failures and is used for routes that never change or are not able to be learned dynamically. Static routing information consists of any of the following items:

- **Default Gateway** – Packets are routed to default (authoritative) gateways when the packet destination cannot be found in the routing table.

- **Default Subnet Gateways** – If you are using subnetted networks, you can define a separate default gateway for each subnetted network.

- **Static Network/Subnet Routes** – For each destination with a fixed route, configure the next hop and distance to the destination.

Default Gateway

Routers send packets having unknown destinations (destinations not present in the routing table) toward the default gateway. A default gateway is configured in the router by specifying the next hop to use to get to the default gateway and the cost of sending packets to the default gateway.

In the following example, the next hop toward the default gateway is 192.9.1.4 and the cost of sending a packet to the default gateway is 5.

```
IP Config> set default network-gateway
Default gateway [0.0.0.0]? 192.9.1.4
gateway's cost [0]? 5
```

Default gateways can be learned and advertised by both the OSPF and RIP protocol. For the OSPF protocol, a router can be configured to advertise itself as the default gateway with the following OSPF command:

```
OSPF Config> enable/disable AS-boundary-routing
```

The RIP protocol can be configured so that it advertises knowledge of the default gateway (if it has any) to its neighbors. RIP can also be configured so that a learned default gateway overrides (or does not override) a statically configured default gateway. These configuration tasks are accomplished with the following two commands:

```
IP Config> enable/disable sending default-routes
IP Config> enable/disable override default
```

Finally, a router that runs EGP can be configured to advertise itself (through the OSPF and RIP protocol) as the default gateway whenever it has EGP-learned routes in its routing tables. For OSPF, this is accomplished through the OSPF **enable/disable AS-boundary-routing** command. For RIP, the following commands are used:

```
IP Config> enable originate-default
IP Config> set advertised default-metric
```

Default Subnet Gateways

There can be a default subnet gateway configured for each subnetted network that the router knows about. When the router attempts to forward a packet to a destination belonging to the subnetted network, but that destination cannot be found in the routing table, the packet is forwarded instead to the default subnet gateway.

Configuring default subnet gateways is the same as configuring the preceding default network gateway. The only difference is that you must specify the subnetted network on the command line. For example, to create a default subnet gateway for the subnetted network 18.0.0.0, you can use the following command:

```
IP Config> set default subnet-gateway
Default gateway [0.0.0.0]? 128.185.123.22
gateway's cost [0]? 2
```

The above example specifies that the next hop to the subnet default gateway is 128.185.123.22, and that the cost of routing a packet to the default subnet gateway is 2.

Static Network/Subnet Routes

Configure static routes for those destinations that cannot be discovered by the dynamic routing protocols. The destination is described by an IP network/subnet number (**dest-addr**) and the destination's address mask (**mask**). The route to the destination is described by the IP address of the first hop router to use (**1st-hop**) and the cost of routing a packet to the destination (**cost**). To create/modify/delete a static route, use the commands:

```
IP Config> add route dest-addr mask 1st-hop cost
IP Config> change route dest-addr mask 1st-hop cost
IP Config> delete route dest-addr mask
```

Routes dynamically learned through the OSPF and RIP protocols can override static routes. For the RIP protocol, you can disable this override behavior. See the RIP section of this chapter concerning the **enable/disable override static-routes** commands.

Enabling ARP Subnet Routing

If there are hosts on attached subnetted networks that do not support IP subnetting, use Address Resolution Protocol (ARP) subnetting routing (described in RFC 1027). When the router is configured for ARP subnet routing, it replies by proxy to ARP requests for destination (off the LAN if the router is the best route to the destination). For proper operation, all routers attached to a LAN containing subnetting-ignorant hosts are configured for ARP subnet routing.

To enable ARP subnet routing, use the following command:

```
IP Config> enable ARP-subnet-routing
```

Enabling RFC 925 ARP Subnet Routing

Some IP hosts ARP for all destinations, whether or not they are attached to the local network segment. For these hosts, ARP subnet routing is not enough and you must see the proxy ARP functionality specified in RFC 925 instead. RFC 925 ARP routing is a subset of ARP subnet routing.

To enable RFC 925 ARP routing, use the following command:

```
IP Config> enable RFC-925
```

Setting Up IP Access Control

The IP access control system allows the IP forwarder to control packet forwarding based on source and destination IP addresses, IP protocol number, and by port number for the TCP and UDP protocols. This can control access to particular classes of IP address and services.

The IP access control system is based on one global ordered list of inclusive and exclusive access control entries. If access control is enabled, each IP packet being originated, forwarded, or received, is subject to the access control list. Each entry in the list may be inclusive or exclusive, permitting or denying forwarding. Each entry has fields for source and destination IP address, optional IP protocol number, and optional port number for UDP and TCP.

For each received packet, the headers are compared to all specified fields in each entry in the list in turn. If the entry matches the packet and the entry is inclusive, the packet is forwarded. If the entry is exclusive, the packet is dropped. If no entry matches after going through the entry list, the packet is dropped.

Each entry has an IP address mask and result pair for both the source and destination IP address. An address is logically “AND-ed” with the mask, and compared to the result. For example, a mask of 255.0.0.0 with a result of 26.0.0.0 matches any address with 26 in the first byte. A mask of 255.255.255.255 with a result 192.67.67.20 matches only the IP host 192.67.67.20. A mask of 0.0.0.0 with a result of 0.0.0.0 is a wildcard, and matches any IP address.

Each entry may also have an optional IP protocol number range. This applies to the protocol byte in the IP header. Any IP packet with a protocol value within the specified range matches. A range of 0 to 255 matches all IP packets. The commonly used protocol numbers are 1 for ICMP, 6 for TCP, 8 for EGP, 17 for UDP, and 89 for OSPF.

Each entry may also have an optional port number range. This applies only to TCP and UDP packets, since the port number is part of the TCP and UDP headers. Any TCP or UDP packet with a destination port number within the specified range matches. (TCP and UDP use the same port numbers.) A range of 0 to 65535 disables port filtering. Some commonly used port numbers are 21 for FTP, 23 for Telnet, 25 for SMTP, 513 for rlogin, 520 for RIP, and 6000 for X.

The following example allows any host to send packets to the SMTP TCP socket on 192.67.67.20.

```
IP Config> add access-control inclusive 0.0.0.0 0.0.0.0
           192.67.67.20 255.255.255.255 6 6 25 25
```

The next example prevents any host on subnet 1 of Class B network 150.150.0.0 from sending packets to hosts on subnet 2 of Class B network 150.150.0.0. (assuming a 1-byte subnet mask).

```
IP Config> add access-control exclusive 150.150.1.0
           255.255.255.0 150.150.2.0 255.255.255.0 0 255 0 65535
```

This command allows the router to send and receive all RIP packets.

```
IP Config> add access-control inclusive 0.0.0.0 0.0.0.0 0.0.0.0
           0.0.0.0 17 17 520 520
```

This command allows the router to send and receive all OSPF packets.

```
IP Config> add access-control inclusive 0.0.0.0 0.0.0.0
           0.0.0.0 0.0.0.0 89 89
```

If IP access control is enabled, you must be careful with packets that the router originates and receives. Be sure not to filter out the RIP or OSPF packets being sent or received by the router. The easiest way to do this is to add a wildcard inclusive entry as the last in the access control list. Alternately, you can add specific entries for RIP and/or OSPF, perhaps with restrictive addresses and masks. Note that some OSPF packets are sent to the Class D multicast addresses 224.0.0.5 and 224.0.0.6, which is important if address checking is being done for routing protocols. See the **add** command section in this chapter for more information about access control.

If you have certain IP networks/subnets that you do not want to forward packets to, nor distribute routing information about, it is best to specify those networks as filters (this is more efficient than the access control mechanism). To add a network filter, use the following command:

```
IP Config> add filter IP-address IP-mask
```

It is recommended that you filter to local loopback network 127.0.0.0 so as not to propagate packets destined as a loopback. Use the following command:

```
IP Config> add filter 127.0.0.0 255.0.0.0
```

This is the end of the steps for setting up IP. You can now operate your router as an internet router between two or more IP networks.

The BOOTP Forwarding Process

BOOTP (documented in RFC 951) is a bootstrap protocol used by a diskless workstation to learn its IP address and the location of its boot file and boot server. BOOTP requests/replies are forwarded at the application level (UDP). They are not forwarded at the network level. This means that the IP header changes as the packet is forwarded. The workstation broadcasts the request in a UDP packet to the routers, and in turn, the routers forward the packets to BOOTP servers.

The following terms are useful when discussing the BOOTP forwarding process:

- BOOTP client – the diskless workstation
- BOOTP servers – the boot host (with UNIX daemon bootpd or DOS version available from FTP software)
- BOOTP relay agent or BOOTP forwarder – your router

The following steps outline an example of the BOOTP forwarding process:

1. The BOOTP client copies its Ethernet address (or appropriate MAC address) into a BOOTP packet and broadcasts it onto the local LAN. BOOTP is running on top of UDP.
2. The local BOOTP relay agent receives the packet and checks to see if the packet is formatted correctly and that the maximum number of application hops did not expire. It also checks to see if the client tried long enough.

Note: If multiple hops are required before reaching the BOOTP agent, the packet is routed normally through IP. All other routers do not examine the packet to determine whether it is a BOOTP packet.

If this is the case:

3. The Local BOOTP agent forwards a separate BOOTP request to each of its configured BOOTP servers. The BOOTP request is the same as the one that was initially sent by the client except that it has a new IP header with the relay agent's IP address copied into the body of the BOOTP request.

4. The BOOTP server receives the request and looks up the client's Ethernet address in its database. If found, it formats a BOOTP reply containing the client's IP address and boot file name. The reply is then sent to the BOOTP relay agent.

5. The BOOTP relay agent receives the reply and makes an entry in its ARP table for the client and then forwards the reply to the station.

6. The station then continues to boot using TFTP, using the information in the BOOTP reply packet.

Enabling/Disabling BOOTP Forwarding

To enable or disable BOOTP forwarding on the router, enter the following command line at the IP configuration prompt.

```
IP Config> enable/disable bootp
```

When enabling BOOTP, you are prompted for the following values:

- Maximum number of application hops you want the BOOTP request to go. This is the maximum number of BOOTP relay agents that can forward the packet. This is NOT the maximum number of IP hops to the BOOTP server. A typical value for this parameter is 1.

- Number of seconds you want the client to retry before the BOOTP request is forwarded. A typical value for this parameter is 0.

Note: This parameter is not commonly used.

After accepting a BOOTP request, the router forwards the BOOTP request to each BOOTP server. If there are multiple servers configured for BOOTP, the transmitting server replicates the packet.

Configuring a BOOTP Server

To add a BOOTP server to the router's configuration, enter the following command at the IP configuration prompt:

```
IP Config> add BOOTP-SERVER [IP address of server]
```

Multiple servers can be configured. In addition, if only the network number of the server is known or if multiple servers reside on the same network segment, a broadcast address can be configured for the server.

IP Configuration Commands

This section summarizes and then explains all IP configuration commands. These commands allow you to modify the IP protocol behavior to meet your specific requirements. Some amount of configuration is necessary to produce a fully functional IP router. Enter IP configuration commands at the `IP config>` prompt.

Table 13–1 IP Configuration Command Summary

Command	Function
? (Help)	Lists the configuration commands or lists the actions associated with specific commands.
Add	Adds to the IP configuration information. Interface addresses can be added, along with access controls, filters, EGP exchange information and EGP neighbors.
Change	Modifies information that was originally entered with the **add** command.
Delete	Deletes IP configuration information that were entered with the **add** command.
Disable	Disables certain IP features that were turned on by the **enable** command.
Enable	Enables IP features such as ARP subnet routing, EGP, originate default, directed broadcasts, BOOTP, and the various RIP flags controlling the sending and receiving of RIP information.
List	Displays IP configuration items.
Move	Changes the order of access control records.
Set	Establishes IP configuration modes such as the type of access control and the format of broadcast addresses. Also sets IP parameters such as default routers and the size of the IP routing table.
Update	Updates information that was originally entered with the **add** command.
Exit	Exits the IP configuration process.

? (Help)

List the commands that are available from the current prompt level. You can also enter a **?** after a specific command name to list its options.

Syntax: ?

Example: **?**

```
LIST
CHANGE
DELETE
DISABLE
ENABLE
ADD
SET
MOVE
UPDATE
EXIT
```

Example: **add ?**

```
ACCEPT-RIP-ROUTE
ACCESS-CONTROL
ADDRESS
BOOTP-SERVER
EGP-AS-INFO
EGP-NEIGHBOR
FILTER
INPUT-INTERCHANGE
OUTPUT-INTERCHANGE
PACKET–FILTER
ROUTE
```

Add

Add IP information to your configuration. This command lets you add interface addresses, access controls, filters, EGP exchange information, and EGP neighbors.

Syntax: add accept-rip-route . . .
access-control . . .
address . . .
bootp-server
egp-as-info . . .
egp-neighbor . . .
filter . . .
input-interchange . . .
output-interchange . . .
packet–filter . . .
route . . .

accept-rip-route ***IP-network/subnet***

Allows an interface to accept a RIP route when input filtering is enabled for an interface. You can prompt the list of networks/subnets that are already entered using the **list rip-routes-accept** command. You can enable the input filtering of RIP routes on a per-IP-interface basis. This is done separately for network-level routes (for example, a route to 10.0.0.0) and for subnet-level routes (for example, a route to 128.185.0.0). To enable input filtering of network-level routes on an IP interface, use the **disable dynamic nets** command. To enable input filtering of subnet-level routes, use the **disable dynamic subnets** command.

Example: **add accept-rip-route 10.0.0.0**

access-control ***type IP-source source-mask IP-dest dest-mask [first-protocol last-protocol] [first-port last- port]***

Adds an access control entry to the end of the access control list. This allows you to describe a class of packets to forward or drop, depending on the type of the entry.

The length and order of the IP access control list can affect the performance of the IP forwarder.

This command adds an IP access control entry to the end of the list. Each entry must be assigned the following: type, IP source, source-mask, IP destination, and destination-mask fields. The *type* must either be inclusive or exclusive. The *IP-source* and *IP-dest* fields are in the form of IP addresses in dotted decimal notation. Optionally, you may specify an IP protocol number range with the *first-protocol* and *last-protocol* fields, which are an inclusive range of IP protocols that match this entry. If a range of protocols was specified, you may specify a TCP and UDP port number range with the *first-port* and *last-port fields*, which are an inclusive range of TCP and UDP ports that matches this entry.

Note: Before access-control can become effective, you must enable it with the **set access-control on** command.

```
Example: add access-control inclusive 0.0.0.0 0.0.0.0
         192.67.67.20         255.255.255.255 6 6 25 25
```

address ***interface-number IP-address address-mask***

Assigns an IP address to one of the router's hardware network interfaces. A hardware network interface does not receive or transmit IP packets until it has at least one IP address.

You must specify an IP address together with its subnet mask. For example, if the address is on a class B network, using the third byte for subnetting, the mask is 255.255.255.0. Use the **list devices** command to obtain the appropriate command *interface-number*. Serial lines do not need addresses. Such lines are called *unnumbered*. However, you must still enable them for IP traffic using the **add address** command. The address then used is 0.0.0.*n*, where *n* is the *interface-number.*

```
Example: add address 0 128.185.123.22 255.255.255.0
```

bootp-server ***server-IP-address***

Adds a BOOTP server to a network configuration. Acting as a boot relay agent, your router accepts and forwards BOOTP requests to the BOOTP server. BOOTP is a bootstrap protocol used by a diskless workstation to learn its IP address and the location of its boot file and boot server.

Note: Before the **list all** command can display the bootp server address, you must enable bootp forwarding with the **enable bootp-forwarding** command.

Example: **add bootp-server 128.185.123.22**

egp-neighbor ***neighbor-IP-address neighbor-AS***

Adds an initial EGP neighbor to the router's IP configuration. If the EGP protocol is enabled, it attempts to contact each router configured as an EGP neighbor.

Each EGP neighbor is configured together with the neighbor's Autonomous System number. The neighbor's AS number is then verified before an EGP connection to the neighbor is fully established. The set of routes that are exchanged with the EGP neighbor is configured on a neighboring-AS basis. If two of the router's EGP neighbors belong to the same neighboring AS, the same set of routes is exchanged with each neighbor.

For more information about the route exchange, see the **add egp-as-info** command. The router can form EGP neighbor relationships with neighbors that have not been configured with the **add egp-neighbor** command. In this case, the neighbor's AS membership cannot be verified, and the set of routes exchanged with such a neighbor is governed by the ''default" EGP route exchange (the exchange configured for the special AS number 0).

Example: **add egp-neighbor 10.0.0.7 1**

egp-as-info ***neighboring-AS interchange-flag [default-metric]***

Defines the type of EGP route exchange that takes place when communicating with a neighboring Autonomous System.

The EGP route exchange is defined in a bidirectional fashion. One direction, called **in**, defines the set of routes that, when received from the neighboring AS, are readvertised through OSPF and RIP. The other direction, called **out**, defines the set of routes that is advertised through EGP to the neighboring AS.

The configured *interchange-flag* indicates if EGP route exchange is free (all routes exchanged) in a particular direction. The *interchange-flag*'s permissible values are **in**,**out**, and **off**.

If the exchange is free in one direction, you can configure a default metric (*default-metric*) for the (re)advertisements. In the **in** direction, this means that OSPF and RIP readvertises all routes with the given cost. In the **out** direction, this means that all routes in the routing table are advertised with the given cost. Setting the default cost to –1 means that the readvertisement or advertisement does not change the metrics: A cost of –2 means that the specified network is not added to the routing table, and a cost –3 places the network in the routing table, but does advertise it to other routers.

When EGP route exchange is not free in a particular direction, it is table driven. The table for the **in** direction is called the Input Exchange Table. The table for the **out** direction is called the Output Exchange Table. See the commands **add input-interchange** and **add output-interchange** for information about updating these tables.

Even when the *interchange-flag* is set to **out**, EGP routes that were originally received from a particular AS are not be readvertised back to that AS.

When the EGP route exchange is set to be free in one direction, the entries in the appropriate Input/Output Exchange Table override the *default-metric* on a per-route basis.

When the *neighboring-AS* is specified as 0, the default EGP exchange is defined. This controls EGP route exchange with all neighboring ASs that were not specified in the **add egp-as-info** commands. The following example reads: readvertise through OSPF and RIP all EGP routes received from Autonomous System 47. The default cost to use in the readvertisements is 10.

Example: **add egp-as-info 47 IN 10**

filter ***dest-IP-address address-mask***

Designates an IP network/subnet to be filtered. IP packets are not forwarded to filtered networks/subnets, nor routing information be disseminated concerning such destinations. Packets destined for filtered network/subnets are simply discarded.

You must specify a filtered network/subnet together with its subnet mask. For example, to filter a subnet of a class B network, using the third byte for subnetting, the mask is 255.255.255.0.

Using the filter mechanism is more efficient than IP access controls, although not as flexible. Filters also affect the operation of the IP routing protocols, unlike access controls.

Example: **add filter 127.0.0.0 255.0.0.0**

input-interchange ***neighboring-AS IP-network metric***

Adds an IP network to the list of routes that, when received from a neighboring Autonomous System, is readvertised through OSPF and RIP. You can configure a separate list for each neighboring AS.

The list of routes specified by this command is called the neighboring AS's Input Exchange Table. Each route is specified by its IP network number and the cost that OSPF and RIP readvertised.

If the metric is set to –1, the readvertised cost is identical to the cost received by the EGP protocol. If the metric is set to –2, the route is not installed in the routing table. If the metric is –3, the route is added to the routing table, but is not advertised to other routers. Before creating the Input Exchange Table, use the **add egp-as-info** command to define the neighboring AS's *interchange-flag*. When the *interchange-flag* is set to **in**, all routes received from the neighboring AS (not just those specified in the Input Exchange Table) are readvertised by OSPF and RIP. In this case, adding routes to the Input Exchange Table can specify the metric used for the readvertisement on a route-to-route basis (overriding the default value specified in the **add egp-as-info** command).

When the *neighboring-AS* is specified as 0, the route is added to the default Input Exchange Table. This table controls EGP route readvertisement for all neighboring ASs that have not been specified in **add egp-as-info** commands.

The example below reads as follows: when receiving an EGP route to 10.0.0.0 from Autonomous System 47, readvertise it through OSPF and RIP with a cost of 5.

Example: **add input-interchange 47 10.0.0.0 5**

output-interchange ***neighboring-AS source-AS IP-network metric***

Adds an IP network to the list of routes that EGP advertises to the neighboring Autonomous System. You can configure a separate list for each neighboring AS.

The list of routes specified by this command is called the neighboring AS's Output Exchange Table. Each route is specified by its IP network number and the cost that EGP advertises.

If the metric is set to –1, the cost advertised by EGP is identical to the route's routing table cost.

In any case, routes originally received from a particular Autonomous System are not readvertised back to that same AS.

When the *neighboring-AS* is specified as 0, the route is added to the default Output Exchange Table. This table lists routes that EGP sends to all neighboring ASs that were not specified in the **add egp-as-info** commands.

Entering a value other than 0 as the *source-AS* indicates that the route is advertised only if the route was received from the specified AS. Entering 0 indicates that the route is advertised regardless of the *source-AS.*

The example below reads as follows: if a route to network 8.0.0.0 exists and was learned from AS 7, advertise it to Autonomous System 47 with a cost of 3.

```
Example: add output-interchange 47 7 8.0.0.0 3
```

packet–filter *filter–name*

Adds one or more packet filters and the corresponding access controls.

```
Example: add packet–filter micon
```

route *IP-network/subnet IP-mask next-hop cost*

Adds a static network/subnet routes to the router's IP configuration. When dynamic routing information is not available for a particular destination, static routes are used.

The destination is specified by an IP address *(IP-network/subnet*) together with an address mask (*IP-mask*). For example, if the destination is a subnet of a class B network, and the third byte of the IP address is used as the subnet portion, the address mask is set to 255.255.255.0.

The route to the destination is specified by the IP address of the next hop (*next-hop*), and the cost (*cost*) of routing the packet to the destination. The next hop must be on the same (sub)net as one of the router's directly connected interfaces.

Example: **add route 17.0.0.0 255.0.0.0 128.185.123.22 6**

Change

Change an IP configuration item previously installed by the **add** command. In general, you must specify the item you want to change, just as you specified the item with the **add** command.

Syntax: change address . . .
egp-as-info . . .
egp-neighbor . . .
filter . . .
input-interchange . . .
output-interchange . . .
route . . .

address *old-address new-address new-mask*

Modifies one of the router's IP interface addresses. You must specify each new address together with the new address' subnet mask. This command can also be used to change an existing address' subnet mask.

Example: **change address 192.9.1.1 128.185.123.22 255.255.255.0**

egp-as-info *as-id new-interchange [default-metric]*

Modifies the interchange flag associated with a neighboring AS (*as-id*). If the interchange flag is set to either in or out, you must specify a default metric for route advertisement in that direction.

Example: **change egp-as-info 32 in 3**

egp-neighbor *nbr-ip-address new-as*

Modifies the configured EGP-neighbor's AS membership.

Example: **change egp-neighbor 192.9.1.3 47**

filter *destination new-mask*

Modifies the subnet mask associated with a filtered network/subnet. Networks that are filtered become black holes. No packets are forwarded to them; nor is routing information distributed about them.

Example: **change filter 127.0.0.0 255.0.0.0**

input-interchange *as-id net-number new-cost*

Modifies the cost associated with a network appearing in an neighboring AS's (*as-id*) Input Exchange Table.

Example: **change input-interchange 32 8.0.0.0 6**

output-interchange *as-id source-as net-number new-cost*

Modifies the cost associated with a *source-as/network* pair appearing in an neighboring AS's (*as-id*) Output Exchange Table.

Example: **change output-interchange 32 47 128.185.0.0 1**

route *destination new-mask new-1st-hop new-cost*

Modifies either the subnet mask, next hop, or the cost associated with a configured static network/subnet route.

Example: **change route 10.0.0.0 255.0.0.0 128.185.123.18 6**

Delete

Delete an IP configuration item previously installed by the **add** command. In general, you must specify the item you want to delete, just as you specified the item with the **add** command.

Syntax: delete accept-rip-route . . .
access-control . . .

address . . .
bootp-server
default network/subnet-gateway . . .
egp-as-info . . .
egp-neighbor . . .
filter . . .
input-interchange . . .
ip–host–only–default . . .
output-interchange . . .
packet–filter . . .
route . . .

accept-rip-route *net-number*

Removes a route from the list of networks that the RIP protocol always accepts.

Example: **delete accept-rip-route 10.0.0.0**

access-control *record-number*

Deletes one of the access control records.

Example: **delete access-control 2**

address *ip-interface-address*

Deletes one of the router's IP interface addresses.

Example: **delete address 128.185.123.22**

bootp-server *server-IP-address*

Removes a BOOTP server from an IP configuration.

Example: **delete bootp-server 128.185.123.22**

default network/subnet-gateway *[subnetted network]*

Deletes either the default gateway or the default subnet gateway for the specified subnetted network.

Example: **delete default subnet-gateway 128.185.0.0**

egp-as-info *as-id*

Deletes the route exchange description for the specified neighboring AS (*as-id*). Without this record, you cannot configure Input or Output Exchange Tables for the neighboring AS. The default tables are used instead (specified by setting *as-id* equal to 0).

Example: **delete egp-as-info 32**

egp-neighbor *nbr-ip-address*

Deletes an initial EGP neighbor.

Example: **delete egp-neighbor 10.0.0.7**

filter *destination*

Deletes one of the router's filtered networks.

Example: **delete filter 127.0.0.0**

input-interchange *as-id net-number*

Deletes a network from a neighboring AS's (*as-id*) Input Exchange Table.

Example: **delete input-interchange 32 8.0.0.0**

ip–host–only–default

Deletes the default gateway used by the router when in host–only mode.

Example: **delete ip–host–only–default**

output-interchange *as-id source-as net-number*

Deletes a source AS or network pair from a neighboring AS's (*as-id*) Output Exchange Table.

Example: **delete output-interchange 32 47 128.185.0.0**

packet–filter *filter–name*

Deletes one or more packet filters and the corresponding access controls.

Example: **delete packet–filter micon**

route *destination*

Deletes one of the router's configured static routes.

Example: **delete route 10.0.0.0**

Disable

Disable IP features previously enabled by the **add** or **enable** command.

Syntax: disable arp-subnet-routing
bootp-forwarding
directed-broadcast
egp
egp-readvertise
override default/static-routes . . .
per–packet–multipath . . .
receiving rip . . .
receiving dynamic nets/subnets . . .
rfc925-routing
rip
sending default/net/subnet/static . . .

arp-subnet-routing

Turns off the IP feature called ARP subnet routing or proxy ARP. When enabled, it deals with hosts that have no IP subnetting support. This is the default and the generally recommended setting.

Example: **disable arp subnet routing**

bootp-forwarding

Turns off the BOOTP relay function.

Example: **disable bootp-forwarding**

directed-broadcast

Disables the forwarding of IP packets whose destination is a non-local (for example, remote LAN) broadcast address. The source host originates the packet as a unicast where it is then forwarded as a unicast to a destination subnet and "exploded" into a broadcast. You can use these packets to locate network servers.

Note: Forwarding and exploding cannot be disabled separately.

Example: **disable directed-broadcast**

egp

Turns off the EGP protocol.

Example: **disable egp**

egp-readvertise

Prevents EGP from readvertising routes that were originally learned from EGP.

Example: **disable egp-readvertise**

override default/static-routes ***ip-interface-address***

Prevents an RIP default route received on interface *ip-interface-address* from being installed as the router's default route. The **disable override static-routes** command prevents RIP routes received on interface *ip-interface-address* from overriding any of the router's static routes.

Example: **disable override default 128.185.123.22**

per–packet–multipath

Turns off the per–packet–multipath feature, which is enabled by default. With the feature enabled, IP can use as many as four equal–cost paths to a destination subnet, which are selected in round robin fashion. Disabling the feature causes IP to use a single path.

Example: **disable per–packet–multipath**

receiving rip ***ip-interface-address***

Prevents any RIP packets form being received on interface *ip-interface-address.*

Example: **disable receiving rip 128.185.123.22**

receiving dynamic nets/subnets ***ip-interface-address***

Ensures that RIP updates receiving on the interface *ip-interface-address* accept only those network level routes entered by the **add accept-rip-route** command. The **disable receiving dynamic subnets** command produces the analogous behavior for subnet routes.

Example: **disable receiving dynamic nets 128.185.123.22**

rfc925-routing

Turns off RFC 925 routing. When this is enabled, the router replies by proxy to all ARP requests for remote destinations that are best reached through the router.

Example: **disable rfc925-routing**

sending default/net/subnet/static ***ip-interface-address***

Prevents the router from advertising a default route in RIP updates sent out the interface ip-interface-address. The other three flags controlling the RIP routes sent out an interface are **net-routes**, **subnet-routes**, and **static-routes**. You can turn these off individually. A route is advertised if it is specified by any of the enabled flags.

Example: **disable sending net-routes 128.185.123.22**

rip

Turns off the RIP protocol.

Example: **disable rip**

Enable

Activate IP features, capabilities, and information added to your IP configuration.

Syntax: enable
- arp-subnet-routing
- bootp-forwarding
- directed-broadcast
- egp . . .
- egp-readvertise
- originate-default
- override default . . .
- override static-routes . . .
- per–packet–multipath
- receiving rip . . .
- receiving dynamic nets . . .
- receiving dynamic subnets . . .
- rfc925-routing
- rip
- sending default-routes . . .
- sending net-routes . . .
- sending poisoned–reverse–routes . . .
- sending subnet-routes . . .
- sending static-routes . . .

arp-subnet-routing

Turns on the router's ARP subnet routing (sometimes also called Proxy ARP) functionality. This functionality is used when there are subnet-incapable hosts attached to directly-connected IP subnets. The directly connected subnet having subnet-incapable hosts must use ARP for this feature to be useful.

The way ARP subnet routing works is as follows. When a subnet-incapable host wants to send an IP packet to a destination on a remote subnet, it does not realize that it must send the packet to a router. The subnet-incapable host therefore simply broadcasts an ARP request. This ARP request is received by the router. The router responds as the destination (hence the name proxy) if both arp-subnet-routing is enabled and if the next hop to the destination is over a different interface than the interface receiving the ARP request.

If there are no hosts on your LAN that are “subnet-incapable,” do not enable ARP-subnet routing. If ARP subnet routing is needed on a LAN, it is enabled on all routers on that LAN.

Example: **enable arp-subnet-routing**

bootp-forwarding

Turns on BOOTP packet forwarding. In order to use the BOOTP forwarding, you must also add one or more BOOTP servers with the **add bootp-server** command.

Example: **enable bootp-forwarding**

```
Maximum number of forwarding hops [4]?
Minimum seconds before forwarding [0]?
```

Maximum number of forwarding hops	Maximum number of allowable BOOTP agents that can forward a BOOTP request from the client to the server (this is **not** the maximum number of IP hops to the server). Default: 4.
Minimum seconds before forwarding	This parameter is generally not used. Use this parameter when there is a redundant path between the client and the server, and you want to use the secondary path(s) as a standby.

directed-broadcast

Enables the forwarding of IP packets whose destination is a non-local (for example, remote LAN) broadcast address. The packet is originated by the source host as a unicast where it is then forwarded as a unicast to a destination subnet and "exploded" into a broadcast. These packets can be used to locate network servers. This command enables both the forwarding and exploding of directed broadcasts. The IP packet forwarder never forwards link level broadcasts/multicasts, unless they correspond to Class D IP addresses. (See the OSPF **enable multicast-routing** command.) The default setting for this feature is enabled.

Note: Forwarding and exploding cannot be implemented separately. Also, the router does not forward subnet-wide IP broadcasts.

Example: **enable directed-broadcast**

egp *own-as-id*

Enables the router's EGP protocol capabilities. When enabling the router's EGP capability, the router's own Autonomous System number (*own-as-id*) must be specified. These are assigned by Stanford Research Institute's Network Information Center. All routers belonging to the same Autonomous System are configured with the same AS number.

Example: **enable egp 21**

egp-readvertise

Turns on EGP readvertising. This allows the router to readvertise through EGP routes that were originally learned from EGP.

Note: A route can always be advertised through EGP regardless of its origin if it is specified in an Output Exchange Table. (See the IP **add output exchange** command.)

Example: **enable egp-readvertise**

originate-default

Originates a default RIP route whenever the router has EGP-derived routes in its routing table.

RIP advertises such a default route with a metric of 1, unless otherwise specified in the **set default-metric** command. In order for RIP to actually send a default route out a particular interface, you must also invoke the **enable sending default** command for the interface.

