HP-UX Routing Services Administrator's Guide

HP-UX 11i v2

Edition 1

Manufacturing Part Number: B2355-90777

August 2003

U.S.A.

Legal Notices

The information in this document is subject to change without notice.

Hewlett-Packard makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Warranty

A copy of the specific warranty terms applicable to your Hewlett-Packard product and replacement parts can be obtained from your local Sales and Service Office.

U.S. Government License

Proprietary computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notice

Trademark Notices

Contents

About This Document

1. Overview

2. Configuring mrouted

3. Configuring gated

About This Document

This manual describes the various routing daemons supported in the HP-UX 11i v2 operating system.It is one of the five new manuals documenting the Internet Services suite of products. See “Related Documentation” on page 11 for a list of the other new Internet Services manuals. These manuals replace the manual *Installing and Administering Internet Services* (B2355-90685), which was shipped with previous releases of the operating system.

This manual assumes that the HP-UX 11i v2 operating system software and the appropriate files, scripts, and subsets are installed.

Intended Audience

This manual is intended for system and network administrators responsible for managing the Routing Services. Administrators are expected to have knowledge of operating system concepts, commands, and the various routing protocols. It is also helpful to have knowledge of Transmission Control Protocol/Internet Protocol (TCP/IP) networking concepts and network configuration; this manual is not a TCP/IP or a routing tutorial.

HP-UX Release Name and Release Identifier

Each HP-UX 11i release has an associated release name and release identifier. The `uname` (1) command with the `-r` option returns the release identifier. Table 1 shows the releases available for HP-UX 11i.

Table 1 **HP-UX 11i Releases**

Release Identifier	Release Name	Supported Processor Architecture
B.11.11	HP-UX 11i v1	PA-RISC
B.11.20	HP-UX 11i v1.5	Intel® Itanium® Processor Family
B.11.22	HP-UX 11i v1.6	Intel® Itanium® Processor Family
B.11.23	HP-UX 11i v2.0	Intel® Itanium® Processor Family

Publishing History

Table 2 provides, for a particular document, the manufacturing part number, the respective operating systems, and the publication date.

Table 2 **Publishing History Details**

Document Manufacturing Part Number	Operating System Supported	Publication Date
B2355-90110	10.x	June 1996
B2355-90147	11.0	October 1997
B2355-90685	11.11 11.20 11.22	December 2000
B5969-4360	11.22	April 2002

What Is in This Document

HP-UX Routing Services Administrator's Guide is divided into chapters, each of which contain information about configuring the routing services.

Table 3 describes the content in more detail.

Table 3 **Document Organization**

Chapter	Description
Overview	Presents an overview of the Routing Services and the various protocols that they support.
Configuring mrouted	Describes how to configure `mrouted` and various configuration commands in `mrouted`.
Configuring gated	Describes how to configure `gated` on RIP, OSPF, and RDP protocols. This chapter also describes how to specify tracing options, route preference, and some troubleshooting measures in `gated`.

Related Documentation

For more information about the Internet Services suite of products, see the following books:

- *HP-UX Internet Services Administrator's Guide*

  Provides an overview of the Internet Services products and describes how to install and configure them on your HP-UX 11i v2 operating system. You can access this manual at the following URL:

  `http://www.docs.hp.com/hpux/netcom/index.html#Internet%20Services`

- *HP-UX Mailing Services Administrator's Guide*

  Provides information about the Mail User Agents (`elm`, `mailx`, `mail`) and Mail Transport Agent (Sendmail) used in the HP-UX 11i v2 operating system. This manual also contains a description of configuring and administering Sendmail on your system. You can access this manual at the following URL:

  http://www.docs.hp.com/hpux/netcom/index.html#Internet%20Services

- *HP-UX IP Address and Client Management Administrator's Guide*

  Provides an overview of the IP address and client management implementations on the HP-UX 11i v2 operating system, where BIND, DHCPv6, and SLP deal with client management, and NTP deals with IP address management. You can access this manual at the following URL:

  `http://www.docs.hp.com/hpux/netcom/index.html#Internet%20Services`

- *HP-UX Remote Access Services Administrator's Guide*

  Provides information about the Remote Access Services available in the HP-UX 11i v2 operating system: r-commands, WU-FTP, and `telnet`. You can access this manual at the following URL:

  `http://www.docs.hp.com/hpux/netcom/index.html#Internet%20Services`

- Request for Comments (RFC)

Many sections of this manual refer to RFCs for more information about certain networking topics. These documents publicize Internet standards, new research concepts, and status memos about the Internet. You can access the full range of RFC documents and more information about the Internet Engineering Task Force (IETF) at the following URL:

http://www.ietf.org/rfc.html

You can obtain additional information about `mrouted` and IP multicast routing from the following RFC (Request for Comment) documents:

— RFC 1075: *Distance-Vector Multicast Routing Protocol*

— RFC 1112: *Host Extensions for IP Multicasting*

- Other Documents

  HP does not maintain and own the following information. As such, their content and availability are subject to change without notice.

  The MBONE FAQ

  The Multicast Backbone (MBONE) is a virtual network implemented on top of the physical Internet. It supports routing of IP multicast packets. It originated as a cooperative, volunteer effort to support experimentation in audio and video teleconferencing over the Internet. You can find an HTML-formatted version of the MBONE FAQ at the URL:

  http://www.ripe.net/rite/wg/mbone/eu-faq.html

- iknow Topics of Interest

  HP iknow Topics of Interest describe some networking concepts and tasks, as well as other topics. You can find these documents on the HP-UX networking communications home page at the following URL

  `http://docs.hp.com/iknow`

Typographical Conventions

This document uses the following typographic conventions:

`audit` (5) An HP-UX manpage. In this example, *audit* is the name and *5* is the section in the *HP-UX Reference*. On the Web and on the Instant Information CD, it may be a hot link to the manpage itself. From the HP-UX command line, you can enter “`man audit`” or “`man 5 audit`” to view the manpage. See `man` (1).

Book Title
The title of a book. On the Web and on the Instant Information CD, it may be a hot link to the book itself.

`ComputerOut`
Text displayed by the computer.

`Command`
A command name, qualified command phrase, daemon, file, or option name.

$
The system prompt for the Bourne, Korn, and POSIX shells.

#
The superuser prompt.

`Variable`
The name of a variable that you may replace in a command or function or information in a display that represents several possible values.

[ ] { }
In syntax definitions, square brackets indicate items that are optional and braces indicate items that are required.

`(Ctrl+A)`
This symbol indicates that you hold down the first named key while pressing the key or mouse button that follows the plus.

Bold
The defined use of an important word or phrase.

HP Encourages Your Feedback

HP welcomes any comments and suggestions you have on this manual.

You can send your comments in the following ways:

- Internet electronic mail: netinfo_feedback@cup.hp.com
- Using a feedback form located at the following URL:

  http://docs.hp.com/assistance/feedback.html

Please include the following information along with your comments:

- The full title of the manual and the part number. (The part number appears on the title page of printed and PDF versions of a manual.)
- The section numbers and page numbers of the information on which you are commenting.

- The version of HP-UX that you are using.

1 Overview

A **router** is a device that has multiple network interfaces and that transfers Internet Protocol (IP) packets from one network or subnet to another within an internetwork. In many IP-related documents, this device is also referred to as a gateway. The term router is used in this manual. The router stores all the routing information in the form of a routing table. Routing tables contain the routes to reach a particular network, and also identify the router to which the datagram packet can be passed for this purpose. The routing tables must contain the latest routing information. Routing protocols perform the task of updating the routing tables with the latest routing information.

The primary function of a routing protocol is to exchange routing information with other routers. Routing daemons perform the task of exchanging routing information with other routers. The routing daemons supported on the HP-UX 11i v2 operating system are `mrouted` and `gated` 3.5.9.

A detailed description of the routing daemons, their configuration and troubleshooting information is provided in this manual.

This chapter contains information about the following topics:

- "The mrouted Routing Daemon" on page 17
- "The gated Routing Daemon" on page 22

The mrouted Routing Daemon

mrouted (pronounced “M route D”) is a routing daemon that forwards IP multicast datagrams, within an autonomous network, through routers that support IP multicast addressing. mrouted implements the Distance-Vector Multicast Routing Protocol (DVMRP). The ultimate destination of multicast datagrams are host systems that are members of one or more multicast groups.

Multicasting enables a client to establish one-to-many and many-to-many communication with other hosts and is used extensively in networking applications such as audio and video teleconferencing, where multiple hosts need to communicate with each other simultaneously.

NOTE You cannot use System Administration Manager (SAM) to configure mrouted.

mrouted routes multicast datagram packets only on certain network interfaces, such as EISA Ethernet (lan2) and EISA FDDI (from a provider other than HP), and the interface types vary depending on the system platform.

When you install the HP-UX 11i v2 operating system, mrouted is automatically installed on your system.

For more information on mrouted, type `man 1m mrouted` at the HP-UX prompt.

Multicasting Overview

This section describes the multicast routing protocol implemented in mrouted, and the multicast addresses and groups.

DVMRP Protocol

mrouted implements the Distance-Vector Multicast Routing Protocol (DVMRP). You can use DVMRP, an Interior Gateway Protocol (IGP), to route multicast datagrams within an autonomous network. The primary purpose of DVMRP is to maintain the shortest return paths to the source of the multicast datagrams. You can achieve this by using topological knowledge of the network to implement a multicast forwarding algorithm called Truncated Reverse Path Broadcasting (TRPB).

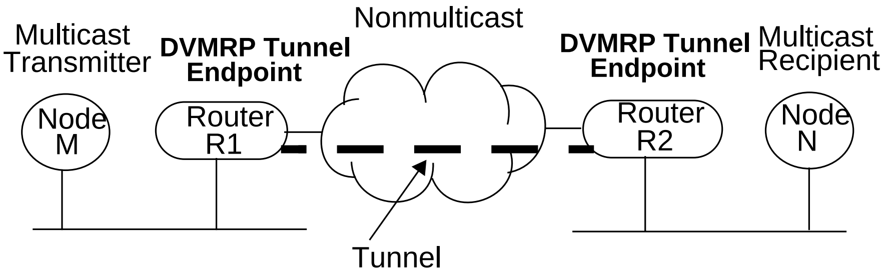
mrouted structures routing information in the form of a **pruned** broadcast delivery tree that contains routing information. mrouted structures routing information only to those subnets that have members of the destination multicast group. In other words, each router determines which of its virtual network interfaces are in the shortest path tree. In this way, DVMRP can determine if an IP multicast datagram needs to be forwarded. Without such a feature, the network bandwidth can easily be saturated with the forwarding of unnecessary datagrams.

Because DVMRP routes only multicast datagrams, you must handle routing of unicast or broadcast datagrams using a separate routing process.

To support multicasting across subnets that do not support IP multicasting, DVMRP provides a mechanism called **tunnelling**. Tunnelling forms a virtual point-to-point link between pairs of mrouted routers by encapsulating the multicast IP datagram within a standard IP unicast datagram using the IP-in-IP protocol (IP protocol number 4). This unicast datagram, containing the multicast datagram, is then routed through the intervening routers and subnets. When the unicast datagram reaches the tunnel destination, which is another mrouted router, the unicast datagram is stripped away and the mrouted daemon forwards the multicast datagram to its destinations.

Figure 1-1 shows a tunnel formed between a pair of mrouted routers.

Figure 1-1 Tunnel Made with mrouted Routers

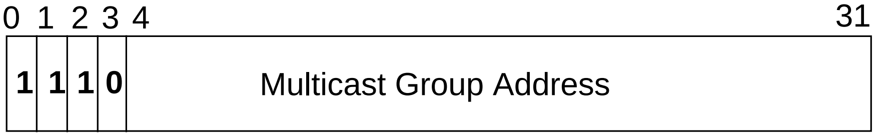


In this figure, the mrouted router R1 receives a multicast packet from node M. Because R1 is configured as one end of a tunnel, R1 encapsulates the IP multicast packet in a standard unicast IP packet addressed to the mrouted router R2. The packet, now treated as a normal IP packet, is sent through the intervening nonmulticast network to R2. R2 receives the packet and removes the outer IP header, thereby restoring the original multicast packet. R2 then forwards the multicast packet through its network interface to node N.

IP Multicast Addresses

An IP Internet address can be either a 32-bit or a 128-bit address. Each host on the Internet is assigned a unique IP address. There are four classes of IP addresses: Class A, Class B, Class C, and Class D. Class D IP addresses are identified as IP multicast addresses. Class A, Class B, and Class C IP addresses are composed of two parts: a network ID (**netid**) and a host ID (**hostid**). Class D IP addresses are structured differently, as shown in Figure 1-2.

Figure 1-2 **Class D IP Multicast Address Format**



The first 4 bits (0 through 3) identify the address as a multicast address. Bits 4 through 31 identify the **multicast group**. Multicast addresses are in the range 224.0.0.0 through 239.255.255.255. Addresses 224.0.0.0 through 224.0.0.255 are reserved, and address 224.0.0.1 is permanently assigned to the **all hosts group**. The all hosts group is used to reach all the hosts on a local network that participate in IP multicasting. The addresses of other permanent multicast groups are published in RFC 1060 (*Assigned Numbers*, March 1990).

You can use IP multicast addresses only as destination addresses, and they must never appear in the source address field of a datagram. Internet Control Message Protocol (ICMP) error messages are not generated for multicast datagrams.

Because IP Internet addressing is a software manifestation of the underlying physical network, you must map IP addresses to physical addresses that the hardware comprising the network understands.

