GNU Zebra User's Guide

LynxOS 4.0

DOC-0391-00

Product names mentioned in *GNU Zebra User's Guide* are trademarks of their respective manufacturers and are used here for identification purposes only.

Printed in the United States of America.

Contents

Preface

GNU Zebra provides TCP/IP routing management for networked systems, supporting RIPv1, RIPv2, OPSFv2, and BGP-4 protocols. Zebra's unique modular design uses separate processes for each protocol, providing greater reliability and flexibility. Modules are independent from each other, and can be upgraded separately. Additionally, in the case of a module failure, the rest of the routing system can continue to function normally.

For More Information

For more information on the features of LynxOS, refer to the following printed and online documentation.

- *Release Notes*

  This printed document contains late-breaking information about the current release.

- *LynxOS Installation Guide*

  This manual supports the initial installation and configuration of LynxOS and the X Windows System.

- *LynxOS User's Guide*

  This document contains information about basic system administration and kernel level specifics of LynxOS. It contains a "Quick Starting" chapter and covers a range of topics, including tuning system performance and creating kernel images for embedded applications.

- Online information

  Information about commands and utilities is provided online in text format through the **man** command. For example, a user wanting

information about the GNU compiler would use the following syntax, where **gcc** is the argument for information about the GNU compiler:

```
man gcc
```

More recent versions of the documentation listed here may also be found online.

Typographical Conventions

The typefaces used in this manual, summarized below, emphasize important concepts. All references to file names and commands are case sensitive and should be typed accurately.

Kind of Text	Examples
Body text; *italicized* for emphasis, new terms, and book titles	Refer to the *LynxOS User's Guide*.

Environment variables, file names, functions, methods, options, parameter names, path names, commands, and computer data
Commands that need to be highlighted within body text, or commands that must be typed as is by the user are **bolded**.
	`ls`
	`-l`
	`myprog.c`
	`/dev/null`
	`login: myname`
	`# cd /usr/home`

Text that represents a variable, such as a file name or a value that must be entered by the user	`cat filename` `mv file1 file2`

Blocks of text that appear on the display screen after entering instructions or commands	`Loading file /tftpboot/shell.kdi into 0x4000`
	`.....`
	`File loaded. Size is 1314816`
	`Copyright 2000 LynuxWorks, Inc. All rights reserved.`
	
	`LynxOS (ppc) created Mon Jul 17 17:50:22 GMT 2000`
	`user name:`

Keyboard options, button names, and menu sequences	**Enter, Ctrl-C**

Special Notes

The following notations highlight any key points and cautionary notes that may appear in this manual.

NOTE: These callouts note important or useful points in the text.

CAUTION! Used for situations that present minor hazards that may interfere with or threaten equipment/performance.

Technical Support

LynuxWorks Technical Support is available Monday through Friday (holidays excluded) between 8:00 AM and 5:00 PM Pacific Time (U.S. Headquarters) or between 9:00 AM and 6:00 PM Central European Time (Europe).

The LynuxWorks World Wide Web home page provides additional information about our products, Frequently Asked Questions (FAQs), and LynuxWorks news groups.

LynuxWorks U.S. Headquarters

Internet: `support@lnxw.com`
Phone: (408) 979-3940
Fax: (408) 979-3945

LynuxWorks Europe

Internet: `tech_europe@lnxw.com`
Phone: (+33) 1 30 85 06 00
Fax: (+33) 1 30 85 06 06

World Wide Web

`http://www.lynuxworks.com`

CHAPTER 1 Zebra Overview

Zebra is a TCP/IP-based routing manager with support for several protocols, including: RIPv1, RIPv2, OSPFv2, OSPFv3, and BGP-4. Zebra also supports special BGP 4+ behaviors: Route Reflector and Route Server. With an SNMP daemon that supports SMUX protocol, Zebra can also provide the MIB for the routing protocol.

Zebra's advanced software architecture provides a high quality, multiserver routing system. With an interactive user interface for each protocol, Zebra uses common commands for each protocol. This design allows users to easily add new protocol daemons to Zebra. The Zebra library is a programmatic interface for C-language applications. Zebra is an open source, royalty-free software distribution licensed under the GNU General Public License (GPL).

How Zebra Works

When sending and receiving information to the Internet, TCP/IP packets pass through many routers using TCP/IP routing protocols.

A system with Zebra installed acts as a router, delivering TCP/IP packets. With Zebra, the system exchanges routing information with other routers using specific routing protocols. Zebra uses this information to update the kernel routing table to ensure that the correct data is sent to the correct destination.

In addition to the routing protocol support, Zebra can configure the address and flags of an interface, static routes and so on. For smaller networks, a sub network, or an xDSL connection, the Zebra routing software can be used to implement basic routing. The only requirements are the configuration of the interfaces and addition of commands for static and/or default routes. For larger, more complicated networks, Zebra offers dynamic routing protocol support for autonomous protocols such as RIP (Routing Information Protocol), OSPF (Open Shortest Path First) or BGP (Border Gateway Protocol).

UNIX-based router configurations are traditionally handled by the `ifconfig` and `route` commands. These commands require `root` privileges. Zebra, however, is configured through two user modes: Normal mode and Enable mode. In Normal mode, users can only view system status (similar to `netstat`), while an Enable mode user can change the Zebra system configuration. These independent UNIX accounts assist the router administrator by allowing users to access router information.

Zebra System Architecture

Traditional routing software uses a single process which provides all of the routing protocol functions. However, Zebra is comprised of several independent daemons that work together to build a routing table. Each of these daemons run under separate processes. In addition, there can be several protocol-specific routing daemons added to `zebra` (the routing manager).

The `ripd` daemon handles the RIP protocol, while the `ospfd` daemon supports OSPF v2. The `bgpd` daemon supports the BGP-4 protocol. The zebra kernel table manager changes the kernel routing table and redistributes routes between different routing protocols. Adding new routing protocol daemons to the routing system is easily accomplished without affecting any other software. Users only need to run the protocol daemon associated with the routing protocols in use. Thus, a specific daemon can send routing reports to a central routing console.

Zebra also allows multiple instances of the same protocol daemon to run on the same system.

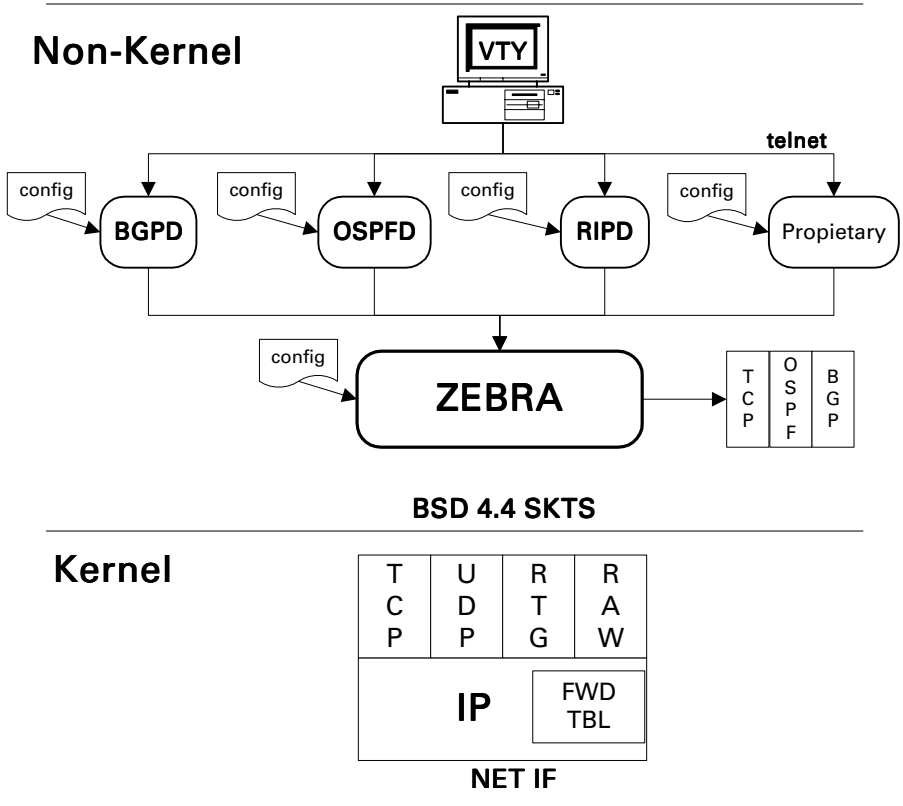


Figure 1-1: Zebra Architecture

vtysh- Integrated User Interface Shell

Each daemon uses its own configuration file and terminal interfaces. Configuring a static route must be done in the `zebra` configuration file. Configuring a BGP network must be done in the `bgpd` configuration file. Editing these files separately is inefficient. Zebra provides an integrated user interface shell called `vtysh`. `vtysh` connects to each daemon with a UNIX domain socket and acts as a proxy for user input.

Autonomous Routing Protocols

An Autonomous Routing Protocol is an Internet specification that allows routers to discover routes, maintain route tables, and communicate route changes. RIP v1, v2, OPSF2, and BGP are all autonomous routing protocols.

Manually Inserted Routes

Zebra allows for the manual insertion of IP routes.

IPv4 Address Notation Conventions

IP addresses are represented in a “dotted decimal notation” of four integers that range from 0 to 255. Each of these integers (also called octets) represents 8 bits of a 32-bit IPv4 address, with the first integer establishing the IP address class.

Table 1-1: IP Address Classes and Ranges

Class	Range
A	1.0.0.0 through 127.255.255.255
B	128.0.0.0 through 191.255.255.255
C	192.0.0.0 through 223.255.255.255
D (Multicast)	224.0.0.0 through 239.255.255.255
E (Experimental)	240.0.0.0 through 247.255.255.255

Subnets

Some IPv4 networks are considered “classless”. These classless IP networks use a 32-bit “subnet mask” to further define the IP network. Subnet masks allow an IP address range to be subdivided, creating additional IP addresses on a network, and facilitating routing services. All IP addresses assigned today are classless.

Subnet masks are written either in decimal dotted notation, or are appended to the IP address in their bit value. This bit value constitutes the number of bits in the subnet mask.

For example, the IP address denoted as 10.0.0.0/8 indicates that the subnet mask uses 8 bits, and is noted as 255.0.0.0 in decimal dotted notation. An IP address of

10.0.0.0/24 uses a 24-bit subnet mask and is noted as 255.255.255.0 in decimal dotted notation.

In this document, IP addresses may be noted with or without the "/" notation for specifying IP addresses and subnet masks.

IPv6

The IPv6 protocol addresses technical limitations of IPv4. Most notably is the increase in IP address space, which has changed from 32 to 128 bits per address.

IPv4 32-bit addresses are represented in dotted-decimal format divided along 8-bit boundaries. IPv6 IP addresses are 128-bit address divided along 16-bit boundaries, and each 16-bit block is converted to a 4-digit hexadecimal number and separated by colons. For example:

```
200A:00A3:2C5B:0000:02FF:FF00:FE38:934A
```

NOTE: The IPv6 and IPsec protocols for LynxOS are not included with the standard LynxOS package. These components are available for purchase separately. For information on these products, please contact your LynuxWorks sales representative.

The IPv6 functionality of Zebra described in this manual applies to LynxOS systems configured with IPv6 support. Additional information on IPv6 can be found in the *LynxOS Networking Guide.*

Installing Zebra

Zebra installation instructions are provided in the *LynxOS Installation Guide*.

GNU Zebra Components

The following table describe the GNU Zebra components.

Table 1-2: GNU Zebra Components

Component	Description
`zebra`	Zebra management daemon
`zebra.conf.sample`	`zebra` sample configuration file
`ripd`	RIP daemon
`ripd.conf.sample`	`ripd` sample configuration file
`ospfd`	OSPF daemon
`ospfd.conf.sample`	`ospfd` sample configuration file
`bgpd`	BGP daemon
`bgpd.conf.sample` `bgpd.conf.sample2`	`bgpd` sample configuration files
`vtysh`	Integrated user shell for Zebra

The following table describes the GNU Zebra man pages included with this distribution.

Table 1-3: GNU Zebra man pages

Component	Description
`zebra(1)`	Zebra management daemon
`ripd(1)`	RIP daemon
`ospfd(1)`	OSPF daemon
`bgpd(1)`	BGP daemon
`vtysh(1)`	Integrated user shell for Zebra

Updating /etc/services

Users can update the `/etc/services` file with the ports used with the Zebra routing protocols. The following provides an example `/etc/services` file with the zebra port numbers:

```
#
# zebra interfaces
#
zebrasrv   2600/tcp   # zebra service
zebra      2601/tcp   # zebra vty
ripd       2602/tcp   # RIPd vty
ripngd     2603/tcp   # RIPngd vty
ospfd      2604/tcp   # OSPFd vty
bgpd       2605/tcp   # BGPd vty
ospf6d     2606/tcp   # OSPF6d vty
```

Figure 1-2: Zebra Ports configured in /etc/services

Supported RFCs

Below is the list of currently supported routing protocol RFCs:

Table 1-4: Supported RFCs

RFC	Description
RFC1058	Routing Information Protocol. C.L. Hedrick. Jun-01-1998
RFC1771	A Border Gateway Protocol 4 (BGP-4.) Y. Rekhter & T. Li. March 1995.
RFC1997	BGP Communities Attribute. R. Chandra, P. Traina & T. Li. August 1996.
RFC2283	Multiprotocol Extensions for BGP-4. T. Bates, R. Chandra, D. Katz, Y. Rekhter. February 1998.
RFC2328	OSPF v2. J. Moy. April 1998.
RFC2453	RIP v2. G. Malkin. November 1998.
RFC2796	BGP Route Reflection An alternative to full mesh IBGP T. Bates R. Chandrasekeran. June 1996.
RFC1227	SNMP MUX protocol and MIB. M.T. Rose. May-01-1991.

Table 1-4: Supported RFCs (Continued)

RFC	Description
RFC 1657	Definitions of Managed Objects for the Fourth Version of the Border Gateway Protocol (BGP-4) using SMIv2. S. Willis, J.Burruss, J. Chu, Editor. July 1994.
RFC1850	OSPF v2 Management Information Base. F. Baker, R. Coltun. November 1995.
RFC1519	CIDR: Address and Assignment Aggregation Strategy
RFC950	Internet Standard Subnetting Procedure
RFC1058	Routing Information Protocol. C.L. Hedrick. Jun-01-1998
RFC1771	A Border Gateway Protocol 4 (BGP-4.) Y. Rekhter & T. Li. March 1995.
RFC1997	BGP Communities Attribute. R. Chandra, P. Traina & T. Li. August 1996.
RFC2080	RIPng for IPv6. G. Malkin, R. Minnear. January 1997.
RFC2283	Multiprotocol Extensions for BGP-4. T. Bates, R. Chandra, D. Katz, Y. Rekhter. February 1998.
RFC2328	OSPF Version 2. J. Moy. April 1998.
RFC2453	RIP Version 2. G. Malkin. November 1998.
RFC2545	Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing. P. Marques, F. Dupont. March 1999.
RFC2740	OSPF for IPv6. R. Coltun, D. Perguson, J.Moy. December 1999.
RFC2796	BGP Route Reflection An alternative to full mesh IBGP T. Bates R. Chandrasekeran. June 1996.
RFC1227	SNMP MUX protocol and MIB. M.T. Rose. May-01-1991.
RFC 1657	Definitions of Managed Objects for the Fourth Version of the Border Gateway Protocol (BGP-4) using SMIv2. S. Willis, J.Burruss, J. Chu, Editor. July 1994.
RFC1850	OSPF Version 2 Management Information Base. F. Baker, R. Coltun. November 1995.