Example: **enable originate-default**

override default *ip-interface-address*

Enables received RIP information to override the router's default gateway. This command is invoked on a per-IP-interface basis. When the **enable override default** command is invoked, default RIP routes received on interface *ip-interface-address* overwrite the router's current default gateway, providing the cost of the new default is cheaper.

Example: **enable override default 128.185.123.22**

override static-routes *ip-interface-address*

Enables received RIP information to override some of the router's statically configured routing information. This command is invoked on a per-IP-interface basis. When the **enable override static-routes** command is invoked, RIP routing information received on interface *ip-interface-address* overwrite statically configured network/subnet routes providing the cost of the RIP information is cheaper.

Example: **enable override static-routes 128.185.123.22**

per–packet–multipath

Enables the feature called per–packet–multipath.

Example: **enable per–packet–multipath**

receiving rip *ip-interface-address*

Modifies the processing of RIP updates that are received on a particular interface. This command has an analogous **disable** command. (See the **disable receiving** command.) This command is enabled by default. The opposite command is easier to describe, and that is done in the following paragraph.

If you invoke the **disable receiving rip** command, no RIP updates are accepted on interface *ip-interface-address* address.

Example: **enable receiving rip 128.185.123.22**

receiving dynamic nets *ip-interface-address*

Modifies the processing of RIP updates that are received on a particular interface. This command has an analogous **disable** command. (See the **disable receiving** command.) This command is enabled by default. The opposite command is easier to describe, and that is done in the following paragraph.

If you invoke the **disable receiving dynamic nets** command, RIP updates received on interface *ip-interface-address* cannot accept any network-level routes unless they were previously specified in an **add accept-rip-route** command.

Example: **enable receiving dynamic nets 128.185.123.22**

receiving dynamic subnets *ip-interface-address*

Modifies the processing of RIP updates that are received on a particular interface. This command has an analogous **disable** command. (See the **disable receiving** command.) This command is enabled by default. The opposite command is easier to describe, and that is done in the following paragraph.

If you invoke the **disable receiving dynamic subnets** command, RIP updates received on interface *ip-interface-address* cannot accept any subnet-level routes unless they were previously specified in an **add accept-rip-route** command.

Example: **enable receiving dynamic subnets 128.185.123.22**

rfc925-routing

Turns on RFC 925 routing. When enabled, the router replies by proxy to all ARP requests for remote destinations that are best reached through the router. Use this command when there are hosts on the LAN that ARP for all destinations, instead of (as is proper) only local destinations.

Example: **enable rfc925-routing**

rip

Enables the router's RIP protocol processing. When the RIP protocol is enabled, use the **enable/disable sending** commands to configure its routing update sending behavior. Its routing update receiving behavior is defined by the **enable/disable receiving** and **enable/disable override** commands.

Example: **enable rip**

sending default-routes ***ip-interface-address***

Determines the contents of RIP updates that are sent out a particular interface. This command has an analogous **disable** command. (See the **disable sending** command.)

The effect of the **enable sending** command is additive. Each separate **enable sending** command specifies that a certain set of routes is advertised from a particular interface. A route is included in a RIP update only if it was included by at least one of the **enable sending** commands. The **enable sending default-routes** command specifies that the default route (if one exists) is included in RIP updates sent out interface *ip-interface-address.*

Example: **enable sending default-routes 128.185.123.22**

Note: Some settings of the **enable sending** commands are redundant. For example, if you invoke **enable sending net-routes** and **enable sending subnet-routes** for a particular interface, there is no need to also specify **enable sending static-routes** (because each static route is either a network-level or subnet route). By default, when you first enable RIP, **sending net-routes** and **sending subnet-routes** are enabled for each interface, while **sending static-routes** and **sending default** are disabled.

sending net-routes ***ip-interface-address***

Determines the contents of RIP updates that are sent out a particular interface. This command has an analogous **disable** command. (See the **disable sending** command.)

The effect of the **enable sending** command is additive. Each separate **enable sending** command specifies that a certain set of routes is advertised from a particular interface. A route is included in an RIP update only if it was included by at least one of the **enable sending** commands. The **enable sending network-routes** command specifies that all network-level routes are included in RIP updates sent out interface *ip-interface-address.* A network-level route is a route to a single class A, B, or C IP network.

Example: **enable sending net-routes 128.185.123.22**

sending poisoned–reverse–routes ***ip-interface-address***

Determines the contents of RIP updates that are sent out a particular interface. This command has an analogous **disable** command. (See the **disable sending** command.)

The effect of the **enable sending** command is additive. Each separate **enable sending** command specifies that a certain set of routes is advertised from a particular interface. A route is included in an RIP update only if it was included by at least one of the **enable sending** commands. The **enable sending poisoned–reverse-routes** command specifies that all network-level routes are included in RIP updates sent out interface *ip-interface-address.* A network-level route is a route to a single class A, B, or C IP network.

Example: **enable sending poisoned–reverse-routes 128.185.123.22**

sending subnet-routes ***ip-interface-address***

Determines the contents of RIP updates that are sent out a particular interface. This command has an analogous **disable** command. (See the **disable sending** command.)

The effect of the **enable sending** command is additive. Each separate **enable sending** command specifies that a certain set of routes is advertised through a particular interface. A route is included in a RIP update only if it was included by at least one of the **enable sending** commands. The **enable sending subnet-routes** command specifies that all subnet routes are included in RIP updates sent out interface *ip-interface-address.* However, a subnet route is included only if *ip-interface-address* connects directly to a subnet of the same IP subnetted network.

Example: **enable sending subnet-routes 128.185.123.22**

sending static-routes ***ip-interface-address***

Determines the contents of RIP updates that are sent out a particular interface. This command has an analogous **disable** command. (See the **disable sending** command.)

The effect of the **enable sending** command is additive. Each separate **enable sending** command specifies that a certain set of routes is advertised through a particular interface. A route is included in a RIP update only if it was included by at least one of the **enable sending** commands. The **enable sending static-routes** command specifies that all statically configured and directly connected routes are included in RIP updates sent out interface *ip-interface-address.*

Example: **enable sending static-routes 128.185.123.22**

List

Display various pieces of the IP configuration data, depending on the particular subcommand invoked.

Syntax: list

- all
- access-controls
- addresses
- bootp
- egp-as-info
- egp-neighbors
- input-interchange . . .
- output-interchange . . .
- protocols
- rip-routes-accept
- routes
- sizes
- tags

all

Prints the entire IP configuration.

Example: **list all**

access-controls

Prints the configured access control mode (inclusive, exclusive, or disabled) and the list of configured access control records. Each record is listed with its record number. This record number can be used to reorder the list with the IP **move access-control** command.

Example: **list access control**

addresses

Prints the IP interface addresses that were assigned to the router, along with their configured broadcast formats.

Example: **list addresses**

bootp

Indicates whether BOOTP forwarding is enabled or disabled, as well as the configured list of BOOTP servers.

Example: **list bootp**

egp-neighbors

Prints the list of initial EGP neighbors that were configured.

Example: **list egp neighbors**

egp-as-info

Prints the configured interchange flags for each of the neighboring Autonomous Systems.

Example: **list egp-as-info**

input-interchange *neighbor-as-id*

Prints the set of routes that, when learned from AS **neighbor-as-id**, are readvertised by the IGPs.

Example: **list input-exchange 32**

output-interchange *neighbor-as-id*

Prints the set of routes that is advertised to Autonomous System *neighbor-as-id* by the EGP protocol.

Example: **list output-exchange 32**

protocols

Prints the configured state of the IP routing protocols (OSPF, RIP, and EGP) along with whether ARP subnet routing is enabled or disabled.

Example: **list protocols**

rip-routes-accept

Prints the set of routes that the RIP routing protocol always accepts. See the IP configuration commands **enable/disable receiving dynamic nets/subnets** for more information.

Example: **`list rip-routes-accept`**

routes

Prints the list of static network/subnet routes that were configured. Also lists any configured default gateways.

Example: **`list routes`**

sizes

Displays the routing table size, reassembly buffer size, and the route cache size.

Example: **`list sizes`**

tags

Displays the per-interface tags that are associated with received RIP information. These tags can be used to group routes together for later readvertisement through EGP where a tag is treated as if it were a route's source AS. (See the IP **add output-exchange** command.) Tags are also propagated by the OSPF routing protocol.

Example: **`list tags`**

Move

Change the order of the access control list. This command places record number *from#* immediately after record number *to#*. After you move the records, they are immediately renumbered to reflect the new order.

Syntax: move access-control *from# to#*

Example: **`move 5 2`**

Set

Set certain values, routes, and formats within your IP configuration.

Syntax: set
- access-control . . .
- advertised default-metric . . .
- broadcast-address . . .
- cache-size
- default network-gateway . . .
- default subnet-gateway . . .
- egp-system-number . . .
- internal-ip-address
- originate–rip–default
- reassembly-size
- router-id . . .
- routing table-size . . .
- tag. . .

access-control *on* or *off*

Allows you to configure the router to enable or disable IP access control.

Example: **`set access-control on`**

advertised default-metric *cost*

Sets the cost that RIP advertises when originating a default route. In order to originate a default route, you must invoke the **enable originate-default** command, and there must be EGP-derived routes present in the router's routing table. If advertised default metric is not set, yet the router was instructed to originate default, it does this with a cost of 1. This command does not affect the cost when a router is simply propagating a default route. The cost advertised in that case is always equal to the routing table cost.

Example: **`set advertised default-metric 3`**

broadcast-address *ip-interface-address style fill-pattern*

Specifies the IP broadcast format that the router uses when broadcasting packets out a particular interface. IP broadcasts are most commonly used by the router when sending RIP update packets.

The *style* parameter can take either the value *local-wire* or the value *network.* Local-wire broadcast addresses are either all ones (255.255.255.255) or all zeros (0.0.0.0). Network style broadcasts begin with the network and subnet portion of the *ip-interface-address.*

You can set the *fill-pattern* parameter to either 1 or 0. This indicates whether the rest of the broadcast address (other than the network and subnet portions, if any) is set to all ones or all zeros.

When receiving the router recognizes all forms of the IP broadcast address.

The example below configures a broadcast address of 255.255.255.255. The second example produces a broadcast address of 192.9.1.0, assuming that the network 192.9.1.0 is not subnetted.

```
Example: set broadcast-address 192.9.1.11 local-wire 1
         set broadcast-address 192.9.1.11 network 0
```

Note: To display the broadcast address setting, issue the **list all** command.

cache-size *entries*

Configures the maximum number entries for the IP routing cache. Default: 64. Maximum: none.

```
Example: set cache-size 64
```

Note: To display the cache size setting, issue the **list sizes** command.

default network-gateway *next-hop cost*

Configures a route to the authoritative router (default gateway). Assume that the router's default gateway has more complete routing information than the router itself.

The route is specified by the IP address of the next hop (*next-hop*) and the distance (*cost*) to the default gateway.

All packets having unknown destinations are forwarded to the authoritative router (default gateway).

```
Example: set default network-gateway 192.9.1.10 10
```

default subnet-gateway *subnetted-network next-hop cost*

Configures a route to a subnetted network's authoritative router (default subnet gateway). You can configure a separate default subnet gateway for each subnetted network.

The IP address of the next hop (*next-hop*) and the distance (*cost*) to the default subnet gateway specify the route.

All packets destined for unknown subnets of a known subnetted network are forwarded to the subnetted network's authoritative router (default subnet gateway).

Example: **set default subnet-gateway 128.185.0.0 128.185.123.22 6**

egp-system-number

Configures the router's Autonomous System number that is used when running the EGP protocol.

Example: **set egp-system-number**

internal-IP-address

Sets the internal IP address that belongs to the router as a whole, and not any particular interface. This address is always reachable regardless of the state of the interface. When the internal IP address and the router ID are set in the same router, the internal IP address has precedence over the router ID. To delete the internal IP address, set the address to 0.0.0.0.

Example: **set internal-ip-address 142.82.10.1**

originate–rip–default

reassembly-size

Configures the size of the buffers that are used for the reassembly of fragmented IP packets. Default: 12000

Note: This parameter is relevant to the EGP routing protocol.

Example: **set reassembly-size 12000**

router-id ***ip-address***

Sets the default IP address used by the router when sourcing various kinds of IP traffic. This address is of particular importance in multicasting. For example, the source address in pings (including multicast pings), traceroute, and tftp packets sent by the router are set to the router ID. In addition, the OSPF router ID are set to the configured router ID.

The router ID must match one of the configured IP interface addresses of the router. If not, it is ignored. When ignored, or just not configured, the default IP address of the router (and its OSPF router ID) is set to the first IP address in the router's configuration.

Note: Configuring a router ID may cause the router's OSPF router ID to change. If this happens, link state advertisements originated by the router before the router ID change persist until they age out (possibly as long as 30 minutes). This may cause an increase in link state database size.

Example: **set router-id 128.185.120.209**

Note: To display the router ID setting, issue the **list all** command.

routing table-size ***number-of-entries***

Sets the size of the router's IP routing table. The default size is 768 entries. Setting the routing table size too small causes dynamic routing information to be discarded. Setting the routing table size too large wastes router memory resources.

Example: **set routing table-size 1000**

tag

Configures the per-interface tags associated with received RIP information. These tags can be used to group routes together for later readvertisement through EGP where a tag is treated as if it were a route's source AS. (See the IP **add output-interchange** command.) Tags are also propagated by the OSPF routing protocol.

Example: **set tag**

Exit

Return to the previous prompt level.

Syntax: exit

Example: **exit**

14

Monitoring IP

This chapter describes the IP console commands.

For more information about IP, refer to the *Routing Protocols Reference Guide*.

Accessing the IP Console Environment

For information about accessing the IP console environment, see Chapter 1.

IP Console Commands

This section summarizes and then explains all the IP console commands. Table 14–1 lists the IP console commands. The commands allow you to monitor the router's IP forwarding process. The monitoring capabilities include the following: configured parameters such as interface address and static routes can be viewed, the current state of the IP routing table can be displayed, and a count of IP routing errors can be listed.

Table 14–1 IP Console Command Summary

Command	Function
? (Help)	Lists the console commands or lists the actions associated with specific commands.
Access controls	List the current IP access control mode, together with the configured access control records.
Cache	Displays a table of all recent routed destinations.
Counters	Lists various IP statistics, including counts of routing errors and packets dropped.
Dump routing tables	Lists the contents of the IP routing table.
EGP-neighbors	Displays the current state of all EGP neighbors.
EGP-routes	Displays the routing information that either can be or was exchanged with a particular EGP neighbor.
Interface addresses	Lists the router's IP interface addresses.
Ping	Sends ICMP Echo Requests to another host once a second and watch for a response. This command can be used to isolate trouble in an internetwork environment.
Route	Lists whether a route exists for a specific IP destination, and if so, the routing table entry that corresponds to the route.
Sizes	Displays the configured sizes of specific IP parameters.
Static routes	Displays the static routes that were configured. This includes the default gateway.
`Traceroute`	Displays the complete path (hop-by-hop) to a particular destination.
Exit	Exits the IP console environment.

? (Help)

List the commands that are available from the current prompt level. You can also enter a **?** after a specific command name to list its options.

Syntax: ?

Example: **?**

```
ACCESS controls
CACHE
COUNTERS
DUMP routing tables
INTERFACE addresses
PING address
ROUTE given address
SIZES
STATIC routes
TRACEROUTE address
EGP-ROUTES
EGP-NEIGHBORS
EXIT
```

Access Controls

Print the access control mode in use together with a list of the configured access control records.

The access control mode is one of the following: *disabled* (meaning that no access control is being done and the access control records are being ignored) or *enabled* (meaning that access control is being done and the access control records are being recognized). When access control is enabled, access control records are scanned in order looking for the first match.

Exclusive (*E*) means that packets matching the access control record are being discarded. Inclusive (*I*) means that packets matching the access control record are being forwarded. When access control is enabled, packets failing to match any access control record are discarded. *Pro* (protocol) indicates the IP protocol number and *Prt* (port) indicates the UDP or TCP port number.

Syntax: access

Example: **access**

```
Access control currently ENABLED
List of access control records:

                                                    Beg End  Beg End
      Source       Mask       Destination   Mask     Pro Pro  Prt Prt
0 I   0.0.0.0      00000000   192.67.67.20  FFFFFFFF 6   6    25  25
1 E   150.150.1.0  FFFFFF00   150.150.2.0   FFFFFFFF 0   255  0   65535
2 I   0.0.0.0      00000000   0.0.0.0       00000000 89  89   0   65535
```

Cache

Display the IP routing cache that contains recently routed destinations. If a destination is not in the cache, the router looks up the destination in the routing information table in order to make a forwarding decision.

Syntax: cache

Example: **cache**

```
Destination     Usage           Next hop

128.185.128.225 1               128.185.138.180 (Eth /0)
192.26.100.42   1               128.185.138.180 (Eth /0)
128.185.124.121 4               128.185.124.121 (Eth /0)
```

Destination	IP destination host.
Usage	Number of packets recently sent to the destination host.
Next hop	IP address of the next router on the path toward the destination host. Also displayed is the network name of the interface used by the sending router to forward the packet.

Counters

Display the statistics related to the IP forwarding process. This includes a count of routing errors, along with the number of packets that were dropped due to congestion.

Syntax: counters

Example: **counters**

```
Routing errors
Count   Type
    0   Routing table overflow
 2539   Net unreachable
    0   Bad subnet number
    0   Bad net number
    0   Unhandled broadcast
58186   Unhandled multicast
    0   Unhandled directed broadcast
 4048   Attempted forward of LL broadcast

Packets discarded through filter  0
IP multicasts accepted:           60592
```

```
IP input packet overflows
   Net    Count
   Eth /0  0
   Eth /1  0
```

Routing table overflow	Lists the number of routes that were discarded due to the routing table being full.
Net unreachable	Indicates the number of packets that were not forwarded due to unknown destinations. This does not count the number of packets that were forwarded to the authoritative router (default gateway).
Bad subnet number	Counts the number of packets or routes that were received for illegal subnets (all ones or all zeroes).
Bad net number	Counts the number of packets or routes that were received for illegal IP destinations (for example, class E addresses).
Unhandled broadcasts	Counts the number of (non-local) IP broadcasts received (these are not forwarded).
Unhandled multicasts	Counts the number of IP multicasts that were received, but whose address was not recognized by the router (these are discarded).
Unhandled directed broadcasts	Counts the number of directed (non-local) IP broadcasts received when forwarding of these packets is disabled.
Attempted forward of LL broadcast	Counts the number of packets that are received having non-local IP addresses but were sent to a link level broadcast address. These are discarded.
Packets discarded through filter	Counts the number of received packets that were addressed to filtered networks/subnets. These are discarded silently.

IP multicasts accepted	Counts the number of IP multicasts that were received and successfully processed by the router.
IP packet overflows	Counts the number of packets that were discarded due to congestion at the forwarder's input queue. These counts are sorted by the receiving interface.

Dump Routing Tables

Display the IP routing table. A separate entry is printed for each reachable IP network/subnet. The IP default gateway in use (if any) is listed at the end of the display.

Syntax: dump

Example: **dump**

```
Type   Dest net         Mask     Cost Age   Next hop(s)
SPE1   0.0.0.0          00000000  4    3    128.185.138.39    (2)
SPF*   128.185.138.0    FFFFFF00  1    1    Eth /0
Sbnt   128.185.0.0      FFFF0000  1    0    None
SPF    128.185.123.0    FFFFFF00  3    3    128.185.138.39    (2)
SPF    128.185.124.0    FFFFFF00  3    3    128.185.138.39    (2)
SPF    192.26.100.0     FFFFFF00  3    3    128.185.131.10    (2)
RIP    197.3.2.0        FFFFFF00  10   30   128.185.131.10
RIP    192.9.3.0        FFFFFF00  4    30   128.185.138.21
Del    128.185.195.0    FFFFFF00  16   270  None


Default gateway in use.

Type Cost Age  Next hop
SPE1 4    3    128.185.138.39

Routing table size: 768 nets (36864 bytes), 36 nets known
```

Type (route type)

Indicates how the route was derived.

- **Sbnt** – Indicates that the network is subnetted; such an entry is a placeholder only.
- **Dir** – Indicates a directly connected network or subnet.
- **RIP** – Indicates the route was learned through the RIP protocol.
- **Del** – Indicates the route was deleted.
- **Stat** – Indicates a statically configured route.
- **EGP** – Indicates routes learned through the EGP protocol.
- **EGPR** – Indicates routes learned through the EGP protocol that are readvertised by OSPF and RIP.
- **Fltr** – Indicates a routing filter.
- **SPF** – Indicates that the route is an OSPF intra-area route.
- **SPIA** – Indicates that it is an OSPF inter-area routes.
- **SPE1**, **SPE2** – Indicates OSPF external routes (types 1 and 2 respectively).
- **Rnge** – Indicates a route type that is an active OSPF area address range and is not used in forwarding packets.

Dest net

IP destination network/subnet.

Mask

IP address mask.

Cost

Route Cost.

Age	For RIP and EGP routes, the time that has elapsed since the routing table entry was last refreshed.
Next Hop	IP address of the next router on the path toward the destination host. Also displayed is the interface type used by the sending router to forward the packet.

An asterisk (*) after the route type indicates the the route has a static or directly connected backup. A percent sign (%) after the route type indicates that RIP updates is always accepted for this network/subnet.

A number in parentheses at the end of the column indicates the number of equal-cost routes to the destination. The first hops belonging to these routes can be displayed with the IP **route** command.

EGP-Neighbors

Display the current EGP state and the configured EGP interchange-flag for each of the router's EGP neighbors. Each EGP neighbor is identified by its IP address. The EGP neighbor states are explained in the EGP specification (RFC 904).

Syntax: egp-neighbor

Example: **egp-neighbor**

```
EGP neighbor 128.185.138.11   State:NEIGHBOR    Flag:in
EGP neighbor 128.185.138.19   State:NEIGHBOR    Flag:out
```

In	Indicates that all routes received from the neighbor are readvertised through OSPF and RIP.
Out	Indicates that all routes found in the routing table are sent to the neighbor.
Off	Indicates that the EGP route exchange is determined completely by the Input and Output Exchange Tables. For more information, see the `IP config>` **add egp-as-info** command.

EGP-Routes

Display the routes that are being sent to and received from an EGP neighbor. The egp neighbor's current state and configured interchange-flag are also listed as in the IP **egp-neighbors** command.

Syntax: egp-routes *egp-neighbor-address*

Example: **egp-routes 128.185.138.11**

```
EGP neighbor 128.185.138.11    State:NEIGHBOR    Flag:in

Routes advertised to EGP neighbor
Destination         Cost         Source AS
128.185.0.0           13              6
192.9.12.0            13              6

Routes obtained from EGP neighbor
Destination         Cost
129.9.0.0             5       Route Advertised through IGP
192.26.101.0          4       Route Advertised through IGP
192.26.100.0          4       Route Advertised through IGP
128.52.0.0            6       Route Advertised through IGP
18.0.0.0              6       Route Advertised through IGP
```

Routes advertised to EGP neighbor	Lists the IP networks that are currently being sent (in EGP poll responses) to the EGP neighbor. The advertised cost is also listed, together with the Autonomous System that originally supplied the route.
Routes obtained from EGP neighbor	Lists the current portion of the IP routing table that was received from the EGP neighbor. Those routes that are readvertised by OSPF and RIP are suitably marked.

Interface Addresses

Display the router's IP interface addresses. Each address is listed together with its corresponding hardware interface and IP address mask.

Hardware interfaces having no configured IP interface addresses are not used by the IP forwarding process; they are listed as **Not an IN net**. There is one exception. Serial lines need not be assigned IP interface addresses in order to forward IP traffic. Such serial lines are called *unnumbered.* They show up as having address 0.0.0.0.

Syntax: interface

Example: **interface**

```
Interface  IP Address(es)   Mask(s)
   SL /0   Not an IN net
   SL /1   0.0.0.0          0.0.0.0
   SL /2   Not an IN net
   Eth /0   128.185.138.19   255.255.255.0
```

Interface	Indicates the hardware type of the interface.
IP addresses	Indicates the IP address of the interface.
Mask	Indicates the subnet mask of the interface.

Ping

Have the router send ICMP Echo Requests to a given destination once a second and watch for a response. This command can be used to isolate trouble in an internetwork environment.

This process is done continuously, incrementing the ICMP sequence number with each additional packet. Matching received ICMP Echo responses are reported with their sequence number and the round trip time. The granularity (time resolution) of the round trip time calculation is usually (depending on platform) on the order of 20 milliseconds. The ping command completes when a character is typed at the console. At that time, a summary of packet loss, round trip time, and number of ICMP destination unreachables received is displayed.

When a multicast address is given as destination, there may be multiple responses printed for each packet sent, one for each group member. Each returned response is displayed with the source address of the responder.

Note: The size of the ping (number of data bytes in the ICMP message, excluding the ICMP header) is 56 bytes, and the TTL used is 60.

Syntax: ping *interface-address*

Example: **ping 128.185.142.11**

```
PING 128.185.142.11: 56 data bytes
64 bytes from 128.185.142.11: icmp_seq=0. time=0. ms
64 bytes from 128.185.142.11: icmp_seq=1. time=0. ms
64 bytes from 128.185.142.11: icmp_seq=2. time=0. ms
64 bytes from 128.185.142.11: icmp_seq=3. time=0. ms
64 bytes from 128.185.142.11: icmp_seq=4. time=0. ms
64 bytes from 128.185.142.11: icmp_seq=5. time=0. ms

----128.185.142.11 PING Statistics----
6 packets transmitted, 6 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 0/0/0
```

Route

Display the route (if one exists) to a given IP destination. If a route exists, the IP addresses of the next hops are displayed, along with detailed information concerning the matching routing table entry. (See the IP **dump** command.)

Syntax: route *ip-destination*

Example: **route 18.10.0.5**

```
Destination:   18.10.0.5
Mask:          255.0.0.0
Route type:    SPE1
Distance:      3
Age:           1
Next hop(s):   128.185.123.18
```

Example: **route 128.185.230.0**

```
Destination:   128.185.230.0
Mask:          255.255.255.0
Route type:    SPF
Distance:      1
Age:           1
Next hop(s):   128.185.230.0
```

Example: **route 128.185.232.0**

```
Destination:   128.185.232.0
Mask:          255.255.255.0
Route type:    RIP
Distance:      3
Age:           0
Next hop(s):   128.185.146.4
```

Sizes

Display the configured sizes of specific IP parameters.

Syntax: sizes

Example: **sizes**

```
Routing table size:           768
Table entries used:           3
Reassembly size:              12000
Largest reassembled pkt:       0
Largest EGP update sent:       0
Size of routing cache:        64
# of cache entries in use:     0
```

Routing table size	Indicates the configured number of entries that the routing table maintains.
Table entries used	Indicates the number entries used from the routing table.
Reassembly buffer size	Indicates the configured size of the reassembly buffer that is used to reassemble fragmented IP packets.
Largest reassembled pkt	Indicates the largest IP packet that this router reassembled.
Size of routing cache	Indicates the configured the size of the routing cache.
# of cache entries in use	Indicates the number of tries currently being used from the cache.

Static Routes

Display the list of configured static routes. Configured default gateways and default subnet gateways are also listed.

Each static route's destination is specified by an address-mask pair. Default gateways appear as static routes to destination 0.0.0.0 with mask 0.0.0.0. Default subnet gateways also appear as static routes to the entire IP subnetted network.

The example below shows a configured default gateway, a configured default subnet gateway (assuming 128.185.0.0 is subnetted), and a static route to network 192.9.10.0.

Syntax: static

Example: **static**

```
Net             Mask            Cost  Next hop
0.0.0.0         0.0.0.0         1     128.185.123.18
128.185.0.0     255.255.0.0     1     128.185.123.22
192.9.10.0      255.255.255.0   10    128.185.123.22
```

Net	Indicates the network address of the route.
Mask	Indicates the subnet mask of the IP address.
Cost	Indicates the cost of using this route.
Next Hop	Indicates the next router a packet passes through using this route.

Traceroute

Display the entire path to a given destination, hop by hop. For each successive hop, **traceroute** sends out three probes, and prints the IP address of the responder, together with the round trip time associated with the response. If a particular probe receives no response, an asterisk is printed. Each line in the display relates to this set of three probes, with the left most number indicating the distance from the router executing the command (in router hops).

The traceroute is done whenever the destination is reached, an ICMP Destination Unreachable is received, or the path length reaches 32 router hops.

When a probe receives an unexpected result, several indications can be printed. "!N" indicates that an ICMP Destination Unreachable (net unreachable) was received. "!H" indicates that an ICMP Destination Unreachable (host unreachable) was received. "!P" indicates that an ICMP Destination Unreachable (protocol unreachable) was received. Since the probe is a UDP packet sent to a strange port, a port unreachable is what we expect. "!" indicates that the destination was reached, but the reply sent by the destination was received with a TTL of 1. This usually indicates an error in the destination, prevalent in some versions of UNIX. The destination is inserting the probe's TTL in its replies. This unfortunately leads to a number of lines consisting solely of asterisks before the destination is finally reached.

Syntax: traceroute *interface-address*

Example: **traceroute 128.185.142.239**

```
TRACEROUTE 128.185.124.110: 56 data bytes
 1 128.185.142.7 16 ms 0 ms 0 ms
 2 128.185.123.22 16 ms 0 ms 16 ms
 3 * * *
 4 * * *
 5 128.185.124.110 16 ms ! 0 ms ! 0 ms !
```

TRACEROUTE	Displays the destination area address and the size of the packet being sent to that address.
1	The first trace showing the destination's NSAP and the amount of time it took the packet to arrive at the destination. The packet is traced three times.
Destination unreachable	Indicates that no route to destination is available.
*1 * * * 2 * * *	Indicates that the router is expecting some form of response from the destination, but the destination is not responding.

Exit

Return to the previous prompt level.

Syntax: exit

Example: **exit**

15

Configuring IPX

This chapter describes how to configure the IPX protocol using the IPX configuration commands.

For additional information about the IPX protocol, refer to the *Routing Protocols Reference Guide*.

Accessing the IPX Configuration Environment

For information on how to access the IPX configuration environment, see Chapter 1 in the *System Software Guide*.

IPX Configuration Commands

This section explains the IPX configuration commands. These commands specify the network parameters for router interfaces transmitting IPX packets. Enter the commands at the `IPX config>` prompt. To activate the commands, restart the router.

Table 15–1 summarizes the IPX configuration commands.

Table 15–1 IPX Configuration Commands Summary

Command	Function
? (Help)	Lists all of the IPX configuration commands or lists the options associated with specific commands.
Add	Adds access control for IPX packets, SAP filters, and IP tunnel addresses.
Delete	Deletes access control for IPX packets, SAP filters, and IP tunnel addresses.
Disable	Disables specific IPX interfaces, IPX over the point-to-point protocol, an IP tunnel, replies to Get Nearest Neighbor requests, NetBIOS filtering for an interface, the router's response to keepalive packets, or globally disables IPX.
Enable	Enables specific IPX interfaces, IPX over the point-to-point protocol, an IP tunnel, replies to Get Nearest Neighbor requests, NetBIOS filtering for an interface, the router's response to keepalive packets, or globally enables IPX.
Frame	Specifies the data link format for Ethernet, token ring, and FDDI interfaces.
List	Displays the current IPX configuration.
Move	Changes the line numbers set when adding access control.
Set	Sets the host number, network number, maximum networks, access control, filters, maximum services parameters, ipxwan over PPP, local and remote cache size, router name, RIP/SAP update intervals for networks and tunnels, and node-id.
Exit	Exits the IPX configuration process and returns to the CONFIG environment.

? (Help)

List the commands that are available from the current prompt level. You can also enter **?** after a specific command name to list its options.

Syntax: ?

Example: **?**

```
DISABLE
ENABLE
EXIT
FRAME
LIST
SET
ADD
DELETE
MOVE access-control
```

Add

Add an access control entry. This determines whether IPX packets are dropped or forwarded. The **add** command also adds an IPX SAP filter and the IP tunnel address.

Syntax: add access-control . . .
filter . . .
ip-tunnel-address . . .

access-control ***type dest-net dest-host dest-socket-range src-net src-host src-socket-range***

Determines whether to pass a packet at the Internet Datagram Protocol (IDP) level. IPX Access controls provide a global access control functionality at the IDP level for the IPX protocol. The access control list is an ordered set of entries. Each entry can be either Inclusive or Exclusive. Each entry has source and destination network numbers, host addresses, and socket ranges.

When a packet is received from a network for the IPX protocol, and IPX access control is enabled, it is checked against the access control list. It is compared with the net/address/socket pairs in the list until there is a match. If there is a match, and the entry is of the Inclusive type, reception of the packet (and potential forwarding) proceeds. If the matching entry is of the Exclusive type, the packet is dropped. If there is no match, the packet is dropped.