Normally, IP multicast addresses are mapped to 802.3 or Ethernet multicast addresses. The IP multicasting addressing scheme, similar to Ethernet's scheme, uses the datagram's destination address to indicate multicast delivery.

When an IP multicast address is mapped to an Ethernet multicast address, the low-order 23 bits of the IP multicast address are placed into the low-order 23 bits of the special Ethernet multicast address. The hexadecimal value of the special Ethernet multicast address is 01-00-5E-00-00-00. The resultant Ethernet address, however, is not unique, because only 23 out of the 28 bits representing the multicast address are used.

Multicast Groups

A **multicast group** comprises hosts with an intention to join the multicast group by listening to the same IP multicast address. Group membership is dynamic, that is, a host may join or leave a group at any time. A host may be a member of one or more groups simultaneously. Additionally, a host is allowed to send multicast datagrams to a group without being a member of the group.

You can assign multicast addresses to transient groups because the multicast address are often temporary. A typical transient group scenario is when users run an application that dynamically registers to specific multicast addresses, which are discarded later when all members of the group have left. Some multicast addresses may be assigned to permanent groups that always exist, even when their membership is empty.

Both hosts and `mrouted` routers that participate in IP multicasting use the Internet Group Management Protocol (IGMP) to communicate multicast group information among themselves. Hosts use IGMP to inform `mrouted` routers that they are joining a group. `mrouted` routers use IGMP to pass multicast routing information to other `mrouted` routers, and to check whether a host is still an active group member.

The underlying TCP/IP stack must support ICMP to participate in IP multicasting. While IGMP defines a standard for communicating information, it does not define a standard for how the multicast information is propagated among multicast routers. Consequently, DVMRP enables multicast routers to efficiently communicate group membership information among themselves. DVMRP uses IGMP messages to carry routing and group membership information. DVMRP also defines IGMP message types that enable hosts to join and leave multicast groups, and that allow multicast routers to query one another for routing information.

The gated Routing Daemon

gated (pronounced “gate D”) is a routing daemon that updates routing tables in internetwork routers. Developed at Cornell University, gated handles the Routing Information Protocol (RIP), External Gateway Protocol (EGP), Border Gateway Protocol (BGP), Open Shortest Path First (OSPF) routing protocol, and the Router Discovery Protocol (RDP), or any combination of these protocols.

Routing protocols are designed to find a path between network nodes. If multiple paths exist for a given protocol, the shorter paths are usually chosen. Each protocol has a cost or a metric that it applies to each path. In most cases, the lower the cost or metric for a given path, the more likely a protocol will choose it.

NOTE You cannot use System Administration Manager (SAM) to configure gated.

Upon startup, gated reads the kernel routing table on the local machine. gated maintains a complete routing table in the user space, and keeps the kernel routing table (in the kernel space) synchronized with this table.

In large local networks, multiple paths often exist to other parts of the local network. You can use gated to maintain nearly optimal routing to other parts of the local network, and to recover from link failures.

Advantages

gated offers the following advantages:

- Dynamic routing eliminates the need to reset routes manually. When network failures occur, routes are automatically rerouted.
- Dynamic routing facilitates adding and administering nodes.
- Dynamic routing lowers the cost of operating complex Internet systems.

- gated translates among several protocols, passing information within or between IP routing domains or autonomous systems. **Autonomous system (AS)** is used here to refer to a group of connected nodes and routers in the same administrative domain that exchange routing information via a common routing protocol.
- gated provides the system administrator flexibility in setting up and controlling network routing. For example, gated can listen to network traffic at specified routers, determine available routes, and update local routing tables accordingly.

Deciding When to Use gated

gated is mostly used in large networks, or in small networks connected to larger wide area networks.

You must run gated on routers (gateways) to send the routing information to other routers. gated supports many routing protocols that allow routers to build and maintain dynamic routing tables. However, gated also supports RIP, which runs on end systems (systems with only one network interface) as well as on routers.

NOTE gated also supports RDP as a client. RDP will replace rdpd.

gated is useful in topologies with multiple routers and multiple paths between parts of the network. gated allows routers to exchange routing information and to change routing information dynamically to reflect topology changes and maintain optimal routing paths.

Alternatively, you can configure IP routes manually with the `route (1M)` command. For end systems in subnets with only one router (gateway) to the Internet, manually configuring a default route is usually more efficient than running gated. For more details on manually manipulating the routing tables, type `man 1M route` at the HP-UX prompt.

When connected to wide area networks, you can use gated to inject local routing information into the wide area network’s routing table.

Routing Protocols

For routing purposes, networks and gateways are logically grouped into autonomous system (AS). Companies and organizations that want to connect to the Internet and form an AS must obtain a unique AS number from the Internet Assigned Numbers Authority (IANA).

An interior gateway protocol distributes routing information within the autonomous system. An exterior gateway protocol distributes general routing information about an autonomous system to other autonomous systems.

Dividing networks into autonomous systems keeps route changes inside the autonomous system from affecting other autonomous systems. When routes change within an autonomous system, the new information need not be propagated outside the autonomous system if it is irrelevant to gateways outside the autonomous system.

gated supports the following interior gateway protocols, as defined in IETF RFCs:

- Routing Information Protocol (RIP) is a common routing protocol used within an autonomous system. A de facto industry standard, it is also used by routed, a service distributed by Berkeley. RIP is not intended for use in wide area network (WAN) applications. There are currently two versions of RIP implementations: Version 1, as defined in RFC 1058, and Version 2, as defined in RFC 1388. gated supports all Version 1 features and most of the features of Version 2. The following Version 2 features are not supported: RIP management information base (MIB) route tag, and route aggregation. gated 3.5.9 supports authentication.

- Open Shortest Path First (OSPF), similar to RIP, is a routing protocol that allows routing information to be distributed between routers in an autonomous system. Each router on the network transmits a packet that describes its local links to all other routers. The distributed database is then built from the collected descriptions. If a link fails, updated information floods the network, allowing all routers to recalculate their routing tables at the same time. OSPF is more suitable than RIP for routing in complex networks with many routers. gated 3.0 supports most of the features of OSPF Version 2, as described in RFC 1247, except the IP type of service (TOS) routing feature. Equal cost multipath routes are limited to one hop per destination, because the HP-UX kernel supports only one gateway per route.

- HELLO is designed to work with routers called Fuzzballs. Most installations use RIP or OSPF instead of HELLO. The HELLO protocol is no longer supported on HP-UX. You can use RIP or OSPF instead, because they are internal routing protocols.

NOTE Do not mix RIP and OSPF protocols within a single network, because the routing information may conflict.

Table 1-1 compares the advantages and disadvantages of the RIP and OSPF protocols.

Table 1-1 Comparison of RIP and OSPF Protocols

RIP	OSPF
Advantage: RIP is easy to configure.	*Disadvantage:* OSPF is complicated to configure and requires network design and planning.
Advantage: An end system (a system with only one network interface) can run RIP in passive mode to listen for routing information.	*Disadvantage:* OSPF does not have a passive mode.
Disadvantage: RIP may be slow to adjust for link failures.	*Advantage:* OSPF is quick to adjust for link failures.

Table 1-1 Comparison of RIP and OSPF Protocols (Continued)

RIP	OSPF
Disadvantage: RIP generates more protocol traffic than OSPF, because it propagates routing information by periodically transmitting the entire routing table to neighbor routers.	*Advantage:* OSPF generates less protocol traffic than RIP, because (i) Each router transmits information only about its links instead of the whole routing table, and (ii) OSPF allows you to divide an autonomous system into areas, each with a designated router that exchanges inter-area routing information with other routers. Intra-area routing information is isolated to a single area.
Disadvantage: RIP is not appropriate for large networks, because RIP packet size increases as the number of networks increases.	*Advantage:* OSPF works well in large networks.

gated supports the following exterior gateway protocols:

- The External Gateway Protocol (EGP) permits a node on the NSFNET **backbone** to exchange information with other backbone nodes about reaching a destination. You can use EGP to communicate routing information between autonomous systems. The EGP protocol will be obsoleted in a future release of HP-UX. Use BGP instead of the EGP protocol. BGP offers more flexibility and requires less bandwidth than EGP.
- The Border Gateway Protocol (BGP) is intended as a replacement for EGP. BGP uses path attributes to select routes. One of the attributes that BGP can pass is the sequence of autonomous systems that must be traversed to reach a destination. gated supports BGP Versions 2, 3, and 4, as described in RFCs 1163 and 1267.

gated also supports the Router Discovery Protocol (RDP), which is neither an interior nor an exterior gateway protocol. RDP is used to inform hosts of the existence of routers to which the hosts can send packets. It is used instead of, or in addition to, a statically configured default router. Router discovery consists of two parts: a server part that runs on routers, and a client part that runs on hosts.

2 Configuring mrouted

This chapter describes how to configure mrouted and the various configuration commands in mrouted. It also provides information on starting and verifying the mrouted installation. A description of the mrouted routing tables is also provided, along with the various multicast routing support tools. This chapter discusses the following topics:

- “How to Configure mrouted” on page 31
- “Starting mrouted” on page 36
- “Verifying mrouted Operation” on page 37
- “Displaying mrouted Routing Tables” on page 38
- “Multicast Routing Support Tools” on page 41

How to Configure mrouted

When the mrouted daemon starts, it automatically reads the default configuration file /etc/mrouted.conf. You can override the default configuration file by specifying an alternate file while invoking mrouted. See “Starting mrouted” on page 36 for more information. If you change the /etc/mrouted.conf file while mrouted is running, issue the following command to reread the configuration file:

```
kill -HUP
```

By default, mrouted automatically configures itself to forward on all multicast-capable interfaces, excluding the loopback interface that has the IFF_MULTICAST flag set. Therefore, you do not need to explicitly configure mrouted, unless you need to configure tunnel links, change the default operating parameters, or disable multicast routing over a specific physical interface.

Configuration Commands

You can define the configuration commands in the /etc/mrouted.conf configuration file. mrouted supports five configuration commands: phyint, tunnel, cache_lifetime, pruning, and name. One or more options are associated with each command.

The syntax of each command is as follows:

```
phyint local-addr [disable] [metric m] [threshold t] [rate_lim
it b]
       [boundary (boundary-name|scoped-addr/mask-len)]
       [altnet network/mask-len]
```

```
tunnel local-addr remote-addr [metric m] [threshold t] [rate_l
imit b]
       [boundary (boundary-name|scoped-addr/mask-len)]
```

```
cache_lifetime ct
```

```
pruning off/on
```

```
name boundary-name scoped-addr/mask-len
```

phyint

You can use the `phyint` command to disable multicast routing on the physical interface identified by the local IP address, *`local-addr`* (see Figure 2-1), or to associate a nondefault `metric` or `threshold` with the specified physical interface. Alternatively, you can replace the local IP address, *`local-addr`*, with the interface name, such as `lan0`. If `phyint` is attached to multiple IP subnets, use the `altnet` option to describe each additional subnet (one `altnet` option for each subnet).

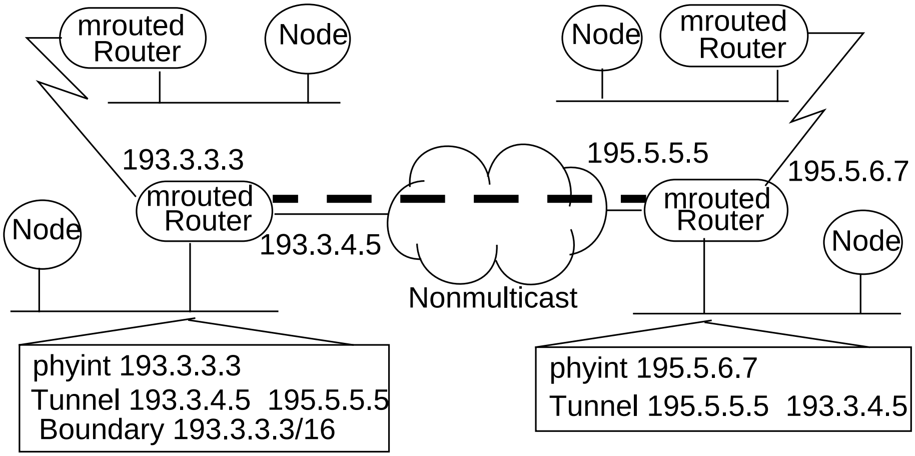
tunnel

You can use the `tunnel` command to establish a tunnel link between the local IP address, *`local-addr`*, and the remote IP address, *`remote-addr`* (see Figure 2-1). You can also use this command to associate a nondefault `metric` or `threshold` value with the tunnel. You can replace the local IP address, *`local-addr`*, with the interface name, such as `lan0`. Similarly, you can replace the remote IP address, *`remote-addr`*, by a host name, but only if the host name has a single IP address associated with it. Before you can use a tunnel, it must be set up in the `mrouted` configuration files of both the `mrouted` routers participating in the tunnel. `mrouted` 3.8 does not support the `srcrt` option. (It provided backward compatibility with older versions of `mrouted` that implemented IP multicast datagram encapsulation using IP source routing.)

NOTE

A phyint command must precede a tunnel command. All the phyint and tunnel command options must be placed on a single line except for the boundary and altnet options, which can begin on a separate line.

Figure 2-1 Multicast Network Example Configuration



```
phyint 193.3.3.3
Tunnel 193.3.4.5  195.5.5.5
 Boundary 193.3.3.3/16
```

```
phyint 195.5.6.7
Tunnel 195.5.5.5  193.3.4.5
```

The metric is the cost, or overhead, associated with sending a datagram on the given interface or tunnel, and is used primarily to influence the choice of routes over which the datagram is forwarded; the larger the value, the higher the cost. You must keep metrics as small as possible, because mrouted cannot route along paths with metrics greater than 31. In general, use a metric value of 1 for all links unless you are specifically attempting to force traffic to take another route. In this case, the metric of the alternate path should be the sum of the metrics on the primary path + 1. The default metric value is 1.