Zebra Resources

Zebra is still beta software and is currently distributed from the Zebra beta `ftp` site:

```
ftp://ftp.zebra.org/pub/zebra
```

Zebra's official web page is located at:

```
http://www.gnu.org/software/zebra/zebra.html
```

The Zebra beta tester web page is available from:

```
http://www.zebra.org/
```

Mailing List

The mailing list for discussions and announcements regarding Zebra is: `zebra@zebra.org`. New snapshot announcements, improvement notes, patches, and other notices are sent to the list.

To subscribe to the Zebra mailing list, send a mail to `majordomo@zebra.org` with a message body that includes only: `subscribe zebra`

To unsubscribe from the list, send a mail to `majordomo@zebra.org` with a message body that includes only: `unsubscribe zebra`

Reporting Bugs

To report a bug, send email to: `bug-zebra@gnu.org`.

When submitting a bug, note the following:

- Provide the results of `netstat -rn` and `ifconfig -a`. Information from Zebra's VTY command `show ip route` will also be helpful.
- Send the configuration file with the report. If arguments are used to the configure script please note that too.

Bug reports are important to improve the quality of Zebra. Zebra is still in the development stage, but please don't hesitate to send a bug report.

CHAPTER 2 *The Zebra Suite*

Zebra Daemons

Zebra uses a number of routing daemons in addition to the `zebra` manager daemon. The router daemons can exist on systems separate from the manager daemon, allowing for a more modular architecture. Each of these daemons listens on a particular port for incoming VTY connections. The routing daemons include:

- `ripd`
- `ospfd`
- `bgpd`

In addition to these routing daemons, Zebra includes the routing manager daemon, `zebra`.

The following sections detail commands common for all of the routing daemons.

Daemon Configurations

Configuration files allow users to write debugging options, edit VTY passwords, change routing daemon configurations, and edit logfile names. This information forms the initial command set for a routing process as it starts.

Configuration files are generally located in:

```
/usr/zebra/etc/
```

Each of these daemons uses its own configuration file. For example, Zebra's default configuration file is:

```
/usr/zebra/etc/zebra.conf
```

The daemon name plus `.conf` is the default configuration file name for the router daemon. To specify a configuration file, use the `-f` or `--config-file` options when starting the daemon.

Basic Configuration Commands

The following table shows the basic configuration commands:

Table 2-1: Basic Configuration Commands

Command	Description
`hostname` *`HOSTNAME`*	Set hostname of the router.
`password` *`PASSWORD`*	Set password for VTY interface. If there is no password, VTY refuses connections.
`enable password` *`PASSWORD`*	Set enable password.

`log stdout`
 	
`no log stdout`
	Set login output to stdout.

`log file` *`FILENAME`*	To log into a file, specify *`FILENAME`* as follows: log file `/usr/zebra/etc/bgpd.log`

`log syslog`
 	
`no log syslog`
	Set login output to syslog.

`write terminal`	Display the current configuration of the VTY interface.
`write file`	Write current configuration to the configuration file.
`configure terminal`	Change to configuration mode. This command is the first step in configuring Zebra.
`terminal length` *`0-512`*	Set terminal display length to *0-512* If length is 0, no display control is performed.
`who`	Display user on VTY.
`list`	List commands.
`service password-encryption`	Encrypt password.
`service advanced-vty`	Enable advanced mode VTY.

Table 2-1: Basic Configuration Commands (Continued)

Command	Description
`service terminal-length` *`0-512`*	Set system wide line configuration. This configuration command applies to all VTY interfaces.
`show version`	Show the current version of Zebra and the build host information.
`line vty`	Enter VTY configuration mode.

`banner motd default`
 	
`no banner motd`
	Set or disable motd banner string printing.

`exec-timeout` *`SECOND`* `exec-timeout` *`MINUTE SECOND`* `no exec-timeout`	Set VTY connection timeout value. When only one argument is specified, that value is used for timeout in seconds. Default timeout value is 10 minutes. If the timeout value is set to zero, there is no timeout. `no-exec-timeout` prevents connection timeout, and is the same as `exec-timeout 0 0`.
`access-class` *`ACCESS-LIST`*	Restrict VTY connections with an access list.

Sample Configuration File

The following is a sample configuration for the `zebra` daemon. The `!` and # are comment characters. If either of these are the first character of the line, the entire line is ignored. In the example below, the password is set to `zebra`.

```
!password:
!
# Zebra configuration file
!
hostname Router
password zebra
enable password zebra
!
log stdout
!
!
```

Common Invocation Options

These options are common to all Zebra daemons:

Table 2-2: Common Invocation Options

Option	Description
`-d` `--daemon`	Run in daemon mode.
`-f` *`FILE`* `--config_file=`*`FILE`*	Set configuration file name.
`-P` *`PORT`* `--vty_port=`*`PORT`*	Set the VTY port number.
`-v` `--version`	Print program version.

Virtual Terminal Interfaces

VTY - Virtual Terminal (aka TeletYpe) Interface is a Command Line Interface (CLI) used to change and/or view the current configuration.

VTY stands for Virtual TeletYpe interface. With VTY, users can connect to the daemon via the `telnet` protocol. VTY can only be accessed if there is a password set. If no password is set on the system, the VTY interface refuses connection.

```
% telnet localhost 2601
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

Hello, this is zebra (version 0.88)
Copyright 1997-2000 Kunihiro Ishiguro
```

User Access Verification

The key **?** is used to look up commands.

```
Password: XXXXX
Router> ?
enable            Turn on privileged commands
exit              Exit current mode and down to previous mode
help              Description of the interactive help system
```

```
list              Print command list
show              Show running system information
who               Display who is on a vty
Router> enable
Password: XXXXX
Router# configure terminal
Router(config)# interface eth0
Router(config-if)# ip address 10.0.0.1/8
Router(config-if)# ^Z
Router#
```

VTY Modes

There are three basic VTY modes:

- VTY View Mode - Read-only
- VTY Enable Mode - Read-write
- VTY Other Modes - Special modes (`tftp`, etc)

Certain commands are restricted to specific VTY modes.

VTY View Mode

This mode is for read-only access to the Command Line Interface (CLI). Users can exit the mode by leaving the system or by entering Enable mode.

VTY Enable Mode

This mode is for read-write access to the CLI. Users can exit the mode by leaving the system or by escaping to View mode.

VTY Other Modes

This mode is used for describing other modes.

VTY CLI Commands

These commands are described in the following three subsections.

- CLI Movement Commands - Cursor movement
- CLI Editing Commands - Changing text
- CLI Advanced Commands - Other commands and session management

CLI Movement Commands

Default Emacs key combinations are used to move the CLI cursor.

NOTE: The **Meta** key varies on different systems with different keyboards. Typically, **Meta** is configured to the **Alt** key, however it can also be configured to **Esc**.

Table 2-3: CLI Movement Commands

Command	Description
Ctrl-F **Right arrow**	Move forward (right) one character
Ctrl-B **Left arrow**	Move backward (left) one character
Meta-F	Move forward one word
Meta-B	Move backward one word
Ctrl-A	Move to the beginning of the line
Ctrl-E	Move to the end of the line

CLI Editing Commands

These commands are used for editing text on a line

NOTE: The **Meta** key varies on different systems with different keyboards. Typically, **Meta** is configured to the **Alt** key, however it can also be configured to **Esc**.

Table 2-4: CLI Editing Commands

Command	Description
Ctrl-H Del	Delete the character before cursor
Ctrl-D	Delete the character after cursor
Meta-D	Forward kill word
Ctrl-W	Backward kill word
Ctrl-K	Kill to the end of the line

Table 2-4: CLI Editing Commands (Continued)

Command	Description
Ctrl-U	Kill line from the beginning, erasing input
Ctrl-T	Transpose character

CLI Advanced Commands

There are several additional CLI commands for command line completions: “insta-help” and VTY session management.

Table 2-5: CLI Advanced Commands

Command	Description
Ctrl-C	Interrupt current input and move to the next line
Ctrl-Z	End current configuration session and move to top node
Ctrl-N **Down arrow**	Move down to next line in the history buffer
Ctrl-P **Up arrow**	Move up to previous line in the history buffer
Tab	Use command line completion

Users can use command line help by typing **`help`** at the beginning of the line. Typing **`?`** at any point in the line displays possible completions of the command.

The zebra Daemon

`zebra` is the routing manager daemon that provides kernel routing table updates, interface lookups, and redistribution of routes between different routing protocols.

Invoking zebra

Besides the common invocation options (see "Common Invocation Options" on page 14), there are `zebra`-specific invocation options:

Table 2-6: zebra Command Options

Command	Description
`-b --batch`	Runs in batch mode. `zebra` parses the configuration file and terminates immediately.
`-k --keep_kernel`	When `zebra` starts, old inserted routes are retained.
`-l --log-mode`	Set verbose logging on.
`-r --retain`	When program terminates, retain routes added by `zebra`.

zebra Interface Commands

The following table details interface commands for the `zebra` daemon. Before using these interface commands, users must first set the interface on which to issue commands. To set the interface, use this command:

```
interface IFNAME
```

where *`IFNAME`* is the name of the interface. After the interface is set, users can issue these commands:

Table 2-7: zebra Interface Commands

Command	Description

`shutdown`
 	
`no shutdown`
	Shutdown, or disable shutdown of current interface.

`ip address` *`ADDRESS`*	Set IP address for the interface.
`description` *`DESCRIPTION ...`*	Set description for the interface.

Table 2-7: zebra Interface Commands (Continued)

Command	Description

`multicast`
 	
`no multicast`
	Enable or disable multicast flags for the interface.

`bandwidth` *`1-10000000`* `no bandwidth` *`1-10000000`*	Set bandwidth value for the interface. This is used in calculating OSPF cost. This command does not set actual device configurations.

zebra Static Route Commands

The following table shows the commands used in Static Routes.

Table 2-8: Static Route Commands

Command	Description
`table` *`TABLENO`*	Select the primary kernel routing table to be used. This only works for kernels supporting multiple routing tables (Linux 2.2.x, for example).
`ip route` *`NETWORK GATEWAY`*	Sets the gateway for `ip route`
`ipv6 route` *`NETWORK GATEWAY`*	Sets the gateway for an IPv6 route

zebra Terminal Mode Commands

The following table shows terminal mode commands for `zebra`:

Table 2-9: zebra Terminal Mode Commands

Command	Description

`show ip route`	Display current routes in the `zebra` database:
	
	`Router# show ip route`
	`Codes: K - kernel route, C - connected, S - static, R - RIP, B - BGP * - FIB route.`
	
	`K* 0.0.0.0/0 203.181.89.241`
	`S 0.0.0.0/0 203.181.89.1`
	`C* 127.0.0.0/8 lo`
	`C* 203.181.89.240/28 eth0`

`show interface`	Display current interface
`show ipv6 route`	Display current IPv6 route
`show ipforward`	Display whether the host IP forwarding function is enabled or not. Almost any UNIX kernel can be configured with IP forwarding disabled. If so, the system is unable to function as a router.
`show ipv6forward`	Display whether IPv6 host forwarding is enabled or not.

ripd

RIP – Routing Information Protocol is a widely deployed Interior Gateway Protocol (IGP). RIP was developed in the 1970s at Xerox Labs as part of the XNS routing protocol. RIP is a distance-vector protocol based on the Bellman-Ford algorithms. As a distance-vector protocol, the RIP router sends periodic updates to its neighbors, allowing the convergence of a known topology. In each update, the distance to any given network is broadcasted to its neighboring router.

`ripd` supports RIP v2 as described in RFC2453 and RIP v1 as described in RFC1058

Starting and Stopping ripd

The default configuration file name of `ripd` is `ripd.conf`. The `ripd` daemon searches the current local directory and `/usr/zebra/etc` for configuration files.

RIP uses port 521 to send and receive RIP packets, so the user must have the capability to bind this port. Generally, this means that the user must have superuser privileges. The RIP protocol requires interface information maintained by the `zebra` daemon. `zebra` must be running before starting `ripd`.

Start `zebra` and `ripd` by typing:

```
# zebra -d
# ripd -d
```

Stop `ripd` with the `kill` command. Find the Process ID (*`pid`*) and kill the daemon with the following commands:

```
# ps -axon | grep ripd
# kill pid
```

Some signals can be used to configure `ripd`:

- `SIGHUP`
  Reload configuration file `ripd.conf`. All configurations are reset. All "learned" routes are removed from the routing table.
- `SIGUSR1`
  Rotate `ripd` logfile
- `SIGINT`
  `ripd` sends signal interrupt
- `SIGTERM`
  `ripd` sweeps all installed RIP routes from the Zebra routing table and then terminates.

In addition to the Common Invocation Options (see "Common Invocation Options" on page 14), `ripd` uses the following command options:

Table 2-10: ripd Command Options

Command	Description
`-r --retain`	When the program terminates, retain manually marked routes in the Zebra Routing Table that were added by `ripd`.

RIP Netmask

The netmask feature of `ripd` supports both v1 and v2 of RIP. RIP v1 does not contain any netmask information. Rather, network classes are used to determine the size of the netmask. Class A networks are assigned an 8 bit mask, Class B networks are assigned a 16 bit mask, and Class C networks are assigned a 24 bit mask. Typically, network masks are assigned to a packet based on the interface that receives the packet.

RIP v2 supports a variable length subnet mask (VLSM). By extending the subnet mask, the mask can be divided and reused. Each subnet can be used for different purposes, such as large and mid-sized LANs and WAN links. `ripd` for Zebra does not support the non-sequential netmasks included in RIP v2.

In a case of similar information with the same prefix and metric, the old information is suppressed. `ripd` does not currently support equal cost multipath routing.

RIP Configuration

The following table details RIP configuration commands:

Table 2-11: RIP Configuration Commands

Command	Description

`router rip`
 	
`no router rip`
	The `router rip` command is used to enable RIP. To disable RIP, use the `no router rip` command. RIP must be enabled before carrying out any of the RIP commands.

`rip version (`*`1`*`|`*`2`*`)`
 	
`no rip version (`*`1`*`|`*`2`*`)`
	RIP can be configured to process either RIP v1 or RIP v2 packets, the default mode is RIP v2. If no version is specified, then the RIP process defaults to RIP v2. In the case that RIP is set to v1, the setting "`Version 1`" is displayed. However, the setting "`Version 2`" is not displayed when RIP v2 is set explicitly or non-explicitly.

Table 2-11: RIP Configuration Commands (Continued)

Command	Description

`network NETWORK`
 	
`no network NETWORK`
	Set the RIP enabled interface to *NETWORK*. The interface with matching *NETWORK* addresses are enabled. These commands either enable or disable RIP interfaces between certain specified network address. For example, if the network for 10.0.0.0/24 is RIP enabled, all addresses from 10.0.0.0 to 10.0.0.255 would be enabled for RIP. The `no network command` disables RIP for the specified network.