Note: You must add entries to the address control list before access control can function.

Note: When enabled, access controls apply to all received packets. If you do not enable reception of RIP (socket 453 hexadecimal) or SAP (socket 452 hexadecimal) packets, the IPX forwarder is non-functional. Always start with the following entry:

```
add access I 0 0 452 453 0 0 0 FFFF
```

The arguments for this command are as follows:

Type	Identifies whether packets are sent or dropped for a specific address or set of addresses. Enter I for Include. This allows the packets to be sent. Enter E for Exclude. This causes the router to discard the packets.
Dest-net	Network number of the destination. Enter the network number in hexadecimal. Zero (0) means all networks.
Dest-host	Host number on the destination network. Enter the host number in hexadecimal. Zero (0) means all hosts on the network.
Dest-socket-range	Two numbers that specify an inclusive range of destination sockets. Enter two hexadecimal numbers between zero (0) and FFFF.
Src-net	Network number of the source. Enter the network number in hexadecimal. Zero (0) means all networks.
Src-host	Host number on the source network. Enter the host number in hexadecimal. Zero (0) means all hosts on the network.
Src-socket-range	Two numbers that specify an inclusive range of source sockets. Enter two hexadecimal numbers between zero (0) and FFFF.

Note: It is not necessary to use access controls and SAP filters for IPX to work in a NetWare environment. Use them only if necessary.

Syntax: add access-control *type dest-net dest-host dest-socket-range src-net src-host src-socket-range*

Example: **add access-control E 201 1 0 FFFF 329 0 451 451**

This access control prevents all nodes on network 329 from accessing the file server with internal network number 201.

filter *hops service-type service-name*

Prevents NetWare bindery overflows for users on large networks by enabling you to determine the number of hops reasonable for a given service. IPX SAP filters allow the protocol to be configured to ignore certain entries in SAP advertisements. This is done to limit the size of the SAP database. This may be necessary due to size limitations in older versions of NetWare file servers. This may also be necessary to limit the amount of SAP data sent across WAN links.

The SAP filters are a global ordered list of filter entries. Each filter entry has a maximum hop count, a service type, and an optional service name. When a SAP response packet is received, each SAP entry is compared with the filter list. If the SAP entry matches an entry in the filter list and is greater than the specified hops, it is ignored and not entered into the local SAP database. If there is no match, the SAP entry is accepted.

The arguments for this command are as follows:

Hops	Maximum number of hops permitted for the service. The range is 0 to FFFF.
Service-type	Numeric service class. Enter a 2-byte number.
Service-name	Identifies a particular function provided by a server. In general, this field is not entered. If you do specify a value, enter a 48-character name.

Syntax: add filter *hops service-type service-name*

Example: **add filter 2 039B NOTES-CHICAGO**

This example filters any SAP advertisements for the Lotus Notes server "NOTES-CHICAGO" at more than 2 hops.

ip-tunnel-address *ip-address*

Used to construct the IPX IP address peer list. You can assign one IP unicast address at a time to form the list.

ip-address	IP unicast or multicast address that makes up the peer list.

Syntax: add ip-tunnel-address ip-address

Example: **add ip-tunnel-address <address>**

Delete

Delete an IPX access control entry, SAP filter entry, or IP tunnel address.

Syntax: delete access-control . . .
filter . . .
ip-tunnel-address . . .

access-control l*ine#*

Discards the access control statement that matches the line number you enter. Run the list command to display the current line numbers.

Syntax: delete access-control *line#*

Example: **delete access-control 2**

filter *hops service-type service-name*

Discards the specified filter statement. You must type the statement exactly as it appears when you run the list command. The arguments are as follows:

Hops	Maximum number of hops permitted for the service.
Service-type	Numeric service class. Enter a 2-byte number.
Service-name	If the entry you are deleting has a name, specify the name.

Example: **delete filter 2 039B NOTES-CHICAGO**

ip-tunnel-address *ip-address*

Deletes an IP address from the IP address peer list.

ip-address	IP unicast or multicast address that makes up the peer list

Syntax: delete ip-tunnel-address ip-address

Example: **delete ip-tunnel-address <ip-address>**

Note: Deleting all addresses from the peer list results in a non-operational tunnel.

Disable

Disable specific IPX interfaces or globally disable the IPX protocol.

Syntax: disable interface . . .
ipx
ipxwan . . .
ip-tunnel
keepalive . . .
netbios . . .
reply-to-get-nearest-server

interface *interface#*

Prevents the router from sending IPX packets over specific interfaces.

Example: **disable interface 2**

ipx

Prevents the router from sending IPX packets over any of the interfaces.

Example: **disable ipx**

ipxwan *interface#*

Prevents IPX from functioning over an interface supporting the point-to-point protocol.

Example: **disable ipxwan interface 2**

ip-tunnel

The disable tunnel command disables IPX on the IP network.

Example: **disable ip-tunnel**

keepalive *interface#*

Prevents the router from responding to keepalive packets on the specified interface.

Example: **disable keepalive 3**

Example: **disable keepalive**
```
Which interface [0]?
```

netbios *interface#*

Prevents the router from filtering NetBIOS packets on the specified interface.

```
Example: disable netbios 3
```

```
Example: disable netbios
         Which interface [0]?
```

reply-to-get-nearest-server

Prevents the router from responding to GET NEAREST SERVER requests from workstations that are attempting to locate a server.

Note: Disable this feature with great caution. Use this command only when there are multiple routers (or servers) on an IPX network and it is known that the "best" server is not behind this router.

```
Example: disable reply 3
```

Enable

Enable specific IPX interfaces or to globally enable the IPX protocol.

Syntax: enable interface . . .
ipx
ipxwan . . .
ip-tunnel
keepalive . . .
netbios . . .
reply-to-get-nearest-server

interface *interface#*

Allows the router to send IPX packets over specific interfaces.

```
Example: enable interface 2
```

ipx

Allows the router to send IPX packets over all of the enabled interfaces.

Example: **enable ipx**

ipxwan ***interface# timeout retry_timer***

Allows the routing of IPX traffic over an interface that supports the point-to-point protocol. This command also queries for a connection timer value and a retry timer value. The enable command prompts for the same parameters as the **set ipxwan** command**.** This allows you to initially set IPXWAN parameters without having to use the set command. If you need to modify preconfigured parameters, then use the **set ipxwan** command.

Example: **enable ipxwan 0 60 60**

ip-tunnel ***net# rip_interval sap_interval***

The enable tunnel command enables IPX on the IP network.

net#	The hexadecimal form of the network number.
rip_interval	The RIP timer interval in seconds. The range is 1 – 1440. The default is 1.
sap_interval	The SAP timer interval in seconds. The range is 1 – 1440. The default is 1.

Syntax: enable ip-tunnel *net# rip_interval sap_interval*

Example: **enable ip-tunnel**

keepalive ***interface# timeout***

Allows the router to respond to keepalive packets on the interface specified by *interface#* for the number of minutes specified in *timeout*. The allowable values for *timeout* are from 0 through 1440 minutes (24 hours) with a default of 0 minutes (no timeout). This command is valid only for dial circuits.

Note: You can use a nonzero timeout to try to overcome file server limitations on the number of connections that it allows. Setting a nonzero timeout value can help to terminate idle connections.

```
Example: enable keepalive 3 60

Example: enable keepalive
         Which interface [0]?
         Timeout (in min) [0]?
```

netbios *interface#*

Allows the router to filter NetBIOS packets on the specified interface.

```
Example: enable netbios 3

Example: enable netbios
         Which interface [0]?
```

reply-to-get-nearest-server

Allows the router to respond to GET NEAREST SERVER requests from workstations that are attempting to locate a server. This is the default setting.

```
Example: enable reply 2
```

Frame

Specify the packet format for IPX interfaces. (Encapsulation can also be set using the CONFIG **network** command.)

Note: When there are incorrect or invalid configuration records, the default frame values are used.

Syntax: frame ethernet_II . . .
ethernet_8022 . . .
ethernet_8023 . . .
ethernet_SNAP . . .
token–ring MSB . . .
token–ring LSB . . .
token–ring_SNAP MSB. . .
token–ring_SNAP LSB. . .
FDDI . . .
FDDI_SNAP . . .

ethernet_*type interface#*

Selects the Ethernet encapsulation format. This is required if you are using NetWare-VMS on the Ethernet, and is often used when there are ISO nodes on the same Ethernet. The following options are available:

- ethernet_II (default of NetWare 4.0 and greater) – Ethernet_II uses Ethernet version 2.0 protocol 81–37.
- ethernet_8022 – Ethernet_8022 uses Ethernet 802.3 with 802.2 SAP E0.
- ethernet_8023 (default of pre-NetWare 4.0 and lower), router default – Ethernet_8023 uses Ethernet 802.3 without any 802.2 header.
- ethernet_SNAP – Ethernet_SNAP uses 802.3, 802.2 with SNAP PID 00–00–00–81–37.

Note: The ethernet_SNAP encapsulation it is not architecturally valid and is not fast-pathed. No cache entries appear for network entries using this encapsulation.

The default value for Ethernet frames is “ethernet_8023.”

Example: **frame ethernet_II 4**

token–ring_*type interface#*

Selects the token ring encapsulation format. The default value is “token–ring MSB.” The following options are available:

- token-ring MSB . . .
- token-ring LSB . . .
- token-ring_SNAP MSB . . .
- token-ring_SNAP LSB . . .

Token-ring MSB uses 802.5, 802.2 SAP E0.

Example: **frame token–ring MSB 3**

Token-ring SNAP uses 802.5, 802.2 with SNAP PID 00–00–00–81–37.

Example: **frame token–ring_SNAP MSB 3**

FDDI_type *interface#*

Selects the FDDI encapsulation format. FDDI_SNAP is the default value for FDDI interfaces. The following options are available:

- FDDI *interface#*
- FDDI_SNAP *interface#*

FDDI uses 802.2 SAP E0.

Example: **frame FDDI 4**

FDDI_SNAP uses 802.2 SNAP PID 00–00–00–81–37.

Example: **frame FDDI_SNAP 4**

List

Display the current IPX configuration.

Syntax: list

Example: **list**

```
IPX globally                enabled
Host number (serial line)   000000000000
Router Name (IPXWAN)
NodeID (IPXWAN)             0
Maximum networks            32
Maximum total alt. route entries        128
Maximum alt. routes per dest. network   3
Maximum services                        32
Maximum Network Cache entries           64
Maximum Local Cache entries             64


List of configured interfaces:
                Frame
Ifc  IPX net #  Encapsulation       SAP nearest server reply   IPXWAN

0    177        FDDI_SNAP              Enabled                 N/A
1    183        TOKEN-RING      MSB    Enabled                 N/A
2    184        TOKEN-RING      MSB    Enabled                 N/A


RIP/SAP Timer Intervals
Net  IPX net #        SAP Interval(Minutes)   RIP Interval(Minutes)
  0       177                   1                       1
  1       183                   1                       1
  2       184                   1                       1
IPX SAP Filter is: disabled
No IPX SAP Filter records in configuration.
IPX Access Controls are: disabled
No IPX Access Control records in configuration.


NetBIOS Filtering Configuration
Net   IPX net #         NetBIOS Filtering
  3         177         Enabled
  4         178         Disable
  5         179         Enabled


Keepalive Configuration
Net   IPX net #         Keepalive                Timeout(Minutes)
  3         177         Enabled                          60
  4         178         Disabled                         60
  5         179         Enabled                        None
```

IPX globally	Indicates if IPX is globally enabled or disabled.
Host number	The host number assigned to IPX. You can change this number with the IPX **set** command.
Router name	The user-assigned router name for IPXWAN.
Node ID	The user-assigned node-id for IPXWAN.
Maximum networks	The size of the IPX RIP routing table, which is the maximum number of IPX networks.
Maximum routes	The number of configured maximum routes, which is the maximum number of routes to IPX networks.
Maximum routes-per-network	The number of configured maximum routes-per-network.
Maximum services	The size of the IPX SAP service table.
Maximum network cache entries	The number of network cache entries.
Maximum local cache entries	The number of local cache entries.
List of configured interfaces	Lists each interface number and its associated IPX network number. It also displays the type of encapsulation enabled for that interface, whether the "get nearest server" feature is enabled, and which interface has IPX enabled for WAN traffic (IPXWAN).
IPX SAP filter	Indicates whether the IPX SAP filter function is enabled or disabled and whether there are any IPX SAP filters in the configuration.
IPX access controls	Indicates whether the IPX access controls are enabled or disabled and whether there are access control records in the configuration.
RIP/SAP Timer Intervals	The delay between the transmission of complete RIP and SAP advertisements on an interface.
IPX Net #	The IPX network number designated for a particular interface.

SAP Interval (Minutes)	The number of minutes between complete SAP advertisements on an interface. The range is 1 through 1440 (24 hours). The default is 1.
RIP Interval (Minutes	The number of minutes between complete RIP advertisements on an interface. The range is 1 through 1440 (24 hours). The default is 1.

Move

Change the line numbers for the access controls. After you move the lines, they are renumbered to reflect the new order.

Syntax: move access–control *line# line#*

Example: **move 5 2**

```
Are you sure this is what you want to do (Yes or [No]):
```

Set

Set the IPX configuration parameters listed below.

Syntax: set
- access–control . . .
- filter . . .
- host–number . . .
- ipxwan . . .
- keepalive . . .
- local–cache size. . .
- maximum alternate–routes–per–destination . . .
- maximum networks . . .
- maximum services . . .
- maximum total–alternate–route–entries . . .
- name . . .
- net–number . . .
- node–id . . .
- remote–cache size
- rip–update–interval . . .
- sap–update–interval . . .
- rip–ip–tunnel–update–interval . . .
- sap–ip–tunnel–update–interval . . .

access–control *toggle*

Turns the access controls globally on or off. Enter **on** or **off** as the toggle value.

Example: **set access-control on**

filter toggle

Turns the IPX SAP filters globally on or off. Enter **on** or **off** as the toggle value.

Example: **set filter on**

host–number ***host#***

Specifies the host number used for serial interfaces running IPX. Each IPX router operating over serial lines must have a unique host number. This is required because the serial lines do not have hardware node addresses from which to build a host number. The host number is a 12-digit hexadecimal number. This number must be unique on each router.

Example: **set host-number 0000000000F4**

ipxwan ***interface# timeout retry_timer***

Sets or modifies an interface to support the routing of IPX traffic over a PPP interface that supports the point-to-point protocol. Before the **set ipxwan** command can be invoked, IPXWAN must be enabled on the interface using the **enable ipxwan** command. This command also queries for a connection timer value and a retry timer value.

Example: **set ipxwan 0 60 60**

keepalive ***interface# timeout***

Sets or modifies an interface specified by *interface#* to respond to keepalive packets for the number of minutes specified in *timeout*. The allowable values for *timeout* are from 0 through 1440 minutes (24 hours) with a default of 0 minutes (no timeout). This command is valid only for dial circuits.

Note: You can use a nonzero timeout to try to overcome file server limitations on the number of connections that it allows. Setting a nonzero timeout value can help to terminate idle connections.

Example: **set keepalive 3 60**

```
Example: set keepalive
         Which interface [0]?
         Timeout (in min) [0]?
```

local–cache size #

Specifies the size of the routing table local cache. The range is 1–10000. The default size is 64. For a description of local and remote cache refer to the *Routing Protocols Reference Guide*.

Example: **set local–cache size 64**

maximum alternate–routes–per–destination #

Specifies the number of alternate routes that you want to assign to a given destination network. The range is 1–4096. The default value is 32.

Example: **set maximum alternate–routes–per–destination 8**

maximum networks #

Specifies the size of IPX's RIP routing table. This reflects the number of networks in the IPX internet on which the router operates. The range of values is 1 – 2048. The default is 32.

Example: **set maximum networks 30**

maximum services #

Specifies the size of IPX's SAP service table. This reflects the number of services (such as file servers or SNA gateways) on the IPX internet on which the router operates. The range is 1 – 2048. The default is 32.

Example: **set maximum services 30**

maximum total–alternate–route–entries #

Specifies the total number of entries available for alternate routes.

Example: **set maximum total–alternate–route–entries 40**

name

Lets you assign a symbolic name to the router. IPXWAN requires that a router have a primary network number and a name. The name can be from 1 to 47 characters in length and can contain the characters "A" through "Z", underscore (_), hyphen (–), and the "at" sign (@).

Example: **set name newyork_accounting**

net–number ***interface# ipx-net#***

Assigns an IPX network number to the associated directly-connected network. Every IPX interface must have a unique network number. The only exception is that serial lines can be assigned network numbers of zero. (Serial lines without network numbers do not pass IPX NETBIOS emulation packets.) The interface number is decimal and the net number is hexadecimal number. The IPX net number is 8 digits in hex (1 – FFFFFFFF).

Example: **set net-number 2 180**

node–id ***primary–net#***

Lets you assign a primary network number. IPXWAN requires a router to have a primary network number and a name. The "node–id" is the primary network number for the router and must be assigned before the exchange of IPXWAN packets can begin. The *primary–net#* must be a 1- to 8-digit hexadecimal number.

This number is for the router as a whole. In NetWare file server terms, it is the "internal" network number. This number must be unique among all the network numbers in the IPX internet.

Example: **set node–id 500**

remote–cache size ***#***

Specifies the size of the routing table remote cache. The range is 1 – 10000. The default size is 32. For a description of local and remote cache refer to the *Routing Protocols Reference Guide*.

Example: **set remote–cache size 64**

```
New IPX remote network cache size [32]?
```

rip–update–interval ***interface# delay#***

Specifies the time delay in minutes between transmissions of complete RIP updates on an interface. The range is 1 through 1440. The default is 1. In the following example the RIP interval on interface 0 is being set to 2 minutes.

Example: **set rip–update–interval 0 2**

sap–update–interval *interface# delay#*

Specifies the time delay in minutes between transmissions of complete SAP updates on an interface. The range is 1 through 1440. The default is 1. In the following example the SAP interval on interface 0 is being set to 2 minutes.

Example: **set sap–update–interval 0 2**

rip–ip–tunnel–update–interval *interface# delay#*

Specifies the time delay in minutes between transmissions of complete RIP updates through an IP tunnel on an interface. The range is 1 through 1440. The default is 1. In the following example, the RIP IP tunnel interval on interface 0 is being set to 2 minutes.

Example: **set rip–ip–tunnel–update–interval 0 2**

sap–ip–tunnel–update–interval *interface# delay#*

Specifies the time delay in minutes between transmissions of complete SAP updates through an IP tunnel on an interface. The range is 1 through 1440. The default is 1. In the following example, the SAP IP tunnel interval on interface 0 is being set to 2 minutes.

Example: **set sap–ip–tunnel–update–interval 0 2**

Exit

Return to the Config> prompt.

Syntax: exit

Example: **exit**

16

Monitoring IPX

This chapter describes how to monitor IPX protocol activity and use the IPX console commands.

For additional information about the IPX protocol, refer to the *Routing Protocols Reference Guide*.

Accessing the IPX Console Environment

For information about accessing the IPX console environment, see Chapter 1.

IPX Console Commands

The IPX console commands allow you to view the parameters and statistics of the interfaces and networks that transmit IPX packets. Table 16–1 provides a summary of console commands.

Enter the IPX console commands at the `IPX>` prompt.

Table 16–1 IPX Console Command Summary

Command	Function
? (Help)	Lists all the IPX console commands or lists the options associated with specific commands.
Access–controls	Lists the status of IPX access controls, the IPX access control statements, and a count of how many times each control statement was followed.
Cache	Lists the current contents of the routing cache.
Configuration	Lists the network numbers of the router interfaces on which IPX is enabled, the IPX frame/encapsulator types and the RIP/SAP timer intervals.
Counters	Displays the number of routing errors, destination errors, and packet overflows.
Disable interface	Disables specific IPX interfaces.
Dump routing tables	Displays the contents of the current IPX RIP routing tables.
Enable interface	Enables specific IPX interfaces.
Filters	Lists the current SAP filters and the state of each filter.
IPXWAN	Lists configuration information about IPX running over a WAN interface through the point-to-point protocol.
Shutdown	Performs an orderly shutdown of IPX functions on all router interfaces.
Sizes	Displays the configured sizes and contents for the local node and remote network caches.
Slist	Displays the contents of the current IPX SAP routing tables.
Exit	Exits the IPX console process and returns to the GWCON environment.

? (Help)

List the commands that are available from the current prompt level. You can also enter **?** after a specific command name to list its options.

Syntax: ?

Example: **?**

```
ACCESS-CONTROLS
CACHE
CONFIG
COUNTERS
DISABLE interface
DUMP routing tables
ENABLE interface
EXIT
FILTERS
IXPWAN
SIZES
SLIST
```

Access Controls

List the status of IPX access controls, the IPX access control statements, and a count of how many times each control statement was followed.

Syntax: access-controls

Example: **access-controls**

```
Access Control currently enabled
List of IPX Access Control records:
# T Dest Net         Host  Sck  Sck  Src Net         Host  Sck Sck Count
1 E      179 123456789ABC 1234 1234      176 000000000000    0   0     0
```

Cache

Display the contents of the IPX routing cache.

Syntax: cache

Example: **cache**

```
Dest Net/Node         Use Count    via Net/Node         via Int
152                    56000       161/000000000006     SL/0
162                    56476       162/000000000000     Eth/0
162/0000C0239F71       56476       162/0000C0239F71     Eth/0
```

The first entry shows that the remote network 152 can be reached over the serial link with an IPX network number of 161. The second entry is the IPX network 162. It is an Ethernet directly attached to the router. This entry is a general local network entry. There is one general local network entry for each of the directly attached networks after they have begun forwarding IPX packets. The last entry is a local entry on an Ethernet. This IPX cache entry was used to send 56,746 packets to the IPX node number 0000C0239F71 on net number 162.

Configuration

List the network, encapsulation information, and RIP/SAP timer intervals of all the router interfaces on which the IPX protocol is enabled.

Syntax: configuration

Example: **configuration**

```
Router Configuration
Net  Name   Type                          Network/Address
  0  Eth/0  SCC Ethernet                   2/08002BB21BFB
  1   SL/0  SCC Serial Line                12/000000000004
  2   SL/1  SCC Serial Line                11/000000000004

IPX Encapsulation/Frame Types
Net  Name   Type                          Encapsulation
  0  Eth/0  SCC Ethernet                   ETHERNET_II
  1   SL/0  SCC Serial Line                N/A
  2   SL/1  SCC Serial Line                N/A

RIP/SAP Timer Intervals
Net  Name   Type                          SAP Interval   RIP Interval
  0  Eth/0  SCC Ethernet                       1             1
  1   SL/0  SCC Serial Line                    1             1
```

Router Configuration

Net	Specifies the interface number.
Name	Specifies the interface name.
Type	Specifies the hardware type of the interface.
Network/ Address	Specifies the user-assigned network number and host number. Except for serial lines, the host number is the node address of the network interface. For serial lines, it is the user-configured IPX host number.
IPX Encapsulation/Frame Types	Displays current configured encapsulation and frame type information by interface.
Net	Specifies the interface number.
Name	Specifies the interface name.
Type	Specifies the hardware type of the interface.

Encapsulation	Displays the encapsulation/frame type configured for a particular interface.
RIP/SAP Timer Intervals	Indicates the delay between the transmission of complete RIP and SAP advertisements on an interface.
Net	Indicates the interface number.
Name	Indicates the interface name.
Type	Specifies the hardware type of the interface.
SAP Interval	Indicates the number of minutes between complete SAP advertisements on an interface. The range is 1 through 1440. The default is 1.
RIP Interval	Indicates the number of minutes between complete RIP advertisements on an interface. The range is 1 through 1440. The default is 1.

Counters

Display the number of routing errors, destination errors, and packet overflows that have occurred. In the example, the counters show no recorded errors.

Syntax: counters

Example: **counters**

```
Routing errors
Count     Type
   0      Unknown
   0      Checksum error
   0      Destination unreachable
   0      Hop count expired
   0      Interface size exceeded

Destination errors
Count      Type
   0       Unknown
   0       Checksum error
   0       Non-existent socket
   0       Congestion

IPX input packet overflows
    Net     Count
   Eth/0    0
   TKR/0    0
```

Disable

Disable specific IPX interfaces from sending IPX packets, or globally disable IP tunneling. The interface can later be re-enabled using the **enable** command.

Syntax: disable *interface #*
ip-tunnel

interface *interface#*

Prevents the router from sending IPX packets over specific interfaces.

Syntax: disable *interface #*

Example: **disable interface 2**

ip-tunnel

The disable tunnel command disables IPX on the IP network.

Syntax: disable ip-tunnel

Example: **disable ip-tunnel**

Dump Routing Tables

Display the contents of the current IPX RIP network routing tables.

Syntax: dump routing tables

Example: **dump routing tables**

The screen displays the following information:

```
Type  Dest net  Hops  Delay  Age(M:S)     via Router
 Dir      124     0      1       0: 0     124/AA0004001A04
 Dir      131     0      1       0: 0     131/00000000001A
 Dir      177     0      1       0: 0     177/00000000001A
 Dir       41     0      1       0: 0      41/4000C90401FA
 Dir      249     0      1       0: 0     249/0000C9084F34
 RIP      250     1      2       0:10     249/0000C9093250
 RIP 2C39ABE9     2      3       0:10     249/0000C9093250
 RIP       BB     1      2       0:50      41/4000C9050971
 RIP        1     2      3       0:50      41/4000C9050971
 RIP       31     2      3       0:50      41/4000C9050971
 RIP      703     1      2       0:20      41/4000C9041243
 RIP      704     1      2       0:30      41/4000C9041243
```

```
12 route entries used out of 32
12 net entries used out of 32
```

Type Specifies one of the following:

- **Dir** – Specifies that this network is directly connected to the router.
- **RIP** – Specifies that this route was provided by the IPX routing protocol.
- **Old** – Specifies that this route has timed out and is no longer being used. The route is kept in this state for a while to inform other routers that the route is bad. After that time, it is removed and is no longer displayed.

Dest net Specifies the destination network number.

Hops Specifies the number of router hops to this destination.

Delay Specifies the estimate of how long it takes for a packet to be transmitted and arrive at its destination. The unit of delay is the number of IBM PC clock ticks to send a 576 byte packet, which is 18.21 clock ticks per second. The minimum delay is 1 unit.

Age Specifies the age of the routing information in minutes and seconds. If an entry in the routing table is not updated, the router does the following:

- After 3 RIP timer intervals have passed, the router declares the route eligible for replacement and is no longer advertised. The route type is then specified as Old. A discussion on RIP timer intervals can be found in the **set rip-update-interval** configuration command section.
- After an additional 60 seconds, the route is garbage-collected and does not appear in the **dump** display.

Via router — Specifies the next hop for packets going to networks that are not directly connected. For directly-connected networks, this is the address of the router interface that transmits the packet.

At the bottom of the display is the number of route and network entries used and the total available. If all the network entries are used, it is likely that the routing table is not large enough. Use the IPX configuration **set maximum networks** command to increase the size.

If all of the route entries are used, then there may be routes to IPX networks that cannot be kept, including new, incoming networks. If you do not want to increase the number of available routes, reduce the number of alternate routes per network.

Enable

Interactively enable IPX on an interface. The interface does not transmit and receive IPX packets unless it has an IPX network number configured and has passed self-test (is up).

Syntax: enable *interface#*
ip-tunnel

interface#

Allows the router to send IPX packets over specific interfaces.

Syntax: enable *interface#*

Example: **enable 2**

ip-tunnel *net# rip_interval sap_interval*

The enable tunnel command enables IPX on the IP network.

net#	The hexadecimal form of the network number.
rip_interval	The rip timer interval in seconds. The range is 1 – 1440. The default is 1.
sap_interval	The sap timer interval in seconds. The range is 1 – 1440. The default is 1.

Syntax: enable ip-tunnel *net# rip_interval sap_interval*

Example: **enable ip-tunnel**

Filters

List the current filters and the state of these filters.

Syntax: filters

Example: **filters**

```
IPX SAP Filter currently enabled
List of IPX SAP Filter records:
Count  Max Hops  Type  Service Name
  0           8     4  ?
  0           1  1234  SomeServer
```

IPXWAN

List the current configuration information for IPX running over a WAN interface through the point-to-point protocol.

Syntax: ipxwan detailed . . .
summary . . .

detailed *intrfc#*

Lists the complete current configuration information for IPX running over a WAN interface through the point-to-point protocol.

Example: **ipxwan detailed 2**

```
Detailed information for IPXWAN link over interface 2, PPP/1
This side is the IPXWAN slave
Neighbor Name: SKYSURF2
Neighbor Node ID: 727299
Negotiated Routing Type: RIP/SAP
Link Delay: 330 1/18th sec ticks
Common Net#: 132
Connection Timeouts: 0
Connection Retries: 0
Timer Requests Sent: 1
Timer Requests Received: 1
Timer Responses Sent: 1
Timer Responses Received: 0
Info Requests Sent: 0
Info Requests Received: 1
Info Responses Sent: 1
Info Responses Received: 0
```

Neighbor Name	Router name of the neighbor as received in the RIP/SAP Information Request Packet.
Neighbor Node ID	Node ID (also known as the primary network number) of the neighbor. This is a IPX network number unique to the entire internetwork. It is a 32-bit quantity.
Negotiated Routing Type	Negotiated routing type. Digital currently supports RIP/SAP. The default is RIP/SAP.
Link Delay	Link delay in 1/18th second ticks calculated by the master. It is a 16-bit quantity. It is always calculated, therefore there is no default.
Common Net#	Network number agreed upon by both ends of the link. This number must be unique to the entire internetwork. It is a 32-bit quantity. There is no default, it must be negotiated.
Connection Timeouts	Number of times the connection timed out. A connection times out periodically if the exchange of IPXWAN packets does not proceed. The timeout period is configurable. The default for the timeout period is 60 seconds.

Connection Retries	Number of times the connection is retried after timing out. The amount of time to wait after timing out before retrying is configurable. It defaults to 60 seconds.
Timer Requests Sent	Number of IPXWAN Timer Request packets sent.
Timer Requests Received	Number of IPXWAN Timer Request packets received.
Timer Responses Sent	Number of IPXWAN Timer Response packets sent.
Timer Responses Received	Number of IPXWAN Timer Response packets received.
Info Requests Sent	Number of IPXWAN Information Request packets sent.
Info Requests Received	Number of IPXWAN Information Request packets received.
Info Responses Sent	Number of IPXWAN Information Response packets sent.
Info Responses Received	Number of IPXWAN Information Response packets received.

summary

Lists a summary of the current configuration information for IPX running over WAN interfaces through the point-to-point protocol.

Example: **ipxwan summary**

```
Net  Name   Common Net#  NodeID  Neighbor Name

6    PPP/1  132          727299  SKYSURF2
```

Net	Network interface number.
Name	Network interface name.
Common Net#	Network number agreed upon by both ends of the link. This number must be unique to the entire internetwork.

NodeID	Node ID (also known as the primary network number) of the neighbor. This is a IPX network number unique to the entire internetwork.
Neighbor Name	Router name of the neighbor as received in the RIP/ SAP Information Request Packet.

Shutdown

Performs an orderly shutdown of IPX functions on all router interfaces.

Syntax: shutdown

Example: **shutdown**

Sizes

List the current size and number of entries in use for the local node and remote network caches.

Syntax: sizes

Example: **sizes**

```
Current IPX cache size:
Remote network cache size(max entries): 64
       2 entries now in use
Local node cache size(max entries): 128
       1 entries now in use
```

Slist

Display the contents of the current IPX SAP tables. This command is similar to the NetWare **slist** command.

Syntax: slist

Example: **slist**

```
State Typ Service Name Hops Age(M:S)  Net/   Host     /Sock
 SAP 0004 PCS12         3   0:50        1/000000000048/0451
 SAP 0004 ACMPCS        3   0:50        1/00000000004A/0451
 SAP 0004 DEVEL2        1   0:50       11/0000000000B4/0451
 SAP 0004 PLANNING      2   0:50       BB/0000000000B7/0451
 SAP 0004 DEVEL         2   0:50       BB/0000000000EE/0451
 SAP 0004 SOFT2         1   0:30      704/000000000094/0451
 SAP 0004 SKYSURF1      2   0: 5 2C39ABE9/000000000001/0451
 SAP 0278 DIRTREE       2   0: 5 2C39ABE9/000000000001/4005
 SAP 026B DIRTREE       2   0: 5 2C39ABE9/000000000001/0045


9 services used out of 32
```

State — Indicates one of the following:

- **SAP** – This service was provided by the SAP protocol.
- **Old** – This service has timed out and is no longer being used. The service is kept in this state for 5 to 10 seconds to inform other routers that the service is bad. After that time, it is removed and no longer displayed.