The threshold is the minimum IP time-to-live (TTL) required for a multicast datagram to be forwarded to a given interface or tunnel. It controls the scope of multicast datagrams. If the TTL value in the datagram is less than the threshold value, the datagram is dropped; if the TTL is greater than or equal to the threshold value, the packet is forwarded. The default threshold value is 1.

The TTL value of forwarded packets is only compared with the threshold value; it is not decremented by the threshold. An application that initiates the IP multicast datagram sets the TTL, and typically represents the number of subnets, or hops, the datagram has to traverse to reach its destination. Every time a multicast datagram passes through a multicast router, the TTL value is decremented by 1. HP recommends that you use the default threshold value unless you have a specific reason to set it otherwise.

In general, all interfaces connected to a particular subnet or tunnel should use the same metric and threshold values for that subnet or tunnel.

You can use the rate_limit option to specify a certain bandwidth in Kbits/second which is allocated to multicast traffic. The default value is 500Kbps on tunnels and 0 (unlimited) on physical interfaces.

You can use the boundary option to configure an interface as an administrative boundary for the specified *boundary-name* or *scoped-addr* (scoped address). You can specify more than one boundary option in the phyint and tunnel commands. Packets belonging to the scoped address, which is an IP multicast group address, are not forwarded on this interface. *mask-len* indicates the number of leading 1s in the mask applied (by means of a bitwise logically AND operation) to the scoped address. For example, the statement boundary 239.2.3.3/16 would result in the mask 255.255.0.0 being calculated by means of an AND operation with 239.2.3.3 to isolate the first two octets, 239.2, of the scoped address. Therefore, all IP multicast addresses beginning with 239.2 will not be forwarded on the specified interface.

The primary use of the boundary option is to allow concurrent use of the same IP multicast addresses on downstream subnets, without interfering with multicast broadcasts using the same IP multicast addresses on subnets that are upstream from the mrouted gateway.

The cache_lifetime value determines the amount of time that a cached multicast route remains in the kernel before timing out. This value is specified in seconds and must be between 300 (5 minutes) and 86400 (24 hours). The default value is 300.

You can use the pruning off command to explicitly configure mrouted as a nonpruning router. When pruning is off, IP multicast datagrams are forwarded to leaf subnets of the broadcast routing tree even when those leaf subnets do not contain members of the multicast destination group. Use only nonpruning mode for testing. The default mode for pruning is `on`.

You can use the `name` command to assign a name (*`boundary-name`*) to a boundary (a *`scoped-addr/mask-len`* pair) to simplify the configuration task.

`mrouted` terminates if it has less than two enabled virtual interfaces (VIFs), where a VIF is either a physical multicast-capable interface or a tunnel. It logs a warning message if all the VIFs are tunnels. HP recommends that you replace such configuration settings with more direct tunnels.

Starting mrouted

You can start `mrouted` from the HP-UX prompt or from within a shell script by issuing the following command:

```
/etc/mrouted [-p] [-c config_file] [-d debug_level]
```

The `-p` option disables pruning by overriding the `pruning on` statement within the `/etc/mrouted.conf` configuration file. You must use this option for testing purposes only.

The `-c` option overrides the default configuration file `/etc/mrouted.conf`. Use *`config_file`* to specify an alternate configuration file.

The `-d` *`debug_level`* option specifies the debug level. *`debug_level`* can be in the range 0 to 3. To know more about *`debug_level`* values, type `man 1M mrouted` at the HP-UX prompt.

By default, `mrouted` always writes warning and error messages to the system log daemon. You can retrieve these messages from the system log file, `syslog.log`, located in the `/var/adm/syslog` directory.

For convenience in sending signals, `mrouted` writes its `pid` to the `/var/tmp/mrouted.pid` file upon startup.

Verifying mrouted Operation

You can use one or more of the following methods to verify mrouted operation:

- Retrieve the **virtual interface table** and the **multicast routing table** to verify if appropriate virtual interfaces (vifs) are configured. See "Displaying mrouted Routing Tables" on page 38 for information on retrieving these tables.
- Retrieve the **routing cache table** to verify if the routing and cache information is appropriate for your configuration of mrouted. See "Displaying mrouted Routing Tables" on page 38 for information on retrieving this table.
- Examine the syslog file /var/adm/syslog/syslog.log for warning and error messages that indicate the status of mrouted. Upon startup, mrouted logs a startup message in the syslog file that indicates the mrouted version number, such as mrouted version 3.8.
- Issue the following ps (process status) command to search for the string "mrouted", using grep, to determine if the mrouted program is running:

  ```
  ps -ef | grep mrouted
  ```

Displaying mrouted Routing Tables

`mrouted` contains three routing tables: the virtual interface table, the multicast routing table, and the multicast routing cache table.

The virtual interface table displays the following topological information for each virtual interface:

- Physical and tunnel interfaces.
- The number of incoming and outgoing packets at each interface.
- The value of specific configuration parameters, such as `metric` and `threshold`.

An example virtual interface table is as follows:

```
Vif   Local                                Metric   Thresh   Flags
      Address

0     36.2.0.8    subnet:       36.2       1        1        querier
                  groups:       224.0.2.1
                                224.0.0.4
                  pkts in:      3456
                  pkts out:     2322323

1     36.11.0.1   subnet:       36.11      1        1        querier
                  groups:       224.0.2.1
                                224.0.1.0
                                224.0.0.4
                  pkts in:      345
                  pkts out:     3456

2     36.2.0.8    tunnel:       36.8.0.77  3        1
```

```
peers:         36.8.0.77   (2.2)

boundaries:    239.0.1

               239.1.2

pkts in:       34545433
```

The multicast routing table displays connectivity information for each subnet from which a multicast datagram can originate.

The multicast routing cache table is a duplicate copy of the kernel forwarding cache table. It contains the status information for multicast destination group-origin subnet pairs.

`mrouted` retrieves these tables by sending an appropriate signal to the `mrouted` daemon. `mrouted` responds to the following signals:

HUP
Restarts `mrouted`. The configuration file is reread each time this signal is evoked.

INT
Terminates `mrouted` by sending messages to all neighboring routers.

TERM
The same as INT.

USR1
Defined as signal 16, dumps the internal routing tables (virtual interface table and multicast routing table) to `/usr/tmp/mrouted.dump`.

USR2
Defined as signal 17, dumps the multicast routing cache tables to `/usr/tmp/mrouted.cache`.

QUIT
Dumps the internal routing tables (virtual interface table and multicast routing table) to `stderr` (only if `mrouted` was invoked with a nonzero debug level).

You can send signals to `mrouted` by issuing the HP-UX `kill` command at the HP-UX prompt. For example:

```
kill -USR1 pid
```

where *`pid`* is the process ID of the `mrouted` daemon.

For more information on the routing tables, type man `1M mrouted` at the HP-UX command prompt, and see the EXAMPLE section.

For more information on signals, type man `1M mrouted` at the HP-UX command prompt, and see the Signals section.

Multicast Routing Support Tools

This section describes various multicast routing support tools.

The mrinfo Tool

mrinfo is a multicast routing tool that requests configuration information from mrouted and prints the information to the standard output. By default, mrouted prints configuration information for its local instance. You can override the default request (the local instance of mrouted) by specifying an alternate router IP address or system name.

For more information, type man 1M mrinfo at the HP-UX command prompt.

The map-mbone Tool

map-mbone is a multicast routing tool that requests multicast router connection information from mrouted and prints the connection map information to the standard output. By default (when no alternate router address is specified), the request message is sent to all the multicast routers on the local network. If map-mbone discovers new neighbor routers from the replies, it sends an identical request to those routers. This process continues till the list of new neighbors is exhausted.

For more information, type man 1M map-mbone at the HP-UX command prompt.

The netstat Tool

You can used netstat to display multicast-related information, including network statistics and multicast routing table contents.

For more information, type man 1M netstat at the HP-UX command prompt.

3 Configuring gated

gated handles multiple routing protocols. You can configure the gated daemon to perform all or any combination of the supported protocols.

The HP-UX 11i v2 operating system supports gated 3.5.9. This chapter contains information about how to configure gated on various routing protocols. It also describes how to specify the tracing options and route preference in gated, and discusses certain troubleshooting measures.

This chapter discusses the following topics:

For additional information on gated, type `man 1M gated.conf` at the HP-UX prompt.

Configuration Overview

Upon startup, gated reads the configuration file to decide how each protocol must be used to manage routing. By default, it uses the configuration file named/etc/gated.conf. Creating the configuration file is usually the responsibility of the system administrator.

The configuration file can include up to eight sections (called **classes**) of configuration **statements**. You can define the statements further with optional **clauses**. The eight classes of configuration statements are:

- Directives — Directives are statements that are immediately acted upon by the gated parser.
- Trace — Trace statement controls gated tracing options.
- Options — Options statements define global gated options.
- Interface — Interface statements define router interface options.
- Definition — Definition statements identify the autonomous system that the router belongs to and **martian** addresses (addresses for which routing information must be ignored).
- Protocol — Protocol statements enable or disable gated protocols and set protocol options.
- Static — Static statements define static routes or default routers that are installed in the kernel routing table.
- Control — Control statements define routes that are imported to the router from other routing protocols, and paths that the router exports to other routing protocols.

For a description of each configuration class and to determine which statements belong to which class, type `man 4 gated.conf` at the HP-UX prompt.

With Version 3.5.9 of gated, the two statements previously in the Trace class (`tracefile` and `traceoptions`) have been combined into a single `traceoptions` statement. Therefore, the `tracefile` statement has been eliminated. Also, some of the global options have been removed, some new global options have been added, and options have been added for some of the protocols. For details about the new syntax, type `man 4 gated.conf` at the HP-UX prompt.

If you do not want to use any of the gated 3.5.9 features added at HP-UX 10.30, and do not have any tracing configured in your gated 3.0 /etc/gated.conf configuration file, you can continue to use your 3.0 configuration file with gated 3.5.9. If you do have tracing configured in your gated 3.0 file, you must run the conv_config conversion tool on the file so that it follows the 3.5.9 syntax (see “Converting the Configuration File from 3.0 to 3.5.9” on page 48). For more information about the 3.5.9 syntax, type man 4 gated.conf at the HP-UX prompt.

To check your gated 3.0 configuration file for compatibility with the 3.5.9 syntax, issue the following command at the command prompt:

```
gated -c [-f config_file_name]
```

You need to specify -f *config_file_name* only if the configuration file you are checking is not the default file.

If you are still running gated 2.0, you must manually edit the /etc/gated.conf file so that it follows the 3.5.9 syntax. The conversion utility that was previously available to migrate from gated 2.0 to 3.0 is no longer available, and you can only use the conv_config tool when migrating from 3.0 to 3.5.9.

Configuring gated

To configure gated, complete the following steps:

1. Issue the following command to create the gated configuration file /etc/gated.conf:

   ```
   gdc newconf
   ```

   If the protocols are not explicitly specified, gated assumes the following:

   ```
   rip yes;
   ospf no;
   ```

2. Determine how you want to configure each routing protocol, then add the appropriate statements for each protocol to the /etc/gated.conf configuration file.

   See the section “Configuring the OSPF Protocol” on page 60 for OSPF routing configuration statements. RIP configuration is described in “Configuring the RIP Protocol” on page 50. For a more detailed description of the configuration statements, type man 4 gated.conf at the HP-UX prompt.

3. Add statements for any additional configuration information. See “Customizing Routes” on page 90, “Specifying Tracing Options” on page 92, and “Specifying Route Preference” on page 94 for other configuration options.

   In particular, you may want to prevent gated from deleting interfaces from the routing table when gated does not receive routing protocol information from that interface. To do this, insert passive interface definitions in the interfaces statements. For example:

   ```
   interfaces {
      interface all passive ;
   } ;
               :
               :
   <protocol statements follow>
   ```

4. If you normally use default routes, you must configure a static default route in the gated configuration file. If the default route is a gateway node, add the following entry to /etc/gated.conf (enter the gateway node’s IP address for *gateway_IP_Address*):

   ```
   static {
      default gateway gateway_IP_Address retain ;
   } ;
   ```

   The default route may be a local interface, such as in topologies that include a proxy ARP server on the local network. If the default route is a local interface, add the following entry to /etc/gated.conf:

   ```
   static {
      default interface local_IP_Address retain ;
   } ;
   ```

   The *local_IP_Address* is the local system’s IP address of the interface or network interface name (that is, lan0, lan1, and so on) that acts as the default route. If you use a proxy ARP server, this is the local address of the interface attached to the same network as the proxy ARP server.

   See “Customizing Routes” on page 90 and the section covering “Common Problems” on page 104 in the section “Troubleshooting gated” on page 101 for more information.

5. To check for syntax errors in the configuration file, run gated with the -c or -C option (gated exits after parsing the configuration file).

NOTE

You can also use the command `gdc checkconf` to parse the `/etc/gated.conf` file for syntax errors. `gdc` issues a message to indicate the parsing errors. If there are any errors, the error output is saved to a file for further inspection. For more details, type `man 1M gdc` at the HP-UX prompt.