`network IFNAME`
 	
`no network IFNAME`
	Set RIP-enabled interface to *IFNAME*. Both the sending and receiving of RIP packets are enabled on the specified port. The `no network IFNAME` command disables RIP on the specified interface.

`neighbor A.B.C.D`
 	
`no neighbor A.B.C.D`
	Specify RIP neighbor. If a neighbor doesn't support multicast, this command is used to specify neighbors. In some cases, not all routers are able to understand multicasting (where packets are sent to a network or a group of addresses). In a situation where a neighbor cannot process multicast routing, it is necessary to establish a direct link between routers. The `neighbor` command allows the network administrator to specify a router as a RIP neighbor. The `no neighbor A.B.C.D` command disables the RIP neighbor.

Below is a simple RIP configuration. Interface `eth0` and any interfaces that match 10.0.0.0/8 are RIP-enabled.

```
!
router rip
network 10.0.0.0/8
network eth0
!
```

Passive Interface Commands

The following table details commands used for setting passive interfaces:

Table 2-12: Passive Interface Commands

Command	Description
`passive-interface` *IFNAME* `no passive-interface` *IFNAME*	This command sets the specified interface to passive mode. When passive mode is set, all receiving packets are processed as normal. `ripd` does not multicast or unicast RIP packets, except in the case where an RIP neighbor is specified.
`version` *VERSION*	Set the RIP process version. *VERSION* can be 1 or 2
`ip rip send version` *VERSION*	*VERSION* can be 1, 2, 1 2. This configuration command overrides the `routers RIP` *VERSION* setting. This command enables the selected interface to send packets with RIP v1, RIP v2, or both. In the case of 1 2, packets are both broadcast and multicast.
`ip rip receive version` *VERSION*	Version setting for incoming RIP packets. This command enables the selected interface to receive packets in RIP v1, RIP v2, or both.

RIP split-horizon Command

The `split-horizon` command improves RIP convergence by preventing the router from advertising routes back to a neighbor that advertised the route first. The table below describes the command used in RIP `split-horizon`.

Table 2-13: RIP split-horizon Command

Command	Description

`ip split-horizon`
 	
`no ip split-horizon`
	Control split-horizon on the interface. Default is `ip split-horizon`. If split-horizon is not performed on the interface, specify `no ip split-horizon`.

Announcing RIP Routes Commands

The following table describes commands used in announcing RIP routes:

Table 2-14: Announcing RIP Route Commands

Command	Description

`redistribute kernel`
 	
`redistribute kernel metric` *`0-16`*
 	
`kernel route-map` *`ROUTE-MAP`*
 	
`no redistribute kernel`
	`redistribute kernel` redistributes routing information from a kernel route entry into the RIP tables. `no redistribute kernel` disables the route.

`redistribute static`
 	
`redistribute static metric` *`0-16`*
 	
`redistribute static route-map` *`ROUTE-MAP`*
 	
`no redistribute static`
	`redistributes static` redistributes routing information from a static route entry into the RIP tables. `no redistribute static` disables the route.

`redistribute connected`
 	
`redistribute connected metric` *`0-16`*
 	
`redistribute connected route-map` *`ROUTE-MAP`*
 	
`no redistribute connected`
	Redistributes a connected route into the RIP tables. This command disables connected routes in the RIP tables. The connected route on RIP enabled interface is announced in default.

`redistribute ospf`
 	
`redistribute ospf metric` *`0-16`*
 	
`ospf route-map` *`ROUTE-MAP`*
 	
`no redistribute ospf`
	`redistribute ospf` redistributes routing information from an OSPF route entry into the RIP tables. `no redistribute ospf` disables the route.

`redistribute bgp`
 	
`redistribute bgp metric` *`0-16`*
 	
`redistribute bgp route-map` *`ROUTE-MAP`*
 	
`no redistribute bgp`
	`redistribute bgp` redistributes routing information from a `bgp` route entry into the RIP tables. `no redistribute bgp` disables the route.

RIP-only Static Route Command

The following table describes the command used to specify RIP-only static routes.

Table 2-15: Static route Command

Command	Description
`route` *`A.B.C.D/M`* `no route` *`A.B.C.D/M`*	This command is specific to Zebra. The `route` command makes a static route inside RIP only. This command should be used by advanced users who are already familiar with the RIP protocol. In most cases, it is recommended to create a static route in Zebra and redistribute it in RIP using `redistribute static`.

RIP Route Filtering Commands

RIP routes can be filtered with the `distribute-list command`.

Table 2-16: RIP Route Filtering Commands

Command	Description

`distribute-list` *`ACCESS_LIST DIRECT`* (*`in`*|*`out`*) *`IFNAME`*	You can apply access lists to the interface with a `distribute-list` command. *`ACCESS_LIST`* is the access list name. *`DIRECT`* is *in* or *out*. If *`DIRECT`* is *in*, the access list is applied to input packets.
	
	The `distribute-list` command can be used to filter the RIP path. `distribute-list` can apply access lists to a chosen interface. Users should first specify the access list, then the name of the access list used in the `distribute-list` command. For example, in the following configuration, `eth0` permits only the paths that match the route 10.0.0.0/8:
	
	`!`
	`router rip`
	`distribute-list private in eth0`
	`!`
	`access-list private permit 10 10.0.0.0/8`
	`access-list private deny any`
	`!`
	
	`distribute-list` can be applied to both incoming and outgoing data.

`distribute-list prefix` *`PREFIX_LIST DIRECT`* (*`in`*|*`out`*) *`IFNAME`*	Users can apply prefix lists to the interface with a `distribute-list` command. *`PREFIX_LIST`* is the prefix list name. The second argument is the direction (*in* or *out*). If *`DIRECT`* is *in* the access list is applied to input packets.

RIP Metric Manipulation Commands

The RIP metric is a value used to measure the distance of the network. `ripd` increments the metric when network information is received. The metric for redistributed routes is set to 1.

Table 2-17: RIP Metric Manipulation Commands

Command	Description
`default-metric` *`1-16`* `no default-metric` *`1-16`*	This command modifies default metric value for redistributed routes. The default value is 1. This command does not affect the connected route, even if it is redistributed by `redistribute connected`. To modify the connected route's metric value, use `redistribute connected metric` or `route-map offset-list`.
`offset-list` *`ACCESS-LIST`* (*`in`*|*`out`*)	Add a metric offset to *`ACCESS-LIST`* for incoming (*in*) or outgoing (*out*) routes.
`offset-list` *`ACCESS-LIST`* (*`in`*|*`out`*) *`IFNAME`*	Add a metric offset to *`IFNAME`* (interface) in *`ACCESS-LIST`* for incoming (*in*) or outgoing (*out*) routes.

RIP Distance Commands

The RIP distance value is used by the `zebra` daemon. The default RIP distance is set to 120. Distance is a measurement of administrative distances between hosts, which is used in selecting shorter and more reliable paths.

Table 2-18: RIP Distance Commands

Command	Description
`distance` *`1-255`* `no distance` *`1-255`*	Set default RIP distance to specified value.

Table 2-18: RIP Distance Commands (Continued)

Command	Description
`distance` *`1-255 A.B.C.D/M`* `no distance` *`1-255 A.B.C.D/M`*	Set default RIP distance to specified value when the route source IP address matches the specified prefix.
`distance` *`1-255 A.B.C.D/M ACCESS-LIST`* `no distance` *`1-255 A.B.C.D/M ACCESS-LIST`*	Set default RIP distance to specified value when the route's source IP address matches the specified prefix and specified `access-list`.

RIP route-map Command

The optional argument `route-map` *`MAP_NAME`* is added to each redistribute statement.

Table 2-19: RIP route-map command

Command	Description

`redistribute static [route-map` *`MAP_NAME`*`]`
 	
`redistribute connected [route-map` *`MAP_NAME`*`]`
	Controls the redistribution of routes between protocols.

Cisco applies a `route-map` before routes that are exported to the RIP route table. In Zebra's current test implementation, `ripd` applies a `route-map` after routes listed in the route table and before routes are announced to the interface (through output filter).

The `route-map` statement is needed to use `route-map` functionality.

route-map Match Statement

The following tables describes `route-map` match statements:

Table 2-20: route-map Match Statement

Command	Description
`match metric` *`METRIC`*	Match if the route has this metric.
`match ip address` *`ACCESS-LIST`*	Match if route destination is permitted by *`ACCESS-LIST`*.

Table 2-20: route-map Match Statement (Continued)

Command	Description
`match ip next-hop` *`A.B.C.D`*	Cisco uses the `access-list ripd` IPv4 address. Match if route has this `next-hop` (listed in the RIP route table).
`match interface` *`NAME`*	This match differs from the Cisco definition. Cisco uses a list of interfaces (*NAME1 NAME2 ... NAMEN*). `ripd` currently allows for only one name. The Cisco notation includes the next-hop of routes (similar to `ip next-hop` statement). `ripd` interprets interface *NAME* as where this route is sent.

Route-map Set Statement

The following table describes `route-map` set statements:

Table 2-21: route-map Set Statement

Command	Description
`set metric` *`0-4294967295`*	Set a metric for the matched route when announcement is sent. The metric value range is large for compatibility reasons.
`set next-hop` *`A.B.C.D`*	Set `next-hop` field.

RIP Authentication Commands

The following command describes RIP Authentication commands:

Table 2-22: RIP Authentication Commands

Command	Description

`ip rip authentication mode md5`
 	
`no ip rip authentication mode md5`
	Set the interface with RIPv2 md5 authentication.

`ip rip authentication mode text`
 	
`no ip rip authentication mode text`
	Set the interface with RIPv2 simple password authentication.

`ip rip authentication string` *`STRING`* `no ip rip authentication string` *`STRING`*	By default, RIP v2 has simple text authentication. This command sets the authentication string. the string must be shorter than 16 characters.

`ip rip authentication key-chain` *`KEY-CHAIN`*
 	
`no ip rip authentication key-chain` *`KEY-CHAIN`*
	Specify keyed md5 chain.
	`!`
	`key chain test`
	`key 1`
	`key-string test`
	`!`
	`interface eth1`
	`ip rip authentication mode md5`
	`ip rip authentication key-chain test`
	`!`

RIP Timer Commands

The following table describes the commands used in setting RIP timers:

Table 2-23: RIP Timer Commands

Command	Description
`timers basic` *`UPDATE TIMEOUT GARBAGE`*	The RIP protocol has several timers. Users can configure the value of these timers with the `timers basic` command. The default settings for the timers are as follows: The `update` timer defaults to 30 seconds. For each update, the RIP process sends an unsolicited response message containing the complete routing table to all neighboring RIP routers. The `timeout` timer defaults to 180 seconds. Upon expiration of the timeout, the route is no longer valid. However, the route is retained in the routing table for a short period of time so neighbors can be notified that the route has been dropped. The `garbage` collect timer defaults to 120 seconds. Upon expiration of the garbage-collection timer, the route is removed from the routing table. The `timers basic` command allows the default values of the timers listed above to be changed.
`no timers basic`	The `no timers basic` command resets the timers to the default settings listed above.

RIP Display Commands

The following table describes commands used to display RIP routes and protocols:

Table 2-24: RIP Display Commands

Command	Description
`show ip rip`	This command displays all RIP routes. For routes received through RIP, this command displays the time the packet was sent and the tag information. This command also displays this information for routes redistributed into RIP.

`show ip protocols`	The command display current RIP status. It includes RIP timer, filtering, version, RIP enabled interface and RIP peer information.
	
	`ripd> **show ip protocols**`
	`Routing Protocol is "rip"`
	`Sending updates every 30 seconds with +/-50%, next due in 35 seconds`
	`Timeout after 180 seconds, garbage collect after 120 seconds`
	`Outgoing update filter list for all interface is not set`
	`Incoming update filter list for all interface is not set default redistribution metric is 1`
	`Redistributing: kernel connected`
	`Default version control: send version 2, receive version 2`
	`  Interface  Send  Recv`
	`Routing for Networks:`
	`  eth0`
	`  eth1`
	`  1.1.1.1`
	`  203.181.89.241`
	`Routing Information Sources:`
	`  Gateway  BadPackets BadRoutes  Distance Last Update`

RIP Debugging Commands

The following table describes RIP debug commands:

Table 2-25: RIP Debugging Commands

Command	Description
`debug rip events`	Debug RIP events. `debug rip` shows RIP events, including packets sent and received, timers, and changes in interfaces.
`debug rip zebra`	Debug RIP between `zebra` communication. This command details the communication between `ripd` and `zebra`. This command displays the addition and deletion of paths to the kernel and the sending and receiving of interface information.
`show debugging rip`	Display RIP debugging option. This command displays the information currently set for `ripd` debug.

ripngd

`ripngd` supports the RIPng protocol as described in RFC2080. It is an IPv6 incarnation of the RIP protocol.

Invoking ripngd

There are no `ripngd`-specific options, however, common options can be specified (see “Common Invocation Options” on page 14).

ripngd Configuration

`ripngd` supports the following commands:

Table 2-26: ripngd Configuration

Command	Description
`router ripng`	Enable `RIPng`.
`flush_timer` *`TIME`*	Set flush timer.
`network` *`NETWORK`*	Set RIPng enable interface by *`NETWORK`*
`network` *`IFNAME`*	SetRIPng Command: `route` *`NETWORK`* Set RIPng static routing announcement of *`NETWORK`*.
`router zebra`	This command is the default and does not appear in the configuration. With this statement, RIPng routes go to the `zebra` daemon.

ripngd Terminal Mode Commands

The following table lists the `ripngd` Terminal Mode Commands:

Table 2-27: ripngd Terminal Mode Commands

Command
`show ip ripng`
`show debugging ripng`
`debug ripng events`
`debug ripng packet`
`debug ripng zebra`

ripngd Filtering Commands

The following table describes commands used in `ripngd` filtering:

Table 2-28: ripngd Filtering Commands

Command	Description
`distribute-list` *`ACCESS_LIST`* `(in|out)` *`IFNAME`*	Users can apply an access-list to the interface using the `distribute-list` command. *`ACCESS_LIST`* is an access-list name. *`DIRECT`* is **in** or **out**. If DIRECT is **in**, the access-list is applied only to incoming packets.

ospfd

`ospfd` provides an OSPF v2 routing protocol as described in RFC2178. OSPF is an IGP (Interior Gateway Protocol). Compared with RIP, OSPF can serve larger networks. In addition, its periods of convergence are shorter. OSPF is widely used in large networks, such as ISP backbones and enterprise networks.

Starting and Stopping ospfd

Start the `ospfd` daemon with this command:

```
# ospfd -d
```

NOTE: `zebra` must be running before starting `ospfd`.

Stop `ospfd` with the `kill` command. Find the Process ID (*pid*) and kill the daemon with the following commands:

```
# ps -axon | grep ripd
# kill pid
```

Configuring ospfd

There are no `ospfd`-specific options, however, common options can be specified (see “Common Invocation Options” on page 14). `ospfd` requires interface information from `zebra`. The `zebra` daemon must be running before invoking `ospfd`.