Typ — The server type in hexadecimal. File servers are type 0004. Other type numbers are assigned by Novell.

Service name — The server's unique name for this type of server. Only the first 30 characters of the 48-character name are printed to conserve space.

Hops — The number of router hops from this router to the server.

Age — Specifies the age of the service information. If an entry in the SAP table is not updated, the router does the following:

- After 3 SAP timer intervals have passed, the service is no longer used, but is broadcast as DEAD. The service state is then specified as DEL. A discussion on SAP timer intervals can be found in the **set sap-update-interval** configuration command section.
- After an addition 60 seconds, the service is garbage-collected and does not appear in the **slist** display.

Net/Host/Sock Specifies the address of the service. The address includes:

- Network number.
- Net host number (the address of the first interface on the network).
- Socket number at which the service can be reached.

At the bottom of the display is the number of entries used and the total available. If all the entries are used, it is likely that the service table is not large enough. Use the IPX configuration **set maximum services** command to increase the size.

Exit

Return to the GWCON(+) prompt.

Syntax: exit

Example: **exit**

17

Configuring OSPF

This chapter describes how to configure the OSPF protocol and how to use the OSPF configuration commands.

For more information about OSPF, refer to the *Routing Protocols Reference Guide*.

Accessing the OSPF Configuration Environment

For information about accessing the OSPF configuration environment, see Chapter 1.

Basic Configuration Procedures

The following sections present information about how to initially configure the OSPF protocol. This information outlines the tasks required to get the OSPF protocol up and running. Information about how to make further configuration changes is explained in the command sections of this chapter.

Before You Begin

Before your router can run the OSPF protocol, you must do the following:

1. Enable the OSPF protocol. In doing so, you must estimate the final size of the OSPF routing domain.
2. Define OSPF areas attached to the router. If no OSPF areas are defined, a single backbone area is assumed.

3. Define the router's OSPF network interfaces. The cost of sending a packet out each interface must be set, along with a collection of the OSPF operating parameters.
4. If you want to forward IP multicasts (IP Class D addresses), enable IP multicast routing capability.
5. If the router interfaces to non-broadcast networks, you must also set the non-broadcast network parameters. This consists of a list of the other OSPF routers that are connected to the non-broadcast network.
6. If you want the router to import routes learned from other routing protocols (EGP, RIP or statically configured routes), you have to enable AS boundary routing. In addition, you must define whether routes are imported as Type 2 or Type 1 externals.
7. If you want to boot a neighboring router over an attached point-to-point interface, the neighbor's IP address must be configured. This is done by defining non-broadcast parameters for the point-to-point interface.

Enabling the OSPF Protocol

When enabling the OSPF routing protocol, you must supply the following two values to estimate the size of the OSPF link state database. These two values are configured identically in all of your OSPF routers.

- Total number of AS external routes that are imported into the OSPF routing domain. A single destination may lead to multiple external routes when it is imported by separate AS boundary routers. For example, if the OSPF routing domain has two AS boundary routers, both importing routes to the same 100 destinations, the number of AS external routes is set to 200.
- Total number of OSPF routers in the routing domain.

To enable the OSPF routing protocol, use the **enable** command as shown in the following example.

```
OSPF Config> enable ospf
Estimated # external routes[0]? 200
Estimated # OSPF routers [0]? 60
```

Defining Attached OSPF Areas

The next step in the configuration process is setting the parameters that define the OSPF areas that are directly attached to the router. If no areas are defined, the router software assumes that all the router's directly attached networks belong to the backbone area (area ID 0.0.0.0).

To set the parameters for an OSPF area, use the **set area** command and respond to the following prompts:

```
OSPF Config> set area
Area number [0.0.0.0]? 0.0.0.1
Authentication type [1]? 1
Is this a stub area? (Yes or No): no
```

- Area number is the OSPF area address. An OSPF area is a contiguous group of networks that is defined by a list of address ranges, each indicated by a combination of the IP address and an address mask. A network belongs to an area if its address is in the list.

- Authentication type (security scheme) to be used in the area. The choices for authentication types are 1, which indicates a simple password; or 0, which indicates that no authentication is necessary for the exchange.

- Stub area designation. If you designate YES,

  – The area does not receive any AS external link advertisements, reducing the size of the area's OSPF database and decreasing memory usage for external routers in the stub area.

  – You cannot configure virtual links through a stub area.

  – You cannot configure a router within the stub area as an AS boundary router.

External Routing in Stub Areas. You cannot configure the backbone as a stub area. External routing in stub areas is based on a default route. Each border area router attaching to a stub area originates a default route for this purpose. The cost of this default route is also configurable in the OSPF **set area** command.

Setting OSPF Interfaces

To set the OSPF parameters for the router's network interfaces, use the **set interface** command.

When responding to the prompts, supply the interface's IP address for each interface in the router and answer the questions that follow. For the parameters listed below you must enter the same value for all routers attached to a common network segment.

- Hello interval
- Dead router interval
- Authentication key (if an authentication type of 1 (simple password) is used)

The first prompt asks for the OSPF area to which the interface attaches. In the following example, suppose that the interface address mask is 255.255.255.0, indicating that the interface attaches to a subnet (128.185.138.0) of network 128.185.0.0. All other OSPF routers attached to subnet 128.185.138.0 must also have their *hello interval* set to 10, *dead router interval* set to 40, and their interface *authentication key* set to xyz_q.

```
OSPF Config> set interface
Interface IP address [0.0.0.0]? 16.24.11.251
Attaches to area [0.0.0.0]?
Retransmission Interval (in seconds) [5]?
Transmission Delay (in seconds) [1]?
Router Priority [1]?
Hello Interval (in seconds) [10]?
Dead Router Interval (in seconds) [40]?
Type Of Service 0 cost [1]? 2
Authentication Key []?
Retype Auth. Key []?
```

Setting Non-Broadcast Network Interface Parameters

If the router is connected to a non-broadcast, multi-access network, such as an X.25 PDN, you have to configure the parameters below to help the router discover its OSPF neighbors. This configuration is only necessary if the router is eligible to become the designated router of the non-broadcast network.

First configure the OSPF poll interval with the following command:

```
OSPF Config> set non-broadcast
Interface IP address [0.0.0.0]? 128.185.138.19
Poll Interval [120]?
```

Then configure the IP addresses of all other OSPF routers that are attached to the non-broadcast network. For each router configured, you must also specify its eligibility to become the designated router.

```
OSPF Config> add neighbor
Interface IP address [0.0.0.0]? 128.185.138.19
IP Address of Neighbor [0.0.0.0]? 128.185.138.21
Can that router become Designated Router [Yes]?
```

Enabling IP Multicast Routing

To enable the routing of IP multicast (class D) datagrams, you must invoke the **enable multicast-routing** command in the OSPF configuration console. This command is described below. When enabling multicast routing, you are also prompted as to whether you want the router to forward multicasts between OSPF areas as well as whether you want the router to forward multicasts between Autonomous Systems.

```
OSPF Config>enable multicast
Inter-area multicasting enabled(Yes or No): yes
Inter-AS multicasting enabled(Yes or No): yes
```

When the above command is first invoked to enable multicast forwarding, multicast is enabled on all OSPF interfaces with default parameters. The interface parameters can be later modified using the OSPF **set interface** command.

Unless the **enable multicast-routing** command is invoked, forwarding of IP multicast datagrams is disabled. In other words, by default the router does not forward IP class D datagrams.

Enabling AS Boundary Routing

To import routes learned from other protocols (EGP, RIP, and statically configured information) into the OSPF domain, enable AS boundary routing. You must do this even if the only route you want to import is the default route (destination 0.0.0.0).

When enabling AS boundary routing, you are asked which external routes you want to import. You can choose to import, or not to import, routes belonging to several categories. The categories are as follows:

- RIP routes
- EGP routes
- BGP routes
- Static routes

- Direct routes

For example, you can choose to import EGP and direct routes, but not RIP or static routes. When you choose to import EGP routes, only the routes that appear in the EGP input exchange tables are actually imported. All routes are imported with cost equal to their routing table cost. They are all imported as either type 1 or type 2 external routes, depending on the routing protocol comparison.

Independent of the external categories, you can also configure whether or not to import subnet routes into the OSPF domain. This configuration item defaults to OFF (subnets not imported).

The metric type used in importing routes determines how the imported cost is viewed by the OSPF domain. When comparing two type 2 metrics, only the external cost is considered in picking the best route. When comparing two type 1 metrics, the external and internal costs of the route are combined before making the comparison.

You are asked whether or not you want to originate an OSPF default route. You can answer always, never, or only if you have EGP routes. If originating a default route when EGP routes are available, you can also choose to originate the default only if EGP routes are received from a particular Autonomous System or if a particular route is received through the EGP.

Combinations of these options are possible. For example, you can set the router so that its default is originated only if a route to 10.0.0.0 is received from AS number 12. Setting the AS number to 0 means “from any AS.” Setting the network number to 0.0.0.0 means “any routes received.”

The syntax of the **enable** command is as follows:

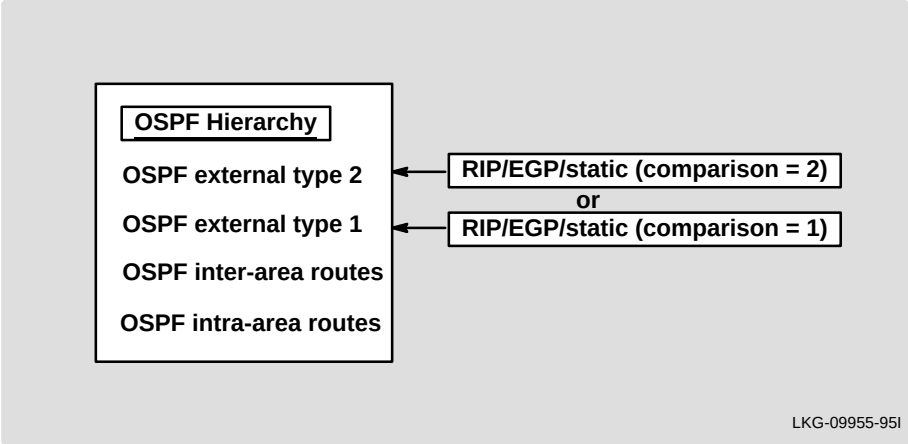
```
OSPF Config>enable as boundary
Import EGP routes? [No]: y
Import BGP routes? [No]:
Import RIP routes? [No]:
Import static routes? [No]:
Import direct routes? [No]: y
Import subnet routes? [No]:
Always originate default route? [No]:
Originate default if EGP/BGP routes available? [No]: y
   From AS number [0]? 12
   To network number [0.0.0.0]? 10.0.0.0
Originate as type 1 or 2 [2]?
Default route cost [1]?
Default forwarding address [0.0.0.0]?
```

Configuring For Routing Protocol Comparisons

If you use a routing protocol in addition to OSPF, or when you change your routing protocol to OSPF, you must set the Routing Protocol Comparison.

OSPF has a 4-level routing hierarchy (see Figure 17–1). The **set comparison** command tells the router where the EGP/RIP/static routes fit in the OSPF hierarchy. The two lower levels consist of the OSPF internal routes. OSPF intra-area and inter-area routes take precedence over information obtained from any other sources, all of which are located on a single level.

Figure 17–1 OSPF Routing Hierarchy



To put the EGP/RIP/static routes on the same level as OSPF external type 1 routes, set the comparison to 1. To put the EGP/RIP/static routes on the same level as OSPF external type 2 routes, set the comparison to 2. The default setting is 2.

For example, suppose the comparison is set to 2. In this case, when RIP routes are imported into the OSPF domain, they are imported as type 2 externals. All OSPF external type 1 routes override received RIP routes, regardless of metric. However, if the RIP routes have a smaller cost, the RIP routes override OSPF external type 2 routes. The comparison values for all of your OSPF routers must match. If the comparison values set for the routers are inconsistent, your routing does not function properly.

The syntax of the **set comparison** command is as follows:

```
OSPF Config> set comparison
Compare to type 1 or 2 externals [2]?
```

Setting Virtual Links

To maintain backbone connectivity you must have all of your backbone routers interconnected either by *permanent* or *virtual* links. Virtual links may be configured between any two area border routers that share a common non-backbone and non-stub area. Virtual links are considered to be separate router interfaces connecting to the backbone area. Therefore, you are asked to also specify many of the interface parameters when configuring a virtual link.

Note: If the router is an ABR that does not directly attach to the backbone area (0.0.0.0), you must create a virtual link. Logically, virtual links belong to area 0.0.0.0 and you must configure ABR to set area 0.0.0.0.

The example below illustrates the configuration of a virtual link. Virtual links must be configured in each of the link's two endpoints. Note that OSPF router IDs are entered in the same form as IP addresses.

```
OSPF Config> set virtual
Virtual endpt.  (Router ID) [0.0.0.0]? 128.185.138.21
Link's transit area [0.0.0.1]? .0.0.1
Retransmission Interval (in seconds) [10]?
Transmission Delay (in seconds) [5]?
Hello Interval (in seconds) [30]?
Dead Router Interval (in seconds) [180]?
Authentication Key []? 3-14159
```

OSPF Router IDs

Every router in an OSPF routing domain must be assigned a 32-bit router ID. The current OSPF implementation sets the OSPF router ID to be the address of the first OSPF interface appearing in the router's configuration.

The OSPF router ID can also be explicitly set by the IP **set router id** command. In this case, the router ID must still be one of the router's IP interface addresses.

Converting from RIP to OSPF

To convert your Autonomous System from RIP to OSPF, install OSPF one router at a time, leaving RIP running. Gradually, all your internal routes shift from being learned through RIP to being learned by OSPF (OSPF routes have precedence over RIP routes). If you want to have your routes look exactly as they did under RIP (in order to check that the conversion is working properly), use hop count as your OSPF metric. This is done by assigning the cost of each OSPF interface to 1.

Remember that the size of your OSPF system must be estimated when the protocol is enabled. This size estimate should reflect the final size of the OSPF routing domain.

After installing OSPF on your routers, turn on AS boundary routing in all those routers that still need to learn routes through other protocols (EGP, RIP, and statically configured routes). Keep the number of these AS boundary routers to a minimum.

Finally, you can disable the receiving of RIP information about all those routers that are not AS boundary routers.

Dynamically Changing Interface Costs

The cost of an OSPF interface can be dynamically changed from the router's console interface. This new cost is flooded quickly throughout the OSPF routing domain, and modifies the routing immediately.

When the router restarts/reloads, the cost of the interface reverts to the value that was configured in SRAM. In future router releases, you are also able to set the cost of an OSPF interface using the SNMP protocol.

OSPF Configuration Commands

Before you can use OSPF, you must configure it using the OSPF configuration commands. The following section summarizes and then explains the OSPF commands. These commands are entered at the `OSPF config>` prompt.

Table 17–1 OSPF Configuration Command Summary

Command	Function
? (Help)	Lists the OSPF configuration commands or lists the options associated with specific commands.
Add	Adds to already existent OSPF information. You can add ranges to areas, and neighbors to non-broadcast networks.
Delete	Deletes OSPF information from SRAM.
Disable	Disables the entire OSPF protocol, AS boundary routing capability, or IP multicast routing.
Enable	Enables the entire OSPF protocol, AS boundary routing capability, or IP multicast routing.
Join	Configures the router to belong to one or more multicast groups.
Leave	Removes the router from membership in multicast groups.
List	Displays OSPF configuration.
Set	Establishes or changes the configuration information concerning OSPF areas, interfaces, non-broadcast networks, or virtual links. This command also allows you to set the way in which OSPF routes are compared to information gained from other routing protocols.
Exit	Exits the OSPF configuration process.

? (Help)

List the commands that are available from the current prompt level. You can also enter a **?** after a specific command name to list its options.

Syntax: ?

Example: **?**

```
ADD
DELETE
DISABLE
ENABLE
EXIT
JOIN
LEAVE
LIST
SET
```

Add

Add more information to already existing OSPF information. With this command you can add ranges to areas as well as neighbors to non-broadcast networks.

Syntax: add range . . .
neighbor . .

range *area# IP-address IP-address-mask*

Adds ranges to OSPF areas. OSPF areas are defined in terms of address ranges. External to the area, a single route is advertised for each address range. For example, if an OSPF area consists of all subnets of the class B network 128.185.0.0, it is defined as consisting of a single address range. The address range is specified as an address of 128.185.0.0 together with a mask of 255.255.0.0. Outside of the area, the entire subnetted network is advertised as a single route to network 128.185.0.0.

Example: **add range 0.0.0.2 128.185.0.0 255.255.0.0**

neighbor

Adds neighbors to non-broadcast networks. If the router is connected to a non-broadcast, multi-access network, such as an X.25 PDN, you have to use this command to help the router discover its OSPF neighbors. This configuration is only necessary if the router is eligible to become the designated router of the non-broadcast network. Configure the IP addresses of all other OSPF routers that are attached to the non-broadcast network. For each router configured, you must also specify its eligibility to become designated router.

Example: **add neighbor**

```
Interface IP address [0.0.0.0]? 128.185.138.19
IP Address of Neighbor [0.0.0.0]? 128.185.138.21
Can that router become Designated Router [Yes]?
```

Delete

Delete OSPF information from SRAM.

Syntax: delete range . . .
area . . .
interface . . .
neighbor . . .
non-broadcast . . .
virtual link

range *area# IP-address*

Deletes ranges from OSPF areas.

Example: **delete range 128.185.0.0 255.255.0.0**

area *area#*

Deletes OSPF areas from the current OSPF configuration.

Example: **delete area 0.0.0.1**

interface *interface-IP-address*

Deletes an interface from the current OSPF configuration.

Example: **delete interface 128.185.138.19**

neighbor

Deletes neighbors on non-broadcast networks from the current OSPF configuration.

Example: **delete neighbor**

```
Interface IP address [0.0.0.0]? 128.185.138.19
IP Address of Neighbor [0.0.0.0]? 128.185.138.21
```

non-broadcast *interface-IP-address*

Deletes non-broadcast network information from the current OSPF configuration.

Example: **delete non-broadcast 128.185.133.21**

virtual-link

Deletes a virtual link. Virtual links can be configured between any two backbone routers that have an interface to a common non-backbone area. Virtual links are used to maintain backbone connectivity and must be configured at both endpoints.

Example: **delete virtual-link**

```
Virtual endpoint (Router ID) [0.0.0.0]?
Link's transit area [0.0.0.1]?
```

Disable

Disable either the entire OSPF protocol or just the AS boundary routing capability.

Syntax: disable as boundary routing
multicast-forwarding
OSPF routing protocol

as boundary routing

Disables the AS boundary routing capability. When disabled, the router does not import external information into the OSPF domain.

Example: **disable as boundary routing**

multicast-forwarding

Disables IP multicast routing on all interfaces. When disabled, the router does not forward IP multicast (Class D) datagrams.

Example: **disable multicast-forwarding**

OSPF routing protocol

Disables the entire OSPF protocol.

Example: **disable OSPF routing protocol**

Enable

Enable either the entire OSPF protocol or just the AS boundary routing capability.

Syntax: enable as boundary routing
multicast-routing
OSPF routing protocol

as boundary routing

Enables the AS boundary routing capability that allows you to import routes learned from other protocols (EGP, RIP, and statically configured information) into the OSPF domain. See the "Basic Configuration Procedures" section of this chapter for more information about using this command.

Example: **enable as boundary routing**

```
Import EGP routes(Yes or No): yes
Import RIP routes(Yes or No): no
Import static routes(Yes or No): no
Import direct routes(Yes or No): yes
Import subnet routes(Yes or No): no
Originate default if EGP routes available []? yes
   From AS number [0]? 12
   To network number [0.0.0.0]? 10.0.0.0
Originate as type 1 or 2 [2]?
Default route cost [1]?
```

multicast-routing

Enables the forwarding of IP multicast (Class D) datagrams. When enabling multicast routing, you are also prompted whether you want to forward IP multicast datagrams between OSPF areas and between Autonomous Systems. To run MOSPF (OSPF with multicast extensions), a router currently running OSPF needs only to use this command. You do not need to re-enter its configuration information.

Example: **enable multicast–routing**

```
Inter-area multicasting enabled (Yes or No): yes
Inter-AS multicasting enabled (Yes or No): yes
```

OSPF routing protocol

Enables the entire OSPF protocol. When enabling the OSPF routing protocol, you must supply the following two values that are used to estimate the size of the OSPF link state database:

- Total number of AS external routes imported into the OSPF routing domain. A single destination may lead to multiple external routes when it is imported by separate AS boundary routers. For example, if the OSPF routing domain has two AS boundary routers, both importing routes to the same 100 destinations, the number of AS external routes is set to 200.

- Total number of OSPF routers in the routing domain.

```
Example: enable OSPF routing protocol

Estimated # external routes[0]? 200
Estimated # OSPF routers [0]? 60
```

Join

Configure the router as a member of a multicast group. When the router is the member of a multicast group, it responds to pings and SNMP queries sent to the group address.

Group membership can also be obtained in a more temporary (and more immediate) way through the **join** command in the OSPF monitoring console.

Syntax: join *multicast-group-address*

```
Example: join  224.185.0.0
```

Leave

Remove a router's membership in a multicast group. This prevents the router from responding to pings and SNMP queries sent to the group address.

Group membership can also be deleted in a more temporary (and more immediate) fashion through the **leave** command in the OSPF monitoring console.

Syntax: leave *multicast-group-address*

```
Example: leave 224.185.0.0
```

List

Display OSPF configuration information.

Syntax: list all
areas
interfaces
non-broadcast
virtual-links

all

Lists all OSPF related configuration information.

Example: **list all**

```
                    --Global configuration--
                OSPF Protocol:          Enabled
                # AS ext. routes:       100
                Estimated # routers:    20
                External comparison:    Type 2
                AS boundary capability: Enabled
                Import external routes: RIP
                Orig. default route:    No (0,0.0.0.0)
                Default route cost:     (1, Type 2)
                Default forward. addr.: 0.0.0.0
                Multicast forwarding:   Disabled

                    --Area configuration--
Area ID          AuType          Stub? Default-cost Import-summaries?
0.0.0.0          0=None           No         N/A           N/A

                    --Interface configuration--
IP address        Area              Cost  Rtrns  TrnsDly  Pri  Hello  Dead
16.24.8.251       0.0.0.0              2      5       1     1     10    40
16.24.11.251      0.0.0.0              2      5       1     1     10    40
17.1.1.251        0.0.0.0              2      5       1     1     10    60
18.1.1.251        0.0.0.0              2      5       1     1     10    60
16.24.10.251      0.0.0.0              2      5       1     1     10    40
135.24.10.251     0.0.0.0              2      5       1     1     10    40
192.24.18.251     0.0.0.0              2      5       1     1     10    40
25.1.1.251        0.0.0.0              2      5       1     1     10    60
```

OSPF protocol	Displays whether OSPF is enabled or disabled.
# AS ext. routes	Displays the estimated number of Autonomous System external routes. The router cannot accept more than this number of AS external routes.
Estimated # routers	Displays the estimated number of routers found in the OSPF configuration.
External comparison	Displays the external route type used by OSPF when importing external information into the OSPF domain and when comparing OSPF external routes to RIP/EGP routes.
AS boundary capability	Displays whether the router imports external routes into the OSPF domain.
Import external	Displays which routes are imported.
Orig default route	Displays whether the router imports a default into the OSPF domain. When the value is “YES”, a non-zero network number is displayed in parentheses. This indicates that the default route originates if and only if a route to that network is available.
Default route cost	Displays the cost and type that are used in the imported default route.
Default forward addr	Displays the forwarding address that is used in the imported default route.
Multicast forwarding	Displays whether IP multicast datagrams is forwarded.
Area-ID	Displays the attached area ID (area summary information)
AuType	Displays the method used for area authentication. “Simple-pass” means a simple password scheme is being used for the area’s authentication.

Stub area	Displays whether or not the area being summarized is a stub area. Stub areas do not carry an external route, resulting in a smaller routing database. However, stub areas cannot contain AS boundary routers, nor can they support configured virtual links.
Interface Configuration	For each interface, its IP address is printed, together with configured parameters. “Area” is the OSPF area to which the interface attaches. “Cost” indicates the TOS 0 cost (or metric) associated with the interface. “Rtrns” is the retransmission interval, which is the number of seconds between retransmissions of unacknowledged routing information. “TrnsDly” is the transmission delay, which is an estimate of the number of seconds it takes to transmit routing information over the interface (it must be greater than 0). “Pri” is the interface’s Router Priority, which is used when selecting the designated router. “Hello” is the number of seconds between Hello Packets sent over the interface. “Dead” is the number of seconds after Hellos cease to be heard that the router is declared down.
Virtual links	Lists all virtual links that were configured with this router as endpoint. “Virtual endpoint” indicates the OSPF Router ID of the other endpoint. “Transit area” indicates the non-backbone area through which the virtual link is configured. Virtual links are considered treated by the OSPF protocol similarly to point-to-point networks. The other parameters listed in the command (“Rtrns,” “TrnsDly,” “Hello,” and “Dead”) are maintained for all interfaces. See the OSPF **list interfaces** command for more information.

areas

Lists all information concerning configured OSPF areas.

Example: **list areas**

```
Area ID:        0.0.0.0
Authentication: 1 (Simple-pass)
Stub area?:     FALSE

Area ID:        0.0.0.1
Authentication: 0 (None)
Stub area?:     FALSE
```

Area-ID	Displays the attached area ID (area summary information).
Authentication	Displays the method used for area authentication. "Simple-pass" means a simple password scheme is being used for the area's authentication.
Stub area	Displays whether or not the area being summarized is a stub area.

interfaces

For each interface, its IP address is printed, together with configured parameters. "Area" is the OSPF area to which the interface attaches. "Cost" indicates the TOS 0 cost (or metric) associated with the interface. "Rtrns" is the retransmission interval, which is the number of seconds between retransmissions of unacknowledged routing information. "TrnsDly" is the transmission delay, which is an estimate of the number of seconds it takes to transmit routing information over the interface (it must be greater than 0). "Pri" is the interface's router priority, which is used when selecting the designated router. "Hello" is the number of seconds between Hello Packets sent out the interface. "Dead" is the number of seconds after Hellos cease to be heard that the router is declared down.

Example: **list interfaces**

```
OSPF interfaces:
IP address       Area     Cost  Rtrns  TrnsDly  Pri Hello Dead
128.185.177.11   0.0.0.0   3      5       1     0     10    60
128.185.142.11   0.0.0.0   4      5       1     1     10    60
128.185.184.11   0.0.0.0   1      5       1     1     10    60
```

non-broadcast

Lists all information related to interfaces connected to non-broadcast networks. For each non-broadcast interface, as long as the router is eligible to become designated router on the attached network, the polling interval is displayed together with a list of the router's neighbors on the non-broadcast network.

Example: **list non-broadcast**

```
Interface Addr      Poll Interval
128.185.235.34      120
```

virtual-links

Lists all virtual links that were configured with this router as endpoint. "Virtual endpoint" indicated the OSPF router ID of the other endpoint. "Transit area" indicates the non-backbone area through which the virtual link is configured. Virtual links are considered treated by the OSPF protocol similarly to point-to-point networks. The other parameters listed in the command ("Rtrns," "TrnsDly," "Hello," and "Dead") are maintained for all interfaces. See the OSPF **list interfaces** command for more information.

Example: **list virtual-links**

```
Virtual links:
Virtual endpoint   Transit area   Rtrns  TrnsDly Hello  Dead
0.0.0.0            0.0.0.1         10        5     30    180
```

Set

Display or change the configuration information concerning OSPF areas, interfaces, non-broadcast networks, or virtual links. This command also allows you to set the way in which OSPF routes are compared to information obtained from other routing protocols.

Syntax: set area
comparison
interface
non-broadcast
virtual-links

area

Sets the parameters for an OSPF area. If no areas are defined, the router software assumes that all the router's directly attached networks belong to the backbone area (area ID 0.0.0.0).

Example: **set area**

```
Area number [0.0.0.0]? 0.0.0.1
Authentication type [1]? 1
Is this a stub area? (Yes or No): no
```

- Area number is the OSPF area address. An OSPF area is a contiguous group of networks that is defined by a list of address ranges, each indicated by a combination of the IP address and an address mask. A network belongs to an area if its address is in the list.

- Authentication type (security scheme) to be used in the area. The choices for authentication types are 1, which indicates a simple password; or 0, which indicates that no authentication is necessary to pass packets.

- Stub area designation. If you designate YES:

  – The area does not receive any AS external link advertisements, reducing the size of your database and decreasing memory usage for routers in the stub area.

- You cannot configure virtual links through a stub area.
- You cannot configure a router within the stub area as an AS boundary routers.

External Routing in Stub Areas. You cannot configure the backbone as a stub area. External routing in stub areas is based on a default route. Each border area router attaching to a stub area originates a default route for this purpose. The cost of this default route is also configurable with the `OSPF config>` **set area** command.

comparison

Tells the router where the EGP/RIP/static routes fit in the OSPF hierarchy. The two lower levels consist of the OSPF internal routes. OSPF internal routes take precedence over information gained from any other sources, all of which are located on a single level.

Example: **set comparison**

```
OSPF Config> set comparison
Compare to type 1 or 2 externals [2]?
```

interface

Sets the OSPF parameters for the router's network interfaces.

Example: **set interface**

```
Attaches to area [0.0.0.0]? 0.0.0.1
Interface IP address [0.0.0.0]? 128.185.138.19
Retransmission Interval (in seconds) [5]?
Transmission Delay (in seconds) [1]? 1
Router Priority [1]? 1
Hello Interval (in seconds) [10]? 10
Dead Router Interval (in seconds) [60]? 40
Type Of Service 0 cost [1]? 5
Authentication Key []? xyz_q
Retype Auth.  Key []? xyz_q
Forward multicast datagrams (Yes or No)? Yes
Forward as data-link unicasts (Yes or No)? No
IGMP polling interval (in seconds) [60]? 60
IGMP timeout (in seconds) [180]? 180
```

When responding to the prompts, supply the IP address for each interface in the router and answer the questions that follow. For the parameters listed, you must enter the same value for all routers attached to a common network.

- Hello interval
- Dead router interval
- Authentication key (if an authentication of 1 is used)

The first prompt asks for the OSPF area to which the interface attaches. For example, suppose that the interface address mask is 255.255.255.0, indicating that the interface attaches to a subnet (128.185.138.0) of network 128.185.0.0. All other OSPF routers attached to subnet 128.185.138.0 must also have their *hello interval* set to 10, *dead router interval* set to 40, and their interface *authentication key* set to xyz_q.

In a multicast routing configuration (multicast was enabled), the MOSPF parameters for each OSPF interface are set to their default values. This means the following:

- Multicast forwarding is enabled.
- Multicast datagrams are forwarded as data-link multicasts.
- IGMP Host Membership is sent out the interface every 60 seconds.
- Local group database entries are removed 180 seconds after IGMP Host Membership reports for the group cease to be received by the interface.

If you want to change the MOSPF parameters, use the **set interface** command. You are queried for multicast parameters (the last five parameters shown in the output display below) only if you first enable multicast forwarding.

On networks that lie on the edge of an Autonomous System, where multiple multicast routing protocols (or multiple instances of a single multicast routing protocol) may exist, you may need to configure forwarding as data-link unicasts to avoid unwanted datagram replication. In any case, for all routers attached to a common network, the interface parameters *forward multicast datagrams* and *forward as data-link unicasts* are configured identically.

non-broadcast

Helps the router discover its OSPF neighbors. This configuration is only necessary if the router is eligible to be the designated router of the non-broadcast network. After using this command you must then configure the IP addresses of all other OSPF routers that are attached to the non-broadcast network. See the **add neighbor** command for more information.

Example: **set non-broadcast**

```
Interface IP address [0.0.0.0]? 128.185.138.19
Poll Interval [120]?
```

virtual-link

Configures virtual links between any two area border routers. To maintain backbone connectivity, you must have all of your backbone routers interconnected either by permanent or virtual links. Virtual links are considered to be separate router interfaces connecting to the backbone area. Therefore, you are asked to also specify many of the interface parameters when configuring a virtual link.

Example: **set virtual link**

```
Virtual endpt.  (Router ID) [0.0.0.0]? 128.185.138.21
Link's transit area [0.0.0.1]? .0.0.1
Retransmission Interval (in seconds) [10]?
Transmission Delay (in seconds) [5]?
Hello Interval (in seconds) [30]?
Dead Router Interval (in seconds) [180]?
Authentication Key []? 3-14159
```

Exit

Return to the previous prompt level.

Syntax: exit

Example: **exit**

18

Monitoring OSPF

This chapter describes the OSPF console commands.