6. Set the environment variable `GATED` to `1` in the file `/etc/rc.config.d/netconf` to invoke `gated` automatically when the system is started.
7. To start `gated`, reboot your system or run the following `gated` startup script:

   ```
   /sbin/init.d/gated start
   ```

   You can also start `gated` by executing the following command:

   ```
   gdc start
   ```

   The following message appears:

   ```
   gated started, pid 25579
   ```

   where

   `25579` is the process ID (`pid`) of the gated process.

Examples of `gated` configuration files are included in the sections “Configuring the OSPF Protocol” on page 60 and “Configuring the RIP Protocol” on page 50. Additionally, they are also included in the `/usr/newconfig/gated/conf` directory.

NOTE

HP recommends that you specify the IP address in dot notation (for example, `a.b.c.d`) for a configuration option such as a router, host, or interface. Host names with multiple IP addresses associated with them produce errors.

Converting the Configuration File from 3.0 to 3.5.9

To convert a `gated` 3.0 configuration file to the `gated` 3.5.9 syntax, complete the following steps:

1. Retain a copy of the gated 3.0 configuration file, because you cannot specify the same file for input and output while running the conv_config conversion tool. For example, if you are using /etc/gated.conf for 3.0, type the following command:

   ```
   cp /etc/gated.conf /etc/gated.conf.30
   ```

2. Issue the following command at the HP-UX prompt:

   ```
   conv_config < input_config_file_name > output_config_file
   ```

   where

   input_config_file_name is the name of the gated 3.0 file you want to convert. You must specify this name, because the tool does not assume that you are converting the default file, /etc/gated.conf.

   output_config_file is the name of the new configuration file for gated 3.5.9. You must specify this name, because the tool does not assume that you are coverting the default file, /etc/gated.conf.

   For example, to convert the gated 3.0 configuration file to gated 3.5.9, issue the following command:

   ```
   conv_config < /etc/gated.conf.30 > /etc/gated.conf
   ```

After running the conversion tool, you can check the new configuration file for compatibility using the gated -c command. See the Note in the section “Configuration Overview” on page 45 for more information.

Configuring the RIP Protocol

RIP uses **hopcount** to determine the shortest path to a destination. Hopcount is the number of routers a packet must pass through to reach its destination. If a path is directly connected, it has the lowest hopcount of 1. If the path passes through a single router, the hopcount increases to 2. Hopcount can increase to a maximum value of 16, which is RIP's **infinity metric**, an indication that a network or node cannot be reached.

If `gated` encounters an unreachable node, it goes into **Holddown** Mode. Holddown Mode stops a node from propagating routing information until the other nodes that it is communicating with stabilize their routing information.

Hosts with only one LAN interface may use the RIP protocol with `gated` to passively listen to routing information when multiple routers on the LAN exist. If only one router on the LAN exists (leaving only one path off the local LAN), you can configure a static route to that router in the `/etc/rc.config.d/net` file, or issue the `route` command manually, instead of running `gated`.

In certain cases, you may not want the traffic to follow a certain path, because it incurs an unacceptable cost or security risk. In these cases, `gated` allows you to assign a metric to each interface. This allows you to select or bypass a path, irrespective of its length or speed.

RIP Protocol Statement

The syntax of the RIP protocol statement is as follows:

```
rip yes|no | on|off [ {
    broadcast|nobroadcast ;
    nocheckzero ;
    preference preference ;
    defaultmetric metric ;
    query authentication [none|[[simple|md5] password]] ;
    interface interface_list
      [noripin]|[ripin] [noripout]|[ripout]
      [metricin metric] [metricout metric]
      [version 1]|[version 2 [multicast|broadcast]]
      [[secondary] authentication [none|[simple|md5] password]
] ;
    [interface ...]
    trustedgateways router_list ;
    sourcegateways router_list ;
    traceoptions traceoptions ;
} ] ;
```

Curly braces ({}) are part of the syntax for the RIP protocol statement. Square brackets ([]) are not part of the syntax; they are used here to indicate optional parameters.

yes (or on) informs gated to enable the RIP protocol at this node and to process RIP packets coming in from other nodes. no (or off) informs gated to disable the RIP protocol at this node. If gated finds fewer than two network interfaces, the node listens to RIP information. If gated finds two or more network interfaces, the node not only listens but also broadcasts or multicasts the RIP information. If you do not specify a RIP statement in your configuration file, rip on is assumed.

The following describes the various options in the RIP statement:

- broadcast specifies that RIP packets are always generated. If the RIP protocol is enabled and more than one interface is specified, broadcast is assumed. Specifying broadcast with only one interface is useful only when propagating static routes or routes learned from other protocols.
- nobroadcast specifies that RIP packets are sent only to routers listed in the sourcegateways clause. If the RIP protocol is enabled, but only one interface is specified, nobroadcast is assumed.
- nocheckzero specifies that the RIP protocol must not check whether the reserved fields in the RIP packets are zero. In RIP Version 1 (as described in RFC 1058), certain reserved fields must be zero; however, this may vary in RIP implementations.
- preference determines the order of routes from other protocols to the same destination in the routing table. gated allows one route to a destination per protocol for each autonomous system. In case of multiple routes, the route used is determined by the value of preference.

  Default: 100

  Range: 0 (most preferred) – 255 (least preferred)
- defaultmetric is the default metric used when propagating routes learned from other protocols.

  Default: 16

Range: 1 – 16

- `query authentication [none|[[simple|md5] *password*]]` specifies the authentication, if any, that is required for query packets that do not originate from routers. If authentication consisting of only a password is required, specify `simple` *password* or *password*. If the required authentication consists of a key that was created with the MD5 algorithm, specify `md5`. Default: `none`.
- `interface` is specified as one of the following (in the order of precedence): an IP address (for example, `193.2.1.36`), a domain or interface name (for example, `lan0` or `lan1`), a wildcard name (for example, `lan*`), or `all` (which refers to all interfaces). You can specify multiple `interface` statements with different clauses. If you specify a clause more than once, the instance with the most specific interface reference is used.
- `noripin` specifies that `gated` does not process any RIP information received through the specified interface. Default: `ripin`.
- `noripout` specifies that `gated` does not send any RIP information through the specified interface. Default: `ripout`.
- `metricin` specifies the incoming metric for all routes propagated to this node through the specified interface.

  Default: kernel interface metric plus 1 (the default RIP hopcount)
- `metricout` specifies the outgoing metric for all routes propagated by this node through the specified interface.

  Default: 0
- `version 1` specifies that RIP Version 1 packets (as defined in RFC 1058) are sent; RIP Version 2 packets (defined in RFC 1388) are sent only in response to a Version 2 poll packet. `version 2` specifies that RIP Version 2 packets are sent to the RIP multicast address or to the broadcast addresses. You can specify how the packets are sent with the `multicast` or `broadcast` clauses. `version 2 multicast` implies that you want to send Version 2 packets (containing subnet mask information). `version 2 broadcast` implies that you want to send Version 2 packets. If you do not specify a version, `version 1` is assumed.

- `[secondary] authentication [none|[simple|md5] *password*]` specifies the type of authentication for RIP Version 2 packets (it is ignored for Version 1 packets). `secondary` indicates that the secondary authentication is defined; otherwise, the primary authentication is defined. If authentication consisting of only a password is required, specify `simple` *`password`* or *`password`* (where *`password`* is a quoted string of 0 – 16 characters). If the required authentication consists of a key that was created with the MD5 algorithm, specify `md5`. Default: `none`. (If you do not specify the `authentication` clause, the default is primary authentication of `none` and no secondary authentication.
- `trustedgateways` provides a list of routers that provide valid RIP routing information; routing packets from other routers are ignored.

  Default: all routers on the attached networks.
- `sourcegateways` specifies routers to which you can send RIP routing packets. If you specify the `nobroadcast` clause, routing updates are sent only to routers listed in the `sourcegateways` clause.
- `traceoptions` enables tracing for the RIP protocol. See “Specifying Tracing Options” on page 92.

Configuration Options

The `-e` and `-a` configuration options help increase the RIP convergent time on the HP-UX operating system. You can set these command-line options in the `/etc/gated.conf` file under the RIP protocol statement.

The `-e` option specifies `route_expiry_time`, which is the expiration time used by RIP to determine route aging. The minimum value is 1 second, and the maximum value is 180 seconds. The default value is 180 seconds.

Using the `-a` option, you can specify the `route_update_time` option. `route_update_time` is the time in seconds required by the RIP protocol to send RIP updates to other nodes on the network. The minimum value is 1 second, and the maximum value is 30 seconds. The default value is 30 seconds.

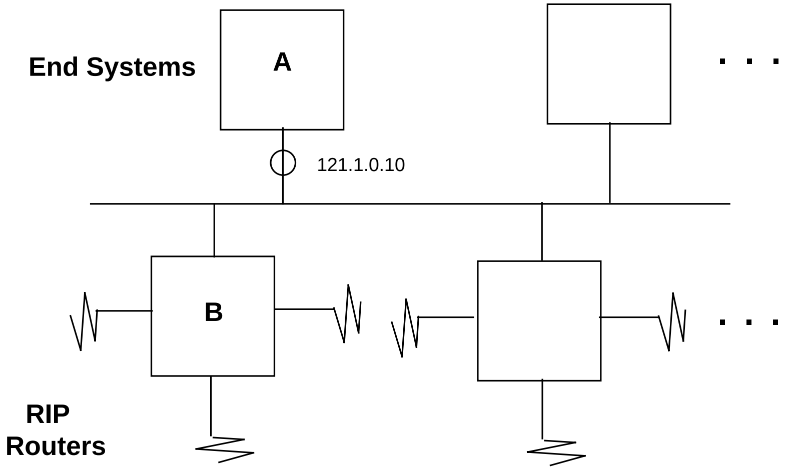
Alternatively, you can manually change the value in the `/etc/gated.conf` file. You can specify the `-e` and `-a` options either on the command line or in the configuration file. Configuration file options override the command-line options.

Simple RIP Configuration

A simple RIP configuration consists of RIP routers and end nodes that listen to information exchanged by the RIP routers, as shown in Figure 3-1.

Figure 3-1 and the accompanying text describe configuration of a single end system (node A) and a RIP router (node B). The configuration is the same for multiple end systems and RIP routers (only node B's configuration is shown here). This example shows only the syntax needed for a simple configuration. A detailed description of the entire RIP protocol statement is given after this example.

Figure 3-1 **Example of Simple RIP Configuration**



A: End System on a LAN with RIP Routers

Set up the configuration file `/etc/gated.conf` in the end system A as follows:

```
rip yes {
    interface 121.1.0.10 version 2 multicast;
};
static {
    default interface 121.1.0.10 preference 255 ;
};
```

With one interface, A can listen to RIP traffic on the network but does not forward routing information. Routers must be multicasting RIP packets on this network for A to learn about them and update its routing table. The first syntax statement enables RIP on node A's interface (121.1.0.10) to multicast routing information. The second statement specifies a static local default route to prevent gated from deleting it.

B: RIP Router

Set up /etc/gated.conf as follows:

```
rip yes {
    interface all version 2 multicast ;
};
```

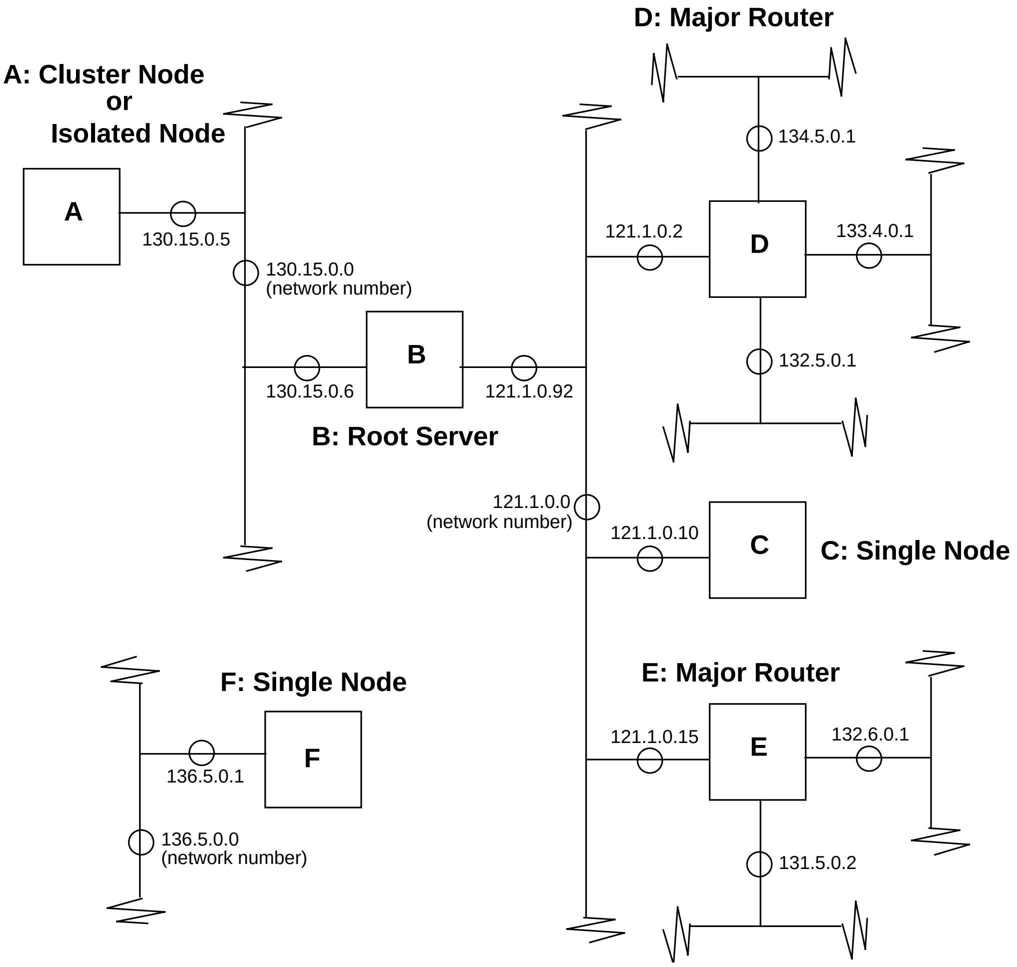
This enables the RIP protocol to multicast routing information on all interfaces.