Like other daemons, `ospfd` is configured in the OSPF specific configuration file, `ospfd.conf`.

OSPF Router Commands

To start the OSPF process an OSPF router must be specified. As of this writing, `ospfd` does not support multiple OSPF processes.

Table 2-29: OSPF Router Commands

Command	Description

`router ospf`
 	
`no router ospf`
	Enable or disable the OSPF process. `ospfd` does not yet support multiple OSPF processes, so specific process numbers cannot be specified.

`ospf router-id` *`A.B.C.D`* `no ospf router-id`	Enable or disable a fixed OSPF router IP address.

`ospf abr-type` *`TYPE`*
 	
`no ospf abr-type` *`TYPE`*
	`TYPE` can be:
	`cisco`
	`ibm`
	`shortcut`
	`standard`

`ospf rfc1583compatibility`
 	
`no ospf rfc1583compatibility`
 	
`passive interface` *`INTERFACE`*
 	
`no passive interface` *`INTERFACE`*
	Enable or disable RFC1583-compatible preferences when choosing between multiple AS-external LSAs advertising the same destination.

`timers spf` *`0-4294967295 0-4294967295`* `no timers spf`	Enable or disable OSPF timer. Time denoted in seconds.

Table 2-29: OSPF Router Commands (Continued)

Command	Description
`refresh group-limit` *`0-10000`* `refresh per-slice` *`0-10000`* `refresh age-diff` *`0-10000`*	Refreshes (Link State Advertisement) LSA. Rate is in seconds.
`auto-cost reference-bandwidth` *`1-4294967`* `no auto-cost reference-bandwidth`	Control OSPF default metrics based on `reference-bandwidth`. Rate is megabits per second.
`network` *`A.B.C.D/M`* `area` *`A.B.C.D`* `network` *`A.B.C.D/M`* `area` *`0-4294967295`* `no network` *`A.B.C.D/M`* `area` *`A.B.C.D`* `no network` *`A.B.C.D/M`* `area` *`0-4294967295`*	This command specifies the OSPF-enabled interface. If the interface has an address of 10.0.0.1/8, then the command below provides network information to the OSPF processes. router ospf network 10.0.0.0/8 area 0 The network command mask length should be the same as the interface address mask.

OSPF Area Commands

The following table describes commands used for setting OSPF area:

Table 2-30: OSPF Area Commands

Command
area *A.B.C.D* range *A.B.C.D/M* no area *A.B.C.D* range *A.B.C.D/M* area *0-4294967295* range *A.B.C.D/M* no area *0-4294967295* range *A.B.C.D/M*
area *A.B.C.D* range *IPV4_PREFIX* suppress no area *A.B.C.D* range *IPV4_PREFIX* suppress area *A.B.C.D* range *IPV4_PREFIX* suppress substitute *IPV4_PREFIX* no area *A.B.C.D* range *IPV4_PREFIX* suppress substitute *IPV4_PREFIX*
area *A.B.C.D* virtual-link *A.B.C.D* no area *A.B.C.D* virtual-link *A.B.C.D* area *0-4294967295* virtual-link *A.B.C.D* no area *0-4294967295* virtual-link *A.B.C.D*
area *A.B.C.D* shortcut no area *A.B.C.D* shortcut area *0-4294967295* shortcut no area *0-4294967295* shortcut
area *A.B.C.D* stub no area *A.B.C.D* stub area *0-4294967295* stub no area *0-4294967295* stub
area *A.B.C.D* stub no-summary no area *A.B.C.D* stub no-summary area *0-4294967295* stub no-summary no area *0-4294967295* stub no-summary

Table 2-30: OSPF Area Commands (Continued)

Command
area *A.B.C.D* default-cost *0-16777215* no area *A.B.C.D* default-cost *0-16777215* area *0-4294967295* default-cost *0-16777215* no area *0-4294967295* default-cost *0-16777215*
area *A.B.C.D* export-list *NAME* no area *A.B.C.D* export-list *NAME* area *0-4294967295* export-list *NAME* no area *0-4294967295* export-list *NAME*
area *A.B.C.D* import-list *NAME* no area A.B.C.D import-list *NAME* area *0-4294967295* import-list *NAME* no area *0-4294967295* import-list *NAME*
area *A.B.C.D* authentication no area *A.B.C.D* authentication area *0-4294967295* authentication no area *0-4294967295* authentication
area *A.B.C.D* authentication message-digest area *0-4294967295* authentication message-digest

OSPF Interface Commands

The following table describes OSPF interface commands:

Table 2-31: OSPF Interface Commands

Command	Description
`ip ospf authentication-key` *`AUTH_KEY`* `no ip ospf authentication-key`	Set OSPF authentication key for a simple password. By setting *`AUTH_KEY`*, all OSPF packets are authenticated. *`AUTH_KEY`* can be up to 8 characters.
`ip ospf message-digest-key` *`KEYID`* `md5` *`KEY`* `no ip ospf message-digest-key`	Set the OSPF authentication key for cryptographic password. The cryptographic algorithm is MD5. *`KEYID`* identifies the secret key used to create the `message digest`. *`KEY`* is the actual `message-digest-key` (up to 16 characters).
`ip ospf cost` *`0-65535`* `no ip ospf cost`	Set link cost for specified interface. The `cost` value is set to router-LSA's metric field, and is used for SPF calculation.
`ip ospf dead-interval` *`1-65535`* `no ip ospf dead-intercal`	Set the number of seconds for `RouterDeadInterval` timer value used for the `wait` timer and `inactivity` timer. This value must be the same for all routers attached to a common network. The default value is 40 seconds.
`ip ospf hello-interval` *`1-65535`* `no ip ospf hello-intercal`	Set number of seconds for `hello-interval` timer value. The value of `hello-interval` determines how often (in seconds) a `Hello` packet is sent on the specified interface. This value must be the same for all routers attached to a common network. The default value is 10 seconds.

`ip ospf network (broadcast|non-broadcast| point-to-multipoint|point-to-point)`
 	
`no ip ospf network`
	Set explicit network type for specified interface.

Table 2-31: OSPF Interface Commands (Continued)

Command	Description
`ip ospf priority` *`0-255`* `no ip ospf priority`	Set `RouterPriority` integer value. Setting a higher value allows the router to be eligible to become the Designated Router. Setting the value to 0 makes the router ineligible to be the Designated Router. The default value is 1.
`ip ospf retransmit-interval` *`1-65535`* `no ip ospf retransmit-interval`	Set number of seconds for `RxmtInterval` timer value. This value is used when retransmitting `Database Description` and `Link State Request` packets. The default value is 5 seconds.

`ip ospf transmit-delay`
 	
`no ip ospf transmit-delay`
	Set number of seconds for `InfTransDelay` value. The age of LSAs is incremented by this value when transmitting. The default value is 1 second.

OSPF Redistribute Route Commands

The following table describes commands used in redistributing routes to OSPF:

Table 2-32: OSPF Redistribute Route Commands

Command
redistribute (kernel|connected|static|rip|bgp) redistribute (kernel|connected|static|rip|bgp) *ROUTE-MAP* redistribute (kernel|connected|static|rip|bgp) metric-type (*1*|*2*) redistribute (kernel|connected|static|rip|bgp) metric-type (*1*|*2*) route-map *WORD* redistribute (kernel|connected|static|rip|bgp) metric *0-16777214* redistribute (kernel|connected|static|rip|bgp) metric *0-16777214* route-map *WORD* redistribute (kernel|connected|static|rip|bgp) metric-type (*1*|*2*) metric *0-16777214* redistribute (kernel|connected|static|rip|bgp) metric-type (*1*|*2*) metric *0-16777214* \ route-map *WORD* no redistribute (kernel|connected|static|rip|bgp)
default-information originate default-information originate metric *0-16777214* default-information originate metric *0-16777214* metric-type (*1*|*2*) default-information originate metric *0-16777214* metric-type (*1*|*2*) route-map *WORD* default-information originate always default-information originate always metric *0-16777214* default-information originate always metric *0-16777214* metric-type (*1*|*2*) default-information originate always metric *0-16777214* metric-type (*1*|*2*) route-map *WORD* no default-information originate distribute-list *NAME* out (kernel|connected|static|rip|ospf no distribute-list *NAME* out (kernel|connected|static|rip|ospf default-metric *0-16777214*
distribute-list *NAME* out (kernel|connected|static|rip|ospf no distribute-list *NAME* out (kernel|connected|static|rip|ospf
default-metric *0-16777214* no default-metric
distance *1-255* no distance *1-255*
distance ospf (intra-area|inter-area|external) *1-255* no distance ospf
router zebra no router zebra

Displaying OSPF Information

The following table shows the commands used to display OPSF information.

Table 2-33: Displaying OSPF Information

Command
`show ip ospf`
`show ip ospf interface` *`INTERFACE`*

`show ip ospf neighbor`
`show ip ospf neighbor` *`INTERFACE`*
`show ip ospf neighbor detail`
`show ip ospf database`

`show ip ospf database (asbr-summary|external|network|router|summary)`
`show ip ospf database (asbr-summary|external|network|router|summary)` *`LINK-STATE-ID`*
`show ip ospf database (asbr-summary|external|network|router|summary) LINK-STATE-ID adv-router` *`ADV-ROUTER`*
`show ip ospf database (asbr-summary|external|network|router|summary) adv-router` *`ADV-ROUTER`*
`show ip ospf database (asbr-summary|external|network|router|summary)` *`LINK-STATE-ID`* `self-originate`
`show ip ospf database (asbr-summary|external|network|router|summary) self-originate`
`show ip ospf database max-age`
`show ip ospf database self-originate`
`show ip ospf refresher`
`show ip ospf route`

OSPF Debugging Commands

The following table shows the commands used to debug OSPF:

Table 2-34: OSPF Debugging Commands

Command
debug ospf packet (hello|dd|ls-request|ls-update|ls-ack|all) (*send*|*recv*) *DETAIL* no debug ospf packet (hello|dd|ls-request|ls-update|ls-ack|all) (*send*|*recv*) *DETAIL*
debug ospf ism no debug ospf ism debug ospf ism (status|events|timers) no debug ospf ism (status|events|timers)
debug ospf nsm no debug ospf nsm debug ospf nsm (status|events|timers) no debug ospf nsm (status|events|timers)
debug ospf lsa no debug ospf lsa debug ospf lsa (generate|flooding|refresh) no debug ospf lsa (generate|flooding|refresh)
debug ospf zebra no debug ospf zebra debug ospf zebra (interface|redistribute) no debug ospf zebra (interface|redistribute)
show debugging ospf

bgpd

`bgpd` is a Border Gateway Protocol 4 (BGP-4) protocol daemon. BGP-4 is described in RFC1771. `bgpd` also supports Multiprotocol Extension for BGP-4 (sometimes known as BGP-4+ or MBGP) which is described in RFC2283.

BGP-4 is a primary EGP (Exterior Gateway Protocols) and is used for inter-domain routing.

Configuring bgpd

The default configuration file for `bgpd` is `/usr/zebra/etc/bgpd.conf`. All of the `bgpd` commands must be configured in `bgpd.conf`.

`bgpd`-specific invocation options are described below. Common options may also be specified (see “Common Invocation Options” on page 14).

Table 2-35: bgpd Options

Option	Description
`-p PORT` `--bgp_port=PORT`	Set the BGP protocol’s port number.
`-r` `--retain`	When program terminates, retain BGP routes added by `zebra`.

BGP Router Commands

The BGP router must first be configured with the `router bgp` command. To configure the BGP router, an AS (Autonomous System) number is required. The AS number provides identification for an Autonomous System. The BGP protocol uses the AS number to detect whether the BGP connection is internal or external.

The AS number is a digit between 1 and 65535. Instructions on using the AS number are described in RFC1930. AS numbers 64512 through 65535 are reserved

for private use. Private AS numbers must not to be advertised on the global Internet.

Table 2-36: BGP Router Commands

Command	Description
`router bgp` *`AS-NUMBER`*	Enables a BGP protocol process with the specified *AS-NUMBER*. After this statement, users can input any BGP Commands. Different BGP processes cannot be created under a different *AS-NUMBER* without specifying multiple instances.
`no router bgp` *`AS-NUMBER`*	Destroys a BGP protocol process with the specified *AS-NUMBER*
`bgp router-id` *`ROUTER-ID`*	This command specifies the `router-id`. If `bgpd` connects to `zebra`, it receives the interface and address information. In that case, the default `router-id` value is set to the largest IP Address of the interfaces. If the router `zebra` is disabled, `bgpd` can't receive interface information, so the `router-id` is set to 0.0.0.0. If this happens, the `router-id` must be set manually.

BGP Peer Commands

The following table describes the commands used in setting BGP Peers:

Table 2-37: BGP Peer Commands

Command	Description

`neighbor` *`PEER`* `remote-as` *`AS-NUMBER`*	Creates a new neighbor whose `remote-as` is *AS-NUMBER*. *`PEER`* is an IPv4 address.
	
	`router bgp 1`
	`  neighbor 10.0.0.1 remote-as 2`
	
	In this case the router in AS-1 is trying to peer with AS-2 at 10.0.0.1.
	
	This command must be the first command used when configuring a neighbor. If `remote-as` is not specified, `bgpd` responds with an error:
	
	can't find neighbor 10.0.0.1

BGP Network Commands

The following table describes the commands used in setting BGP networks:

Table 2-38: BGP Network Commands

Command	Description

`network` *`NETWORK`*
 	
`no network` *`NETWORK`*
	This command adds the announcement network.
	
	`router bgp 1`
	`  network 10.0.0.0/8`
	
	This configuration example says that network 10.0.0.0/8 is announced to all neighbors. Some vendor routers do not advertise routes if they are not present in its IGP routing tables; `bgpd` doesn't care about IGP routes when announcing its routes.

`aggregate-address` *`NETWORK`* `no aggregate-address` *`NETWORK`*	This command specifies an aggregate address.

BGP Redistribute Commands

The following table describes the commands used in redistributing routes to BGP:

Table 2-39: BGP Redistribute Commands

Command	Description
`redistribute kernel`	Redistribute kernel route to BGP process.
`redistribute static`	Redistribute static route to BGP process.
`redistribute connected`	Redistribute connected route to BGP process.
`redistribute rip`	Redistribute RIP route to BGP process.
`redistribute ospf`	Redistribute `OSPF` route to BGP process.

Peer Configuration Commands

The following table describes the commands used to configure BGP peers:

Table 2-40: Peer Configuration Commands

Command	Description

`neighbor` *`PEER`* `shutdown`
 	
`no neighbor` *`PEER`* `shutdown`
	Shutdown *`PEER`*. Users can delete a neighbor's configuration with `no neighbor` *`PEER`* `remote-as` *`AS-NUMBER`*, but all configurations of the neighbor are deleted. Use this syntax to preserve the configuration and drop the BGP peer.

`neighbor` *`PEER`* `ebgp-multihop`
 	
`no neighbor` *`PEER`* `ebgp-multihop`
	Specifies *`PEER`* for `ebgp` multihopping.