For more information about OSPF, refer to the *Routing Protocols Reference Guide.*

Accessing the OSPF Console Environment

For information about accessing the OSPF console environment, see Chapter 1.

OSPF Console Commands

This section summarizes and then explains all the OSPF console monitoring commands. These commands enable you to monitor the OSPF routing protocol. Table 18–1 lists the OSPF console commands.

Enter the OSPF console commands at the `OSPF>` prompt.

Table 18–1 OSPF Console Command Summary

Command	Function
? (Help)	Lists the OSPF console commands or lists the options associated with specific commands.
Advertisement	Displays a link state advertisement belonging to the OSPF database.
Area summary	Displays OSPF area statistics and parameters.
AS external	Lists the AS external advertisements belonging to the OSPF link state database.
Database summary	Displays the advertisements belonging to an OSPF area's link state database.
Dump routing tables	Displays the OSPF routes contained in the routing table.
Interface summary	Displays OSPF interface statistics and parameters.
Join	Configures the router to belong to one or more multicast groups.
Leave	Removes the router from membership in multicast groups.
Mcache	Displays a list of currently active multicast forwarding cache entries.
Mgroups	Displays the group membership of the router's attached interfaces.
Mstats	Displays various multicast routing statistics.
Neighbor summary	Displays OSPF neighbor statistics and parameters.
Routers	Displays the reachable OSPF area-border routers and AS-boundary routers.
Size	Displays the number of LSAs currently in the link state database, categorized by type.
Statistics	Displays OSPF statistics detailing memory and network usage.
Weight	Dynamically changes the cost of an OSPF interface.
Exit	Exits the OSPF console process.

? (Help)

List the commands that are available from the current prompt level. You can also enter a **?** after a specific command name to list its options.

Syntax: ?

Example: **?**

```
ADVERTISEMENT expansion
AREA summary
AS-EXTERNAL advertisements
DATABASE summary
DUMP routing tables
EXIT
INTERFACE summary
JOIN
LEAVE
MCACHE
MGROUPS
MSTATS
NEIGHBOR summary
ROUTERS
SIZE
STATISTICS
WEIGHT
```

Advertisement Expansion

Print the contents of a link state advertisement contained in the OSPF database. For a summary of the router's advertisements use the **database** command.

A link state advertisement is defined by its link state type, link state ID and its advertising router. There is a separate link state database for each OSPF area. Providing an *area-id* on the command line tells the software which database you want to search. Listed below are different kinds of advertisements that depend on the value given for link-state-type:

- **Router links** – Contain descriptions of a single router's interface.
- **Network links** – Contain the list of routers attached to a particular interface.
- **Summary nets** – Contain descriptions of a single inter-area route.

- **Summary AS boundary routers** – Contain descriptions of the route to an AS boundary router in another area.

- **AS external nets** – Contain descriptions of a single route.

- **Multicast group memberships** – Contain descriptions of a particular group's membership in the neighborhood of the advertising router.

Note: Link State IDs, advertising routers (specified by their router IDs), and area IDs take the same format as IP addresses. For example, the backbone area can be entered as 0.0.0.0.

The example below shows an expansion of a router links advertisement. The router's ID is 128.185.184.11. It is an AS boundary router and has three interfaces to the backbone area (all of cost 1). Multicast routing was enabled. Detailed field descriptions are provided with the example shown below.

This command was also enhanced in two ways. First, when displaying router-LSAs and network-LSAs, the reverse cost of each router-to-router link and router-to-transit-network link is displayed, as well as the previously displayed forward cost. This is done because routing of multicast datagrams whose source lies in different areas/ASs is based on reverse cost instead of forward cost. In those cases where there is no reverse link (which means that the link is never used by the Dijkstra), the reverse cost is shown as ''1-way".

In addition, the LSA's OSPF options are displayed in the same manner as they were displayed in the detailed OSPF **neighbor** command.

New group-membership-LSAs can also be displayed. An example follows. The ''LS destination" of each group-membership-LSA is a group address. A router originates a group-membership-LSA for each group that has members on one or more of the router's attached networks. The group-membership-LSA for the group lists those attached transit networks having group members (the type ''2" vertices), and when there are members belonging to one or more attached stub networks, or if the router itself is a member of the multicast group, a type ''1" vertex whose ID is the router's OSPF router ID is included.

Syntax: advertisement *ls-type link-state-id [advertising-router] [area-id]*

Example: **advertisement 1 128.185.184.11 0.0.0.0**

```
LS age:     173
LS options:  E,MC
LS type:     1
LS destination (ID): 128.185.184.11
LS originator:   128.185.184.11
LS sequence no:  0x80000047
LS checksum:     0x122
LS length:       60
Router type:  ASBR,W
# router ifcs:   3
         Link ID:          128.185.177.31
         Link Data:        128.185.177.11
         Interface type:   2
                  No. of metrics: 0
                  TOS 0 metric:   3 (0)
         Link ID:          128.185.142.40
         Link Data:        128.185.142.11
         Interface type:   2
                  No. of metrics: 0
                  TOS 0 metric:   4 (0)
         Link ID:          128.185.184.0
         Link Data:        255.255.255.0
         Interface type:   3
                   No. of metrics: 0
                   TOS 0 metric:   1
```

LS age — Indicates the age of the advertisement in seconds.

LS options — Indicates the optional OSPF capabilities supported by the piece of the routing domain described by the advertisement. These capabilities are denoted by E (processes type 5 externals. When this is not set to the area to which the advertisement belongs was configured as a stub), T (can route based on TOS) and MC (can forward IP multicast datagrams).

LS type — Classifies the advertisement and dictates its contents: 1 (router links advertisement), 2 (network link advertisement), 3 (summary link advertisement), 4 (summary ASBR advertisement), 5 (AS external link) and 6 (group-membership advertisement).

LS destination	Identifies what is being described by the advertisement. Depends on the advertisement type. For router links and ASBR summaries, it is the OSPF router ID. For network links, it is the IP address of the network's designated router. For summary links and AS external links, it is a network/subnet number. For group-membership advertisements, it is a particular multicast group.
LS originator	OSPF router ID of the originating router.
LS sequence number	Used to distinguish separate instances of the same advertisement. Is looked at as a signed 32-bit integer. Starts at 0x80000001, and increments by one each time the advertisement is updated.
LS checksum	A checksum of advertisement contents, used to detect data corruption.
LS length	The size of the advertisement in bytes.
Router type	Indicates the level of functionality of the router. ASBR means that the router is an AS boundary router, ABR that the router is an area border router, and W that the router is a wildcard multicast receiver.
# Router ifcs	The number of router interface described in the advertisement.
Link ID	Indicates what the interface connects to. Depends on Interface type. For interfaces to routers (point-to-point links), the Link ID is the neighbor's router ID. For interfaces to transit networks, it is the IP address of the network designated router. For interfaces to stub networks, it is the network's network/subnet number.

Link Data	4 bytes of extra information concerning the link, it is either the IP address of the interface (for interfaces to point-to-point networks and transit networks), or the subnet mask (for interfaces to stub networks).
Interface type	One of the following: 1 (point-to-point connection to another router), 2 (connection to transit network), 3 (connection to stub network) or 4 (virtual link).
No. of metrics	The number of non-zero TOS values for which metrics are provided for this interface.
TOS 0 metric	The cost of the interface. In parentheses, the reverse cost of the link is given (derived from another advertisement). If there is no reverse link, "1-way" is displayed.

The *LS age, LS options, LS type, LS destination, LS originator, LS sequence no, LS checksum* and *LS length* fields are common to all advertisements. The *Router type* and *# router ifcs* are seen only in router links advertisements. Each link in the router advertisement is described by the *Link ID, Link Data*, and *Interface type* fields. Each link can also be assigned a separate cost for each IP Type of Service (TOS). This is described by the *No. of metrics* and *TOS 0 metric fields* (the router currently does not route based on TOS, and only looks at the TOS 0 cost).

The next example shows an expansion of a group-membership advertisement. A group-membership advertisement for a given group/advertising router combination lists those networks directly attached to the advertising router that have group members. It also lists whether the router itself is a member of the specified group. The example below shows that network 128.185.184.0 has members of group 224.0.1.1.

Example: **adv 6 224.0.1.1 128.185.184.114**

```
For which area [0.0.0.0]?
```

```
LS age:     168
LS options:  E
LS type:     6
LS destination (ID): 224.0.1.1
LS originator:   128.185.184.114
LS sequence no:  0x80000001
LS checksum:     0x7A3
LS length:       28
Vertex type: 2
Vertex ID:   128.185.184.114
```

Vertex type	Describes the object having group members, one of: 1 (the router itself, or stub networks attached to the router) or 2 (a transit network).
Vertex ID	When the vertex type is 1, always the advertising router's ID. When the vertex type is 2, the IP address of the transit network's designated router.

Area Summary

Display the statistics and parameters for all OSPF areas attached to the router.

In the example below, the router attaches to a single area (the backbone area). A simple password scheme is being used for the area's authentication. The router has three interfaces attaching to the area, and has found 4 transit networks, 7 routers and no area border routers when doing the SPF tree calculation for the backbone.

Syntax: area

Example: **area**

```
Area ID      Authentication   #ifcs  #nets  #rtrs  #brdrs
0.0.0.0       Simple-pass       3      4      7      0
```

 # ifcs Indicates the number of router interfaces attached to the particular area. These interfaces are not necessarily functional.

 # nets Indicates the number of transit networks that were found while doing the SPF tree calculation for this area.

 # rtrs Indicates the number of routers that were found when doing the SPF tree calculation for this area.

 # brdrs Indicates the number of area border routers that were found when doing the SPF tree calculation for this area.

AS-external advertisements

List the AS external advertisements belonging to the OSPF routing domain. One line is printed for each advertisement. Each advertisement is defined by the following three parameters: its link state type (always 5 for AS external advertisements), its link state ID (called the LS destination), and the advertising router (called the LS originator).

Syntax: as-external

Example: **as-external**

```
Type LS destination LS originator  Seqno      Age   Xsum
  5  0.0.0.0       128.185.123.22  0x80000084 430   0x41C7
  5  128.185.131.0 128.185.123.22  0x80000080 450  0x71DC
  5  128.185.132.0 128.185.123.22  0x80000080 450  0x66E6
  5  128.185.144.0 128.185.123.22  0x80000002 329  0xF2CA
  5  128.185.178.0 128.185.123.22  0x80000081 450  0x72AA
  5  128.185.178.0 128.185.129.40  0x80000080 382  0xDD28
  5  129.9.0.0     128.185.123.22  0x80000082 451  0x4F30
  5  129.9.0.0     128.185.126.24  0x80000080 676  0x324A
  5  134.216.0.0   128.185.123.22  0x80000082 451  0x505A
  5  134.216.0.0   128.185.126.24  0x80000080 676  0x3374
  5  192.9.3.0     128.185.123.22  0x80000082 451  0xF745
  5  192.9.3.0     128.185.126.24  0x80000080 677  0xDA5F
  5  192.9.12.0    128.185.123.22  0x80000082 452  0x949F
  5  192.9.12.0    128.185.128.41  0x80000080 679  0x31B2
  5  192.26.100.0  128.185.123.22  0x80000081 452  0xFDCD
  5  192.26.100.0  128.185.126.24  0x80000080  21  0xDEE8
                             etc.
              # advertisements:            133
              Checksum total:              0x43CC41
```

Type	Always 5 for AS external advertisements.
LS destination	Indicates an IP network/subnet number. These network numbers belong to other Autonomous Systems.
LS originator	Advertising router.
Seqno, Age, Xsum	It is possible for several instances of an advertisement to be present in the OSPF routing domain at any one time. However, only the most recent instance is kept in the OSPF link state database (and printed by this command). The LS sequence number (Seqno), LS age (Age) and LS checksum fields (Xsum) are compared to see which instance is most recent. The LS age field is expressed in seconds. Its maximum value is 3600.

At the end of the display, the total number of AS external advertisements is printed, along with a checksum total over all of their contents. The checksum total is simply the 32-bit sum (carries discarded) of the individual advertisement's LS checksum fields. This information can be used to quickly determine whether two OSPF routers have synchronized databases.

Database Summary

Display a description of the contents of a particular OSPF area's link state database. AS external advertisements are omitted from the display. A single line is printed for each advertisement. Each advertisement is defined by the following three parameters: its link state type (called Type), its link state ID (called the LS destination) and the advertising router (called the LS originator).

Syntax: database *area-id*

Example: **database 0.0.0.0**

```
Type LS destination LS originator    Seqno     Age  Xsum
  1  128.185.123.22 128.185.123.22 0x80000084  442 0xCE2D
  1  128.185.125.38 128.185.125.38 0x80000082  470 0x344D
  1  128.185.126.24 128.185.126.24 0x80000088 1394 0xCC47
  1  128.185.128.41 128.185.128.41 0x80000082  471 0x16A2
  1  128.185.129.25 128.185.129.25 0x8000008D 1624 0x8B64
  1  128.185.129.40 128.185.129.40 0x8000008A 1623 0xABBE
  1  128.185.136.39 128.185.136.39 0x80000082  469 0x5045
  2  128.185.125.40 128.185.129.40 0x80000049  457 0xA31
  2  128.185.126.25 128.185.129.25 0x80000002 1394 0x56B8
  2  128.185.127.24 128.185.126.24 0x8000007F 1031 0x592D
  2  128.185.129.25 128.185.129.25 0x8000005F 2295 0x8219
  2  128.185.129.40 128.185.129.40 0x80000001 1623 0x12C9
  6  224.0.2.6      128.185.142.9  0x8000003D  232 0x513F
  6  224.0.2.6      128.185.184.11 0x80000003  376 0x2250

              # advertisements:            14
              Checksum total:              0x4BBC2
```

Type — Separate LS types are numerically displayed: type 1 (router links advertisements), type 2 (network links advertisements), type 3 (network summaries), type 4 (AS boundary router summaries), and type 6 (group-membership-LSAs).

LS destination — Indicates what is being described by the advertisement.

LS originator	Advertising router.
Seqno, Age, Xsum	It is possible for several instances of an advertisement to be presenting the OSPF routing domain at any one time. However, only the most recent instance is kept in the OSPF link state database (and printed by this command). The LS sequence number (Seqno), LS age (Age) and LS checksum fields (Xsum) are compared to see which instance is most recent. The LS age field is expressed in seconds. Its maximum value is 3600.

At the end of the display, the total number of advertisements in the area database is printed, along with a checksum total over all of their contents. The checksum total is simply the 32-bit sum (carries discarded) of the individual advertisement's LS checksum fields. This information can be used to quickly determine whether two OSPF routers have synchronized databases.

Note: When comparing multicast-capable to non-multicast routers, the above database checksum (and also # advertisements) will not necessarily match, because non-multicast routers do not handle or store group-membership-LSAs.

Dump Routing Tables

Display all the routes that were calculated by OSPF and are now present in the routing table. Its output is similar in format to the IP console's **dump routing tables** command.

Syntax: dump

Example: **dump**

```
Type   Dest net         Mask     Cost Age  Next hop(s)

Sbnt   16.0.0.0         FF000000  1    0    None
 SPF   16.24.8.0        FFFFFF00  2    2    20.24.12.230
 SPF   16.24.10.0       FFFFFF00  4    4    20.24.12.230
 SPF   16.24.11.0       FFFFFF00  2    2    20.24.12.230
Sbnt   17.0.0.0         FF000000  1    0    None
 SPF   17.1.1.0         FFFFFF00  4    4    20.24.12.230
Sbnt   18.0.0.0         FF000000  1    0    None
 SPF   18.1.1.0         FFFFFF00  4    4    20.24.12.230
Sbnt   20.0.0.0         FF000000  1    0    None
 SPF*  20.24.12.0       FFFFFF00  1    1    Eth/1
Sbnt   21.0.0.0         FF000000  1    0    None
 SPF*  21.24.16.0       FFFFFF00  1    1    Eth/5
 Dir*  21.24.166.0      FFFFFF00  1    0    Eth/5
 Dir*  21.24.167.0      FFFFFF00  1    0    Eth/5
 Dir*  21.24.168.0      FFFFFF00  1    0    Eth/5
 Dir*  21.24.169.0      FFFFFF00  1    0    Eth/5
 Dir*  21.24.170.0      FFFFFF00  1    0    Eth/5
Sbnt   25.0.0.0         FF000000  1    0    None
 SPF   25.24.13.0       FFFFFF00  4    4    20.24.12.230
Sbnt   135.24.0.0       FFFF0000  1    0    None
 SPF   135.24.10.0      FFFFFF00  4    4    20.24.12.230

Routing table size: 768 nets (55296 bytes), 21 nets known
```

Type Indicates destination type. Net indicates that the destination is a network. All other destinations are covered by the OSPF **routers** command.

- **Sbnt** – Indicates that the network is subnetted; such an entry is a placeholder only.
- **Dir** – Indicates a directly connected network or subnet.
- **RIP** – Indicates the route was learned through the RIP protocol.
- **Del** – Indicates the route was deleted.
- **Stat** – Indicates a statically configured route.
- **EGP** – Indicates routes learned through the EGP protocol.
- **EGPR** – Indicates routes learned through the EGP protocol that are readvertised by OSPF and RIP.
- **Fltr** – Indicates a routing filter.
- **SPF** – Indicates that the route is an OSPF intra-area route.
- **SPIA** – Indicates that it is an OSPF inter-area routes.
- **SPE1**, **SPE2** – Indicates OSPF external routes (types 1 and 2 respectively).
- **Rnge** – Indicates a route type that is an active OSPF area address range and is not used in forwarding packets.

Dest net Destination host or network.

Mask Displays the entry's subnet mask.

Cost Age	Displays the route cost.
Next hop(s)	Address of the next router on the path toward the destination host. A number in parentheses at the end of the column indicates the number of equal-cost routes to the destination. The first hops belonging to these routes can be displayed with the IP console's **route** command.

Interface Summary

Display statistics and parameters related to OSPF interfaces. If no arguments are given, a single line is printed summarizing each interface. If an interface's IP address is given, detailed statistics for that interface are displayed.

Syntax: interface *interface-ip-address*

Example: **interface**

```
Ifc Address      Phys    assoc. Area      Type   State   #nbrs  #adjs
16.24.8.251      Eth/1   0.0.0.0         Brdcst   64        1        1
16.24.11.251     Eth/1   0.0.0.0         Brdcst   64        1        1
17.1.1.251       Eth/2   0.0.0.0         Brdcst   64        0        0
25.24.13.251     Eth/3   0.0.0.0         Brdcst   64        0        0
18.1.1.251       Eth/4   0.0.0.0         Brdcst   64        0        0
16.24.10.251     Eth/0   0.0.0.0         Brdcst   64        0        0
135.24.10.251    Eth/0   0.0.0.0         Brdcst   64        0        0

Ifc Address    assoc. Area    Type   State  #nbrs #adjs
128.185.123.22   0.0.0.0      Brdcst   64     0     0
128.185.124.22   0.0.0.0      Brdcst   64     0     0
128.185.125.22   0.0.0.0      Brdcst   16     6     2
```

Ifc Address	Interface IP address.
Phys	The physical interface.
Assoc Area	Attached area ID.
Type	Can be either Brdcst (broadcast, for example, an Ethernet interface), P-P (a point-to-point network, for example, a synchronous serial line), Multi (non-broadcast, multi-access, for example, an X.25 connection) and VLink (an OSPF virtual link).

State	Can be one of the following: 1 (down), 2 (looped back), 4 (waiting), 8 (point-to-point), 16 (DR other), 32 (backup DR) or 64 (designated router).
#nbrs	Number of neighbors. This is the number of routers whose hellos were received, plus those that were configured.
#adjs	Number of adjacencies. This is the number of neighbors in state Exchange or greater. These are the neighbors with whom the router has synchronized or is in the process of synchronization.

Example: **interface 128.185.125.22**

```
Interface address:      128.185.125.22
Attached area:          0.0.0.1
Physical interface:     Eth /1
Interface mask:         255.255.255.0
Interface type:         Brdcst
State:                  32
Designated Router:      128.185.184.34
Backup DR:              128.185.184.11

DR Priority:       1  Hello interval:   10  Rxmt interval:     5
Dead interval:    60  TX delay:          1  Poll interval:     0
Max pkt size:   1500  TOS 0 cost:        1

# Neighbors:       2  # Adjacencies:     2  # Full adjs.:      2
# Mcast floods: 1714  # Mcast acks:    856

MC forwarding:    on  DL unicast:      off  IGMP monitor:     on
# MC data in:  14444  # MC data acc: 13379  # MC data out: 16254
IGMP polls snt: 1316  IGMP polls rcv: 1009  Unexp polls:       9
IGMP reports:   2000  Nbr node: type:    2  ID:   128.185.184.34
```

Interface Address	Interface IP address.
Attached Area	Attached area ID.
Physical interface	Displays physical interface type and number.
Interface Mask	Displays interface subnet mask.

Interface type	Can be either Brdcst (broadcast, for example, an Ethernet interface), P-P (a point-to-point network, for example, a synchronous serial line), Multi (non-broadcast, multi-access, for example, an X.25 connection) and VLink (an OSPF virtual link).
State	Can be one of the following: 1 (Down), 2 (Attempt), 4 (Init), 8 (2-Way), 16 (ExStart), 32 (Exchange), 64 (Loading) or 128 (Full).
Designated Router	IP address of the designated router.
Backup DR	IP address of the backup designated router.
DR Priority	Displays priority assigned to designated router.
Hello interval	Displays the current hello interval value.
Rxmt interval	Displays the current retransmission interval value.
Dead interval	Displays the current dead interval value.
TX delay	Displays the current transmission delay value.
Poll interval	Displays the current poll interval value.
Max pkt size	Displays the maximum size for an OSPF packet sent out this interface.
TOS 0 cost	Displays the interface's TOS 0 cost.
# Neighbors	Number of neighbors. This is the number of routers whose hellos were received, plus those that were configured.
# Adjacencies	Number of adjacencies. This is the number of neighbors in state Exchange or greater.
# Full adj	Number of full adjacencies. The number of full adjacencies is the number of neighbors whose state is Full (and therefore, with which the router has synchronized databases).

# Mcast Floods	Number of link state updates flooded out the interface (not counting retransmissions).
# Mcast acks	Number of link state acknowledgements flooded out the interface (not counting retransmissions).
MC forwarding	Displays whether multicast forwarding was enabled for the interface.
DL unicast	Displays whether multicast datagrams are to be forwarded as data-link multicasts or as data-link unicasts.
IGMP monitor	Displays whether IGMP is enabled on the interface.
# MC data in	Displays the number of multicast datagrams that were received on this interface and then successfully forwarded.
# MC data acc	Displays the number of multicast datagrams that were successfully forwarded.
# MC data out	Displays the number of datagrams that were forwarded out the interface (either as data-link multicasts or data-link unicasts).
IGMP polls sent	Displays the number of IGMP Host Membership Queries that were sent out the interface.
IGMP polls rcv	Displays the number of IGMP Host Membership Queries that were received on the interface.
Unexp polls	Displays the number of unexpected IGMP Host Membership Queries that were received on the interface (received when the router itself was sending them).
IGMP reports	Displays the number of IGMP Host Membership Reports received on the interface.
Nbr node: type and ID	Displays the identity of the upstream node if the router is supposed to receive datagrams on this interface. *Type* here is an integer from 1 to 3, with 1 indicating router, 2 indicating transit net, and 3 indicating stub net.

Join

Establish the router as a member of a multicast group.

This command is similar to the **join** command in the OSPF configuration console with two differences:

- The effect on group membership is immediate when the commands are given from the OSPF monitor (a restart/reload is not required).
- The command keeps track of the number of times a particular group is ‘‘joined.”

When the router is the member of a multicast group, it responds to pings and SNMP queries sent to the group address.

Syntax: join *multicast-group-address*

Example: **join 128.185.00.00**

Leave

Remove a router’s membership in a multicast group. This keeps the router from responding to pings and SNMP queries sent to the group address.

This command is similar to the **leave** command in the OSPF configuration console with two differences:

- The effect on group membership is immediate when the commands are given from the OSPF monitor (a restart/reload is not required).
- The command does not delete group membership until the “leaves” executed equals the number of “joins” previously executed.

Syntax: leave *multicast-group-address*

Example: **leave 128.185.00.00**

Mcache

Display a list of currently active multicast cache entries. Multicast cache entries are built on demand, whenever the first matching multicast datagram is received. There is a separate cache entry (and therefore a separate route) for each datagram source network and destination group combination.

Cache entries are cleared on topology changes (for example, a point-to-point line in the MOSPF system going up or down), and on group membership changes.

Syntax: mcache

Example: **mcache**

```
                MOSPF forwarding cache
Source          Destination     Upstream        #Down     Usage
128.185.142.0   224.0.1.1       128.185.142.11    0       184
```

Source	Source network/subnet of matching datagrams.
Destination	Destination group of matching datagrams.
Upstream	Displays the neighboring network/router from which the datagram is received in order to be forwarded. When this reads as "none," the datagram is never forwarded.
Down	Displays the total number of downstream interfaces/neighbors to which the datagram is forwarded. When this is 0, the datagram is not forwarded.
Usage	Displays the number of received datagrams that matched the cache entry.

There is more information in a multicast forwarding cache entry. A cache entry can be displayed in detail by providing the source and destination of a matching datagram on the command line. If a matching cache entry is not found, one is built. A sample of this command is shown below:

Example: **mcache 128.185.182.9 224.0.1.2**

```
source Net:     128.185.182.0
Destination:    224.0.1.2
Use Count:      472
Upstream Type:  Transit Net
Upstream ID:    128.185.184.114
Downstream:     128.185.177.11 (TTL = 2)
```

In addition to the information shown in the short form of the **mcache** command, the following fields are displayed:

Upstream Type Indicates the type of node from which the datagram must be received in order to be forwarded. Possible values for this field are “none” (indicating that the datagram is not forwarded), “router” (indicating that the datagram must be received over a point-to-point connection), “transit network,” “stub network,” and “external” (indicating that the datagram is expected to be received from another Autonomous System).

Down–stream Prints a separate line for each interface or neighbor to which the datagram is sent. A TTL value is also given, indicating that datagrams forwarded out of or to this interface must have at least the specified TTL value in their IP header. When the router is itself a member of the multicast group, a line specifying “internal Application” appears as one of the downstream interfaces/neighbors.

Mgroups

Display the group membership of the router’s attached interfaces. Only the group membership for those interfaces on which the router is either designated router or backup designated router are displayed.

Syntax: mgroups

Example: **mgroups**

```
               Local Group Database

 Group             Interface                Lifetime (secs)

224.0.1.1         128.185.184.11 (Eth /1)       176
224.0.1.2         128.185.184.11 (Eth /1)       170
224.1.1.1         Internal                     1
```

Group Displays the group address as it was reported (through IGMP) on a particular interface.

Interface Displays the interface address to which the group address was reported (through IGMP).

The router's internal group membership is indicated by a value of "internal." For these entries, the lifetime field (see below) indicates the number of applications that requested membership in the particular group.

Lifetime — Displays the number of seconds that the entry persists if Membership Reports cease to be heard on the interface for the given group.

Mstat

Display various multicast routing statistics. The command indicates whether multicast routing is enabled and whether the router is an inter-area and/or inter-AS multicast forwarder.

Syntax: mstats

Example: **mstats**

```
              MOSPF forwarding:              Disabled
              Inter-area forwarding:         Disabled
              DVMRP forwarding:              Disabled

 Datagrams received:              0  Datagrams (ext source):          0
 Datagrams fwd (multicast):       0  Datagrams fwd (unicast):         0
 Locally delivered:               0  No matching rcv interface:       0
 Unreachable source:              0  Unallocated cache entries:       0
 Off multicast tree:              0  Unexpected DL multicast:         0
 Buffer alloc failure:            0  TTL scoping:                     0

 # DVMRP routing entries:         0  # DVMRP entries freed:           0
 # fwd cache alloc:               0  # fwd cache freed:               0
 # fwd cache GC:                  0  # local group DB alloc:          0
 # local group DB free:           0
```

MOSPF forwarding — Displays whether the router forwards IP MOSPF datagrams.

Inter-area forwarding — Displays whether the router forwards IP multicast datagrams between areas.

DVMRP forwarding — Displays whether the router forwards IP multicast datagrams between Autonomous Systems.

Datagrams received	Displays the number of multicast datagrams received by the router (datagrams whose destination group lies in the range 224.0.0.1 – 224.0.0.255 are not included in this total).
Datagrams (ext source)	Displays the number of datagrams received whose source is outside the AS.
Datagrams fwd (multicast)	Displays the number of datagrams forwarded as data-link multicasts. (This includes packet replications, when necessary. So this count can be greater than the number received.)
Datagrams fwd (unicast)	Displays the number of datagrams forwarded as data-link unicasts.
Locally delivered	Displays the number of datagrams forwarded to internal applications.
No matching rcv interface	Displays the count of those datagrams received by a non-inter-AS multicast forwarder on a non-MOSPF interface.
Unreachable source	Displays a count of those datagrams whose source address was unreachable.
Unallocated cache entries	Displays a count of those datagrams whose cache entries were not created due to resource shortages.
Off multicast tree	Displays a count of those datagrams not forwarded either because there was no upstream neighbor or no downstream interfaces/neighbors in the matching cache entry.
Unexpected DL multicast	Displays a count of those datagrams that were received as data-link multicasts on those interfaces that were configured for data-link unicast.
Buffer alloc failure	Displays a count of those datagrams that could not be replicated because of buffer shortages.

TTL scoping	Indicates those datagrams that were not forwarded because their TTL indicated that they could never reach a group member.
#DVMRP routing entries	
#DVMRP entries freed	
# fwd cache alloc	Indicates the number of cache entries allocated. The current forwarding cache size is the number of entries allocated (“# fwd cache alloc”) minus the number of cache entries freed (“# fwd cache freed”).
# fwd cache freed	Indicates the number of cache entries freed. The current forwarding cache size is the number of entries allocated (“# fwd cache alloc”) minus the number of cache entries freed (“# fwd cache freed”).
# fwd cache GC	Indicates the number of cache entries were cleared because they were not recently used and the cache overflowed.
# local group DB alloc	Indicates the number of local group database entries allocated. The number allocated (“# local group DB alloc”) minus the number freed (“# local group DB free”) equals the current size of the local group database.
# local group DB free	Indicates the number of local group database entries freed. The number allocated (“# local group DB alloc”) minus the number freed (”# local group DB free”) equals the current size of the local group database.

The number of cache hits can be calculated as the number of datagrams received (“Datagrams received”) minus the total of datagrams discarded due to “No matching rcv interface,” “Unreachable source” and “Unallocated cache entries,” and minus “# local group DB alloc.” The number of cache misses is simply “# local group DB alloc.”

Neighbor Summary

Display statistics and parameters related to OSPF neighbors. If no arguments are given, a single line is printed summarizing each neighbor. If a neighbor’s IP address is given, detailed statistics for that neighbor are displayed.

Syntax: neighbor *neighbor-ip-address*

Example: **neighbor**

```
Neighbor addr   Neighbor ID     State  LSrxl  DBsum  LSreq  Ifc
128.185.125.39  128.185.136.39  128      0     0       0    Eth/1
```

Neighbor addr	Displays the neighbor address.
Neighbor ID	Displays the neighbor’s OSPF router ID.
Neighbor State	Can be one of the following: 1 (Down), 2 (Attempt), 4 (Init), 8 (2-Way), 16 (ExStart), 32 (Exchange), 64 (Loading) or 128 (Full).
LSrxl	Displays the size of the current link state retransmission list for this neighbor.
DBsum	Displays the size of the database summary list waiting to be sent to the neighbor.
LSreq	Displays the number of more recent advertisements that are being requested from the neighbor.
Ifc	Displays the interface shared by the router and the neighbor.

Example: **neighbor 128.185.138.39**

The meaning of most of the displayed fields is given in section 10 of the OSPF specification (RFC 1131).

```
Neighbor IP address:    128.185.184.34
OSPF Router ID:         128.185.207.34
Neighbor State:         128
Physical interface:     Eth /1
DR choice:              128.185.184.34
Backup choice:          128.185.184.11
DR Priority:            1
Nbr options:            E,MC

DB summ qlen:      0  LS rxmt qlen:      0  LS req qlen:       0
Last hello:        7

# LS rxmits:     108  # Direct acks:    13  # Dup LS rcvd:   572
# Old LS rcvd:     2  # Dup acks rcv:  111  # Nbr losses:     29
# Adj. resets:    30
```

Neighbor IP addr	Neighbor IP address.
OSPF router ID	Neighbor's OSPF router ID.
Neighbor State	Can be one of the following: 1 (Down), 2 (Attempt), 4 (Init), 8 (2-Way), 16 (ExStart), 32 (Exchange), 64 (Loading) or 128 (Full).
Physical interface	Displays physical interface type and number of the router and neighbor's common network.
DR choice, backup choice, DR priority	Indicate the values seen in the the last hello received from the neighbor.
Nbr options	Indicates the optional OSPF capabilities supported by the neighbor. These capabilities are denoted by E (processes type 5 externals. When this is not set, the area to which the common network belongs was configured as a stub), T (can route based on TOS) and MC (can forward IP multicast datagrams). This field is valid only for those neighbors in state Exchng or greater.