Example of a Large RIP Configuration

Figure 3-2 and the accompanying text describe how to configure gated for the RIP protocol in each node within a networked system.

B, D, and E pass routing information among themselves and update their routes accordingly. C listens to the RIP conversation between B, D, and E, and updates its routes accordingly. If both the routers D and E can provide a path to a network, but the path through router D is shorter, nodes B, C, and E use router D when routing packets to that network. If D goes down, E becomes the new router to that network for the nodes B, C, and E.

Figure 3-2 Example of Large RIP Network



A: Cluster Node (or Isolated Node)

You need not run `gated` at this node, because it is on a LAN with only one router. Set a static default route to the cluster server (B) in the `/etc/rc.config.d/netconf` file as follows:

```
ROUTE_DESTINATION[0]= "default"
ROUTE_GATEWAY[0]= "130.15.0.6"
ROUTE_COUNT[0]= "1"
```

B: Cluster (or Root) Server Node

Run gated to get routing information about the 121.0.0.0 network. Set up /etc/gated.conf as follows:

```
interfaces {
    interface 130.15.0.6 121.1.0.92 passive ;
};
rip yes {
    interface 130.15.0.6 noripout ;
    interface 121.1.0.92 version 2 multicast;
};
static {
    default gateway 121.1.0.2 preference 255 ;
};
```

In the previous example, setting rip to yes is similar to setting rip to broadcast. Both the arguments inform the node to send out RIP packets, because the node has at least two interfaces. To reduce traffic on the 130.15.0.0 LAN, use the noripout option on this interface. This prevents RIP from sending packets on the 130.15.0.0 network.

To isolate the 130.15.0.0 LAN, use the following:

```
export proto rip interface 121.1.0.92 {
    proto direct {
        130.15.0.0 restrict ;
    };
};
```

To further isolate the LAN from the 121.1.0.0 LAN, do not specify any static routes that specify that you can reach the LAN through B. See “Importing and Exporting Routes” on page 97 for more information.

Always specify the passive option with the interface’s IP address. It informs gated to maintain the routes even if no other nodes on the 121.0.0.0 network are using RIP. Without this clause, gated changes the preference of the route to the interface if it does not receive routing information for that interface. The static default route adds the specified default to the kernel routing table. Setting the preference to 255 replaces this route whenever another default route is learned from one of the protocols.

C: End System on a LAN with RIP Routers

Set up /etc/gated.conf as follows:

```
rip yes {
    interface 121.1.0.10 version 2 multicast;
};
static {
    default interface 121.1.0.10 preference 255 ;
};
```

With one interface, C can listen to RIP traffic on the network but does not forward routing information. Routers must be multicasting RIP packets on this network for C to learn about them and to update its routing table.

D: Major Router

Set up /etc/gated.conf as follows:

```
rip yes {
    interface all version 2 multicast ;
};
```

This runs RIP on all attached networks.

E: Major Router

Set up the configuration file /etc/gated.conf as follows:

```
rip yes {
    interface all version 2 multicast;
};
```

Controlling RIP Traffic

This section describes configuration options for RIP routing information sent by gated from the node. Use these options to hide all or part of your network from other networks or to limit network traffic.

The RIP protocol definition in the /etc/gated.conf file contains the following two options for limiting RIP routing information exported by gated:

- The noripout clause in the interface definition informs gated not to send any RIP information through the listed interfaces.
- The sourcegateways clause informs gated to send RIP information directly to the specified routers.

See “RIP Protocol Statement” on page 50 for more information about these clauses.

The options for limiting RIP routing information imported by `gated` in the RIP protocol definition in the `/etc/gated.conf` file are as follows:

- The `noripin` clause in the interface definition informs `gated` not to process RIP information received through the listed interfaces.
- The `trustedgateways` clause informs `gated` to listen to RIP information received only from the specified routers.

See “RIP Protocol Statement” on page 50 for more information about these clauses.

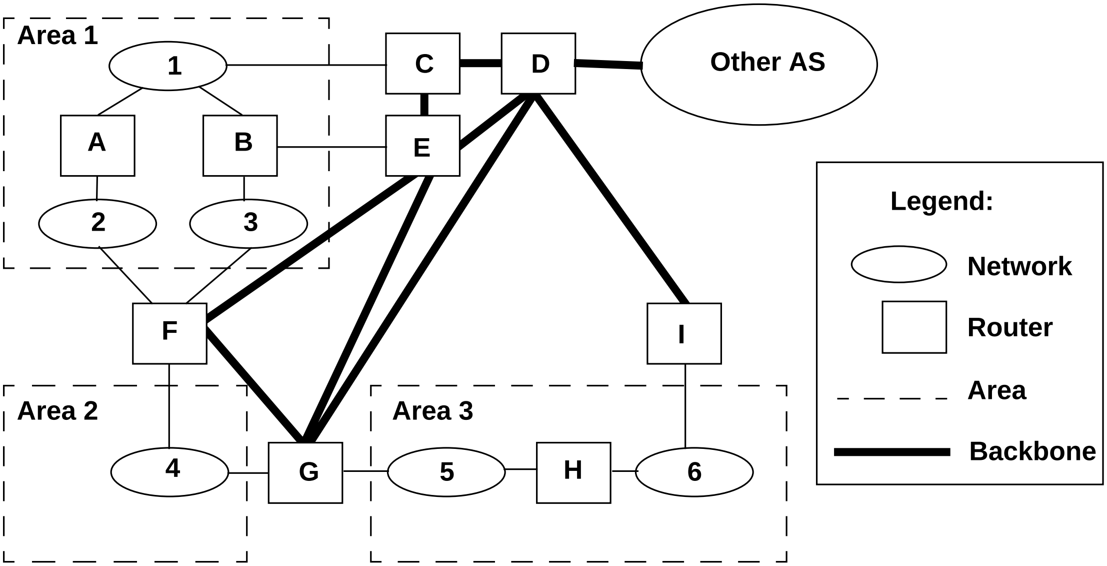
You can also use the `gated` `import` and `export` statements to restrict and control the route information propagated from one routing protocol to another. See “Importing and Exporting Routes” on page 97 for more information.

Configuring the OSPF Protocol

Open Shortest Path First (OSPF) is a link-state routing protocol that distributes routing information between routers in a single autonomous system (AS). Each OSPF router transmits a packet with a description of its local links to all other OSPF routers. The distributed database is built from the collected descriptions. Using the database information, each router constructs its own routing table of shortest paths from itself to each destination in the AS.

OSPF allows you to organize routers, networks, and subnetworks within an AS into subsets called areas. An area is a grouping of logically contiguous networks and hosts. Instead of maintaining a topological database of the entire AS, routers in an area maintain the topology for only the area in which they reside. Therefore, all routers belonging to an area must be consistent in their configuration of the area. The topology of an area is hidden from systems that are not part of the area. The creation of separate areas can help minimize overall routing traffic in the AS. Figure 3-3 shows an example of three separate areas defined for an AS.

Figure 3-3 Areas Defined in an Autonomous System



Internal routers have all their directly connected networks in the same area. In Figure 3-3, routers A, B, and H are internal routers.

Routers that are connected to multiple areas are called **area border routers**. In Figure 3-3, routers F and G are area border routers.

Routers that connect one AS to another are called **AS boundary routers**. In Figure 3-3, router D is an AS boundary router.

Neighbor routers are routers that interface to a common network. OSPF uses its own Hello subprotocol to determine which routers are neighbors. In Figure 3-3, routers A, B, and C are a set of neighbor routers that interface to network 1, while routers A and F are another set of neighbor routers that interface to network 2.

NOTE

The Hello subprotocol used with OSPF is not the same as the `gated` HELLO protocol. The Hello subprotocol is still supported.

Multi-access networks (networks that can be accessed through two or more neighbor routers) must have one of the routers identified as a designated router.

Designated routers initiate OSPF protocol functions on behalf of the network. In Figure 3-3, you can access network 1 through the neighbor routers A, B, or C; one of these routers is elected to become the designated router for network 1.

The set of routers that exchange OSPF protocol packets between areas in an autonomous system is called the **backbone**. In Figure 3-3, routers C, D, E, F, G, and I form an AS backbone that allows protocol packets to travel between the three areas.

OSPF routers exchange various types of **link state advertisements** to build their topological databases. Most link state advertisements are flooded (sent to every router) throughout the attached area. An exception is the link state advertisement sent out by AS boundary routers that describe routes to destinations outside the AS; these advertisements are flooded throughout the AS. Table 3-1 shows the various types of link state advertisements used by the OSPF protocol.

Table 3-1 **Types of Link State Advertisements**

Type	Content	Originated By	Flooded Throughout
Router link	Router’s links to area	Internal and area border routers	Area
Network link	List of routers attached to network	Designated router	Area
Summary link	Routes to destinations outside area but within AS	Area border router	Area
AS external link	Routes to destinations outside AS	AS boundary router	AS

AS boundary routers exchange routing information with routers in other autonomous systems. An AS boundary router can be an area border router or an internal router. It can also be a backbone router, but it is not required that an AS boundary router be a backbone router. An AS boundary router learns about routes other than its attached AS through exchanges with other routing protocols or through configuration information. Each AS boundary router calculates paths to destinations outside of its attached AS. It then advertises these paths to all routers in its AS.

Following are the two levels of routing in an AS:

- **Intra-area routing**, where the source and destination of a packet both reside in the same area. Routing is handled by internal routers.
- **Inter-area routing**, where the source and destination of a packet reside in different areas. Packets travel through an intra-area route from the source to an area border router, then travel an inter-area route on a backbone path between areas. Finally, they travel another intra-area route to the destination.

Planning Your OSPF Configuration

Following is a suggested sequence of steps in planning the OSPF routing in your autonomous system:

1. If your AS exchanges routing information with other autonomous systems, you need to obtain a unique AS number from the Internet Assigned Numbers Authority.
2. Partition the AS into areas. You can partition any interconnected networks into lists of address ranges, with each address range represented as an address-mask pair. The area border routers summarize the area content for each address range and distribute the summaries to the backbone. See “The networks Statement” on page 66 for more information on specifying address ranges.
3. Identify the internal routers for each area. An internal router configuration contains only one area definition.
4. Identify the area border routers and the areas to which they interface. The configuration for each area border router contains multiple area definitions.
5. For each router, determine the interface type for each area. Router interfaces can be multicast, non-broadcast multi-access (NBMA), or point-to-point. See “The interface Statement” on page 67 for more information on router interfaces.

6. For multi-access networks, identify a designated router. For NBMA networks, several routers can be designated router candidates. Designated routers are specified in the interface definitions (see “The interface Statement” on page 67).
7. You must decide if you want to assign a cost to each interface. See “Cost” on page 79 for more information about costs.
8. Designate stub areas. AS external link advertisements are propagated to every router in every area in an AS, except for routers in the configured stub areas. See “Stub Areas” on page 74 for more information
9. Identify backbone routers. The router configuration contains a backbone definition and a virtual link definition, if necessary. See “Defining Backbones” on page 76 for more information
10. Determine if routing packets are authenticated for each area. See “Authentication” on page 77 for more information
11. Identify AS boundary routers. See “AS External Routes (AS Boundary Routers Only)” on page 80 for more information.

Enabling OSPF

The default router identifier used by OSPF is the address of the first interface on the router encountered by `gated`. To set the router identifier to a specific address, specify the `routerid` interface statement in the Definition class of the `/etc/gated.conf` file.

NOTE

You must enable the OSPF protocol only for routers. When the OSPF protocol is enabled for a system, the system is treated as a router, and not a host, by other hosts.

You can enable the OSPF protocol using the `ospf` statement in the Protocol class of the `/etc/gated.conf` file. The clause `yes` (or `on`) informs `gated` to enable the OSPF protocol at this node and to process all OSPF packets arriving from other nodes. If you do not specify an OSPF line in your configuration file, `ospf no` is assumed. The clause `no` (or `off`) informs `gated` to disable the OSPF protocol on this node.

The following is an example to enable OSPF:

```
ospf yes { ... }
```

The following sections explain other statements defined for the OSPF protocol configuration.

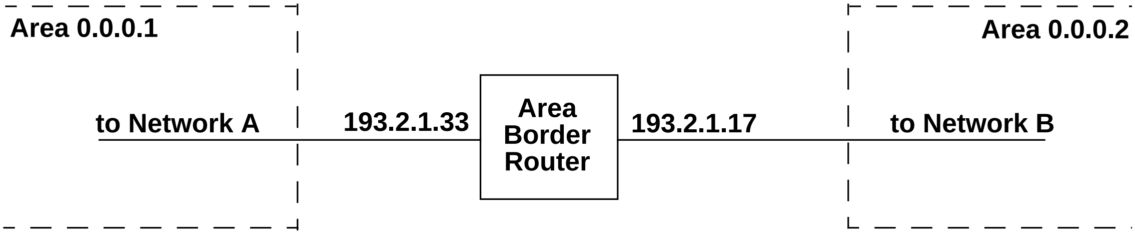
Defining Areas

Each OSPF router is associated with one or more areas. The `area` statement identifies an OSPF area. The value is in the form of a dotted quad, or a number between 1 and 4294967295. To define an area, you must specify the following:

- The addresses of the networks that make up the area.
- The router interfaces used to communicate with the area.