`neighbor` *`PEER`* `version` *`VERSION`* `no neighbor` *`PEER`* `version` *`VERSION`*	Set up the neighbor's BGP version. VERSION can be: 4 4+ 4- BGP version 4 is the default value used for BGP peering. BGP version 4+ means that the neighbor supports Multiprotocol Extensions for BGP-4. BGP version 4- is similar, but uses the old Internet-Draft revision 00's Multiprotocol Extensions for BGP-4. Some routing software is still using this version.

Table 2-40: Peer Configuration Commands (Continued)

Command	Description

`neighbor PEER next-hop-self`
 	
`no neighbor PEER next-hop-self`
	This command specifies an announced route's `next-hop` as equivalent to the address of the BGP process.

`neighbor PEER update-source`
 	
`no neighbor PEER update-source`
	Sets BGP sessions to allow use of any functioning interface for TCP connections.

`neighbor PEER default-originate`
 	
`no neighbor PEER default-originate`
	`bgpd` defaults to not announce the default route (0.0.0.0/0), even if it is in routing table. Use this command to announce default routes.

`neighbor PEER port PORT`
 	
`no neighbor PEER port PORT`
	Sets a specific port for *PEER*.

`neighbor PEER send-community`
 	
`no neighbor PEER send-community`
	Sends *PEER* attribute to BGP community.

`neighbor PEER weight WEIGHT`
 	
`no neighbor PEER weight WEIGHT`
	This command specifies a default *WEIGHT* value for the neighbor's routes.

`neighbor PEER maximum-prefix NUMBER`
 	
`no neighbor PEER maximum-prefix NUMBER`
	Sets the maximum number of prefixes that can be sent to a neighbor.

`neighbor PEER interface IFNAME`
 	
`no neighbor PEER interface IFNAME`
	When connecting to a BGP peer over an IPv6 link-local address, users must specify the *IFNAME* of the interface used for the connection.

Peer Filtering Commands

The following table describes the commands used to filter BGP Peers:

Table 2-41: Peer Filtering Commands

Command	Description
`neighbor PEER distribute-list NAME DIRECT`	This command specifies a `distribute-list` for the peer. `DIRECT` is *in* or *out*.
`neighbor PEER prefix-list NAME DIRECT`	Distribute `PEER` attribute to BGP systems according to `prefix-list`. `DIRECT` is *in* or *out*.
`neighbor PEER filter-list NAME DIRECT`	Distributes `PEER` attribute according to `filter-list`. `DIRECT` is *in* or *out*.
`neighbor PEER route-map NAME DIRECT`	Apply a `route-map` on the neighbor. `DIRECT` must be *in* or *out*.

BGP Terminal Mode Commands

The following table describes the BGP terminal commands:

Table 2-42: BGP Terminal Mode Commands

Command	Description
`show ip bgp NETWORK`	Lists all `bgpd` routes
`show ip bgp regexp AS-REGEX`	Display routes matching AS path regular expression
`show ip bgp summary`	Display status of all BGP connections
`show ip bgp neighbor PEER`	Show status of BGP neighbor
`clear ip bgp PEER`	Clear peers with addresses of *X.X.X.X*
`clear ip bgb PEER soft in`	Clear `peer` using soft reconfiguration
`show debug`	Display debugging information

Table 2-42: BGP Terminal Mode Commands (Continued)

Command	Description

`debug event`
 	
`no debug event`
	Enable or disable BGP event debugging

`debug update`
 	
`no debug update`
	Enable or disable BGP updates debugging

`debug keepalive`
 	
`no debug keepalive`
	Enable or disable BGP keepalives debugging

BGP Log Format

`bgpd` outputs logging information to a terminal or a specified file. It includes routing updates and peer status change information. It also includes date, time, packet type, the peer IP address, and other routing information. The following is an example of the BGP log format:

```
1999/03/29 17:42:18 Update:[202.216.226.1]
130.58.0.0/16 med: 0 lpref: 0 nexthop: 202.216.226.1
aspath: 4691 3561 5119 3576 3782 i
```

Route Reflector Commands

The following table shows the commands used in reflecting routes:

Table 2-43: Route Reflector Commands

Commands
`bgp cluster-id` *`A.B.C.D`*

`neighbor` *`PEER`* `route-reflector-client`
 	
`no neighbor` *`PEER`* `route-reflector-client`

Route Server

Many ISPs are connected to each other by external BGP peers. Normally, these external BGP connections are created by full mesh methods. This method, however, has a scaling problem.

Route Server is a method used to resolve this problem. Each ISP's BGP router is a peer to a Route Server. The Route Server sends BGP information to other BGP routers. By applying this method, the number of BGP connections is reduced from O(n*(n-1)/2) to O(n).

Unlike a normal BGP router, a Route Server must have several routing tables for managing the different routing policies of each BGP router. The routing tables are called a BGP view. `bgpd` can work as normal BGP router, a Route Server, or both at the same time.

Multiple BGP Instance Commands

To enable multiple view function of `bgpd`, the multiple instance feature must be enabled before running the command.

Table 2-44: Multiple BGP Instance Commands

Command	Description
`bgp multiple-instance`	Enable BGP multiple instance feature. After this feature is enabled, users can make multiple BGP instances or multiple BGP views.
`no bgp multiple-instance`	Disable BGP multiple instance feature. This feature cannot be disabled when multiple BGP instances or views exist.

BGP Instance and View Commands

BGP instance is a normal BGP process. The result of the route selection is sent to the kernel routing table. Users can setup different AS at the same time when the BGP multiple instance feature is enabled. Routing Policy

Table 2-45: BGP Instance and View Commands

Command	Description

`router bgp` *`AS-NUMBER`*	Make a new BGP instance.
	`bgp multiple-instance`
	`!`
	`router bgp 1`
	`  neighbor 10.0.0.1 remote-as 2`
	`  neighbor 10.0.0.2 remote-as 3`
	`!`
	`router bgp 2`
	`  neighbor 10.0.0.3 remote-as 4`
	`  neighbor 10.0.0.4 remote-as 5`
	
	The BGP view is similar to a normal BGP process without the route selection appended to the kernel routing table. The BGP view is only for exchanging BGP routing information.

`router bgp` *`AS-NUMBER`* `view` *`NAME`*	Make a new BGP view. Any arbitrary word can be used for `NAME`. The route selection for this view is not sent to the kernel routing table. With this command, users can setup a Route Server:
	
	`bgp multiple-instance`
	`!`
	`router bgp 1 view 1`
	`  neighbor 10.0.0.1 remote-as 2`
	`  neighbor 10.0.0.2 remote-as 3`
	`!`
	`router bgp 2 view 2`
	`  neighbor 10.0.0.3 remote-as 4`
	`  neighbor 10.0.0.4 remote-as 5`

Users can set different routing policies for a peer. For example, these peers use different filters:

```
bgp multiple-instance
!
router bgp 1 view 1
  neighbor 10.0.0.1 remote-as 2
  neighbor 10.0.0.1 distribute-list 1 in
!
router bgp 1 view 2
  neighbor 10.0.0.1 remote-as 2
  neighbor 10.0.0.1 distribute-list 2 in
```

In this example, the BGP update from peer 10.0.0.1 is sent to both BGP view 1 and view 2. When the update is inserted into view 1, `distribute-list 1` is applied. However, if the the update is inserted into view 2 `distribute-list 2` is applied.

BGP Display Command

To display the routing table of the BGP view, the view name must be specified.

Table 2-46: BGP Display Command

Command	Description
`show ip bgp view` *`NAME`*	Display routing table of BGP view *`NAME`*

Dump BGP Packet and Table Commands

The table below describes commands used in BGP packet and table dumping:

Table 2-47: Dump BGP Packet and Table Commands

Command	Description
`dump bgp all` *`PATH`* `dump bgp all` *`PATH INTERVAL`*	Dump all BGP packet and events to *`PATH`* file.
`dump bgp updates` *`PATH`* `dump bgp updates` *`PATH INTERVAL`*	Dump BGP updates to *`PATH`* file.
`dump bgp routes` *`PATH`* `dump bgp routes` *`PATH`*	Dump whole BGP routing table to *`PATH`*. This is a heavy process.

Multiple BGP Protocol Extension Commands

BGP includes a Multiprotocol Extension which extends the BGP protocol to support IPv6 and Multicast routing. If BGP is used to exchange IPv6 routing information, it is called BGP-4+. When BGP is used to exchange multicast routing information, it is called MBGP.

`bgpd` supports Multiprotocol Extension for BGP. If a remote peer supports the protocol, bgpd can exchange multicast and IPv6 routing information.

Traditional BGP does not have a feature to detect the capability of the remote peer, which can make Multiprotocol Extension difficult to implement. `draft-ietf-idr-bgp4-cap-neg-04.txt` proposes a feature called capability negotiation. bgpd uses this capability negotiation to detect the remote peers capability. If the peer is configured as an IPv4 unicast neighbor, `bgpd` does not send the capability negotiation packet.

By default, zebra brings up peering with minimal common capabilities of both sides. For example, local routers have unicast and multicast capabilities, and remote routers have only unicast capabilities. In this case, the local router establishes the connection with unicast-only capability. If there is no common capability, zebra sends an unsupported capability error. and resets the connection.

If `capability` must match a remote peer, use the `strict-capability-match` command.

Table 2-48: Multiple BGP Protocol Expansion Commands

Command	Description

`neighbor` *PEER* `strict-capability-match`
 	
`no neighbor` *PEER* `strict-capability-match`
	Strictly compare `remote capability` and `local capability`. If `capability` is different, send an `unsupported capability` error and reset the connection.
	
	Users may want to disable sending `capability negotiation OPEN message optional` parameter to the peer when remote peer does not implement `capability negotiation`. Use the `dont-capability-negotiate` command to disable this feature.

`neighbor` *PEER* `dont-capability-negotiate`
 	
`no neighbor` *PEER* `dont-capability-negotiate`
	Suppresses sending `capability negotiation` as `OPEN message optional` parameter to the peer. This command only affects the peer if it is configured for IPv4 unicast configuration.
	
	If the remote peer does not have the `capability negotiation` feature, the remote peer does not send `capability`. In this case, BGP configures the peer with `configured capability`.
	
	Users may prefer a locally configured `capability` more than negotiated `capability`, even if a remote peer sends `capability`. If the peer is configured by `override-capability`, `bgpd` ignores the received `capability`. It then overrides `negotiate capability` with the configured value.

`neighbor` *PEER* `dont-capability-negotiate`
 	
`no neighbor` *PEER* `dont-capability-negotiate`
	Override the result of `capability negotiate` with local configuration. Ignore remote peer's `capability` value.

vtysh

`vtysh` is an integrated command shell of the Zebra software. For information on specific `vtysh` commands, please see Appendix C, “VTY Key Index”.

CHAPTER 3 *Additional Features*

Zebra Filtering

Zebra provides several flexible filtering features. Filtering is used for both input and output of the routing information. Once filtering is defined, it can be applied in any direction.

ip access-list Commands

The following table describes commands used in configuring IP access lists:

Table 3-1: ip access-list Commands

Command	Description

`access-list` *`NAME`* `permit` *`IPV4-NETWORK`*
 	
`access-list` *`NAME`* `deny` *`IPV4-NETWORK`*
	Basic filtering is created with an `access-list` as shown in the following example.
	`access-list filter deny 10.0.0.0/9`
	`access-list filter permit 10.0.0.0/8`

ip prefix-list Commands

`ip prefix-list` provides a powerful prefix-based filtering mechanism. Adding to `access-list` functionality, `ip prefix-list` has a prefix length range specification and sequential number specification. Users can add or delete a prefix-based filter to an arbitrary point of `prefix-list` using sequential numbers.

If no `ip prefix-list` is specified, it acts as a permit. Once the `ip prefix-list` is defined, then no match is performed and the default `deny` is applied.

Table 3-2: ip prefix-list Commands

Command	Description
`ip prefix-list` *NAME* (*permit*|*deny*) *PREFIX* [le *LEN*] [ge *LEN*] `ip prefix-list` *NAME* `seq` *NUMBER* (*permit*|*deny*) [le *LEN*] [ge *LEN*]	`ip prefix-list` are created with these commands.

Table 3-3: ip prefix-list Command Descriptions

`seq`	`seq NUMBER` can be set either automatically or manually. In the case that sequential numbers are set manually, the user may pick any number less than 4294967295. In the case that sequential numbers are set automatically, the sequential numbers increase by a factor of five (5) per list. If a list with no specified sequential number is created after a list with a specified sequential number, the list automatically picks the next multiple of five (5) as the list number. For example, if a list with number 2 already exists and a new list with no specified number is created, the next list will be numbered 5. If lists 2 and 7 already exist and a new list with no specified number is created, the new list will be numbered 10.
`le`	`The le option` specifies a "lesser than" `prefix` length. Used in conjunction with `ge`, `le` is used to determine a range of values. The `prefix` list is applied if the `prefix` length is less than or equal to the `le prefix` length.
`ge`	The `ge` command specifies a "greater than" `prefix` length. Used in conjunction with `le`, `ge` is used to determine a range of values. The `prefix` list is applied if the `prefix` length is greater than or equal to the `ge prefix` length.

Lesser than or equal to `prefix` numbers and greater than or equal to `prefix` numbers can be used together. The order of the `le` and `ge` commands does not matter.

If a `prefix list` is created with a different sequential number, but with the exact same rules as a previous list, an error results. However, no error results if sequential number and the rules are exactly the same.

If a list with the same sequential number as a previous list is created, the new list overwrites the old.

Matching of `ip prefix` is performed from the smaller sequential number to the larger. The matching stops once any rule is applied.

In the case of `no le` or `ge` command, the `prefix` length must exactly match the length specified in the `prefix-list`.