DBsumm qlen	Indicates the number of advertisements waiting to be summarized in Database Description packets. It is zero except when the neighbor is in state Exchange.
LS rxmt qlen	Indicates the number of advertisements that were flooded to the neighbor, but not yet acknowledged.
LS req qlen	Indicates the number of advertisements that are being requested from the neighbor in state Loading.
Last hello	Indicates the number of seconds since a hello was received from the neighbor.
# LS rxmits	Indicates the number of retransmissions that occurred during flooding.
# direct acks	Indicates responses to duplicate link state advertisements.
# Dup LS rcvd	Indicates the number of duplicate retransmissions that occurred during flooding.
# Old LS rcvd	Indicates the number of old advertisements received during flooding.
# Dup acks rcvd	Indicates the number of duplicate acknowledgements received.
# Nbr losses	Indicates the number of times the neighbor transitioned to Down state.
# Adj. resets	Counts entries to state ExStart.

Routers

Display all router routes that were calculated by OSPF and are now present in the routing table. With the **dump routing tables** command, the *Net* field indicates that the destination is a network. The **routers** command covers all other destinations.

Syntax: routers

Example: **routers**

```
DType RType Destination       Area           Cost       Next hop(s)

  BR  SPF   20.24.12.230     0.0.0.0          1         20.24.12.230
Fadd  SPF   20.24.12.230     0.0.0.0          1         0.0.0.2
  BR  SPF   16.24.8.251      0.0.0.0          2         20.24.12.230
ASBR  SPIA  19.24.9.252      0.0.0.0          3         20.24.12.230
```

DType	Indicates destination type. Net indicates that the destination is a network, ASBR indicates that the destination is an AS boundary router, and BR indicates that the destination is an area border router, and Fadd indicates a forwarding address (for external routes).
RType	Indicates route type and how the route was derived. SPF indicates that the route is an intra-area route (comes from the Dijkstra calculation), SPIA indicates that it is an inter-area route (comes from considering summary link advertisements).
Destination	Destination router's OSPF ID. For Type D entries, one of the router's IP addresses is displayed (which corresponds to a router in another AS).
Area	Always displayed as 0.0.0.0.
Cost	Displays the route cost.
Next hop	Address of the next router on the path toward the destination host. A number in parentheses at the end of the column indicates the number of equal-cost routes to the destination.

Size

Display the number of LSAs currently in the link state database, categorized by type.

Syntax: size

Example: **size**

```
# Router-LSAs:           7
# Network-LSAs:          6
# Summary-LSAs:          14
# Summary Router-LSAs:   2
# AS External-LSAs:      44
# Group-membership-LSAs: 21
```

Statistics

Display statistics generated by the OSPF routing protocol. The statistics indicate how well the implementation is performing, including its memory and network utilization. Many of the fields displayed are confirmation of the OSPF configuration.

Syntax: statistics

Example: **statistics**

```
S/W version:            2.1
OSPF Router ID:         128.185.184.11
External comparison:    Type 2
AS boundary capability: Yes
Import external routes: EGP RIP STA DIR SUB
Orig. default route:    No (0,0.0.0.0)
Default route cost:     (1, Type 2)
Default forward. addr:  0.0.0.0

Attached areas:                2  Estimated # external routes:    300
Estimated # OSPF routers:    100  Estimated heap usage:         76000
OSPF packets rcvd:         60822  OSPF packets rcvd w/ errs:    28305
Transit nodes allocated:    1728  Transit nodes freed:           1715
LS adv. allocated:          7394  LS adv. freed:                 7313
Queue headers alloc:         224  Queue headers avail:            224

# Dijkstra runs:             391  Incremental summ. updates:        0
Incremental VL updates:        0  Buffer alloc failures:            0
Multicast pkts sent:       49487  Unicast pkts sent:              557
LS adv. aged out:              0  LS adv. flushed:                521
```

S/W version	Displays the current OSPF software revision level.
OSPF Router ID	Displays the router's OSPF ID.
External comparison	Displays the external route type used by the router when importing external routes.
AS boundary capability	Displays whether external routes are imported.
Import external routes	Displays which external routes are imported.

Orig default route	Displays whether the router will advertise an OSPF default route. If the value is Yes and a non-zero number is displayed in parentheses, then a default route is advertised only when a route to the network exists.
Default route cost	Displays the cost and type of the default route (if advertised).
Default forward addr	Displays the forwarding address specified in the default route (if advertised).
Attached areas	Indicates the number of areas that the router has active interfaces to.
Estimated heap usage	Rough indication of the size of the OSPF link state database (in bytes).
Transit nodes	Allocated to store router links and network links advertisements.
LS adv.	Allocated to store summary link and AS external link advertisements.
Queue headers	Form lists of link state advertisements. These lists are used in the flooding and database exchange processes; if the number of queue headers allocated is not equal to the number freed, database synchronization with some neighbor is in progress.
# Dijkstra runs	Indicates how many times the OSPF routing table was calculated from scratch.
Incremental summ updates, incremental VL updates	Indicate that new summary link advertisements caused the routing table to be partially rebuilt.

Buffer alloc failures.	Indicate buffer allocation failures. The OSPF system recovers from temporary lack of packet buffers.
Multicast pkts sent	Covers OSPF hello packets and packets sent during the flooding procedure.
Unicast pkts sent	Covers OSPF packet retransmissions and the Database Exchange procedure.
LS adv. aged out	Counts the number of advertisements that have hit 60 minutes. Link state advertisements are aged out after 60 minutes. Usually, they are refreshed before this time.
LS adv. flushed	Indicates number of advertisements removed (and not replaced) from the link state database.
Incremental ext. updates.	Displays number of changes to external destinations that are incrementally installed in the routing table.

Weight

Change the cost of one of the routers OSPF interfaces. This new cost is immediately flooded throughout the OSPF routing domain, causing routes to be updated accordingly.

The cost of the interface will revert to its configured cost whenever the router is restarted or reloaded. To make the cost change permanent, you must reconfigure the appropriate OSPF interface after invoking the **weight** command. This command causes a new router links advertisement to originate, unless the cost of the interface does not change.

Syntax: weight *ip-interface-address new-cost*

Example: **`weight 128.185.124.22 2`**

Exit

Return to the previous prompt level.

Syntax: exit

Example: **`exit`**

19

Configuring SNMP

This chapter describes the SNMP configuration commands and includes the following sections:

For more information about SNMP, refer to the *Routing Protocols Reference Guide.*

Accessing the SNMP Configuration Environment

For information about accessing the SNMP configuration environment, see Chapter 1.

SNMP Configuration Commands

This section summarizes and then explains all the SNMP configuration commands.

Table 19–1 lists the SNMP configuration commands. Table 19–2 lists the commands and command options. The SNMP configuration commands allow you to specify network parameters for router interfaces that transmit SNMP packets. The information you specify takes effect immediately with the exception of the set trap command.

Enter the SNMP configuration commands at the `SNMP config>` prompt.

Table 19–1 SNMP Configuration Commands Summary

Command	Function
? (Help*)*	Lists all the SNMP configuration commands or lists the options associated with specific commands.
Add	Adds a community to the list of SNMP communities, an IP address with mask to a community, or a subtree to a MIB view.
Delete	Removes a community from the list of SNMP communities, an IP address with mask from a community, or a subtree from a MIB view.
Enable/Disable	Enables/disables SNMP protocol and standard traps associated with named communities.
List	Displays the current communities with their associated access modes, enabled traps, IP addresses, and views. Also displays all views and their associated MIB subtrees.
Set	Sets a community's access mode or view. A community's access modes is one of the following: • Read and trap generation • Read, write and trap generation • Trap generation only Also allows setting of trap UDP port.
Exit	Exits the SNMP configuration process and returns to the CONFIG environment.

Table 19–2 SNMP Configuration Commands Options Summary

Command	Param 1	Param 2	Param 3	Param 4	Default
add	community	<name>			None
	address	<comm_name>	<ipAddress>	<ipMask>	
	sub_tree	<view_text_name>	<oid>		
delete	community	<name>			
	address	<comm_name>	<ipAddress>		
	sub_tree	<view_text_name>	<oid>		
disable	snmp				
	trap	all	<comm_name>		
		cold_start	<comm_name>		
		warm_start	<comm_name>		
		link_down	<comm_name>		
		link_up	<comm_name>		
		auth_fail	<comm_name>		
		egp	<comm_name>		
		enterprise	<comm_name>		
enable	snmp				
	trap	all	<comm_name>		
		cold_start	<comm_name>		
		warm_start	<comm_name>		
		link_down	<comm_name>		
		link_up	<comm_name>		
		auth_fail	<comm_name>		
		egp	<comm_name>		
		enterprise	<comm_name>		

(continued on next page)

Table 22–2 SNMP Commands Options Summary (Cont.)

Command	Param 1	Param 2	Param 3	Param 4	Default
list	all				
	community	access			
		traps			
		address			
		view			
	views				
set	community	access	read_trap	<comm_name>	
			write_read_trap	<comm_name>	
			trap_only	<comm_name>	
		view	<comm_name>	all	
				<view>	
	trap_port	<udpPort#>			
exit					

? (Help)

List the commands that are available from the current prompt level. You can also enter **?** after a specific command name to list its options.

Syntax: ?

Example: **?**

```
ADD
DELETE
DISABLE
ENABLE
LIST
EXIT
```

Add

Add a community name to the list of SNMP communities, add an address to a community, or assign a portion of the MIB (subtree) to a view.

Syntax: add address
community
sub_tree

address

Note: SNMP requests may arrive for any of the routers' addresses.

You may specify one or more address for a community. You must enter the command each time you want to add another address.

If you specify no addresses for a community, requests are handled from any host. The addresses also specify those hosts that receive the traps. If no addresses are specified, no traps are generated.

Example: **add address <community name> <ip address> <ip mask>**

```
Community Name [trap]?
New Address [0.0.0.0]?
```

community

Use the **add community** command to create a community with read_trap access, a view of all, allows all IP addresses access, and all traps disabled.

Note: The **add community** command no longer allows you to select access type or trap control. Use the **set community access** command to assign access types to existing SNMP communities.

Example: **add community <community name>**

```
Community Name  []?
```

Community Name Specifies the name of community (32 characters maximum). Characters such as spaces, tabs, or <esc> key sequences are not accepted.

sub_tree

Use the **add sub_tree** command to add a portion of the MIB to a view or to create a new view. The **add sub_tree** command is used to manage MIB views. More than one subtree can be added to a view defined by <view_text_name>. To create a new MIB view, issue the **add sub_tree** command with the new view name.

Note: You must assign a view to one or more communities using the **set community view** command to have it take effect.

Example: **add sub_tree**

```
View Name  [system-only]?
MIB OID name [1.3.6.1.2.1.1]?
```

View Name	Specify the name of the view (32 visual characters maximum. Characters such as spaces, tabs, or <esc> key sequences are not accepted.
MIB OID	Specifies the MIB Object ID for the sub_tree. This must be entered as a numeric value, not a symbolic value.

Delete

Delete:

- An address from a community.
- A community and all of its addresses.
- A subtree from a view.

Syntax: delete address
community
sub_tree

address

Removes an address from a community. You must supply the name.

Example: **delete address <community name> <ip address>**

community

Removes a community and its IP addresses. You must supply the community name.

Example: **delete community <community name>**

sub_tree

Removes a MIB or a portion of the MIB from a view. You must supply the name of the subtree. If all subtrees are deleted, the MIB view is also deleted and all references to it from any associated SNMP communities are removed.

Example: **delete sub_tree <object identifier>**

Disable

Disable the snmp protocol or specified traps on the router.

Syntax: disable snmp
trap

snmp

Disables SNMP

Example: **disable snmp**

trap

Disables specified traps or all traps. You must specify the trap type from the options shown below.

Example: **disable trap <trap_type> <community name>**

Trap Type	Description
all	Disables all traps in a specified community.
cold_start	Disables cold start traps in a specified community. A cold start trap (0) means that the transmitting router is reinitializing and that the agent's configuration or the protocol entity implementation may be altered.
warm_start	Disables warm start traps in a specified community. A warm start trap (1) means that the transmitting router is reinitializing, but the configuration or protocol implementation remains the same.
link_down	Disables link_down traps in a specified community. A link_down trap (2) recognizes a failure in one of the communication links represented in the agent's configuration. The link_down trap–PDU contains the name and value of the ifIndex instance for the affected link as the first element of its variable–bindings.
link_up	Disables link_up traps in a specified community. A link_up trap recognizes that a previously inactive link in the network has come up. The link_up trap–PDU contains the name and value of the ifIndex instance for the affected link as the first element of its variable–bindings.
auth_fail	Disables authentication failure traps for a specified community. Authentication failure traps recognize that the sending entity is the addressee of a protocol message that is not properly authenticated.

(continued on next page)

Trap Type	Description
egp	Disables egp neighbor loss traps in a specified community. EGP Neighbor Loss traps recognize that an EGP neighbor and peer are marked down and no longer a peer. The egpNeighborLoss trap–PDU contains the name and value of the egpNeighAddr instance for the affected neighbor as the first element of its variable–bindings.
enterprise	Disables enterprise specific traps in a specified community. Enterprise specific traps recognize that some enterprise specific event has occurred. The specific–trap field identifies the particular trap that occurred.

Enable

Enable the snmp protocol or specified traps on the router.

Syntax: enable snmp
trap

snmp

Enables SNMP

Example: **enable snmp**

trap

Enables specified traps or all traps. You must specify the trap type from the options shown below.

Example: **enable trap <trap_type> <community name>**

Trap Type	Description
all	Enables all traps in a specified community.
cold_start	Enables cold start traps in a specified community. A cold start trap (0) means that the transmitting router is reinitializing and that the agent's configuration or the protocol entity implementation may be altered.
warm_start	Enables warm start traps in a specified community. A warm start trap (1) means that the transmitting router is reinitializing, but the configuration or protocol implementation remains the same.
link_down	Enables link_down traps in a specified community. A link_down trap (2) recognizes a failure in one of the communication links represented in the agent's configuration. The link_down trap–PDU contains the name and value of the ifIndex instance for the affected link as the first element of its variable–bindings.
link_up	Enables link_up traps in a specified community. A link_up trap recognizes that a previously inactive link in the network has come up. The link_up trap–PDU contains the name and value of the ifIndex instance for the affected link as the first element of its variable–bindings.
auth_fail	Enables authentication failure traps for a specified community. Authentication failure traps recognize that the sending entity is the addressee of a protocol message that is not properly authenticated.

(continued on next page)

Trap Type	Description
egp	Enables egp neighbor loss traps in a specified community. EGP Neighbor Loss traps recognize that an EGP neighbor and peer are marked down and no longer a peer. The egpNeighborLoss trap–PDU contains the name and value of the egpNeighAddr instance for the affected neighbor as the first element of its variable–bindings.
enterprise	Enables enterprise specific traps in a specified community. Enterprise specific traps recognize that some enterprise specific event has occurred. The specific–trap field identifies the particular trap that occurred.

List

Display the current configuration of SNMP communities, access modes, traps, and network addresses, and views.

Syntax: list all
community
views

list all

Displays the current configuration of SNMP communities for Access, Traps, Address, and View. See the description for the **list community** command on the next page for details on the options.

Example: **list all**

```
      Community Name_______      _______ Access_____ __
public                           Read, Trap
johnp                            Read, Write, Trap
trap                             Trap Only
snmp                             Read, Trap
```

```
      Community Name______     IP Address     IP Mask
public                         All            N/A
jonhp                          16.24.10.98    255.255.255.0
trap                           16.24.10.98    255.255.255.0
snmp                           All            N/A

      Community Name______          Enabled Traps____ __
public                         None
johnp                          None
trap                           Cold Restart, Warm Restart,
                               Link Down, Link Up,
                               Authentication Failure,
                               EGP Neighbor Loss, Enterprise Specific
snmp                           None

      Community Name______         View____ __
public                         All
johnp                          All
trap                           All
snmp                           mibii-snmp

         View Name_____  _          Sub-Tree
mibii-snmp                     1.3.6.1.2.1.11
```

list community

Displays the specified options for all communities. Options are access, traps, address, view.

Example: **list community <option>**

Access	Displays the access modes for all communities.
Address	Displays the network address for all communities.
Traps	Displays the types of traps generated for all communities.
View	Displays the MIB view for all communities.

list community access

Example: **list community access**

```
      Community Name______        Access____ __
public                         Read, Trap
johnp                          Read, Write, Trap
trap                           Trap Only
snmp                           Read, Trap
```

list community address

Example: **list community address**

Community Name	IP Address	IP Mask
public	All	N/A
jonhp	16.24.10.98	255.255.255.0
trap	16.24.10.98	255.255.255.0
snmp	All	N/A

list community traps

Example: **list community traps**

Community Name	Enabled Traps
public	None
johnp	None
trap	Cold Restart, Warm Restart, Link Down, Link Up, Authentication Failure, EGP Neighbor Loss, Enterprise Specific
snmp	None

list community view

Example: **list community view**

Community Name	View
public	All
johnp	All
trap	All
snmp	mibii-snmp

list views

Displays the current views for a specified SNMP community.

Example: **list views**

View Name	Sub-Tree
mibii-snmp	1.3.6.1.2.1.11

Set

Assign a MIB view to a community or to set the SNMP udp port numbers.

Syntax: set community access
view
trap_port

community *access*

Use the **set community access** command to assign one of three access types to a community. You must supply the name of the community and the network address (in the standard *a.b.c.d.* notation).

Example: **set community access <options> <comm_name>**

read_trap	Sets read access and trap generation to the named community.
write_read_trap	Sets write and read trap access to the community specified.
trap_only	Sets the trap port to the named community.

community view

Use the **set community view** command to assign a MIB view to a community.

Example: **set community view <comm_name> <options>**

all	Assigns all supported MIB views to the named community. All is the default.
view	Assigns a specified MIB view to the named community.

trap_port

Use the **set trap_port** command to specify a UDP prot number to send traps to the trap port. The default is the standard port.

Example: **set trap_port <udpport#>**

UDP Port Number	Specifies a User Datagram Protocol port other than the standard UDP port (default # 162).

Exit

Return to the `Config>` prompt.

Syntax: exit

`Example:` **`exit`**

20

Monitoring SNMP

This chapter describes the SNMP console commands.

For more information about SNMP, refer to the *Routing Protocols Reference Guide*.

Accessing the SNMP Console Environment

For information about accessing the SNMP console environment, see Chapter 1.

SNMP Console Commands

This section summarizes and then explains all of the SNMP console commands.

Table 20–1 lists the SNMP console commands. Enter the SNMP console commands at the `SNMP>` prompt.

Table 20–1 SNMP Console Command Summary

Command	Function
? (Help)	Lists all the SNMP console commands or lists the options associated with specific commands.
List	Displays the current configuration of SNMP communities, authentication types, access modes, traps, and network addresses.
Statistics	Displays statistics about the number of defined variables and the size of the MIB.
Exit	Exits the SNMP console process and returns to the GWCON environment.

? (Help)

List the commands that are available from the current prompt level. You can also enter a **?** after a specific command name to list its options.

Syntax: ?

Example: **?**

```
LIST
STATISTICS
EXIT
```

List

Display the current configuration of SNMP communities, access modes, traps, and network addresses, and views.

Syntax: list all
community
views

list all

Displays the current configuration of SNMP communities for Access, Traps, Address, and View. See the description for the **list community** command on the next page for details on the options.

Example: **list all**

```
      Community Name______      ______ Access____ __
public                          Read, Trap
johnp                           Read, Write, Trap
trap                            Trap Only
snmp                            Read, Trap

      Community Name______      IP Address     IP Mask
public                          All            N/A
jonhp                           16.24.10.98    255.255.255.0
trap                            16.24.10.98    255.255.255.0
snmp                            All            N/A
```

```
      Community Name______      Enabled Traps____ __
public                          None
johnp                           None
trap                            Cold Restart, Warm Restart,
                                Link Down, Link Up,
                                Authentication Failure,
                                EGP Neighbor Loss, Enterprise Specific
snmp                            None

      Community Name______      View____ __
public                          All
johnp                           All
trap                            All
snmp                            mibii-snmp

         View Name_____  _      Sub-Tree
mibii-snmp                      1.3.6.1.2.1.11
```

list community

Displays the specified options for all communities. Options are access, traps, address, view.

Example: **list community <option>**

Access	Displays the access modes for all communities.
Address	Displays the network address for all communities.
Traps	Displays the types of traps generated for all communities.
View	Displays the MIB view for all communities.

list community access

Example: **list community access**

```
      Community Name______      Access____ __
public                          Read, Trap
johnp                           Read, Write, Trap
trap                            Trap Only
snmp                            Read, Trap
```

list community address

Example: **list community address**

Community Name	IP Address	IP Mask
public	All	N/A
jonhp	16.24.10.98	255.255.255.0
trap	16.24.10.98	255.255.255.0
snmp	All	N/A

list community traps

Example: **list community traps**

Community Name	Enabled Traps
public	None
johnp	None
trap	Cold Restart, Warm Restart, Link Down, Link Up, Authentication Failure, EGP Neighbor Loss, Enterprise Specific
snmp	None

list community view

Example: **list community view**

Community Name	View
public	All
johnp	All
trap	All
snmp	mibii-snmp

list views

Displays the current views for a specified SNMP community.

Example: **list views**

View Name	Sub-Tree
mibii-snmp	1.3.6.1.2.1.11

Statistics

Display the statistics about the number of defined variables and the size of the MIB. The statistics can change only when the load or hardware configuration changes.

Syntax: statistics

Example: **statistics**

```
Number of defined variables = 231
Size of MIB = 14320 bytes
```

Exit

Return to the previous prompt level.

Syntax: exit

Example: **exit**

21

Configuring Bandwidth Reservation

This chapter describes how to access the Bandwidth Reservation System (BRS) configuration prompt and the available bandwidth reservation configuration commands.

Displaying the Bandwidth Reservation Configuration Prompt

To access bandwidth reservation configuration commands and to configure bandwidth reservation on your router, perform the following steps:

1. At the `*` prompt, enter **talk 6**.
2. At the `Config>` prompt, type **feature brs**.
3. At the `BRS Config>` prompt, type **interface #**.
4. At the `BRS [i 0] Config>` prompt, type **enable**. (This is the interface prompt level, and the interface number is zero in this instance.)
5. For Frame Relay interfaces select PVCs using the **circuit** command. At the `[BRS i 0] [dlci 16] Config>` prompt, type **enable.** (This is the circuit prompt, and the circuit number is 16 in this example.)
6. Restart your router.
7. Repeat Steps 1 through 4 to configure bandwidth reservation for the particular interface that you have enabled.
8. At the `BRS [i 0] Config>` prompt, configure the bandwidth reservation parameters for the selected interface by using the appropriate configuration commands discussed in this chapter. If this is a Frame Relay interface, configure circuit classes at this prompt.

9. For Frame Relay interfaces, select PVCs using the **circuit** command. At the `[BRS i 0] [dlci 16] Config>` prompt configure the bandwidth reservation parameters for the selected circuit using configuration commands discussed in this chapter. (This is the circuit prompt, and the circuit number is 16 in this example.)

10. Restart your router.

The **talk 6** (**t 6**) command lets you access the configuration process.

The **feature brs** command lets you access the BRS configuration process. You can enter this command by using either the feature name (brs) or number (1).

The **interface #** command selects the particular interface that you want to configure for bandwidth reservation. Before configuring bandwidth reservation, you must select the interface to be configured. In Step 4, the prompt indicates that the selected interface's number is zero.

You must enable bandwidth reservation for the selected interface and restart your router before configuring the particular interface.

To return to the `Config>` prompt at any time, enter the **exit** command at the `BRS config>` prompt.

Bandwidth Reservation Configuration Commands

Table 21–1 describes the bandwidth reservation configuration commands. The commands marked by an asterisk are used only with Frame Relay. (The asterisk is not part of the command.)

Table 21–1 Bandwidth Reservation Configuration Commands

Command	Function
? (Help)	Displays the bandwidth reservation configuration commands or lists options for specific commands (if available).
Add-circuit-class*	Sets the name of a circuit class and its percentage of bandwidth.
Add-class	Allocates a designated amount of bandwidth to a user-defined bandwidth class.
Assign	Assigns a protocol or filter to a reserved class.
Assign-circuit*	Assigns a specified circuit to the specified circuit class.
Change-circuit-class*	Changes the percentage of the bandwidth to be used by the group of circuits assigned to the designated class.
Change-class	Changes the amount of bandwidth configured for a bandwidth class.
Circuit #	Selects the DLCI of a Frame Relay permanent virtual circuit.
Clear-block	Clears the current reservation configuration from SRAM. (**Note:** This command requires a router restart.)
Deassign	Restores a specified protocol or filter to its default class and priority.
Deassign-circuit*	Deassigns the specified circuit from the circuit class to which it was assigned.
Default-circuit-class*	Assigns the name of the default circuit class.
Default-class	Sets the default class and priority to a desired value.
Del-circuit-class*	Deletes the specified circuit class.
Del-class	Deletes a previously configured bandwidth class from the specified interface.
Disable	Disables bandwidth reservation on the interface or Frame Relay circuit. (**Note:** This command requires a router restart.)

(continued on next page)

Command	Function
Enable	Enables bandwidth reservation on the interface or Frame Relay circuit. (**Note:** This command requires a router restart.)
Interface	Selects the serial interface that runs bandwidth reservation. Use this command to enable BRS on an interface. **Note:** This command must be entered BEFORE using any other configuration commands.
List	Displays the currently defined bandwidth classes by their guaranteed percentage rates and priority queuing values stored in the SRAM display. Also displays the assigned protocols and filters. (For Frame Relay, this command provides two levels of information.)
Show	Displays the currently defined bandwidth classes stored in RAM. (For Frame Relay, this command provides two levels of information.)
Tag	Assigns a class and priority to a filter that was tagged during the configuration of the MAC filtering feature.
Untag	Removes the tag/tag name relationship and the tag name from the list of assignable filters.
Exit	Exits from one BRS level to another or exits the bandwidth reservation configuration process.

Except for the commands marked with an asterisk, which are only for Frame Relay, the commands in Table 21-1 are the same for configuring bandwidth reservation for the Proteon Serial Line protocol, the Point-to-Point protocol (PPP), Frame Relay, Integrated Services Digital Network (ISDN), and V.25 *bis*.

Note: When the **clear-block**, **disable**, **enable**, **list,** and **show** commands are issued from within the BRS interface level, they affect or list the bandwidth reservation information configured for the selected interface. When these commands are issued from within the BRS circuit level, they affect only the FR bandwidth reservation information configured for the permanent virtual circuit (PVC.).

Note: Before using the bandwidth reservation commands, keep the following in mind:

- You must use the **interface** command to select a serial interface before you use any other configuration commands. (BRS configuration enforces this.)

- The Class-name parameter is case-sensitive.
- To view the current class names, use the **list** or **show** command.

? (Help)

At the BRS prompt, use the **? (help)** command to list the available commands from the current prompt level. You can also enter **?** after a specific command name to list its options.

Syntax: ?

Example: **?**

At the `BRS [i #] [dlci #] Config>` and the `BRS Config>` prompts, the following commands are listed:

```
INTERFACE
LIST
EXIT
```

At the `BRS [i #] Config>` prompt (for non-FR), the following commands are listed:

```
ENABLE
DISABLE
ADD-CLASS
DEL-CLASS
CHANGE-CLASS
DEFAULT-CLASS
ASSIGN
DEASSIGN
INTERFACE
LIST
SHOW
CLEAR-BLOCK
TAG
UNTAG
EXIT
```

At the `BRS [i #] Config>` prompt (for FR), the following commands are listed:

```
ENABLE
DISABLE
CIRCUIT
ADD-CIRCUIT-CLASS
DEL-CIRCUIT-CLASS
CHANGE-CIRCUIT-CLASS
DEFAULT-CIRCUIT-CLASS
ASSIGN-CIRCUIT
DEASSIGN-CIRCUIT
LIST
SHOW
CLEAR-BLOCK
EXIT
```

Add-circuit-class

Use the **add-circuit-class** command at the interface level to allocate a designated amount of bandwidth to be used by the group of Frame Relay circuits assigned to the circuit class.

Syntax: add-circuit-class *class-name* *%*

Example: **`add-circuit-class alpha 10`**

Here *class-name* is the ASCII string assigned as the name of the circuit class, and % is a percentage of the bandwidth - between 1 and 100 - of the interface.

Add-class

Allocate a designated amount of bandwidth to a user-defined bandwidth class.

Syntax: add-class *class-name* *%*

Example: **`add test 20`**

Here *class-name* is the ASCII string assigned as the name of the bandwidth class, and % is a percentage of the bandwidth of the interface or Frame Relay circuit.

Assign

Assign specified tags, protocol packets, or filters to a given class and priority within that class. The four priority types include:

- Urgent
- High
- Normal (the default priority)
- Low

Syntax: assign *protocol* or *TAG* or *filter* *class-name*

```
Example: assign AP2 test
           priority <URGENT/HIGH/NORMAL/LOW>: [NORMAL]? low
           protocol AP2 maps to class test with priority LOW
```

Assign-circuit

Use the **assign-circuit** command at the interface level to assign the specified circuit (DLCI) to the specified circuit class.

Syntax: assign-circuit *#* *class-name*

```
Example: assign-circuit 16 pubs
```

Change-circuit-class

Use the **change-circuit-class** command at the interface level to change the percentage of the bandwidth to be used by the group of circuits assigned to the circuit class.

Syntax: change-circuit-class *class-name* *%*

```
Example: change-circuit-class alpha 20
```

Change-class

Change the amount of bandwidth configured for a bandwidth class.

Syntax: change-class *class-name* or *class#* *%*

```
Example: change test 10
```

Circuit

Use the **circuit** command to select the DLCI of a Frame Relay PVC for configuration. This command can only be issued from the BRS interface configuration prompt (`BRS [i #] Config>`).

Syntax: circuit *permanent-virtual-circuit#*

`Example:` **`circuit 16`**

When the FR circuit is enabled, the following commands may be used at the circuit prompt:

- **add-class**
- **assign**
- **change-class**
- **clear-block**
- **deassign**
- **default-class**
- **del-class**
- **disable**
- **exit**
- **list**
- **show**
- **tag**
- **untag**

Clear-block

Clear the current bandwidth reservation configuration from SRAM for the current interface or Frame Relay PVC. This command requires a router restart.

Syntax: clear-block

Example: **clear-block**

```
You are about to clear BRS configuration information
Are you sure you want to do this (Yes or No): y
BRS [i 0] Config>
```

Deassign

Restore a specified protocol, TAG, or filter to its default class and priority.

Syntax: deassign *protocol* or *TAG* or *filter*

Example: **deassign IP**

Deassign-circuit

Use the **deassign-circuit** command at the interface level to deassign the specified circuit (DLCI) from the circuit class to which it was previously assigned.

Syntax: deassign-circuit *permanent-virtual-circuit#*

Example: **deassign 16**

Default-circuit-class

Use the **default-circuit-class** command at the interface level to select the name of the default circuit class.

Syntax: default-circuit-class *class-name*

Example: **default-circuit-class group**

Default-class

Set the default class and priority to a desired value. If no value was previously assigned, system default values are used. Otherwise, the last previously assigned value is used.

Syntax: default-class *class-name* or *class# priority*

Example: **default-class test normal**

Del-circuit-class

Use the **del-circuit-class** command at the interface level to delete the specified bandwidth class.

Syntax: del-circuit-class *class-name*

Example: **del-circuit-class group**

Del-class

Delete a previously configured bandwidth class from the specified interface or Frame Relay circuit.

Syntax: del-class *class-name* or *class#*

Example: **del-class ip**

Disable

Disable bandwidth reservation on the interface or Frame Relay circuit. This command requires a router restart.

To verify that bandwidth reservation is disabled, enter the **list** command.

Syntax: disable

Example: **disable**

Enable

Enable bandwidth reservation on the interface or Frame Relay circuit. This command requires a router restart.

Syntax: enable

Example: **enable**

Interface

Select the serial interface to which bandwidth reservation configuration commands are to be applied. Bandwidth reservation is supported on routers running the Proteon Serial Line protocol, PPP, Frame Relay, V.25 *bis*, and ISDN interfaces.