The configuration of an area border router contains multiple area definitions; a different router interface is defined for each area. Figure 3-4 shows an example of an area border router that is connected to area 0.0.0.1 through interface 193.2.1.33 and to area 0.0.0.2 through interface 193.2.1.17.

Figure 3-4 Area Border Router Configuration Example



The following is an example of the area definitions in the router’s `/etc/gated.conf` file:

```
ospf yes {
      area 0.0.0.1 {
        interface 193.2.1.33 {
               ...
        } ;
       } ;
       area 0.0.0.2 {
         interface 193.2.1.17 {
               ...
         } ;
       } ;
   } ;
```

You can define various characteristics for an area and interfaces. The following sections describe the configuration statements that you can use in defining an area.

The networks Statement

The `networks` statement defines the address ranges that forms an OSPF area. This definition applies only to area border routers, where multiple areas are specified, and is required only if you need to compress a number of subnets using a network mask.

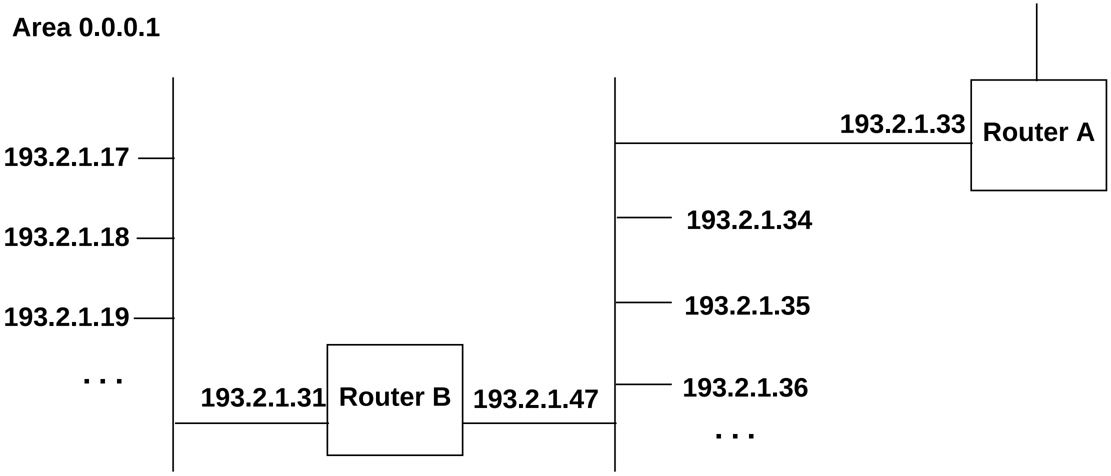
A network address followed by a hexadecimal bit mask specifies an IP address range in the `network` statement. For example, the following address range begins with the network address 193.2.1.16 and includes the first 15 addresses in that network (193.2.1.17 through 193.2.1.31):

```
193.2.1.16 mask 0xfffffff0
```

You can specify many separate networks in an address range. Area border routers advertise a single route for each address range.

Figure 3-5 shows an example of a router that is connected to area 0.0.0.1 through the interface 193.2.1.33. The attached network consists of addresses 193.2.1.33 through 193.2.1.47. The other network in the area consists of addresses 193.2.1.17 through 193.2.1.31.

Figure 3-5 Network Configuration Example



The following is an example of the network definition in the Router A's /etc/gated.conf file:

```
ospf yes
     area 0.0.0.1
       networks {
          193.2.1.16 mask 0xfffffff0 ;
          193.2.1.32 mask 0xfffffff0 ;
          } ;
          interface 193.2.1.33 {
             ...
          } ;
     } ;
 ...
```

The interface Statement

The interface statement in the OSPF protocol definition specifies the interface to use while communicating with the specified networks. You can specify the interface with an address (for example, 193.2.1.36), a domain or interface name (for example, lan0 or lan1), a wild card name (for example, lan*), or all (the order of precedence is address, name, wild card name, all). You can specify multiple interface statements using different clauses. If you specify a clause more than once, the instance with the most specific interface reference is used.

You can specify the cost clause optionally to define a cost of sending a packet on the interface. This cost is advertised as the link cost for this interface. See "Cost" on page 79 for more information on setting interface costs.

You can also enable or disable the interface definition. If you do not explicitly specify disable, an interface definition is enabled by default.

OSPF supports the following types of network interfaces:

- A multicast (or broadcast) network is a network that supports two or more attached routers and allows a single message to be addressed to a set of network nodes at the same time. An example of a multicast network is an Ethernet LAN.
- A non-broadcast multi-access (NBMA) network is a network that supports multiple attached routers, but does not support broadcasting of messages. An example of an NBMA network is an X.25 PDN.

- A point-to-point network is a network that joins a single pair of routers. An example of a point-to-point network is a 56-KB serial line.

The following sections describe each type of interface.

Multicast Interfaces On multicast networks, an OSPF router dynamically detects its neighbor routers through the OSPF Hello message. The following statements are defined for a multicast interface:

- `retransmitinterval` is the number of seconds between retransmission of link states, database descriptions, and link state request packets. This value must exceed the expected round-trip delay between any two routers in the network. A sample value for a LAN is 5 seconds.

  Default: None (you must specify a value)

  Range: Integer between 0 – 65535

- `transitdelay` is the number of seconds to transmit a Link State Update Packet over this interface. This value must take into account the transmission and propagation delays for the interface. It must be greater than 0. A sample value for a LAN is 1 second.

  Default: None (you must specify a value)

  Range: Integer between 1 – 65535

- `priority` specifies the priority of the router to be the designated router. You must configure this value only for interfaces to multi-access networks. This value specifies the priority of the router to be the designated router. When two routers attached to a network attempt to be the designated router, the one with the higher router priority value takes precedence.

  Default: None (you must specify a value for multi-access networks)

  Range: 8-bit unsigned integer between 0 – 255. 0 means that the router is ineligible to become a designated router on the attached network.

- `hellointerval` specifies the time interval (in seconds) for the transmission of OSPF Hello packets. Smaller intervals ensure that changes in network topology are detected faster. A sample value for an X.25 network is 30 seconds. A sample value for a LAN is 10 seconds.

Default: None (you must specify a value)

Range: Integer between 0 – 255

NOTE The `hellointerval` value must be the same for all OSPF routers.

- `routerdeadinterval` specifies the time interval (in seconds) for which the Hello packets are not received from a router before it is considered down or inactive by its neighbors. This value must be a multiple of the `hellointerval` value.

  Default: None (you must specify a value)

  Range: 0 – 65535

NOTE The `routerdeadinterval` value must be the same for all OSPF routers.

- You can use the password `authkey` to validate the protocol packets received on the router interface. The value can be 1 to 8 decimal digits separated by periods, a 1-byte to 8-byte hexadecimal string preceded by `0x`, or a string of 1 to 8 characters in double quotes.

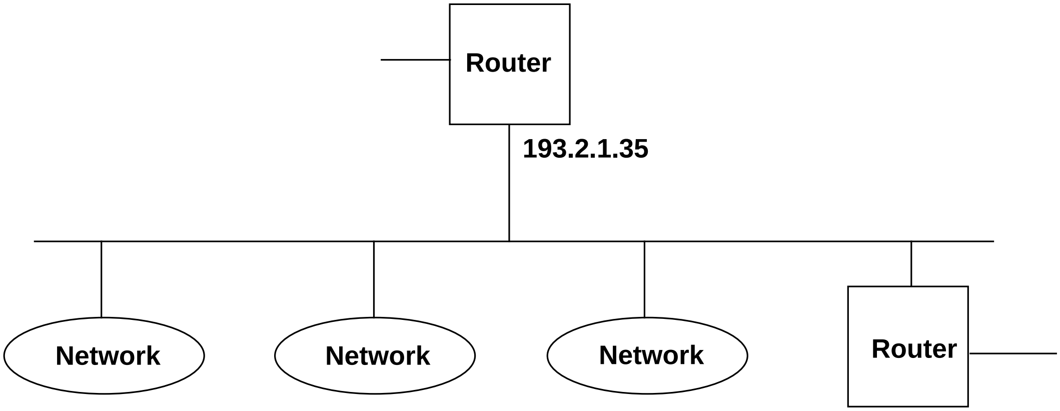
  Default: None

  Range: Up to 8 bytes

NOTE To set an `authkey` value, you must set the `authtype` clause to `1` or `simple` for the area. See “Authentication” on page 77 for more information on using OSPF authentication.

Figure 3-6 shows an example of a router that is connected to a multicast network through the interface 193.2.1.35.

Figure 3-6 Multicast Router Interface Example



The following is an example of the multicast interface definition in the router's `/etc/gated.conf` file:

```
interface 193.2.1.35 cost 5 {
      enable ;
      priority 15 ;
      hellointerval 5 ;
      routerdeadinterval 20 ;
      retransmitinterval 10 ;
    } ;
```

Non-Broadcast Multi-Access (NBMA) Interface On NBMA networks, you must supply the configuration information, including the routers that are attached to the network, so that the OSPF's Hello protocol communicates with neighbor routers. An NBMA interface definition applies to both X.25 network interfaces and systems that do not support IP multicasting. You can define an NBMA interface using the multicast `interface` statements, with the following additions:

- You must specify the clause `nonbroadcast` in the `interface` statement.
- `pollinterval` specifies a rate at which hellos are sent when a neighboring router becomes inactive (a router is considered inactive when hellos are not received from the router for the amount of time

specified by the `routerdeadinterval` definition). The value of `pollinterval` must be larger than the value of `hellointerval`. A sample value for an X.25 network is 2 minutes.

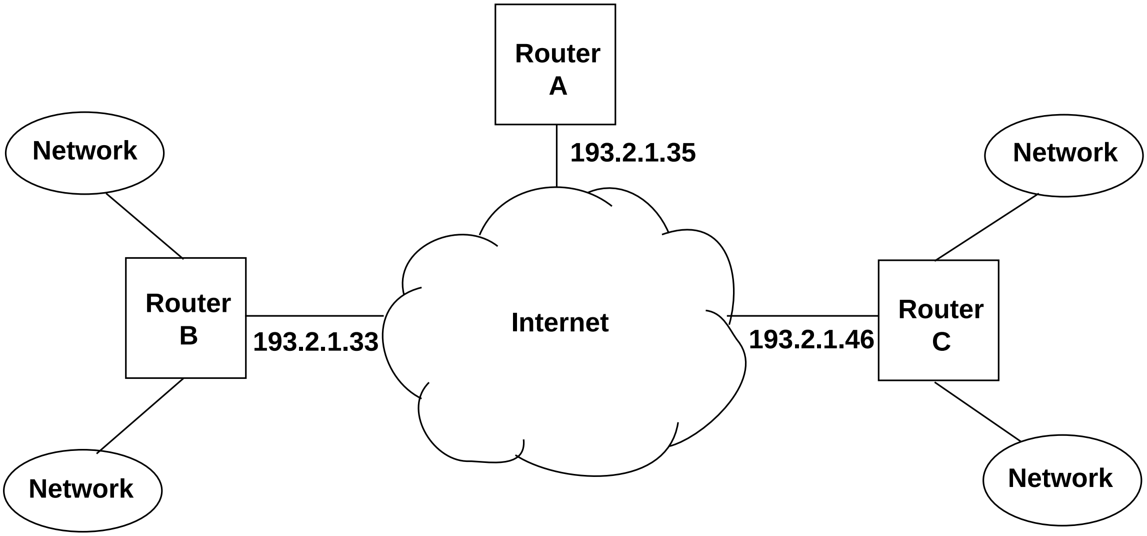
Default: None (you must specify a value)

Range: 0 – 255

- `routers` specify the list of routers attached to the non-broadcast network. Routers are defined by their IP interface addresses. You must define the routers that are eligible to be designated routers as `eligible`.

Figure 3-7 shows an example of a router (A) that is connected to an NBMA network through the interface 193.2.1.35. Two other routers are also attached to the network: router B is connected through the interface 193.2.1.33 and C is connected through the interface 193.2.1.46. B and C are eligible to be designated routers.

Figure 3-7 Non-Broadcast Router Interface Example



The following is an example of the non-broadcast interface definition in router A’s `/etc/gated.conf` file:

```
interface 193.2.1.35 nonbroadcast cost 5 {
      routers {
        193.2.1.33 eligible ;
        193.2.1.46 eligible ;
      } ;
      priority 15 ;
      hellointerval 5 ;
      routerdeadinterval 20 ;
      retransmitinterval 10 ;
      pollinterval 20 ;
} ;
```

Point-to-Point Interfaces On point-to-point networks, an OSPF router dynamically detects its neighbor router by sending OSPF Hello packets. The following statements are defined for a point-to-point interface:

- `retransmitinterval` is the time interval (in seconds) between the retransmission of link states, database descriptions, and link state request packets. This value must exceed the expected round-trip delay between any two routers in the network. A sample value for a LAN is 5 seconds.

  Default: None (you must specify a value)

  Range: 0 – 65535

- `hellointerval` specifies the time interval (in seconds) between transmission of OSPF Hello packets. Smaller intervals ensure that changes in network topology are detected faster. A sample value for an X.25 network is 30 seconds. A sample value for a LAN is 10 seconds.

  Default: None (you must specify a value)

  Range: 0 – 255

NOTE The `hellointerval` value must be the same for all OSPF routers.