Table 3-4: no ip prefix-list Command

Command
`no ip prefix-list NAME`

ip prefix-list Description Commands

The following table describes the commands used in setting `ip prefix-list` descriptions:

Table 3-5: ip prefix-list Description Commands

Command	Description
`ip prefix-list NAME description DESC`	This command adds a description to the prefix list.
`no ip prefix-list NAME description DESC`	Deletes the description from a prefix list. It is possible to use the command without a description.

ip prefix-list Sequential Number Control Commands

The following table describes commands used in setting `ip prefix-list` sequential number control:

Table 3-6: ip prefix-list Sequential Number Control Commands

Command	Description
`ip prefix-list sequence-number`	With this command, the `ip prefix list` sequential number is displayed. This is the default behavior.
`no ip prefix-list sequence-number`	With this command, the `ip prefix list` sequential number is not displayed.

ip prefix-list Display Commands

The following table describes commands used to display the `ip prefix-lists`:

Table 3-7: ip prefix-list Display Commands

Command	Description
`show ip prefix-list`	Display all `ip prefix-lists`.
`show ip prefix-list` *`NAME`*	Show `ip prefix-list`, can be used with a prefix list name.
`show ip prefix-list` *`NAME`* `seq` *`NUM`*	Show `ip prefix-list`, can be used with a prefix list name and sequential number.
`show ip prefix-list` *`NAME A.B.C.D/M`*	If the command `longer` is used, all prefix lists with prefix lengths equal to or longer than the specified length are displayed. If the command `first match` is used, the first prefix length match is displayed.
`show ip prefix-list` *`NAME A.B.C.D/M`* `longer`	Display all ip prefix-list entries more specific than *`NAME`*.
`show ip prefix-list` *`NAME A.B.C.D/M`* `fisrt-match`	Display first ip prefix-list entry matching *`NAME`*.
`show ip prefix-list summary`	Display summary of `ip prefix-list`.
`show ip prefix-list summary` *`NAME`*	Display summary of `ip prefix-list` *`NAME`*.
`show ip prefix-list detail`	Display detailed `ip prefix-list`.
`show ip prefix-list detail` *`NAME`*	Display detailed `ip prefix-list` entry for *`NAME`*.

clear ip prefix-list Counter Commands

The following table describes commands used in clearing `ip prefix-list` counters:

Table 3-8: clear ip prefix-list Counter Commands

Command	Description
`clear ip prefix-list`	Clears the counters of all `ip prefix-list`. `clear ip prefix-list` can be used with a specified name and prefix.
`clear ip prefix-list` *`NAME`*	Clears the counter of the *`NAME`* `prefix-list`
`clear ip prefix-list` *`NAME A.B.C.D/M`*	Clears the counter of the *`NAME`* in a specific `prefix-list` and IP address

ip community-list Command

The following table describes the command used in setting `ip community list`:

Table 3-9: ip community list Command

Command
`ip community-list` *`NAME TYPE COMMUNITY`*

as-path access-list Command

The following table shows the command used in setting `as-path access-list`:

Table 3-10: as-path access-list Command

Command
`ip as-path access-list` *`NAME TYPE AS_PATH`*

route-map

`route-map` is a useful function in Zebra. There is a `match` and `set` statement used to define `route-map`.

```
route-map test permit 10
match ip address 10
set local-preference 200
```

In this example, the route `match ip access-list` number is defined as 10 and the `local-preference` value is set to 200.

route-map Command

The following table shows the command used in setting `route-map` commands:

Table 3-11: route-map Command

Command
`route-map` *`ROUTE-MAP-NAME`* `permit` *`PRIORITY`*

route-map match Command

The following table describes commands used in matching route maps:

Table 3-12: route-map match Command

Command	Description
`match ip address` *`ACCESS_LIST`*	Match the specified *`ACCESS_LIST`*
`match ip next-hop` *`IPV4_ADDR`*	Match the specified *`IPV4_ADDR`*
`match aspath` *`AS_PATH`*	Match the specified *`AS_PATH`*
`match metric` *`METRIC`*	Match the specified *`METRIC`*
`match community` *`COMMUNITY_LIST`*	Match the specified *`COMMUNITY_LIST`*

route-map set Command

The following table describes commands used in setting `route-map`:

Table 3-13: route-map set Command

Command	Description
`set ip next-hop` *`IPV4_ADDRESS`*	Set the BGP `next-hop` address
`set local-preference` *`LOCAL_PREF`*	Set the BGP `local-preference`.
`set weight` *`WEIGHT`*	Set the route's *`WEIGHT`*
`set metric` *`METRIC`*	Set the BGP attribute *`METRIC`*
`set aspath prepend` *`AS_PATH`*	Set the BGP *`AS_PATH`* to prepend
`set community` *`COMMUNITY`*	Set the BGP *`COMMUNITY`* attribute
`set ipv6 next-hop global` *`IPV6_ADDRESS`*	Set the BGP-4+ global IPv6 nexthop address.
`set ipv6 next-hop local` *`IPV6_ADDRESS`*	Set the BGP-4+ link local IPv6 nexthop address.

Kernel Interface

There are two different methods for reading kernel routing table information:

- updating kernel routing tables
- looking up interfaces.

Several methods are available for obtaining kernel information:

- `ioctl`

  The `ioctl` method is a traditional way for reading or writing kernel information. `ioctl` can be used to look up interfaces and for modify interface addresses, flags, `mtu` settings and other types of information. Also, `ioctl` can insert and delete kernel routing table entries.

- `sysctl`

  `sysctl` can lookup kernel information using MIB (Management Information Base) syntax. Normally, it only provides a way of getting information from the kernel. So, it is preferred to change kernel information using another method, such as `ioctl`.

- `proc` filesystem

  `proc` filesystem provides an easy way of obtaining kernel information.

- routing socket

  LynxOS uses this method by default.

- netlink

SNMP Support

SNMP (Simple Network Managing Protocol) is a widely implemented feature for collecting network information from routers and hosts. Zebra itself does not support SNMP functionality. However, in conjunction with an SNMP agent, Zebra provides routing protocol for MIBs.

Zebra uses the SMUX protocol (RFC1227) to communicate with the SNMP agent. There are several SNMP agents which support SMUX. It is recommended to use the latest net-SNMP software. Zebra is tested with `ucd-snmp-4.1.pre1.tar.gz`.

After installing net-snmp, `smuxpeer` must be configured. Refer to the following sample configuration file:

```
/usr/local/share/snmp/snmpd.conf
================================
smuxpeer 1.3.6.1.6.3.1 test
```

SMUX Commands

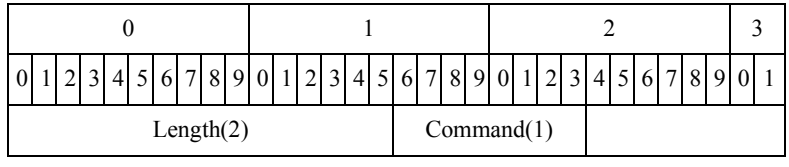
The following describes commands used to configure `smux`.

Table 3-14: SMUX commands

Command	Description
`smux peer` *`OID`* `no smux peer` *`OID`*	Sets SMUX peer OID (Object ID)
`smux peer` *`OID PASSWORD`* `no smux peer` *`OID PASSWORD`*	! smux peer .1.3.6.1.6.3.1 test !

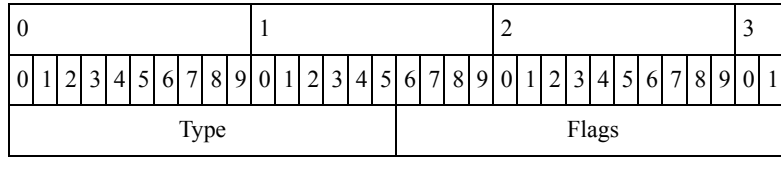
APPENDIX A *Zebra Protocol*

The Zebra Protocol is used to link a protocol daemon and `zebra`. Each protocol daemon sends selected routes to the `zebra` daemon. The `zebra` daemon manages which route is installed into a forwarding table. Below is a common header of the Zebra Protocol.



Length is the total packet length, including this header length, so the minimum length is three. The command is `Zebra Protocol`.

```
ZEBRA_INTERFACE_ADD                1
ZEBRA_INTERFACE_DELETE             2
ZEBRA_INTERFACE_ADDRESS_ADD        3
ZEBRA_INTERFACE_ADDRESS_DELETE     4
ZEBRA_INTERFACE_UP                 5
ZEBRA_INTERFACE_DOWN               6
ZEBRA_IPV4_ROUTE_ADD               7
ZEBRA_IPV4_ROUTE_DELETE            8
ZEBRA_IPV6_ROUTE_ADD               9
ZEBRA_IPV6_ROUTE_DELETE           10
ZEBRA_REDISTRIBUTE_ADD            11
ZEBRA_REDISTRIBUTE_DELETE         12
ZEBRA_REDISTRIBUTE_DEFAULT_ADD    13
ZEBRA_REDISTRIBUTE_DEFAULT_DELETE 14
ZEBRA_IPV4_NEXTHOP_LOOKUP         15
ZEBRA_IPV6_NEXTHOP_LOOKUP         16
```

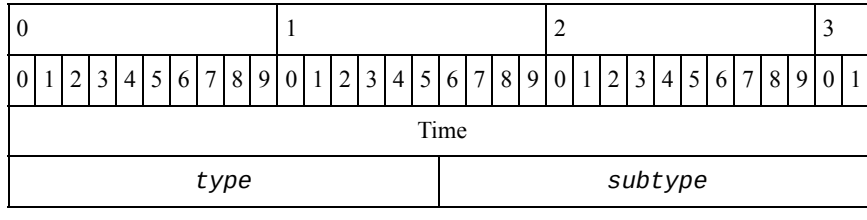


APPENDIX B *Packet Binary Dump Format*

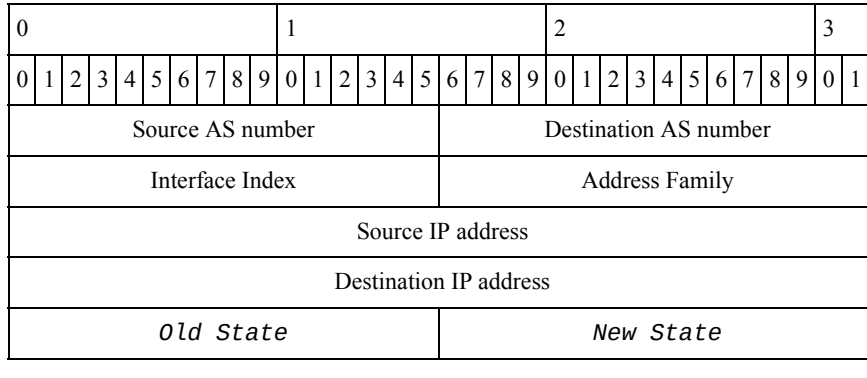
Zebra can dump routing protocol packets into files with binary format (also see “Dump BGP Packet and Table Commands” on page 55).

The MRT header format is used for backwards compatibility with the MRT dump logs. The binary format should also be defined to support IPv4 addresses as socket addresses and / or routing entries.

This is the common header format, the same as that of MRT.

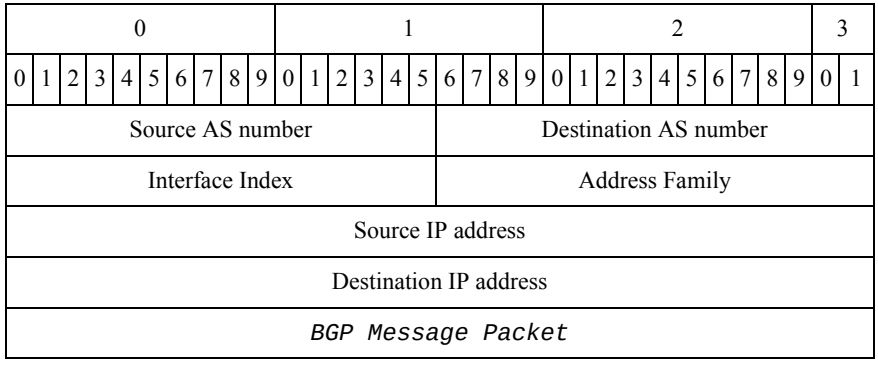


If *type* is `PROTOCOL_BGP4MP`, *subtype* is `BGP4MP_STATE_CHANGE`, and `Address Family == IP (version 4)`



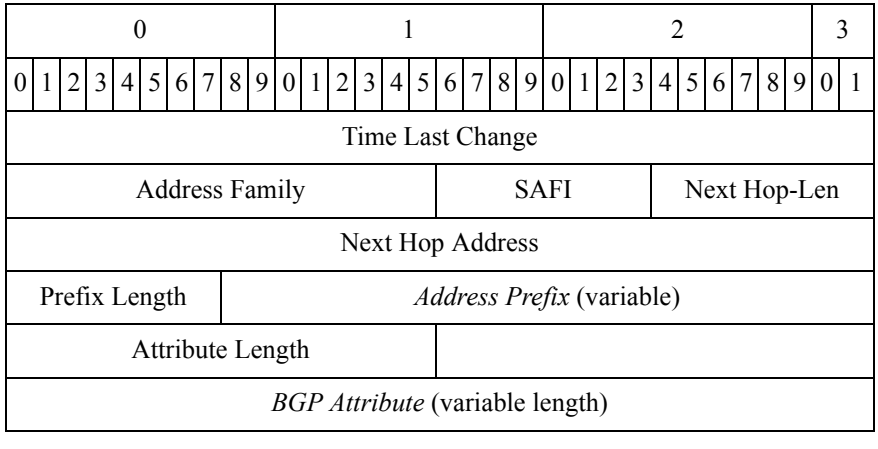
Where *State* is the value defined in RFC1771.

If *type* is `PROTOCOL_BGP4MP`, *subtype* is `BGP4MP_MESSAGE`, and `Address Family == IP (version 4)`.



Where *BGP Message Packet* is the whole contents of the BGP4 message including header portion.

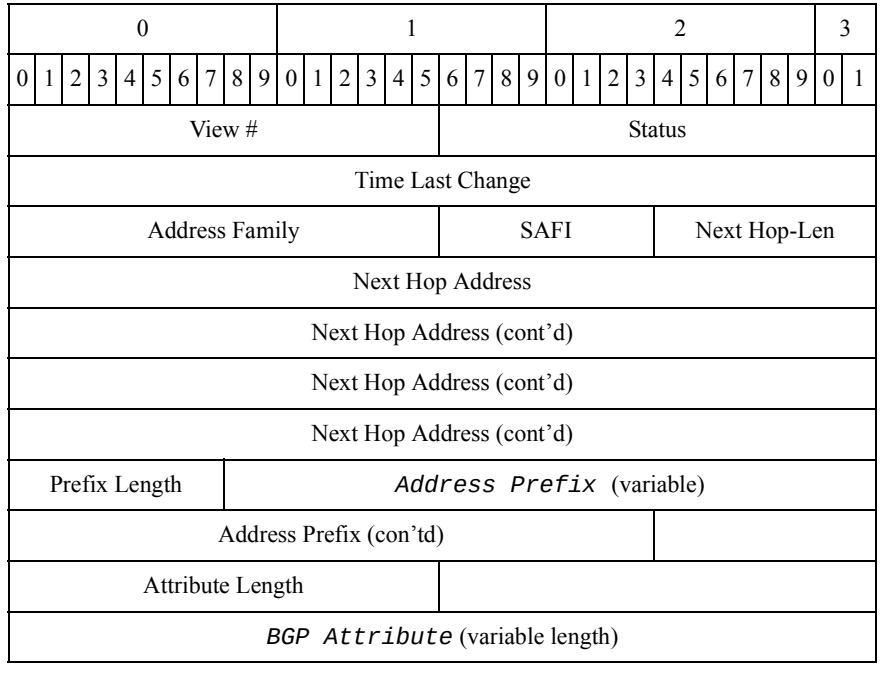
If *type* is `PROTOCOL_BGP4MP`, *subtype* is `BGP4MP_ENTRY`, and `Address Family == IP (version 4)`.



Where *BGP Message Packet* is the whole contents of the BGP4 message, including header portion.

If *type* is `PROTOCOL_BGP4MP`, *subtype* is `BGP4MP_MESSAGE`, and `Address Family == IP version 6`

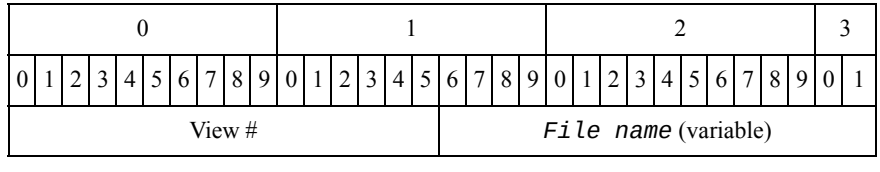
BGP4 Attribute must not contain `MP_UNREACH_NLRI`



.