Note: To enter bandwidth reservation commands for a new interface, you must enter this command BEFORE using any other bandwidth reservation configuration commands. If you have exited the bandwidth reservation prompt and want to return to make bandwidth reservation changes to a previously configured interface, you must again enter this command first.

To configure bandwidth reservation on a particular interface, at the `BRS Config>` prompt, enter the number of the interface that supports the particular protocol or feature. You can then use BRS configuration commands as described in this chapter.

Syntax: interface *interface#*

Example: **`interface 2`**

List

Display currently defined bandwidth classes by their guaranteed percentage rates and priority queuing values stored in SRAM. This command also displays all assigned protocols and filters.

Syntax: list

Example: **`list`**

Depending on the prompt at which you issue the **list** command, various outputs appear. You can issue the **list** command from the following example prompts:

`BRS Config>`
`BRS [i 1] Config>` (for PPP interface 1)
`BRS [i 0] Config>` (for FR interface 0)
`BRS [i 0] [dlci 16] Config>` (for circuit 16 on FR interface 0)

For example, the following output appears when you issue the list command at the `BRS Config>` prompt:

```
Bandwidth Reservation is available for 2 interfaces.
Interface      Type           State
---------      ----           -----
       0       FR             Enabled
       1       PPP            Enabled
```

The **list** command is very similar to the **show** command. However, **show** displays current settings from the active RAM.

Note: For Frame Relay, there are two levels of this command: the interface level and the circuit level.

Show

Display currently defined bandwidth classes stored in RAM.

Syntax: show

`Example:` **`show`**

Depending on the prompt at which you issue the **show** command, various outputs are displayed. You can issue the **show** command from the following prompts:
`BRS [i 1] Config>` (for PPP interface 1)
`BRS [i 0] Config>` (for FR interface 0)
`BRS [i 0] [dlci 16] Config>` (for circuit 16 on FR interface 0)

Tag

Assign a class and priority to a filter that was tagged during the configuration of the MAC filtering feature. The command requires a filter tag number (configured in MAC filtering), to reference the tag in bandwidth reservation. Refer to the Using MAC Filtering chapter in the *Bridging Configuration Guide.*

Up to five tagged MAC addresses can be set from 1 to 5. TAG1 is searched for first, then TAG2, and so on up to TAG5.

Any newly added address filter can then be assigned a tag (as any other protocol or filter) with the **assign** command. See the **assign** command in this chapter for more information.

Syntax: tag *tag#*

`Example:` **`tag 3`**

Untag

Remove the tag/tag name relationship and the tag name from the list of assignable filters.

A tag can only be removed if it is not assigned to any class.

Syntax: untag *tag#*

Example: **untag 3**

Exit

Use the **exit** command to do the following:

- Return from the circuit level to the interface level.
- Return from the interface level to the `BRS Config>` level.
- Return from the `BRS Config>` level to the `Config>` level.

Syntax: exit

Example: **exit**

22

Monitoring Bandwidth Reservation

This chapter describes how to access the Bandwidth Reservation System (BRS) monitoring prompt and the available monitoring commands.

Displaying the Bandwidth Reservation Monitoring Prompt

To access bandwidth reservation monitoring commands and to monitor bandwidth reservation on your router, do the following:

1. At the OPCON prompt (*), type **t 5**.
2. At the GWCON prompt (+), type **feature brs**.
3. At the BRS> prompt, type **interface #**. (Enter the number [#] of the interface that you want to monitor.)
4. For Frame Relay (only), type **circuit #** or issue one of the circuit class monitoring commands (for example **counters-circuit-class).**
5. At the prompt, type the appropriate monitoring command**.** (Refer to the "Bandwidth Reservation Monitoring Commands" section.)

The **talk 5** (**t 5**) command lets you access the monitoring process.

The **feature brs** command lets you access the BRS monitoring process. You can enter this command by using either the feature name (brs) or number (1).

The **interface #** command selects the particular interface that you want to monitor for bandwidth reservation.

The **circuit #** command selects the DLCI of a Frame Relay permanent virtual circuit (PVC).

The **counters #** command allows you to display statistics on BRS traffic for the selected interface.

To return to the GWCON prompt at any time, type **exit** at the BRS> prompt.

Once you access the bandwidth reservation monitoring prompt (BRS>), you can enter any of the specific monitoring commands described in Table 22–1.

Bandwidth Reservation Monitoring Commands

This section explains the bandwidth reservation monitoring commands. You enter the commands at the BRS> prompt.

Table 22–1 Bandwidth Reservation Monitoring Commands

Command	Function
? (Help)	Displays all the bandwidth reservation commands or lists sub-command options for specific commands (if available).
Circuit	Selects the DLCI of a Frame Relay permanent virtual circuit (PVC). To monitor Frame Relay bandwidth reservation traffic, you must be at the circuit prompt level.
Clear	Clears the current reservation counters and stores them as **last** command counters. Counters are listed by class usage.
Clear-circuit-class	Clears the reservation counters for all the circuit classes of the interface.
Counters	Displays the current counters.
Counters-circuit-class	Displays the current counters for all the circuit classes of the interface
Interface	Selects the serial interface to run bandwidth reservation. **Note:** You must enter this command BEFORE you use any other bandwidth reservation monitoring commands.
Last	Displays the last saved statistics.
Last-circuit-class	Displays the last saved statistics for all the circuit classes of the interface.
Exit	Exits the bandwidth reservation monitoring process.

? (Help)

List the commands that are available from the current prompt level. You can also enter **?** after a specific command name to list its options.

Syntax: ?

Example: **?**

At the BRS> prompt:

```
INTERFACE
EXIT
```

At the non-Frame Relay `BRS [i #]>` prompt:

```
COUNTERS
CLEAR
LAST
EXIT
```

For Frame Relay, at the `BRS [i #]>` prompt:

```
COUNTERS-CIRCUIT-CLASS
CLEAR-CIRCUIT-CLASS
LAST-CIRCUIT-CLASS
CIRCUIT
EXIT
```

For Frame Relay, at the `BRS [i #] [dlci #]>` prompt:

```
COUNTERS
CLEAR
LAST
EXIT
```

Circuit

Select the DLCI of a Frame Relay PVC for monitoring. This command can only be issued from the BRS interface monitoring prompt (`BRS [i #]>`).

Syntax: circuit *permanent-virtual-circuit #*

Example: **`circuit 16`**

After the FR circuit is selected, the following commands can be used at the circuit prompt:

```
COUNTERS
CLEAR
LAST
EXIT
```

Clear

Clear from RAM the current bandwidth reservation counters for the selected interface or Frame Relay circuit, and store them as counters that can be made available by the **last** command.

Syntax: clear

Example: **`clear`**

Clear-circuit-class

Enter the **clear-circuit-class** command at the `BRS [i #]>` prompt. It clears the current bandwidth reservation counters for the circuit classes of the selected Frame Relay interface. This command clears the counters from RAM and stores them as counters that you can display with **last-circuit-class.**

Syntax: clear-circuit-class

Example: **`clear`**

Counters

Display statistics describing bandwidth reservation traffic for the selected interface or Frame Relay circuit according to the configured classes.

Syntax: counters

Example: **`counters`**

Counters-circuit-class

Enter the **counters-circuit-class** command at the `BRS [i #]>` prompt. It displays statistics describing bandwidth reservation traffic for the circuit classes of the selected Frame Relay interface.

Syntax: counters-circuit-class

Example: **`counters-circuit-class`**

```
            Bandwidth Reservation Circuit Class Counters
            Interface 0

   Class         Pkt Xmit     Bytes Xmit     Bytes Ovfl

   DEFAULT         103          57692              0
   new            2149        1730056              0
   CLASS 2           0              0              0


   TOTAL          2252        1787748              0
```

Interface

Select the serial interface to which bandwidth reservation monitoring commands are to be applied. Bandwidth reservation is supported on routers running the Proteon Serial Line protocol (PSL), PPP, Frame Relay, V.25 *bis*, and Integrated Services Digital Network (ISDN) interfaces.

Note: To enter bandwidth reservation commands for a new interface, you must enter this command BEFORE using any other bandwidth reservation monitoring commands. If you have exited the bandwidth reservation monitoring prompt (BRS>) and want to return to monitor bandwidth reservation, you must again enter this command first.

To monitor bandwidth reservation on a particular interface, at the BRS> prompt, type the number of the interface. You can then use bandwidth reservation monitoring commands as described in this chapter.

Syntax: interface *interface#*

Example: **interface 0**

Last

Display the last saved bandwidth reservation statistics. The statistics are displayed in the same format as they are for the **counters** command.

Syntax: last

Example: **last**

Last-circuit-class

Enter the **last-circuit-class** command at the BRS [i #]> prompt. It displays the last saved bandwidth reservation statistics for the circuit classes of the selected Frame Relay interface. The statistics are displayed in the same format as they are for the **counters-circuit-class** command.

Syntax: last

Example: **last-circuit-class**

Exit

Use the **exit** command to:

- Return from the Frame Relay circuit level to the interface level.
- Return from the interface level to the `BRS>` level.
- Return from the `BRS>` level to the `GWCON (+)` prompt.

Syntax: exit

`Example:` **`exit`**

23

Configuring BGP4

This chapter describes how to configure the Border Gateway Protocol (BGP) using the BGP configuration commands.

Border Group Protocol Overview

BGP is an exterior gateway routing protocol used to exchange network reachability information among autonomous systems (ASs). An AS is essentially a collection of routers and endnodes that operate under a single administrative organization. Within each AS, routers and endnodes share routing information using an interior gateway protocol. The interior gateway protocol may be either RIP, OSPF, or Integrated IS-IS.

BGP was introduced in the Internet in the late 1980s to facilitate the loop-free exchange of routing information between autonomous systems (ASs). Based on Classless Inter-Domain Routing (CIDR), BGP has since evolved to support the *aggregation* and *reduction* of routing information.

In essence, CIDR is a strategy designed to address the following problems:

- Exhaustion of Class B address space
- Routing table growth

CIDR eliminates the concept of address classes, and provides a method for summarizing *n* different routes into single routes. This significantly reduces the amount of routing information that BGP routers must store and exchange.

Note: Digital only supports the latest version of BGP, BGP4, which is defined in RFC 1654. All references to BGP in this chapter and on the interface of Digital's routers are to BGP4, and do not apply to previous versions of BGP.

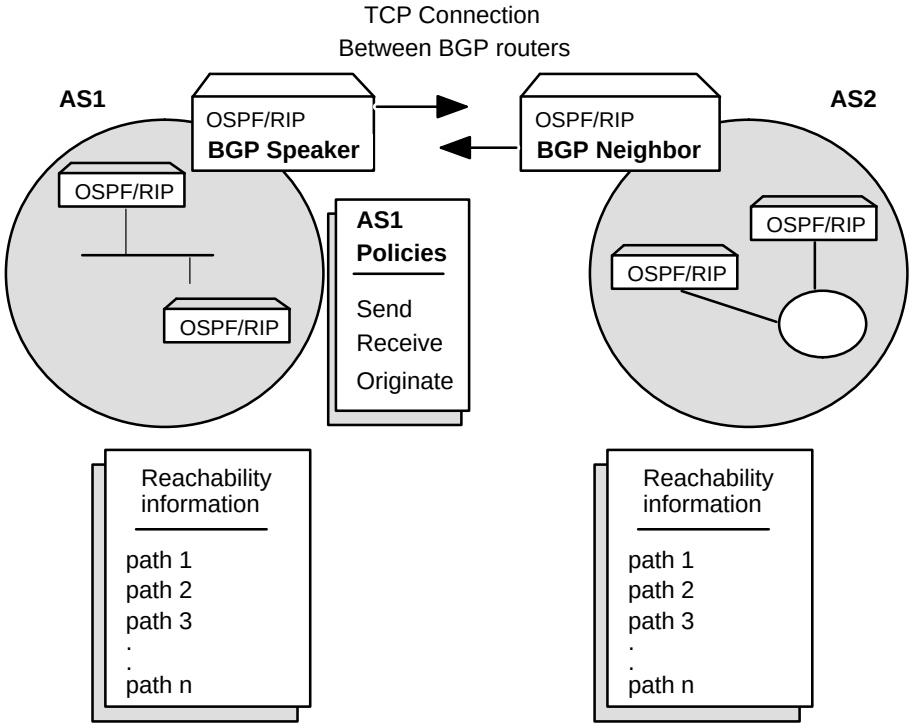
How BGP Works

BGP is not a routing protocol, but a reachability protocol. In essence, BGP routers selectively collect and advertise reachability information to and from BGP neighbors in their own and other autonomous systems (ASs). Reachability information consists of the sequences of AS numbers that form the paths to particular BGP speakers, and the list of IP addresses that can be reached via each advertised path. An *AS* is a administrative group of networks and routers that share reachability information using one or more Interior Gateway Protocols (IGPs), such as RIP or OSPF.

Routers that run BGP are called BGP speakers. These routers function as servers with respect to its BGP neighbors (its clients). Each BGP router opens a passive TCP connection on port 179, and listens for incoming connections from neighbors at this well-known address. The router also opens active TCP connections to enabled BGP neighbors. This TCP connection enables BGP routers to share and update reachability information with neighbors in the same or other ASs. Connections between BGP speakers in the same AS are called internal BGP (IBGP) connections, while connections between BGP speakers in different ASs are external BGP (EBGP) connections. A single AS may have one or many BGP connections to outside ASs.

Figure 23–1 shows two ASs. The BGP speaker in AS1 attempts to establish a TCP connection with its neighbor in AS2. After this connection is established, the routers can share reachability information.

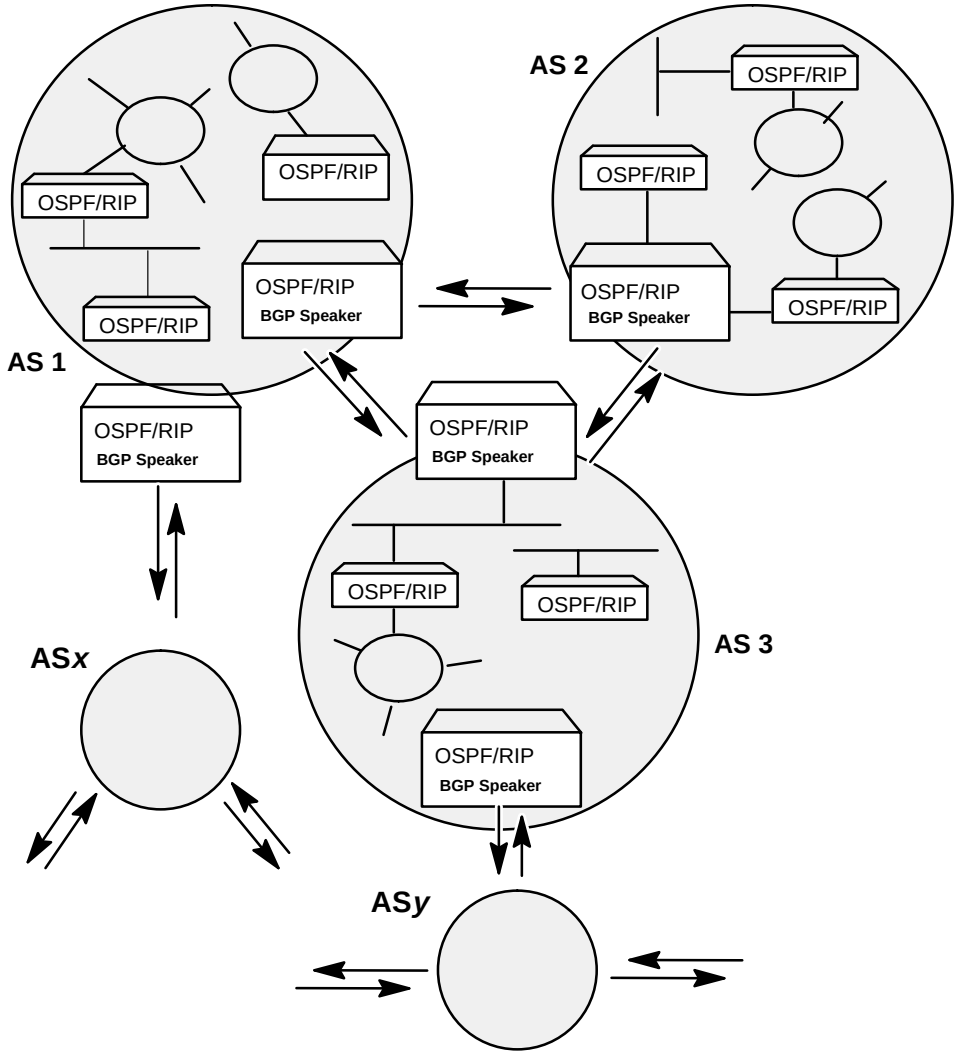
Figure 23–1 BGP Connections between Two Autonomous Systems



Once a the BGP Speaker in AS1 establishes a TCP connection with its BGP neighbor in AS2, the two routers can selectively exchange reachability information. The information each router sends or accepts is determined by policies defined for each router.

While the ASs shown in Figure 23–1 have only one BGP router, each may have multiple connections to other ASs. As an example of this, Figure 23–2 shows three interconnected ASs. AS1 has three BGP connections to outside ASs: one to AS2, one to AS3 and one to ASx. Similarly, AS3 has connections to AS1, AS2 and to ASy.

Figure 23–2 BGP Connections between Three Autonomous Systems



BGP relationships between three autonomous systems. Note that AS1 and AS3 have two BGP speakers.

Originate, Send and Receive Policies

Decisions about which reachability information to advertise (send) and which to accept (receive) are made on the basis of explicitly defined policy statements. Digital's BGP implementation supports three types of policy statements:

- Originate Policies
- Send Policies
- Receive Policies

Once a TCP connection is established, the BGP speaker shown in Figure 23–2 can send its entire routing table to its BGP neighbor in AS2. However, for security or other reasons, it may not be desirable to send reachability information on each network to AS2. Similarly, it may not be desirable for AS2 to receive reachability information on each network in AS1.

Note: Before you can send or receive information, you must establish policies.

BGP Messages

BGP routers use four kinds of messages to communicate with their neighbors: OPEN, KEEP ALIVE, UPDATE, and NOTIFICATION messages.

OPEN

Open messages are the first transmitted when a link to a BGP neighbor comes up and establishes a connection.

KEEP ALIVE

Keep alive messages are used by BGP routers to inform one another that a particular connection is alive and working.

UPDATE

Update messages contain the interior routing table information. BGP speakers only send update messages when there is a change in their routing tables.

NOTIFICATION

Nofication messages are sent whenever a BGP speaker detects a condition that forces it to terminate an existing connection. These messages are advertised before the connection is transmitted.

Setting Up BGP

Setting up BGP involves three basic steps:

1. Enabling BGP

   Enabling BGP requires you to specify the BGP router's unique AS Number. AS numbers are assigned by Stanford Research Institute Network Information Center.

2. Defining BGP Neighbors

   BGP Neighbors are BGP routers with which a BGP speaker establishes a TCP connection. Once neighbors are defined, connections to them are established by default.

3. Defining Policies

   The policies you establish determine which routes are imported and exported by the BGP speaker.

   You can set up policies for different purposes. See Sample Policy Definitions on page 23–8 for more information

The sections that follow explain each of these steps in detail.

Enabling BGP

You enable BGP using the **enable BGP speaker** command as shown.

```
BGP Config>enable BGP speaker
AS [0]? 167
TCP segment size [1024]?
```

The *AS number* must be greater than zero, but less than or equal to 65535.

The TCP segment size must be greater than zero, but less than or equal to 65535. The default value is 1024. This number represents the maximum segment size BGP uses for passive TCP connections.

Defining BGP Neighbors

After enabling a BGP speaker, you must define its neighbors. BGP neighbors can be internal or external. Internal neighbors exist in the same AS, and do not need to have a direct connection to one another. External neighbors exist in different ASs. These must have a direct connection to one another.

To define internal or external BGP neighbors, use the **add neighbor** command. You must specify the IP address of the neighbor, and assign an AS number to the neighbor as shown below. Internal neighbors must have the same AS number as the BGP speaker.

```
BGP Config>add neighbor 192.0.190.178
AS [0]? 178
Init timer [12]? 30
Connect timer [120]?
Hold timer [90]? 30
TCP segment size [1024]? 512
```

Adding a BGP neighbor automatically enables it, causing the BGP speaker to send out a connection request to the neighbor.

Adding Policies

Proteon's BGP implementation supports three policy commands:

- **Originate Policy** – Enables you to select the internal gateway protocol (IGP) networks to export. These policies apply to routes to which the BGP speaker is directly connected; that is, routes that are local to the BGP speaker.

- **Receive Policy** – Enables you to select the route information to import from BGP peers

- **Send Policy** – Enables you to select the route information to export to peers. Note that exportable route information can include information collected from neighboring ASs, as well as the routes that originate in the IGP.

Sample Policy Definitions

This section provides a set of examples of some specific policies you can set up for a BGP speaker.

You define all policies using the BGP **add** command. See page 23–11 for the syntax of the **add** command.

Originate Policy Examples

Include All Routes for Advertisement

This example includes all routes in the BGP speaker's IGP routing table for advertisement. In this sense, you can view this command as the default originate policy statement for BGP.

Notice that the command specifies a range of addresses, rather than a single (exact) address.

```
BGP Config>add originate-policy inclusive
Network Prefix [0.0.0.0]?
Network Mask [0.0.0.0]?
Address Match (Exact/Range) [Exact]? range
Tag [0]?
```

Exclude a Range of Routes

This example also specifies a range, but in this case the goal is to prevent the BGP Speaker from advertising addresses in this range to its neighbors.

This example excludes all routes in the range 194.10.16.0 and 194.10.31.255 from the BGP routing table, which in turn prevents them from being advertised.

```
BGP Config>add originate-policy exclusive
Network Prefix [0.0.0.0]? 194.10.16.0
Network Mask [0.0.0.0]? 255.255.240.0
Address Match (Exact/Range) [Exact]? range
Tag [0]?
```

Receive Policy Examples

Import all Routes from All BGP Neighbors

This example ensures that the BGP speaker import all routes from all of its neighbors into its IGP routing table.

```
BGP Config>add receive-policy inclusive
Network Prefix [0.0.0.0]?
Network Mask [0.0.0.0]?
Address Match (Exact/Range) [Exact]? range
Originating AS# [0]?
Adjacent AS# [0]?
IGP-metric [0]?
```

IGP-metric specifies the metric value with which the accepted routes are imported into the speaker's IGP routing table. You are only prompted to enter a value for IGP-metric when setting up a policy for route inclusion.

Block Specific Routes from a Transit AS

This example prevents the BGP speaker from importing any routes originating at AS 168 from neighboring AS 165. You might use this command if you do not want the BGP speaker to receive any routes from AS 168 for security reasons

```
BGP Config>add receive-policy exclusive
Network Prefix [0.0.0.0]?
Network Mask [0.0.0.0]?
Address Match (Exact/Range) [Exact]? range
Originating AS# [0]? 168
Adjacent AS# [0]? 165
```

Send Policy Examples

Restrict Route Advertisement to a Specific AS

This example restricts the BGP speaker. The speaker cannot advertise routes in the address range 143.116.0.0 to 143.116.255.255, that originate from AS 165, to autonomous system 168.

```
BGP Config>add send exclusive
Network Prefix [0.0.0.0]? 143.116.0.0
Network Mask [0.0.0.0]? 255.255.0.0
Address Match (Exact/Range) [Exact]? range
Tag [0]? 165
Adjacent AS# [0]? 168
```

Advertise All Known Routes

This example ensures that the BGP speaker advertises all routes originated from its IGP, and all routes learned from its neighboring autonomous systems.

```
BGP Config>add send-policy inclusive
Network Prefix [0.0.0.0]?
Network Mask [0.0.0.0]?
Address Match (Exact/Range) [Exact]? range
Tag [0]?
Adjacent AS# [0]?
```

BGP Commands

This section explains all BGP configuration commands. These commands allow you to modify the BGP protocol behavior to meet your specific requirements. Some amount of configuration is necessary to produce a fully functional BGP router. Enter BGP configuration commands at the `BGP config>` prompt.

Table 23–1 summarizes the BGP configuration commands.

Table 23–1 BGP Command Summary

Command	Function
? (Help)	Lists the configuration or monitoring commands or lists the actions associated with specific commands.
Add	Add BGP neighbors.
Change	Modifies information that was originally entered with the **add** command.
Clear	Erases the BGP configuration.
Delete	Deletes BGP configuration information that was entered with the **add** command.
Disable	Disables certain BGP features that were turned on by the **enable** command.
Enable	Enables BGP speakers or BGP neighbors.
List	Displays BGP configuration items.
Exit	Exits the process.

? (Help)

List the commands that are available from the current prompt level. You can also enter **?** after a specific command name to list its options.

Syntax: ?

`Example:` **`?`**

```
ADD
CHANGE
CLEAR
DELETE
DISABLE
ENABLE
LIST
EXIT
```

Add

Add BGP information to your configuration.

Syntax: add

aggregate . . .

neighbor . . .

no-receive asnum . . .

originate-policy . . .

receive-policy . . .

send-policy. . .

add aggregate *network prefix* *network mask*

The **add aggregate** command causes the BGP speaker to aggregate a block of addresses and advertise a single route to its BGP neighbors. You must specify the network prefix common to all the routes being aggregated and its mask.

The following example illustrates how to aggregate a block of addresses from 194.10.16.0 through 194.10.31.255.

Example: **add aggregate**

```
Network Prefix [0.0.0.0]? 194.10.16.0
Network Mask [0.0.0.0]? 255.255.240.0
```

When you add an aggregate definition, remember to define a policy to block the aggregated routes from being exported. If you do not, the router supports both the individual routes and the aggregate you have defined.

add neighbor ***neighbor IP address as# init timer connect timer hold timer keep alive timer tcp segment size***

Use the **add neighbor** command to define a BGP neighbor. The neighbor can be internal to the BGP speaker's AS, or external. An internal neighbor must exist on the same network as the speaker.

Example: **add neighbor**

```
Neighbor address [0.0.0.0]? 192.0.251.165
AS [0]? 165
Init timer [12]?
Connect timer [120]?
Hold timer [90]?
TCP segment size [1024]?
```

Neighbor address	Address of the neighbor you wish to peer with. It may be within your own autonomous system or in another autonomous system. If it's an external neighbor, both BGP speakers must share the same network. There is no such restriction for internal neighbors.
AS	Your own autonomous system number for internal neighbor or neighbor's autonomous system number.
Init Timer	Specifies the amount of time the BGP speaker waits to initialize resources and reinitiate transport connection with the neighbor in case the speaker has previously transitioned to IDLE state due to an error. If the error persists, this timer increases exponentially. The default is 12 seconds.
Connect Timer	The amount of time the BGP speaker waits to reinitiate transport connection to its neighbor, if the TCP connection fails while in either CONNECT or ACTIVE state. In the mean time, the BGP speaker continues to listen for any connection that may be initiated by its neighbor. The default is 120 seconds.

Hold Timer	The length of time the BGP speaker waits before assuming that the neighbor is unreachable. Both neighbors exchange the configured information in OPEN message and chose the smallest of the two timers as their negotiated Hold Timer value. The default is 90 seconds. Once neighbors have established BGP connection, they exchange KeepAlive messages at frequent intervals to ensure that the connection is still alive and the neighbors are reachable. The KeepAlive timer interval is calculated to be one third of the negotiated hold timer value. Hence the hold timer value must be either zero or at least threeseconds. Note that on switched lines, you may wish to have the Hold Timer value of zero to save bandwidth by not sending KeepAlives at frequent intervals.
TCP Segment Size	The maximum data size that may be exchanged on the TCP connection with a neighbor. This value is used for active TCP connection with the neighbor. It defaults to 1024, but can be set up to 65535.

add no-receive asnum

Use the **add no-receive asnum** to exclude updates from a particular AS.

```
Example: add no-receive
            Enter AS: [0]? 178
```

add originate-policy *(exclusive/ inclusive) network prefix network mask address match (Exact/Range) tag*

Use the **add originate-policy** command to create a policy that determines whether a specific address, or range of addresses, can be imported to the BGP speaker's routing table from the IGP routing table.

Exclusive	Exclusive policies prevent route information from being included in the BGP speaker's routing table.
Inclusive	Inclusive policies ensure that specific routes are included in the BGP speaker's routing table.

Network prefix	The network prefix for the addresses being affected.
Address match	The address, or range of addresses, that is affected by the policy statement.
Tag	The value that was set for a particular AS. All tag values match that of the AS from which they were learned.

The following example includes all routes in the BGP speaker's IGP routing table to be advertised.

Example: **add originate-policy exclusive**

```
Network Prefix [0.0.0.0]?
Network Mask [0.0.0.0]?
Address Match (Exact/Range) [Exact]? range
Tag [0]?
```

See page 23–8 for detailed examples of this policy command.

add receive-policy *(exclusive/ inclusive) network prefix network mask address match originating as# adjacent as# igp-metric (inclusive only)*

Use the **add receive-policy** command to determine what routes are imported to the BGP speaker's routing table.

Example: **add receive-policy exclusive**

```
Network Prefix [0.0.0.0]? 10.0.0.0
Network Mask [0.0.0.0]?  255.0.0.0
Address Match (Exact/Range) [Exact]? range
Originating AS# [0]? 168
Adjacent AS# [0]? 165
```

See page 23–8 for detailed examples of this policy command.

add send-policy *(exclusive/ inclusive) network prefix network mask address match tag adjacent as#*

Use the **add send-policy** command to create policies that determine which of the BGP speaker's learned routes are readvertised. These routes may be internal or external to the BGP speaker's AS.

Example: **add send exclusive**

```
Network Prefix [0.0.0.0]? 180.220.0.0
Network Mask [0.0.0.0]? 255.255.0.0
Address Match (Exact/Range) [Exact]? range
Tag [0]?
Adjacent AS# [0]? 25
```

See page 23–9 for detailed examples of this policy command.

Change

Change an BGP configuration item previously installed by the **add** command.

Syntax: change

- aggregate . . .
- neighbor . . .
- originate-policy . . .
- receive-policy . . .
- send-policy. . .

change aggregate *index# network prefix network mask*

This example changes the current aggregate (aggregate 1). The change causes aggregate 1 to use a different network prefix and mask to aggregate all routes in the address range from 128.185.0.0 to 128.185.255.255.

Example: **change aggregate 1**

```
Network Prefix [128.185.0.0]? 128.128.0.0
Network Mask [255.255.0.0]? 255.192.0.0
```

change neighbor ***neighbor IP address as# init timer connect timer hold timer keep alive timer tcp segment size***

The following example changes the value of the hold timer to zero for neighbor 192.0.251.165.

Example: **change neighbor 192.0.251.165**

```
AS [165]?
Init timer [12]?
Connect timer [60]?
Hold timer [12]? 0
TCP segment size [1024]?
```

change originate-policy ***index# (exclusive/ inclusive) network prefix network mask address match tag***

Use the **change originate-policy** command to alter an existing originate policy definition

This example alters the BGP speaker's originate policy. Rather than excluding networks with prefix 194.10.16.0 from the IGP routing table, the policy now includes all routes.

Example: **change originate-policy**

```
Enter index of originate-policy to be modified [1]?
Policy Type (Inclusive/Exclusive) [Exclusive]? inclusive
Network Prefix [194.10.16.0]? 0.0.0.0
Network Mask [255.255.240.0]? 0.0.0.0
Address Match (Exact/Range) [Range]?
Tag [0]?
```

change receive-policy ***index# (exclusive/inclusive) network prefix network mask address match originating as# adjacent as# igp-metric (inclusive only)***

Use the **change receive-policy** command to alter an existing receive policy definition.

This example adds a restriction to the BGP speaker's receive-policy. Rather than import route information from every BGP peer into its IGP routing table, it now prevents routes from AS 165 from being imported.

Example: **change receive-policy**

```
Enter index of receive-policy to be modified [1]?
Policy Type (Inclusive/Exclusive) [Inclusive]? exclusive
Network Prefix [0.0.0.0]?
Network Mask [0.0.0.0]?
Address Match (Exact/Range) [Range]?
Originating AS# [0]?
Adjacent AS# [0]? 165
```

change send-policy ***index# (exclusive/ inclusive) network prefix network mask address match tag adjacent as#***

Use the **change send-policy** command to alter an existing send policy to one that is more inclusive, or more exclusive.

This example adds a restriction to the BGP speaker's send policy. The restriction ensures that all routes in the address range 194.10.16.0 to 194.10.31.255 are excluded when advertising to autonomous system 165.