- routerdeadinterval specifies the time interval (in seconds) for which the Hello packets are not received from a router before it is considered down or inactive by its neighbors. This value must be a multiple of the hellointerval value.

  Default: None (you must specify a value)

  Range: 0 – 65535

NOTE The routerdeadinterval value must be the same for all OSPF routers.

You can define a point-to-point interface with or without a nonbroadcast clause. If you specify the nonbroadcast clause, then you must define the pollinterval statement.

- pollinterval specifies a rate at which hellos are sent when a neighboring router becomes inactive (a router is considered inactive when hellos have not been received from the router for the amount of time specified by the routerdeadinterval definition). The value of pollinterval must be larger than the value of hellointerval. A sample value for an X.25 network is 2 minutes.

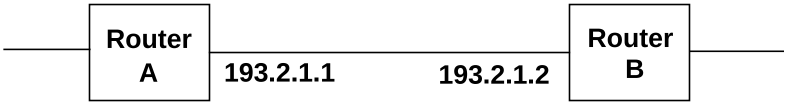
  Default: None (you must specify a value)

  Range: 0 – 255

  If the device at the other end of the point-to-point network is not an OSPF router, you can prevent sending Hello packets to that OSPF router using the stubhosts statement. stubhosts specifies the IP address or domain name of the non-OSPF host. The cost of sending a packet to the host must also be specified (in most cases, the host has only a single connection to the network, so the cost configured has no effect on routing).

Figure 3-8 shows an example of a router (A) that is connected to a non-broadcast, point-to-point network through interface 193.2.1.1.

Figure 3-8 Point-to-Point Router Interface Example



The following is an example of the interface definition in router A's `/etc/gated.conf` file:

```
interface 193.2.1.1 nonbroadcast cost 5 {
      hellointerval 30 ;
      routerdeadinterval 30 ;
      retransmitinterval 30 ;
      pollinterval 30 ;
} ;
```

If the router A is connected to a multicast, point-to-point network, you must omit the `nonbroadcast` clause and the `pollinterval` statement.

Stub Areas

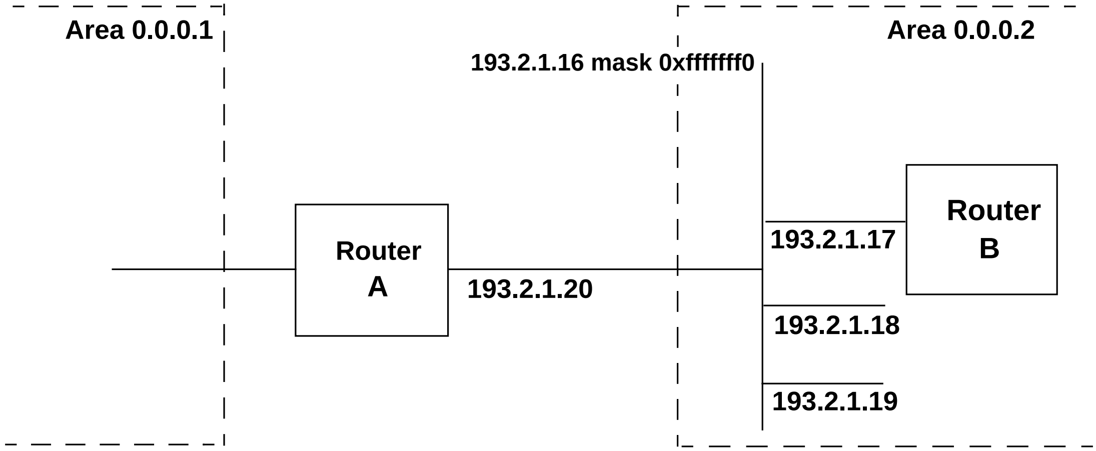
By default, AS external link advertisements (routes to destinations outside the AS) are propagated to every router in every area in the AS. You can configure certain OSPF areas as stub areas. AS external link advertisements are not flooded through stub areas. This reduces the size of the topology database that must be maintained by internal routers in the stub area and reduces the protocol traffic through the area. For example, if all the inter-area traffic for an area must go through a single router, then it is not necessary for all routers in the area to receive inter-area routing information.

An area border router advertises a default route in the stub area as the summary of all the IP destinations that are reachable outside the AS. Summary link advertisements (routes to destinations outside the area but within the AS) are still sent into the stub area.

The `stub` statement specifies that the area is a stub area. You can optionally define a `cost` clause to specify the cost associated with the default route that is advertised in the stub area.

Figure 3-9 shows an example of an area border router that is connected to area 0.0.0.2 through interface 193.2.1.20. Because all traffic in and out of the area 0.0.0.2 must pass through router A, it is not necessary for the area's internal routers, such as router B, to receive inter-area routing information.

Figure 3-9 Area Border Router Configuration Example



The following is an example of the stub area definition in the router's `/etc/gated.conf` file:

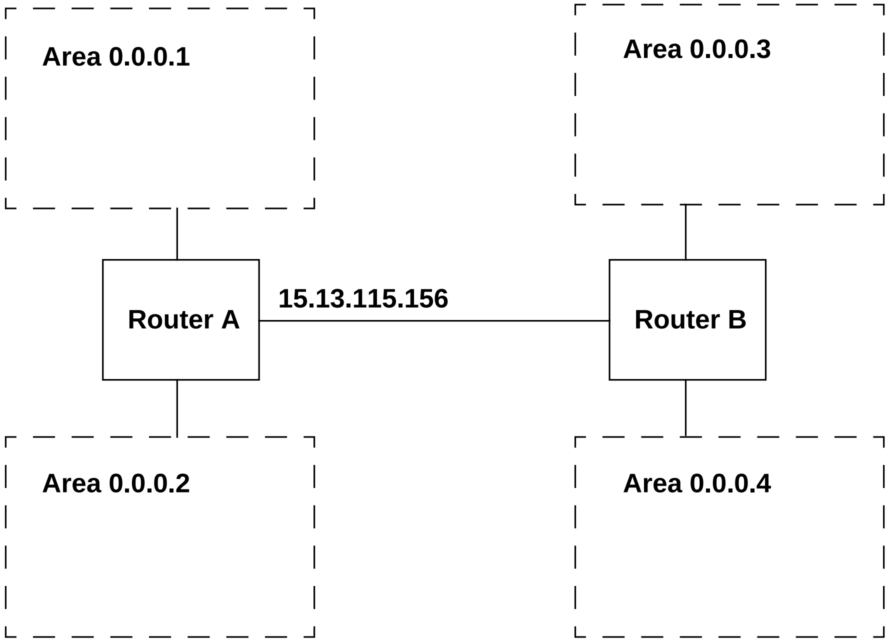
```
OSPF yes {
  area 0.0.0.2 {
    stub cost 5 ;
    networks {
      193.2.1.16 mask 0xfffffff0 ;
    } ;
    interface 193.2.1.20 nonbroadcast cost 5 {
      enable ;
      routers {
        193.2.1.17 eligible ;
       } ;
      priority 5 ;
      hellointerval 5 ;
      routerdeadinterval 20 ;
      retransmitinterval 10 ;
      pollinterval 20 ;
    } ;
  } ;
} ;
```

Defining Backbones

The OSPF backbone distributes routing information between areas. You can define backbones with the same statements and clauses as areas. You need not define the `stub` statement for a backbone. The backbone statement is used to define a router as a backbone router. If an OSPF internal or area boarder router is also a backbone router, the backbone statement must follow the `area` statements in the `/etc/gated.conf` file. Whenever you configure an area border router (a router connected to multiple areas), you must provide the backbone.

Figure 3-10 shows an example of two area border routers that form part of a backbone. Router A has interfaces to both area 0.0.0.1 and area 0.0.0.2, while router B has interfaces to areas 0.0.0.3 and 0.0.0.4. Router A is connected to router B through interface 15.13.115.156.

Figure 3-10 Backbone Configuration Example



The following is an example of the backbone router definition in router A's `/etc/gated.conf` file:

```
backbone {
      interface 15.13.115.156 {
        enable ;
        transitdelay 20 ;
        priority 20 ;
        hellointerval 30 ;
        routerdeadinterval 120 ;
        retransmitinterval 60 ;
        } ;
      } ;
```

If the router is directly attached via a point-to-point interface to a host that is not running OSPF, you can prevent sending OSPF Hello packets to the host. Do this by specifying the subhost statement with the host's address. You can optionally define the cost.

NOTE Backbones must be directly connected or contiguous. In some gated implementations, you can configure a virtual link to join non-contiguous backbone routers. HP-UX systems do not support virtual links.

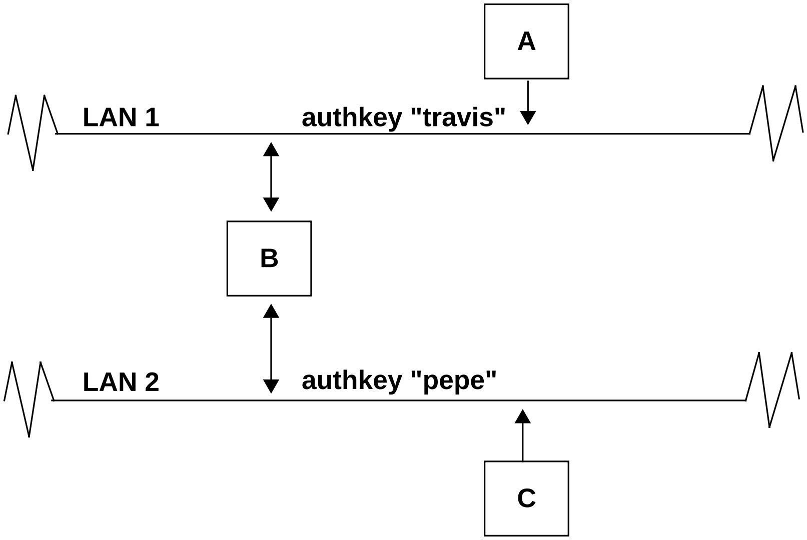
Authentication

The OSPF protocol allows you to authenticate packets containing routing information. You can configure the authentication on a per-area basis; You can use different authentication methods in different areas.

gated supports a simple password authentication method. You can also choose to have no authentication. You can use the authtype statement to define the authentication method used for the area. 0 or none specifies that routing exchanges in the area are not authenticated. 1 or simple specifies that network passwords of up to 64 bits (8 characters) are used to authenticate packets received from routers in the area.

In the simple password authentication method, all routers that interface to a given network use the same password. The password is defined by the authkey statement in the router's interface definition. If you do not configure a router with the same password as other routers in the network, other networks discard the router's packets. The password is configured on a per-interface basis. If a router has interfaces to more than one network, different passwords can be configured. This is illustrated in Figure 3-11.

Figure 3-11 Simple Password Authentication



The following example shows an `authtype` statement that enables a simple password authentication for the routers in the area and an `authkey` statement in the interface definition that defines a password (`travis`) to validate protocol packets received by the router:

```
area 0.0.0.1 {
    authtype simple ;
    networks {
      193.2.1.16 mask 0xfffffff0 ;
      193.2.1.32 mask 0xfffffff0 ;
      } ;
    interface 193.2.1.35 nonbroadcast cost 5 {
      routers {
        193.2.1.33 eligible ;
        193.2.1.46 eligible ;
      } ;
      priority 15 ;
      enable ;
      hellointerval 5 ;
      routerdeadinterval 20 ;
```

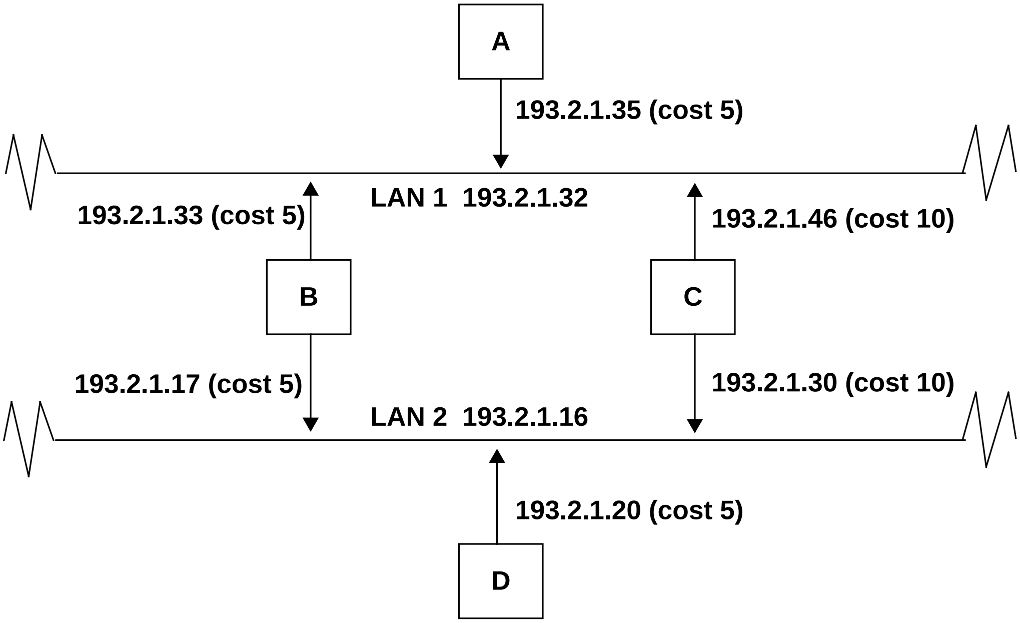
```
            retransmitinterval 10 ;
            pollinterval 20 ;
            authkey  " travis "  ;
        } ;
} ;
```

Cost

The outbound side of each router interface is associated with a configurable cost. Lower cost interfaces are more likely to be used in forwarding data traffic. Cost values are assigned at the discretion of the network or system administrator. While the value is arbitrary, it must be a function of throughput or capacity of the interface: the higher the value, the lower the throughput or capacity. Thus, the interfaces with the highest throughput or capacity must be assigned lower cost values than other interfaces. Interfaces from networks to routers have a cost of 0.