If *BGP Attribute* has MP_REACH_NLRI field, it must have zero length NLRI, for example, MP_REACH_NLRI has only *Address Family*, *SAFI* and *next-hop* values.

If *type* is PROTOCOL_BGP4MP and *subtype* is BGP4MP_SNAPSHOT,



The file specified in *File Name* contains all routing entries, which are in the format of subtype == BGP4MP_ENTRY.

Constants:

```
/* type value */
      #define MSG_PROTOCOL_BGP4MP 16
      /* subtype value */
      #define BGP4MP_STATE_CHANGE 0
      #define BGP4MP_MESSAGE 1
      #define BGP4MP_ENTRY 2
      #define BGP4MP_SNAPSHOT 3
```

If type is PROTOCOL_BGP4MP, subtype is BGP4MP_STATE_CHANGE, and Address Family == IP version 6

0										1										2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Source AS number										Destination AS number																			
Interface Index										Address Family																			
Source IP address																													
Source IP address (Cont'd)																													
Source IP address (Cont'd)																													
Source IP address (Cont'd)																													
Destination IP address																													
Destination IP address (Cont'd)																													
Destination IP address (Cont'd)																													
Destination IP address (Cont'd)																													
Old State										New State																			

If type is `PROTOCOL_BGP4MP`, subtype is `BGP4MP_MESSAGE`, and `Address Family == IP version 6`

0										1										2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Source AS number										Destination AS number																			
Interface Index										Address Family																			
Source IP address																													

Source IP address (Cont’d)
Source IP address (Cont’d)
Source IP address (Cont’d)
Destination IP address
Destination IP address (Cont’d)
Destination IP address (Cont’d)
Destination IP address (Cont’d)
BGP Message Packet

APPENDIX C *VTY Key Index*

The following table details the keys used in the VTY interface.

NOTE: The Emacs **Meta** key varies on different systems with different keyboards. Typically, **Meta** is configured to the **ALT** key, however it can also be configured to **ESC**.

Table D- 1: VTY Key Index

Menu	
DEL	CLI Editing Commands.
Down Arrow	CLI Advanced Commands.
Left Arrow	CLI Movement Commands.
Right Arrow	CLI Movement Commands.
Tab	CLI Advanced Commands.
Up Arrow	CLI Advanced Commands.
?	CLI Advanced Commands.
Ctrl-A	CLI Movement Commands.
Ctrl-B	CLI Movement Commands.
Ctrl-C	CLI Advanced Commands.
Ctrl-D	CLI Editing Commands.
Ctrl-E	CLI Movement Commands.

Table D- 1: VTY Key Index (Continued)

	Menu
Ctrl-F	CLI Movement Commands.
Ctrl-G	CLI Editing Commands.
Ctrl-K	CLI Editing Commands.
Ctrl-N	CLI Advanced Commands.
Ctrl-P	CLI Advanced Commands.
Ctrl-T	CLI Editing Commands.
Ctrl-U	CLI Editing Commands.
Ctrl-W	CLI Editing Commands.
Ctrl-Z	CLI Advanced Commands.
Meta-B	CLI Movement Commands.
Meta-D	CLI Editing Commands.
Meta-F	CLI Movement Commands.

APPENDIX D *Command Index*

Table C - 1: Command Index

Command	Description
`access-class` *`ACCESS-LIST`*	Basic Config Commands
`access-list` *`NAME`* `deny` *`IPV4-NETWORK`*	IP Access List
`access-list` *`NAME`* `permit` *`IPV4-NETWORK`*	IP Access List
`aggregate-address` *`NETWORK`*	BGP network
`area` *`0-4294967295`* `authentication`	OSPF area
`area` *`0-4294967295`* `authentication message-digest`	OSPF area
`area` *`0-4294967295`* `export-list` *`NAME`*	OSPF area
`area` *`0-4294967295`* `import-list` *`NAME`*	OSPF area
`area` *`0-4294967295`* `range` *`A.B.C.D/M`*	OSPF area
`area` *`0-4294967295`* `shortcut`	OSPF area
`area` *`0-4294967295`* `stub`	OSPF area
`area` *`0-4294967295`* `stub no-summary`	OSPF area
`area` *`0-4294967295`* `virtual-link` *`A.B.C.D`*	OSPF area
`area` *`A.B.C.D`* `authentication`	OSPF area
`area` *`A.B.C.D`* `authentication message-digest`	OSPF area
`area` *`A.B.C.D`* `default-cost` *`0-16777215`*	OSPF area

Table C - 1: Command Index (Continued)

Command	Description
area *A.B.C.D* export-list *NAME*	OSPF area
area *A.B.C.D* import-list *NAME*	OSPF area
area *A.B.C.D* range *A.B.C.D/M*	OSPF area
area *A.B.C.D* range *IPV4_PREFIX* substitute *IPV4_PREFIX*	OSPF area
area *A.B.C.D* range *IPV4_PREFIX* suppress	OSPF area
area *A.B.C.D* shortcut	OSPF area
area *A.B.C.D* stub	OSPF area
area *A.B.C.D* stub no-summary	OSPF area
area *A.B.C.D* virtual-link *A.B.C.D*	OSPF area
auto-cost refrence-bandwidth *1-4294967*	OSPF router
bandwidth *1-10000000*	Interface Commands
banner motd default	Basic Config Commands
bgp cluster-id *A.B.C.D*	Route Reflector
bgp multiple-instance	Multiple instance
bgp router-id *ROUTER-ID*	BGP router
clear ip bgp *PEER*	BGP terminal mode commands
clear ip bgp *PEER* soft in	BGP terminal mode commands
clear ip prefix-list	Clear counter of ip prefix-list
clear ip prefix-list *NAME*	Clear counter of ip prefix-list
clear ip prefix-list *NAME A.B.C.D/M*	Clear counter of ip prefix-list
configure terminal	Basic Config Commands
debug event	BGP terminal mode commands

Table C - 1: Command Index (Continued)

Command	Description
`debug keepalive`	BGP terminal mode commands
`debug ospf ism`	Debugging OSPF
`debug ospf ism (status|events|timers)`	Debugging OSPF
`debug ospf lsa`	Debugging OSPF
`debug ospf lsa (generate|flooding|refresh)`	Debugging OSPF
`debug ospf nsm`	Debugging OSPF
`debug ospf nsm (status|events|timers)`	Debugging OSPF
`debug ospf packet (hello|dd|ls-request|ls-update|ls-ack|all) (send|recv) [detail]`	Debugging OSPF
`debug ospf zebra`	Debugging OSPF
`debug ospf zebra (interface|redistribute)`	Debugging OSPF
`debug rip events`	RIP Debug Commands
`debug rip packet`	RIP Debug Commands
`debug rip zebra`	RIP Debug Commands
`debug ripng events`	`ripngd` Terminal Mode Commands
`debug ripng packet`	`ripngd` Terminal Mode Commands
`debug ripng zebra`	`ripngd` Terminal Mode Commands
`debug update`	BGP terminal mode commands
`default-information originate` *`1`*	Redistribute routes to OSPF
`default-information originate`	How to Announce RIP route
`default-information originate always`	Redistribute routes to OSPF
`default-information originate always metric` *`0-16777214`*	Redistribute routes to OSPF

Table C - 1: Command Index (Continued)

Command	Description
`default-information originate always metric 0-16777214 metric-type (1|2)`	Redistribute routes to OSPF
`default-information originate always metric 0-16777214 metric-type (1|2) route-map WORD`	Redistribute routes to OSPF
`default-information originate metric 0-16777214`	Redistribute routes to OSPF
`default-information originate metric 0-16777214 metric-type (1|2)`	Redistribute routes to OSPF
`default-information originate metric 0-16777214 metric-type (1|2) route-map WORD`	Redistribute routes to OSPF
`default-metric 0-16777214`	Redistribute routes to OSPF
`default-metric 1-16`	RIP Metric Manipulation
`description DESCRIPTION ...`	Interface Commands
`distance 1-255 1`	Redistribute routes to OSPF
`distance 1-255`	RIP distance
`distance 1-255 A.B.C.D/M`	RIP distance
`distance 1-255 A.B.C.D/M ACCESS-LIST`	RIP distance
`distance ospf (intra-area|inter-area|external) 1-255`	Redistribute routes to OSPF
`distribute-list ACCESS_LIST (in|out) IFNAME`	`ripngd` Filtering Commands
`distribute-list ACCESS_LIST DIRECT IFNAME`	Filtering RIP Routes
`distribute-list NAME out (kernel|connected|static|rip|ospf`	Redistribute routes to OSPF
`distribute-list prefix PREFIX_LIST (in|out) IFNAME`	Filtering RIP Routes
`dump bgp all PATH`	Dump BGP packet and table
`dump bgp all PATH INTERVAL`	Dump BGP packet and table

Table C - 1: Command Index (Continued)

Command	Description
`dump bgp routes` *PATH*	Dump BGP packet and table
`dump bgp updates` *PATH*	Dump BGP packet and table
`dump bgp updates` *PATH INTERVAL*	Dump BGP packet and table
`enable password` *PASSWORD*	Basic Config Commands
`exec-timeout` *MINUTE*	Basic Config Commands
`exec-timeout` *MINUTE SECOND*	Basic Config Commands
`flush_timer` *TIME*	`ripngd` Configuration
`hostname` *HOSTNAME*	Basic Config Commands
`interface` *IFNAME*	Interface Commands
`interface` *IFNAME* `area` *AREA*	OSPF6 router
`ip address` *ADDRESS*	Interface Commands
`ip as-path access-list` *NAME TYPE AS_PATH*	AS Path Access List
`ip community-list` *NAME TYPE COMMUNITY*	IP Community List
`ip ospf authentication-key` *AUTH_KEY*	OSPF interface
`ip ospf cost` *1-65535*	OSPF interface
`ip ospf dead-interval` *1-65535*	OSPF interface
`ip ospf hello-interval` *1-65535*	OSPF interface
`ip ospf message-digest-key` *KEYID* `md5` *KEY*	OSPF interface
`ip ospf network (broadcast|non-broadcast|point-to-multipoint|point-to-point)`	OSPF interface
`ip ospf priority` *0-255*	OSPF interface
`ip ospf retransmit-interval` *1-65535*	OSPF interface
`ip ospf transmit-delay`	OSPF interface
`ip prefix-list` *NAME* `(permit|deny)` *PREFIX* `/ [le` *LEN*`] [ge` *LEN*`]`	IP Prefix List

Table C - 1: Command Index (Continued)

Command	Description
`ip prefix-list NAME description DESC`	`ip prefix-list` description
`ip prefix-list NAME seq NUMBER (permit|deny) PREFIX [le LEN] [ge LEN]`	IP Prefix List
`ip prefix-list sequence-number`	`ip prefix-list` sequential number control
`ip rip authentication key-chain KEY-CHAIN`	RIP Authentication
`ip rip authentication mode md5`	RIP Authentication
`ip rip authentication mode text`	RIP Authentication
`ip rip authentication string STRING`	RIP Authentication
`ip rip receive version VERSION`	RIP Configuration
`ip rip send version VERSION`	RIP Configuration
`ip route NETWORK GATEWAY`	Static Route Commands
`ip split-horizon`	RIP Configuration
`ipv6 nd prefix-advertisement IPV6PREFIX`	Router Advertisement
`ipv6 nd send-ra`	Router Advertisement
`ipv6 ospf6 cost COST`	OSPF6 interface
`ipv6 ospf6 dead-interval DEADINTERVAL`	OSPF6 interface
`ipv6 ospf6 hello-interval HELLOINTERVAL`	OSPF6 interface
`ipv6 ospf6 priority PRIORITY`	OSPF6 interface
`ipv6 ospf6 retransmit-interval RETRANSMITINTERVAL`	OSPF6 interface
`ipv6 ospf6 transmit-delay TRANSMITDELAY`	OSPF6 interface
`ipv6 route NETWORK GATEWAY`	Static Route Commands
`line vty`	Basic Config Commands
`list`	Basic Config Commands

Table C - 1: Command Index (Continued)

Command	Description
`log file FILENAME`	Basic Config Commands
`log stdout`	Basic Config Commands
`log syslog`	Basic Config Commands
`match aspath AS_PATH`	Route Map Match Command
`match community COMMUNITY_LIST`	Route Map Match Command
`match interface NAME`	RIP route-map
`match ip address`	RIP route-map
`match ip address ACCESS_LIST`	Route Map Match Command
`match ip next-hop IPV4_ADDR`	Route Map Match Command
`match ip next-hot A.B.C.D`	RIP route-map
`match metric METRIC`	Route Map Match Command
`match metric N`	RIP route-map
`multicast`	Interface Commands
`neighbor A.B.C.D`	RIP Configuration
`neighbor PEER default-originate`	Peer configuration
`neighbor PEER description ...`	Peer configuration
`neighbor PEER distribute-list NAME [in|out]`	Peer filtering
`neighbor PEER dont-capability-negotiate`	Multiple Protocol Extension for BGP
`neighbor PEER ebgp-multihop`	Peer configuration
`neighbor PEER filter-list NAME [in|out]`	Peer filtering
`neighbor PEER interface IFNAME`	Peer configuration
`neighbor PEER maximum-prefix NUMBER`	Peer configuration
`neighbor PEER next-hop-self`	Peer configuration

Table C - 1: Command Index (Continued)

Command	Description
`neighbor PEER override-capability`	Multiple Protocol Extension for BGP
`neighbor PEER port PORT`	Peer configuration
`neighbor PEER prefix-list NAME [in|out]`	Peer filtering
`neighbor PEER remote-as AS-NUMBER`	BGP peer
`neighbor PEER route-map NAME [in|out]`	Peer filtering
`neighbor PEER route-reflector-client`	Route Reflector
`neighbor PEER send-community`	Peer configuration
`neighbor PEER shutdown`	Peer configuration
`neighbor PEER strict-capability-match`	Multiple Protocol Extension for BGP
`neighbor PEER update-source`	Peer configuration
`neighbor PEER version VERSION`	Peer configuration
`neighbor PEER weight WEIGHT`	Peer configuration
`network A.B.C.D/M area 0-4294967295`	OSPF router
`network A.B.C.D/M area A.B.C.D`	OSPF router
`network IFNAME 1`	`ripngd` Configuration
`network IFNAME`	RIP Configuration
`network NETWORK 1`	BGP network
`network NETWORK 2`	`ripngd` Configuration
`network NETWORK`	RIP Configuration
`no aggregate-address NETWORK`	BGP network
`no area 0-4294967295 authentication`	OSPF area
`no area 0-4294967295 export-list NAME`	OSPF area
`no area 0-4294967295 import-list NAME`	OSPF area

Table C - 1: Command Index (Continued)