Example: **change send-policy**

```
Enter index of send-policy to be modified [1]?
Policy Type (Inclusive/Exclusive) [Inclusive]? exclusive
Network Prefix [0.0.0.0]? 194.10.16.0
Network Mask [0.0.0.0]? 255.255.240.0
Address Match (Exact/Range) [Range]?
Tag [0]?
Adjacent AS# [0]? 165
```

Clear

Erases the complete BGP configuration.

Syntax: clear

Delete

Use the **delete** command to delete an IP configuration item previously installed by the **add** command.

Syntax: delete

aggregate . . .

neighbor . . .

no-receive . . .

originate-policy . . .

receive-policy . . .

send-policy. . .

delete aggregate *index#*

You must specify the index number of the aggregate you want to delete. The index number is equivalent to the AS number.

Example: **delete aggregate 1**

delete neighbor *neighbor IP address*

Use this command to delete a BGP neighbor. You must specify the neighbor's network address.

Example: **delete neighbor 192.0.251.165**

delete no-receive *as*

Use this command to delete the no-receive policy set up for a particular AS. You must specify the AS number.

Example: **delete no-receive 168**

delete originate-policy *index#*

Use this command to delete a specific originate policy. You must specify the index number associated with the policy.

```
Example: delete originate-policy 2
```

delete receive-policy *index#*

Use this command to delete a specific receive policy. You must specify the index number associated with the policy.

```
Example: delete receive-policy
           Enter index of receive-policy to be deleted [1]?
```

delete send-policy *index#*

Use this command to delete a specific send policy. You must specify the index number associated with the policy.

```
Example: delete send-policy 4
```

Disable

Disable a previously enabled BGP neighbor or speaker. Note that neighbors are implicitly enabled whenever added with the **add** command.

Syntax: disable
BGP speaker
neighbor . . .

disable bgp speaker

```
Example: disable bgp speaker
```

disable neighbor *neighbor IP address*

```
Example: disable neighbor 192.0.190.178
```

Enable

Activate the BGP features, capabilities, and information added to your BGP configuration.

Syntax: enable

BGP speaker

neighbor . . .

enable bgp speaker ***as#*** ***tcp segment size***

Use the **enable bgp speaker** command to enable the BGP protocol.

Example: **enable bgp speaker**

```
AS [0]? 165
TCP segment size [1024]?
```

enable neighbor ***neighbor IP address***

Use this command to enable a BGP neighbor.

Example: **enable neighbor 192.0.190.178**

Exit

Leave the BGP configuration module and return to the Config> prompt.

Syntax: exit

Example: **exit**

List

Display various pieces of the IP configuration data, depending on the particular subcommand invoked.

Syntax: list

- aggregate
- all
- BGP speaker
- neighbor
- no-receive
- originate-policy
- receive-policy
- send-policy

list aggregate

Use the **list aggregate** command to all aggregated routes defined with the **add aggregate** command.

Example: **list aggregate**

```
Aggregation:
Index    Prefix              Mask
1         194.10.16.0       255.255.240.0
```

list all

Use the **list all** command to list the BGP neighbors, policies, aggregated routes, and no-receive-as records in the current BGP configuration.

Example: **list all**

```
                BGP Protocol:          Enabled
               AS:                    167
               TCP-Segment Size:      1024
Neighbors and their AS:
                                    Init    Conn    Hold    TCPSEG
Address            State     AS     Timer   Timer   Timer   Size
128.185.250.168    ENABLD    168    12      60      12      1024
192.0.251.165      ENABLD    165    12      60      12      1024

Receive-Policies:
Index  Type  Prefix       Mask      Match  OrgAS  AdjAS IGPmetric
1      INCL  0.0.0.0      0.0.0.0   Range  0      0      0

Send-Policies:
Index  Type  Prefix       Mask      Match  Tag    AdjAS
1      INCL  0.0.0.0      0.0.0.0   Range  0      0

Originate-Policies:
Index  Type  Prefix       Mask             Match  Tag
1      EXCL  194.10.16.0  255.255.240.0    Range  0

Aggregation:

Index  Prefix             Mask
1      194.10.16.0        255.255.240.0
No no-receive-AS records in configuration.
```

list bgp speaker

Use the **list bgp speaker** command to derive information on the BGP speaker. The information provided is shown below.

Example: **list BGP speaker**

```
            BGP Protocol:          Enabled
            AS:                    165
            TCP-Segment Size:      1024
```

list neighbor

Use the **list neighbor** command to derive information on BGP neighbors.

Example: **list neighbor**

```
Neighbors and their AS:
                                    Init    Conn    Hold    TCPSEG
Address            State     AS     Timer   Timer   Timer   Size
128.185.252.168    ENABLD    168    12      60      12      1024
192.0.190.178      DISBLD    178    12      60      12      1024
192.0.251.167      ENABLD    167    12      60      12      1024
```

list no-receive

Use the **list no-receive** command to derive information on *no-receive-AS* definitions that were added to the BGP configuration.

Example: **list no-receive**

```
AS-PATH with following ASs will be discarded:
AS  178
AS  165
```

list originate-policy *all index prefix*

Use the **list originate-policy** command to derive information on the originate policies that were added to the BGP configuration.

Example: **list originate-policy**

```
Originate-Policies:
Index  Type  Prefix          Mask            Match  Tag
1      EXCL  194.10.16.0     255.255.240.0   Range  0
2      INCL  0.0.0.0         0.0.0.0         Range  0
```

list receive-policy adj-as-number *all* or *index* or *prefix*

Use the **list receive-policy** command to derive information on the receive policies that were added to the BGP configuration. You can display all receive policies defined for an AS, or display policies by index or prefix number.

Example: **list receive-policy**

```
Receive-Policies:
Index  Type  Prefix      Mask        Match  OrgAS  AdjAS  IGPmetric
1      EXCL  0.0.0.0     0.0.0.0     Range  178    165
2      INCL  0.0.0.0     0.0.0.0     Range  0      0      0
```

list send-policy adj-as-number *all* or *index* or *prefix*

Use the **list send-policy** command to display information on send policies defined for specified ASs. You can display all send policies defined for an AS, or display policies by index or prefix number.

Example: **list send-policy**

```
Send-Policies:
Index  Type  Prefix      Mask           Match Tag    AdjAS
1      EXCL  194.10.16.0 255.255.240.0  Range   0    165
2      INCL   0.0.0.0          0.0.0.0  Range   0    0
```

24

Monitoring BGP4

This chapter describes how to monitor the Border Gateway Protocol (BGP) using the BGP monitoring commands.

BGP Commands

This section explains all BGP monitoring commands. These commands allow you to modify the BGP protocol behavior to meet your specific requirements. Enter BGP monitoring commands at the `BGP>` prompt.

Table 23–1 summarizes the BGP monitoring commands.

Table 24–1 BGP Command Summary

Command	Function
? (Help)	Lists the monitoring commands or lists the actions associated with specific commands.
Destinations	Displays all entries in the BGP routing table.
Neighbors	Displays currently active neighbors.
Paths	Displays all available paths in the database.
Sizes	Displays the number of entries in various databases.
Exit	Exits the process.

? (Help)

List the commands that are available from the current prompt level. You can also enter **?** after a specific command name to list its options.

Syntax: ?

Example: **?**

```
DESTINATIONS
NEIGHBORS
PATHS
SIZES
EXIT
```

Destinations

Dump all BGP routing table entries, or to display information on routes advertised to, or received from, specified BGP neighbor addresses (destinations).

Syntax: destinations *net address/net address net mask*

advertised-to *network address*

received-from *network address*

Example: **destinations**

```
Network         Mask       NextHop         MED AAG AGRAS ORG AS-Path
128.185.0.0     FFFF0000   192.0.251.165   0   No  0     IGP
142.4.0.0       FFFF0000   192.0.190.178   0   No  0     IGP seq[178]
143.116.0.0     FFFF0000   128.185.252.168 0   No  0     IGP seq[168]
192.0.190.0     FFFFFF00   192.0.251.165   0   No  0     IGP
192.0.251.0     FFFFFF00   192.0.251.165   0   No  0     IGP
194.10.16.0     FFFFF000   192.0.251.167   0   No  167   IGP seq[167]
```

MED — Specifies a multi-exit discriminator value, used to discriminate among multiple entry/exit points to the same AS.

AAG — Indicates whether the route is an aggregate or not. Values are YES or NO.

AGRAS — The number of the AS that aggregated the route.

destinations *net address*

Displays detailed information on the specified route or destination network. The command shows how a specific route was learned, the best path to a specific destination, the metric associated with the route, and other information.

Example: **destinations 142.4.0.0**

```
Network         Mask     NextHop         MED AAG AGRAS ORG AS-Path
142.4.0.0       FFFF0000 192.0.251.165   0   No  0     IGP
seq[165-178]Dest:142.4.0.0, Mask:FFFF0000, Age:180, Upd#:13,
LastSent:0001:53:32 Eligible paths: 2

PathID: 8 - (Best Path)
        ASpath: seq[165-178]
        Origin: IGP, Pref: 507, LocalPref: 0
        Metric: 0, Weight: 0, MED: 0
        NextHop: 192.0.251.165  , Neighbor: 192.0.251.165
        AtomicAggr: No

PathID: 21
        ASpath: seq[168-165-178]
        Origin: IGP, Pref: 505, LocalPref: 0
        Metric: 0, Weight: 0, MED: 0
        NextHop: 128.185.250.168, Neighbor: 128.185.250.168
        AtomicAggr: No
```

ASpath Enumeration of ASs along the path.

- **seq:** Sequence of ASs in order in the path
- **set:** Set of ASs in the path.

Origin Indicates the originator of the destination. This is either EGP, IGP, or Incomplete (originated by some other means not known).

LocalPref Indicates the originating router's degree of preference for the destination.

Metric Specifies the path metric with which the route is imported.

Weight Specifies the path weight.

MED Specifies a multi-exit discriminator value, used to discriminate among multiple entry/exit points to the same AS.

NextHop	Indicates the address of the router to use as the forwarding address for destinations reachable via the given path.
AtomicAggr	Indicates whether the router advertising the path has included the path in an atomic-aggregate.

destinations *net address net mask*

Displays detailed information on the specified route or destination network. The command shows how a specific route was learned, the best path to a specific destination, the metric associated with the route, and other information.

This command is useful in cases where multiple network addresses have the same prefix and different masks. In such cases, specifying the network mask narrows the scope of the information presented.

Example: **destinations 194.10.16.0 255.255.240.0**

```
Dest:194.10.16.0, Mask:FFFFF000, Age:0, Upd#:3, LastSent:0002:00:00

Eligible paths: 1
PathID: 0 - (Best Path)
        ASpath:
        Origin: IGP, Pref: 0, LocalPref: 0
        Metric: 0, Weight: 0, MED: 0
        NextHop: 194.10.16.167  , Neighbor: 194.10.16.167
        AtomicAggr: No, Aggregator AS167/194.10.16.167
```

destinations advertised-to *net address*

Lists all routes advertised to the specified BGP neighbor.

Example: **destinations advertised-to**

```
BGP neighbor address [0.0.0.0]? 192.0.251.165

        Destinations advertised to BGP neighbor 192.0.251.165

Network       Mask     NextHop         MED AAG AGRAS ORG AS-Path
194.10.16.0   FFFFF000 194.10.16.167   0   No  167   IGP
192.0.190.0   FFFFFF00 192.0.251.165   0   No  0     IGP seq [165]
142.4.0.0     FFFF0000 192.0.251.165   0   No  0     IGPseq [165-178]
143.116.0.0   FFFF0000 128.185.250.168 0   No  0     IGP seq [168]
```

destinations received-from *net address*

Lists all routes received from the specified BGP neighbor.

Example: **destinations received-from**

```
BGP neighbor address [0.0.0.0]? 128.185.250.167

        Destinations obtained from BGP neighbor 128.185.250.167

Network     Mask     NextHop         MED AAG AGRAS ORG AS-Path
194.10.16.0 FFFFF000 128.185.250.167 0   No  167   IGP seq[167]
192.0.190.0 FFFFFF00 128.185.250.167 0   No  0     IGP seq[167-165]
142.4.0.0   FFFF0000 128.185.250.167 0   No  0     IGP seq[167-165-178]
```

Exit

Leave the BGP configuration module and return to the Config> prompt.

Syntax: exit

Example: **exit**

Neighbors

Display information on all active BGP neighbors.

Syntax: neighbors *internet address*

Example: **neighbors**

```
IP-Address       State        DAY-HH:MM:SS  BGP-ID           AS     Upd#

128.185.252.168  Established  000-00:48:52  128.185.142.168  168     16
192.0.190.178    Established  000-02:01:49  142.4.140.178    178     16
192.0.251.167    Established  000-02:01:45  194.10.16.167    167     16
```

IP-Address	Specifies the IP address of the BGP neighbor.	
State	Specifies the state of the connection. Possible states are:	
	Connect	Waiting for the TCP connection to the neighbor to be completed.
	Active	In the event of TCP connection failure, the state is changed to Active, and the attempt to acquire the neighbor continues.
	OpenSent	In this state OPEN was sent, and BGP waits for an OPEN message from the neighbor.
	OpenConfirm	In this state a KEEPALIVE was sent in response to neighbor's OPEN, and waits for a KEEPALIVE/NOTIFICATION from the neighbor.
	Established	A BGP connection was successfully established, and can now start to exchange UPDATE messages.
BGP-ID	Specifies the neighbor's BGP Identification number.	
AS	Specifies the neighbor's AS number.	
Upd#	Specifies the sequence number of the last UPDATE message sent to the neighbor.	

Neighbor *internet-address*

Use the **neighbor** command to display detailed data on a particular BGP neighbor.

Example: **neighbor 192.0.251.167**

```
Active Conn: Sprt:1026  Dprt:179    State: Established KeepAlive/Hold Time:
4/12
Passve Conn: None
TCP connection errors: 0          TCP state transitions: 0

BGP Messages:     Sent       Received                    Sent       Received
Open:             1          1          Update:          11         11
Notification:     0          0          KeepAlive:       1828       1830
Total Messages:   1840       1842

Msg Header Errs:  Sent       Received                    Sent       Received
Conn sync err:    0          0          Bad msg length:  0          0
Bad msg type:     0          0

Open Msg Errs:    Sent       Received                    Sent       Received
Unsupp versions:  0          0          Unsupp auth code: 0          0
Bad peer AS ident:0          0          Auth failure:    0          0
Bad BGP ident:    0          0          Bad hold time:   0          0

Update Msg Errs:  Sent       Received                    Sent       Received
Bad attr list:    0          0          AS routing loop:  0          0
Bad wlkn attr:    0          0          Bad NEXT_HOP atr: 0          0
Mssng wlkn attr:  0          0          Optional atr err: 0          0
Attr flags err:   0          0          Bad netwrk field: 0          0
Attr length err:  0          0          Bad AS_PATH attr: 0          0
Bad ORIGIN attr:  0          0

Total Errors:     Sent       Received                    Sent       Received
Msg Header Errs:  0          0          Hold Timer Exprd: 0          0
Open Msg Errs:    0          0          FSM Errs:        0          0
Update Msg Errs:  0          0          Cease:           0          0
```

Paths

Use the BGP **paths** command to display the paths stored in the path description data base.

Syntax: paths

Example: **paths**

```
PathId   NextHop    MED  AAG   AGRAS  RefCnt   ORG   AS-Path
0       10.2.0.3    0   No     0      2      IGP
4       192.2.0.2   0   No     0      2      IGP    seq[2]
5       192.2.0.2   0   No     2      1      IGP    seq[2]
6       192.2.0.2   0   No     0      1      IGP    seq[2-1]
7       10.2.0.168  0   No     0      4      IGP
8       192.3.0.1   0   No     0      2      IGP    seq[1]
9       192.2.0.2   0   No     2      1      IGP    seq[2]
10      10.2.0.3    0   No     0      1      IGP
```

PathId	Path identifier
NextHop	The address of the router to use as the forwarding address for the destinations that can be reached via the given path.
MED	The Multi Exit Descriminator used to discriminate among multiple entry/exit points to the same AS.
AAG	Indicates whether the path was atomic-aggregated, that is, the router that is advertising the given path has selected a less specific route over the more specific one when presented with overlapping routes.
AGRAS	Indicates the AS number of the BGP speaker that aggregated the routes.
RefCnt	Indicates the number of path entities referring to the descriptor.
ORG	Specifies the originator of the advertised destinations in the given path: either EGP, IGP, or Incomplete (originated by some other means not known).
AS-Path	Enumeration of ASs along the path. • seq: Sequence of ASs in order in the path. • set: Set of ASs in the path.

Sizes

Use the BGP **sizes** command to display the number of entries stored in the various data bases.

Syntax: sizes

Example: **sizes**

```
# Paths:                                11
# Path descriptors:                      7
Update sequence#:                       22
# Routing tbl entries (allocated):       6
# Current tbl entries (not imported):    0
# Current tbl entries (imported to IGP): 3
```

Paths	Total number of eligible paths for all the routes in the BGP routing table.
Path descriptors	Total number of path descriptors in the database used to hold common path information.

Update sequence#	Indicates the current update sequence number.
Routing tbl entries (allocated)	Indicates the number of entries in BGP routing table.
Current tbl entries (not imported)	Indicates the number of BGP routes not imported into IGP.
Current tbl entries (imported to IGP)	Indicates the number of BGP routes imported into IGP.

A

SNMP Objects

This appendix summarizes the SNMP object supported by the router software. The base is MIB II (Management Information Base II) which is specified in RFC 1213. The groups from MIB II which have been implemented are the following:

- System Group
- Interfaces Group
  - Object ifInNUcastPkts is always 0, those packets are counted in object ifInUcastPkts
  - Object ifOutNUcastPkts is always 0, those packets are counted in object ifOutUcastPkts
- Address Translation Group – no objects are settable
- IP Group
  - Object ipDefaultTTL is settable
  - Routing table objects are not settable
- ICMP Group
- UDP Group
- EGP Group
- Transmission Group
- SNMP Group

The TCP Group B is not implemented. In addition, a proprietary MIB is implemented, with the following groups:

```
proteon          = { iso(1) org(3) dod(6) internet(1) private(4) enterprise(1) 1 }

admin            = { proteon(1) 1 }
objid            = { admin(1) 1 }
status           = { admin(1) 2 }
els              = { admin(1) 3 }

xface            = { proteon(1) 2 }
generic          = { xface(2) 1 }
tokenr           = { xface(2) 2 }
ieth             = { xface(2) 4 }
peth             = { xface(2) 5 }
comsl            = { xface(2) 6 }
gwsl             = { xface(2) 7 }
x25              = { xface(2) 8 }
fddi             = { xface(2) 12}
ceth             = { xface(2) 13}
csl              = { xface(2) 14}
seth             = { xface(2) 15}
```

B

Packet Sizes

This appendix discusses the sizes of packets for the various networks and protocols that the p45xx supports. This appendix contains the following sections:

- General Issues
- Network-specific Size Limits
- Protocol-specific Size Limits
- Changing Maximum Packet Sizes

General Issues

For the purposes of this discussion, the packets that the routers handle consist of user data and header information.

The amount of user data within a packet is limited by the amount of header information on the packet. The amount of header information minimally depends on:

- The network-types over which the packet must travel.
- The protocols in use on these networks.

The following factors affect the size of the packet contents:

- Length of the Data-Link header information that the current network type and interface require the packet to have.

- Length of the trailer information (if any) that the current network type and interface require the packet to have.

On any given network, the sum of the maximum data size together with header and trailer sizes will equal the network's maximum packet size. When routing between networks of different maximum packet size, fragmentation of the packet may result.

Network-Specific Size Limits

Given the information in the previous section, the maximum amount of **Network** layer data supported by each **Data Link** layer (network interface) can be determined.

Table B–1 lists the packet sizes with defaults.

Table B–1 Network-Specific Packet Size Limits

Network Type (Data Link)	Network Layer max packet size (bytes)	Length of Network Header	Information Trailer
FDDI	4479	20	0
Ethernet	1500	18	4
Serial Line	2046*	2	0

* Default

The maximum packet size is the maximum amount of data the protocol forwarder can pass to the device.

Note: These numbers correspond to the MTUs in 4.2 BSD UNIX.

For an IP packet, this includes the IP header, the UDP or TCP header, and all data. For a DECnet packet, this includes the Routing header (long format data packet), the NSP header, and any data. For an XNS packet, this includes the IDP header, the SPP or PXP header, and any data.

The packet size in use is displayed when the router's GWCON **memory** command is used. The "Pkt" size is the Network layer packet size. The Hdr (header) and Tlr (trailer) sizes depend on the networks and their network interfaces.

Protocol-Specific Size Limits

This section explains the protocol-specific size limits.

IP Packet Lengths

The IP protocol specifications do not require a host IP implementation to accept IP packets of more than 576 octets. Router IP implementations *must* accommodate IP packets of any length up to the limits imposed by the network-specific packets in use.

Furthermore, router IP performs transparent fragmentation and reassembly of packets that would otherwise exceed network-specific length restrictions, as required by the IP specification.

Packet size mismatches do not cause connectivity problems. However, fragment reassembly does pose a performance penalty, so fragmentation should be avoided whenever possible.

DECnet Packet Lengths

In the default configuration and according to the specifications, all DECnet routers must forward packets that are 576 bytes long. It is allowable to configure hosts to use larger packets, up to the limits imposed by Ethernet. Not all Digital routers support packets larger than 576 bytes, especially those based on PDP–11s, such as the DECSA (Digital Ethernet Router Server). The DECnet–VAX buffer sizes should not be increased from 576 bytes if there are any PDP–11 routers in the DECnet network.

The maximum possible packet size in DECnet is 1498 bytes because DECnet adds two bytes of length to the Ethernet Data Link. This provides information on the network length of all packets, even when they are shorter than the minimum Ethernet packet size of 60 bytes (excluding 4 byte CRC).

The DECnet forwarder keeps internal maximums on the packet sizes for each network (circuit) that are two bigger than the minimum of 1498 packet size for that network interface.

If a packet is too large to be forwarded in DECnet, it is dropped. An event 4.3 (oversized packet loss) is logged, and the executor Oversized Packet Loss counter is incremented.

Changing Maximum Packet Sizes

Normally, the router automatically sets the maximum Network layer packet size to the size of the largest possible packet on all the connected networks. It then adds any headers and trailers required by the networks to determine the internal buffer size, which is larger than the Network layer size.

Some networks allow you to configure maximum packet sizes. Configuring maximum packet sizes affects the size of buffers used on the router and this in turn affects the number of buffers available for a given memory size. Routers automatically determine what size buffer it is going to need. You can change the maximum Network layer packet size that the router handles by using the **set packet-size** command. Do not use this command unless specifically directed to by Customer Service.

C

Comparison of Protocols

This appendix compares some of the well-known protocols that your router supports. It is provided as a memory aid and is not meant as a reference.

Protocol Comparison Table

Table C–1 compares the protocols.

Table C–1 Comparison Protocols

ISO OSI Model	TCP/IP	Xerox	DECnet	Other	IPX	OSI
7 Application	Application		CTERM			
6 Presentation	(Telnet, FTP,		DAP			
5 Session	TFTP, SGMP)		Session			
4 Transport	Transport (TCP, UDP)	SPP PXP	NSP		PXP SPX	TP
3 Network	Internet (IP, RIP, EGP, ICMP)	XNS	Routing	ARP	IPX RIP SAP	CLNP ES–IS IS–IS
2 Data Link	Local Net	Ethernet		PPP		
1 Physical				HDLC		

Key to Protocols

Table C–2 is a key to the protocols.

Table C–2 Protocol Key

Protocol	Description
AP	Authentication Protocol. Used with the Simple Gateway Monitoring Protocol (SGMP) to validate requests for router statistics.
CLNP	Connection-Less Network Protocol.
EGP	External Gateway Protocol. An IP routing protocol.
ES-IS	End System-Immediate System protocol (ISO 9542). Used between an end system and an immediate system to provide configuration information and route redirection information.
FTP, TFTP	File Transfer Protocol; Trivial File Transfer Protocol.
ICMP	Internet Control Message Protocol. Used to send network level error and control messages between routers and hosts.
IP	Internet protocol. IP is a widely-used standard transport protocol. IP is the routers' basic protocol. IP leaves some error-checking to higher-level (end-to-end) protocols.
IPX	Internet Exchange Packet Protocol.
IS-IS	Immediate System-Immediate System protocol (ISO 10589). Used by immediate systems to communicate with each other.
RIP	Routing Information Protocol (Routing protocols are used to determine network topology and data paths). RIP is the common IP routing protocol. XNS also has a routing protocol called RIP.
SGMP	Simple Gateway Monitoring Protocol. Used to obtain statistics in machine-readable form from the routers.
SNMP	Simple Network Monitoring Protocol. Used to obtain statistics in machine-readable form from the routers.
TCP	Transport Control Protocol. An end-to-end (host-to-host) protocol that is often used with IP. Useful for sending streams of data. Uses checksums, acknowledgements, and timeouts to ensure the correct delivery and sequence of data.

D

Digital MIB Support

This appendix describes the MIBs or portions of MIBs contained in the Digital-Router-SNMP-Agent.

Digital supports the following standard MIBs shown in Table D–1.

Table D–1 Standard MIBs

MIB	Exceptions	RFC Number
Bridge	• dot1dStaticTable • dot1dTpFdbTable	RFC 1286
DLSw	See the following sections.	None
Ethernet	None	RFC 1623
FDDI	None	RFC 1285
Frame Relay	None	RFC 1315
MIB2	• TCP group	RFC 1213
OSPF	None	RFC 1253
PPP	See the following sections.	RFC 1471

(continued on next page)

Table D–1 (Cont.) Standard MIBs

MIB	Exceptions	RFC Number
RS–232 Serial Line	None	RFC 1317
Token Ring	None	RFC 1231

DLSw MIB

The full text of the DLS extensions is contained in IBM®'s enterprise tree in the 6611 MIB. Refer to *6611 Network Processor Network Management Reference*, IBM manual number GC30–3567–01 for more information.

Table D–2 lists the subgroups within the DLS MIB that Digital supports. Table D–3 lists the DLSw extensions within a subgroup that are supported.

Table D–2 DLSw MIB Tables Supported

DLSw Group Attributes	Supported	Not Supported
Virtual Ring Segment Number	✓	
Filter Types		✓
Participating Router Table	✓	
SNA Local Filter Frame Table		✓
SNA Remote Filter Frame Table		✓
NETBIOS Local Name Filter Table		✓
NETBIOS Remote Name Filter Table		✓
SNA Default Destination Table		✓
NETBIOS Default Destination Table		✓
SNA Station Table		✓
Circuit Table	✓	

Table D–3 DLSw MIB Objects Supported

DLSw Table Name	DLSw Object Name	Supported	Not Supported
Participating Router Table			
	IBM DLS Router Address	✓	
	IBM DLS Router Status	✓	
	IBM DLS Router Defined By	✓	
	IBM DLS Router In Frames		✓
	IBM DLS Router Out Frames		✓
Circuit Table			
	IBM DLS Cir If Index	✓	
	IBM DLS Cir Src Address	✓	
	IBM DLS Cir Src Sap	✓	
	IBM DLS Cir Dest Address	✓	
	IBM DLS Cir Dest Sap	✓	
	IBM DLS Cir Partner Router Address	✓	
	IBM DLS Cir Local Link State	✓	
	IBM DLS Cir Local Link Sub State	✓	
	IBM DLS Cir Local Link Routing	✓	
	IBM DLS Cir Local Link Test Cmds Sent		✓
	IBM DLS Cir Local Link Test Cmds Fail		✓
	IBM DLS Cir Local Link Test Cmds Rcv		✓
	IBM DLS Cir Local Link Data Pkt Sent	✓	
	IBM DLS Cir Local Link Data Pkt Resent		✓
	IBM DLS Cir Local Link Max Cont Resent		✓
	IBM DLS Cir Local Link Data Pkt Rcv	✓	
	IBM DLS Cir Local Link Invalid Pkt Rcv		✓
	IBM DLS Cir Local Link Adp Rcv Err		✓

(continued on next page)

Table D–3 (Cont.) DLSw MIB Objects Supported

DLSw Table Name	DLSw Object Name	Supported	Not Supported
Circuit Table (cont.)			
	IBM DLS Cir Local Link Adp Send Err		✓
	IBM DLS Cir Local Link Rcv Inactive Timeouts		✓
	IBM DLS Cir Local Link Cmd Polls Sent		✓
	IBM DLS Cir Local Link Cmd Repolls Sent		✓
	IBM DLS Cir Local Link Cmd Cont Repolls		✓
	IBM DLS Cir Local Address		✓

PPP MIB

Table D–4 lists the group attributes of the PPP MIB supported by Digital. This section describes the level of support provided for extensions of the Link Control Protocol of the Point-to-Point Protocol. These extensions are defined by the Internet standard RFC 1471.

Table D–4 PPP MIB Groups Supported

PPP Group Attributes	Supported	Not Supported
PPP Link Group (Link Status Group only)	✓	
PPP Link Quality Reporting Group		✓
PPP Link Quality Reporting Extensions Group		✓
PPP IP Group		✓
PPP Bridge Group		✓
PPP Security Group		✓

Table D–5 lists the attributes supported within the PPP Link Group specified in RFC 1471.

Table D–5 PPP Link Group Attributes Supported

PPP Link Group Attributes	Supported	Not Supported
PPP Link Status Physical Index[1]	✓	
PPP Link Status Bad Addresses	✓	
PPP Link Status Bad Controls	✓	
PPP Link Status Packet Too Longs	✓	
PPP Link Status BadFCSs	✓	
PPP Link Status Local MRU	✓	
PPP Link Status Remote MRU	✓	
PPP Link Status Local To Peer ACC Map	✓	
PPP Link Status Peer To Local ACC Map	✓	
PPP Link Status Local To Remote Protocol Compression	✓	
PPP Link Status Remote to Local Protocol Compression	✓	
PPP Link Status Local To Remote ACC Compression	✓	
PPP Link Status Remote To Local ACC Compression	✓	
PPP Link Status Transmit Fcs Size	✓	
PPP Link Status Receive Fcs Size	✓	
PPP Link Config Table (Run-time SNMP Attributes)		✓

[1]Each Serial Line that supports PPP has two entries in the Interface table. One entry is associated with the physical hardware link over which PPP is running. The other entry represents the PPP layer running over that link.

For example, if the router contains four interfaces, two of which run PPP over RS–232 ports, the number of interfaces reported in the Interface Table is six. Six interface numbers are defined, four for the network interfaces and an additional two designating the PPP–layer links.

Index

Symbols

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

R

#

#

HOW TO ORDER ADDITIONAL DOCUMENTATION

DIRECT TELEPHONE ORDERS

In Continental USA
call 800–DIGITAL

In Canada
call 800–267–6215

In New Hampshire
Alaska or Hawaii
call 603–884–6660

In Puerto Rico
call 809–754–7575 x2012

ELECTRONIC ORDERS (U.S. ONLY)

Dial 800–234–1998 with any VT100 or VT200
compatible terminal and a 1200 baud modem.
If you need assistance, call 1–800–DIGITAL.

DIRECT MAIL ORDERS (U.S. AND PUERTO RICO*)

U. S. SOFTWARE SUPPLY BUSINESS
DIGITAL EQUIPMENT CORPORATION
10 Cotton Road
Nashua, New Hampshire 03063–1260

DIRECT MAIL ORDERS (Canada)

DIGITAL EQUIPMENT OF CANADA LTD.
940 Belfast Road
Ottawa, Ontario, Canada K1G 4C2
Attn: A&SG Business Manager

INTERNATIONAL

DIGITAL
EQUIPMENT CORPORATION
A&SG Business Manager
c/o Digital's local subsidiary
or approved distributor

Internal orders should be placed through the Software Services Business (SSB)
Digital Equipment Corporation, Westminster, Massachusetts 01473

*Any prepaid order from Puerto Rico must be placed
with the Local Digital Subsidiary:
809–754–7575 x2012

Distributed Routing Software
Routing Protocols User's Guide
AA–QL2DC–TE

READER'S COMMENTS

What do you think of this manual? Your comments and suggestions will help us to improve the quality and usefulness of our publications.

Please rate this manual:

	Poor				Excellent
Accuracy	1	2	3	4	5
Readability	1	2	3	4	5
Examples	1	2	3	4	5
Organization	1	2	3	4	5
Completeness	1	2	3	4	5

Did you find errors in this manual? If so, please specify the error(s) and page number(s).

_____

_____

_____

_____

General comments:

_____

_____

_____

_____

Suggestions for improvement:

_____

_____

_____

_____

Name _____ Date _____

Title _____ Department _____

Company _____ Street _____

City _____ State/Country _____ Zip Code _____

DO NOT CUT – FOLD HERE AND TAPE

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY LABEL
FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

digital™

Shared Engineering Services

550 King Street
Littleton, MA 01460–1289

DO NOT CUT – FOLD HERE