Command	Description
no area *0-4294967295* range *A.B.C.D/M*	OSPF area
no area *0-4294967295* shortcut	OSPF area
no area *0-4294967295* stub	OSPF area
no area *0-4294967295* stub no-summary	OSPF area
no area *0-4294967295* virtual-link *A.B.C.D*	OSPF area
no area *0-4294967295* authentication	OSPF area
no area *A.B.C.D* default-cost *0-16777215*	OSPF area
no area *A.B.C.D* export-list *NAME*	OSPF area
no area *A.B.C.D* import-list *NAME*	OSPF area
no area *A.B.C.D* range *A.B.C.D/M*	OSPF area
no area *A.B.C.D* range *IPV4_PREFIX* substitute *IPV4_PREFIX*	OSPF area
no area *A.B.C.D* range *IPV4_PREFIX* suppress	OSPF area
no area *A.B.C.D* shortcut	OSPF area
no area *A.B.C.D* stub	OSPF area
no area *A.B.C.D* stub no-summary	OSPF area
no area *A.B.C.D* virtual-link *A.B.C.D*	OSPF area
no auto-cost refrence-bandwidth	OSPF router
no bandwidth *1-10000000*	Interface Commands
no banner motd	Basic Config Commands
no bgp multiple-instance	Multiple instance
no debug event	BGP terminal mode commands
no debug keepalive	BGP terminal mode commands
no debug ospf ism	Debugging OSPF

Table C - 1: Command Index (Continued)

Command	Description
`no debug ospf ism (status|events|timers)`	Debugging OSPF
`no debug ospf lsa`	Debugging OSPF
`no debug ospf lsa (generate|flooding|refresh)`	Debugging OSPF
`no debug ospf nsm`	Debugging OSPF
`no debug ospf nsm (status|events|timers)`	Debugging OSPF
`no debug ospf packet (hello|dd|ls-request|ls-update|ls-ack|all) (send|recv) [detail]`	Debugging OSPF
`no debug ospf zebra`	Debugging OSPF
`no debug ospf zebra (interface|redistribute)`	Debugging OSPF
`no debug update`	BGP terminal mode commands
`no default-information originate`	Redistribute routes to OSPF
`no default-metric`	Redistribute routes to OSPF
`no default-metric` *`1-16`*	RIP Metric Manipulation
`no distance` *`1-255 1`*	Redistribute routes to OSPF
`no distance` *`1-255`*	RIP distance
`no distance` *`1-255 A.B.C.D/M`*	RIP distance
`no distance` *`1-255 A.B.C.D/M ACCESS-LIST`*	RIP distance
`no distance ospf`	Redistribute routes to OSPF
`no distribute-list` *`NAME`* `out (kernel|connected|static|rip|ospf`	Redistribute routes to OSPF
`no exec-timeout`	Basic Config Commands
`no ip ospf authentication-key`	OSPF interface
`no ip ospf cost`	OSPF interface
`no ip ospf dead-interval`	OSPF interface
`no ip ospf hello-interval`	OSPF interface

Table C - 1: Command Index (Continued)

Command	Description
`no ip ospf message-digest-key`	OSPF interface
`no ip ospf network`	OSPF interface
`no ip ospf priority`	OSPF interface
`no ip ospf retransmit interval`	OSPF interface
`no ip ospf transmit-delay`	OSPF interface
`no ip prefix-list` *`NAME`*	IP Prefix List
`no ip prefix-list` *`NAME`* `description` *`DESC`*	`ip prefix-list` description
`no ip prefix-list sequence-number`	`ip prefix-list` sequential number control
`no ip rip authentication key-chain` *`KEY-CHAIN`*	RIP Authentication
`no ip rip authentication mode md5`	RIP Authentication
`no ip rip authentication mode text`	RIP Authentication
`no ip rip authentication string` *`STRING`*	RIP Authentication
`no ip split-horizon`	RIP Configuration
`no log stdout`	Basic Config Commands
`no log syslog`	Basic Config Commands
`no multicast`	Interface Commands
`no neighbor` *`A.B.C.D`*	RIP Configuration
`no neighbor` *`PEER`* `default-originate`	Peer configuration
`no neighbor` *`PEER`* `description ...`	Peer configuration
`no neighbor` *`PEER`* `dont-capability-negotiate`	Multiple Protocol Extension for BGP
`no neighbor` *`PEER`* `ebgp-multihop`	Peer configuration
`no neighbor` *`PEER`* `interface` *`IFNAME`*	Peer configuration

Table C - 1: Command Index (Continued)

Command	Description
`no neighbor PEER maximum-prefix NUMBER`	Peer configuration
`no neighbor PEER next-hop-self`	Peer configuration
`no neighbor PEER override-capability`	Multiple Protocol Extension for BGP
`no neighbor PEER route-reflector-client`	Route Reflector
`no neighbor PEER shutdown`	Peer configuration
`no neighbor PEER strict-capability-match`	Multiple Protocol Extension for BGP
`no neighbor PEER update-source`	Peer configuration
`no neighbor PEER weight WEIGHT`	Peer configuration
`no network A.B.C.D/M area 0-4294967295`	OSPF router
`no network A.B.C.D/M area A.B.C.D`	OSPF router
`no network IFNAME`	RIP Configuration
`no network NETWORK 1`	BGP network
`no network NETWORK`	RIP Configuration
`no ospf abr-type TYPE`	OSPF router
`no ospf rfc1583compatibility`	OSPF router
`no ospf router-id`	OSPF router
`no passive interface INTERFACE`	OSPF router
`no passive-interface IFNAME`	RIP Configuration
`no redistribute (kernel|connected|static|rip|bgp)`	Redistribute routes to OSPF
`no redistribute bgp`	How to Announce RIP route
`no redistribute connected`	How to Announce RIP route
`no redistribute kernel`	How to Announce RIP route

Table C - 1: Command Index (Continued)

Command	Description
`no redistribute ospf`	How to Announce RIP route
`no redistribute static`	How to Announce RIP route
`no rouer rip`	RIP Configuration
`no route` *`A.B.C.D/M`*	How to Announce RIP route
`no router bgp` *`AS-NUMBER`*	BGP router
`no router ospf`	OSPF router
`no router zebra`	Redistribute routes to OSPF
`no shutdown`	Interface Commands
`no smux peer` *`OID`*	SMUX configuration
`no smux peer` *`OID PASSWORD`*	SMUX configuration
`no timers basic`	RIP Timers
`no timers spf`	OSPF router
`offset-list` *`ACCESS-LIST`* `(in|out)`	RIP Metric Manipulation
`offset-list` *`ACCESS-LIST`* `(in|out) IFNAME`	RIP Metric Manipulation
`ospf abr-type` *`TYPE`*	OSPF router
`ospf rfc1583compatibility`	OSPF router
`ospf router-id` *`A.B.C.D`*	OSPF router
`passive interface` *`INTERFACE`*	OSPF router
`passive-interface` *`IFNAME`*	RIP Configuration
`password` *`PASSWORD`*	Basic Config Commands
`redistribute (kernel|connected|static|rip|bgp)`	Redistribute routes to OSPF
`redistribute (kernel|connected|static|rip|bgp) metric` *`0-16777214`*	Redistribute routes to OSPF

Table C - 1: Command Index (Continued)

Command	Description
`redistribute (kernel|connected|static|rip|bgp) metric *0-16777214* route-map *WORD*`	Redistribute routes to OSPF
`redistribute (kernel|connected|static|rip|bgp) metric-type (1|2)`	Redistribute routes to OSPF
`redistribute (kernel|connected|static|rip|bgp) metric-type (1|2) metric *0-16777214*`	Redistribute routes to OSPF
`redistribute (kernel|connected|static|rip|bgp) metric-type (1|2) metric *0-16777214* route-map *WORD*`	Redistribute routes to OSPF
`redistribute (kernel|connected|static|rip|bgp) metric-type (1|2) route-map *WORD*`	Redistribute routes to OSPF
`redistribute (kernel|connected|static|rip|bgp) *ROUTE-MAP*`	Redistribute routes to OSPF
`redistribute bgp`	How to Announce RIP route
`redistribute bgp metric *0-16*`	How to Announce RIP route
`redistribute bgp route-map *ROUTE-MAP*`	How to Announce RIP route
`redistribute connected *1*`	Redistribute to BGP
`redistribute connected *2*`	Redistribute routes to OSPF6
`redistribute connected`	How to Announce RIP route
`redistribute connected metric *0-16*`	How to Announce RIP route
`redistribute connected route-map *ROUTE-MAP*`	How to Announce RIP route
`redistribute kernel *1*`	Redistribute to BGP
`redistribute kernel`	How to Announce RIP route
`redistribute kernel metric *0-16*`	How to Announce RIP route
`redistribute kernel route-map *ROUTE-MAP*`	How to Announce RIP route

Table C - 1: Command Index (Continued)

Command	Description
`redistribute ospf 1`	Redistribute to BGP
`redistribute ospf`	How to Announce RIP route
`redistribute ospf metric 0-16`	How to Announce RIP route
`redistribute ospf route-map ROUTE-MAP`	How to Announce RIP route
`redistribute rip`	Redistribute to BGP
`redistribute ripng`	Redistribute routes to OSPF6
`redistribute static 1`	Redistribute to BGP
`redistribute static 2`	Redistribute routes to OSPF6
`redistribute static`	How to Announce RIP route
`redistribute static metric 0-16`	How to Announce RIP route
`redistribute static route-map ROUTE-MAP`	How to Announce RIP route
`refresh age-diff 0-10000`	OSPF router
`refresh group-limit 0-10000`	OSPF router
`refresh per-slice 0-10000`	OSPF router
`route A.B.C.D/M`	How to Announce RIP route
`route NETWORK`	`ripngd` Configuration
`route-map ROUTE-MAP-NAME permit PRIORITY`	Route Map Command
`router bgp AS-NUMBER 1`	BGP instance and view
`router bgp AS-NUMBER`	BGP router
`router bgp AS-NUMBER view NAME`	BGP instance and view
`router ospf`	OSPF router
`router ospf6`	OSPF6 router
`router rip`	RIP Configuration
`router ripng`	`ripngd` Configuration

Table C - 1: Command Index (Continued)

Command	Description
`router zebra <1`	Redistribute routes to OSPF
`router zebra`	`ripngd` Configuration
`router-id A.B.C.D`	OSPF6 router
`service advanced-vty`	Basic Config Commands
`service password-encryption`	Basic Config Commands
`service terminal-length 0-512`	Basic Config Commands
`set as-path prepend AS_PATH`	Route Map Set Command
`set community COMMUNITY`	Route Map Set Command
`set ip next-hop IPV4_ADDRESS`	Route Map Set Command
`set ipv6 next-hop global IPV6_ADDRESS`	Route Map Set Command
`set ipv6 next-hop local IPV6_ADDRESS`	Route Map Set Command
`set local-preference LOCAL_PREF`	Route Map Set Command
`set metric 0-4294967295`	RIP route-map
`set metric METRIC`	Route Map Set Command
`set next-hop A.B.C.D`	RIP route-map
`set weight WEIGHT`	Route Map Set Command
`show debug`	BGP terminal mode commands
`show debugging ospf`	Debugging OSPF
`show debugging rip`	RIP Debug Commands
`show debugging ripng`	`ripngd` Terminal Mode Commands
`show interface`	`zebra` Terminal Mode Commands
`show ip bgp NETWORK`	BGP terminal mode commands
`show ip bgp neighbor PEER`	BGP terminal mode commands

Table C - 1: Command Index (Continued)

Command	Description
`show ip bgp regexp` *AS-REGEXP*	BGP terminal mode commands
`show ip bgp summary`	BGP terminal mode commands
`show ip bgp view` *NAME*	Displaying the BGP view
`show ip ospf`	Showing OSPF information
`show ip ospf database`	Showing OSPF information
`show ip ospf database (asbr-summary|external|network|router|summary)`	Showing OSPF information
`show ip ospf database (asbr-summary|external|network|router|summary) adv-router` *ADV-ROUTER*	Showing OSPF information
`show ip ospf database (asbr-summary|external|network|router|summary)` *LINK-STATE-ID*	Showing OSPF information
`show ip ospf database (asbr-summary|external|network|router|summary)` *LINK-STATE-ID* `adv-router` *ADV-ROUTER*	Showing OSPF information
`show ip ospf database (asbr-summary|external|network|router|summary)` *LINK-STATE-ID* `self-originate`	Showing OSPF information
`show ip ospf database (asbr-summary|external|network|router|summary) self-originate`	Showing OSPF information
`show ip ospf database max-age`	Showing OSPF information
`show ip ospf database self-originate`	Showing OSPF information
`show ip ospf interface` *INTERFACE*	Showing OSPF information
`show ip ospf neighbor`	Showing OSPF information
`show ip ospf neighbor detail`	Showing OSPF information
`show ip ospf neighbor` *INTERFACE*	Showing OSPF information
`show ip ospf neighbor` *INTERFACE* `detail`	Showing OSPF information

Table C - 1: Command Index (Continued)

Command	Description
`show ip ospf refresher`	Showing OSPF information
`show ip ospf route`	Showing OSPF information
`show ip prefix-list`	Showing ip prefix-list
`show ip prefix-list detail`	Showing ip prefix-list
`show ip prefix-list detail` *`NAME`*	Showing ip prefix-list
`show ip prefix-list` *`NAME`*	Showing ip prefix-list
`show ip prefix-list` *`NAME A.B.C.D/M`*	Showing ip prefix-list
`show ip prefix-list` *`NAME A.B.C.D/M`* `first-match`	Showing ip prefix-list
`show ip prefix-list` *`NAME A.B.C.D/M`* `longer`	Showing ip prefix-list
`show ip prefix-list` *`NAME`* `seq` *`NUM`*	Showing ip prefix-list
`show ip prefix-list summary`	Showing ip prefix-list
`show ip prefix-list summary` *`NAME`*	Showing ip prefix-list
`show ip protocols`	Show RIP Information
`show ip rip`	Show RIP Information
`show ip ripng`	`ripngd` Terminal Mode Commands
`show ip route`	`zebra` Terminal Mode Commands
`show ipforward`	`zebra` Terminal Mode Commands
`show ipv6 ospf6` *`INSTANCE_ID`*	Showing OSPF6 information
`show ipv6 ospf6 database`	Showing OSPF6 information
`show ipv6 ospf6 interface`	Showing OSPF6 information
`show ipv6 ospf6 neighbor`	Showing OSPF6 information
`show ipv6 ospf6 request-list A.B.C.D`	Showing OSPF6 information

Table C - 1: Command Index (Continued)

Command	Description
`show ipv6 route`	`zebra` Terminal Mode Commands
`show ipv6 route ospf6`	Showing OSPF6 information
`show ipv6forward`	`zebra` Terminal Mode Commands
`show version`	Basic Config Commands
`shutdown`	Interface Commands
`smux peer` *`OID`*	SMUX configuration
`smux peer` *`OID PASSWORD`*	SMUX configuration
`table` *`TABLENO`*	Static Route Commands
`terminal length` *`0-512`*	Basic Config Commands
`timers basic` *`UPDATE TIMEOUT GARBAGE`*	RIP Timers
`timers spf` *`0-4294967295 0-4294967295`*	OSPF router
`version` *`VERSION`*	RIP Configuration
`who`	Basic Config Commands
`write file`	Basic Config Commands
`write terminal`	Basic Config Commands

Index

K

L

M

O

P

R

S

T

V

Z