Figure 3-12 shows an example network where costs are specified for each interface.

Figure 3-12 Cost Configuration Example



In Figure 3-12, there are two possible packet routes between nodes A and D: one route goes through node B and the other route goes through node C. The cost of each route is calculated as follows:

Node A to node B and node B to node D: 5+5 = 10

Node A to node C and node C to node D: 5+10 = 15

The lowest cost OSPF path between nodes A and D is therefore through node B. However, packets are rerouted through node C if there is a link failure between node B and LAN 2.

You can optionally define `cost` in the following places in the `/etc/gated.conf` file:

- In a `defaults` statement in the OSPF protocol configuration, which applies only to AS boundary routers. This cost definition applies to routes to destinations outside the AS. These routes may have been derived from other routing protocols, such as EGP. See “AS External Routes (AS Boundary Routers Only)” on page 80 for more information.
- In the `export` statement in the Control class in the `/etc/gated.conf` file, which applies only to AS boundary routers. This cost definition applies to routes that are exported from the AS boundary router to routers in other autonomous systems.
- In the stub area definition of the OSPF protocol configuration. This cost definition specifies the cost of the default summary link that is advertised in the area.
- In the `stubhosts` definition of the OSPF protocol configuration. This cost definition specifies the cost of a point-to-point interface between the router and a non-OSPF host.
- In the `subhosts` definition of the OSPF protocol configuration. This cost definition specifies the cost of a point-to-point interface between the backbone router and a non-OSPF host.

AS External Routes (AS Boundary Routers Only)

AS external (ASE) routes are paths to destinations that are outside the AS. Most ASE routes are routes to specific destinations. AS boundary routers specify the ASE routes through another routing protocol, such as EGP, or through configured routes.

gated supports the use of route information from other autonomous systems that use other routing protocols, such as EGP. AS boundary routers send AS external link advertisements and flood the AS with advertisements (with the exception of configured stub areas). A single AS external link advertisement is sent for each external route that the AS boundary router has learned about.

Externally defined routing information and OSPF routing information are maintained separately. In addition, you can tag the externally defined routing information, to identify and store the source of the information along with the route information.

Statements in the Control class of the /etc/gated.conf file control the importing of routes from routing protocols to a gated forwarding table and the exporting of routes from the gated forwarding table. See "Importing and Exporting Routes" on page 97 for more information.

You must specify the defaults statements in the OSPF protocol configuration only for AS boundary routers. These statements specify how external routing information is handled by the OSPF protocol. You can define the following in the defaults statements:

- preference specifies the preference value given to the ASE routes imported from other autonomous systems. The preference value determines the order of routes to the same destination in the routing table. gated allows one route to a destination per protocol for each AS. In case of multiple routes, the route used is determined by the lowest preference value (see "Specifying Route Preference" on page 94 for more information). If a preference value is not specified, ASE routes are imported with a preference of 150.

  Default: 150

  Range: 0 (most preferred) – 255 (least preferred)

- cost specifies the cost associated with an OSPF route that is exported to other AS boundary routers.

  Default: 0

  Range: 0 – 65535

- tag specifies an OSPF tag placed on all routes exported by gated into OSPF. You can tag each external route by the AS boundary router to identify the source of the routing information. The tag value can be an unsigned 31-bit number. You can also specify tag as *as_tag*, where *as_tag* is an unsigned 12-bit number that is automatically assigned.

- type determines how ASE routes imported into OSPF are treated. Type 1 routes must be routes from internal gateway protocols with external metrics that are directly comparable to OSPF metrics. When OSPF selects a route, OSPF uses the type 1 route's external metric and adds the OSPF internal cost to the AS border router. Type 2 routes must be routes from external gateway protocols with metrics that are not comparable to OSPF metrics. When OSPF selects a route, OSPF ignores a type 2 route's metric and uses only the OSPF internal cost to the AS border router.

  Default: 1

- exportlimit specifies the rate at which ASE routes are imported into the gated routing table for each exportinterval.

  Default: 100 (ASE routes)

  Range: 0 – 65535

- exportinterval specifies the interval (in seconds) between ASE exports into OSPF.

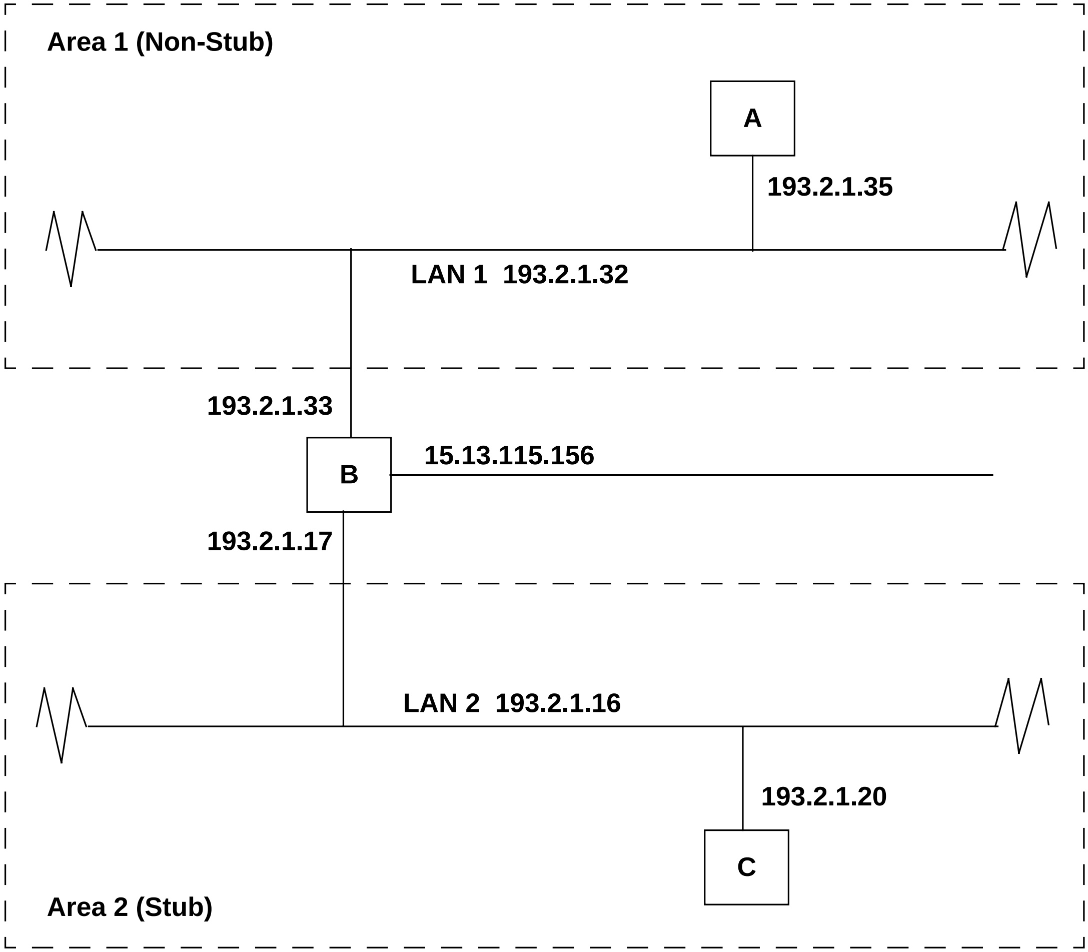
  Default: 1 (second)

  Range: 0 – 2147483647

Sample OSPF Configuration

Figure 3-13 shows an example of two areas. Area 1 is a non-stub area, while area 2 is configured as a stub area. Node B is an area border router between the two areas.

Figure 3-13 OSPF Sample Configuration



A: Internal Router (Non-Stub Area)

Set up `/etc/gated.conf` as follows:

```
# Router A Configuration (non-stub area)
OSPF yes {
  area 0.0.0.1 {
    interface 193.2.1.35 cost 5 {
      priority 5 ;
      enable ;
      hellointerval 5 ;
      routerdeadinterval 20 ;
      retransmitinterval 10 ;
    } ;
  } ;
} ;
```

The configuration for the internal router A is for a multicast interface. For an NBMA interface, you can set the configuration in `/etc/gated.conf` as follows:

```
# Router A Configuration (non-stub area)
OSPF yes {
  area 0.0.0.1 {
    interface 193.2.1.35 nonbroadcast cost 5 {
      routers {
        193.2.1.33 eligible ;
   } ;
      priority 5 ;
      enable ;
      hellointerval 5 ;
      routerdeadinterval 20 ;
      retransmitinterval 10 ;
      pollinterval 20 ;
    } ;
  } ;
} ;
```

NOTE If you use IP multicasting in an area, every router and all the intermediate network devices in that area must support IP multicasting.

B: Area Border Router

Set up `/etc/gated.conf` as follows:

```
OSPF yes {
  defaults {
    cost 5 ;
  } ;
  area 0.0.0.1 {
    interface 193.2.1.33 cost 5 {
      priority 15 ;
      enable ;
      hellointerval 5 ;
      routerdeadinterval 20 ;
      retransmitinterval 10 ;
    } ;
 } ;

area 0.0.0.2 {
    interface 193.2.1.17 cost 5 {
      priority 15 ;
      enable ;
      hellointerval 5 ;
      routerdeadinterval 20 ;
      retransmitinterval 10 ;
    } ;
  } ;

backbone {
  interface 15.13.115.156 cost 5 {
    enable ;
    priority 10 ;
    hellointerval 5 ;
    routerdeadinterval 20 ;
    retransmitinterval 10 ;
  } ;
} ;
} ;
```

C: Internal Router (Stub Area)

Set up /etc/gated.conf as follows:

```
OSPF yes {
  area 0.0.0.2 {
    stub cost 5 ;
    interface 193.2.1.20 cost 5 {
      priority 5 ;
      enable ;
      hellointerval 5 ;
      routerdeadinterval 20 ;
      retransmitinterval 10 ;
    } ;
  } ;
} ;
```

The routing table on node A contains routes to 193.2.1.32 and 193.2.1.16. The routing table on node C in the stub area contains routes only to LAN 1 and a default router.

Accessing the OSPF MIB

HP's `gated` also provides `ospfagt`, an OSPF Simple Management Network Protocol (SNMP) subagent that supports the OSPF MIB (Management Information Base) (see RFC 1253). The `ospfagt` subagent works with the HP SNMP Agent, `snmpdm`. If you use an SNMP manager utility to manage your network, such as HP's OpenView Network Node Manager, you can also use HP's OSPF SNMP subagent.

To start `ospfagt` automatically during system bootup, set the environment variable `OSPFMIB` to `1` in the file `/etc/rc.config.d/netdaemons`.

To manually start `ospfagt`, type the following command at the command prompt:

```
/usr/sbin/ospfagt
```

`gated` must be running before `ospfagt` is started. Both `gated` and `ospfagt` must be running to retrieve OSPF MIB objects.

To load the OSPF MIB, select `Load/Unload SNMP:MIBS ...` from the Options menu of OpenView.

Configuring RDP

You can use Router Discovery Protocol (RDP), a standard protocol, to inform hosts of the presence of routers to which they can send packets. You can also use RDP instead of host wiretapping routing protocols (for example, RIP). It is used instead of, or in addition to, having statically configured default routes in hosts.

RDP consists of two portions: the **server** portion, which runs on routers, and the **client** portion, which runs on hosts. `gated` treats these portions as separate protocols; therefore, you can enable only one of them at a time. These portions are described in detail in the subsequent sections.

For a description of the RDP configuration statements, type `man 4 gated.conf` at the HP-UX prompt.

RDP Server

The RDP server runs on routers, and announces the routers' existence to hosts periodically by multicasting or broadcasting a **router advertisement**. The advertisement is sent over an RDP server enabled physical interface. Each router advertisement contains a list of all addresses on a physical interface and their preference for being used as a default router. You can configure the length of time (the lifetime) for which addresses must remain on the list.

At first, router advertisements occur every few seconds, and then, they start occurring few minutes. You can configure the minimum and maximum intervals for router advertisements to occur. Also, a host can send a **router solicitation**, requesting an advertisement. The router responds with a unicast router advertisement unless a multicast or broadcast advertisement is due to occur.

On hosts that support IP multicasting, router advertisements are sent by default to the all-hosts mulicast address `224.0.0.1`. You can also configuration RDP to use broadcasting to send router advertisements. This is useful when a particular host does not support IP multicasting, or when one or more hosts on an attached network do not support IP multicasting. If router advertisements are sent to the all-hosts multicast address, or if an interface is configured for the limited-broadcast address `255.255.255.255`, the advertisements contain all the IP addresses configured on the physical interface. If advertisements are sent to a net or subnet broadcast, only that network's or subnet's address is included in the advertisement.

An example of the `routerdiscovery server` statement is as follows:

```
routerdiscovery server yes {
   interface lan1 lan2
   maxadvinterval 5 ;
   address 193.2.1.17 193.2.1.33 193.2.1.46
   broadcast
   preference 50 ;
} ;
```

In the example, the server is enabled on the physical interfaces `lan1` and `lan2`, and the IP addresses 193.2.1.17, 193.2.1.33, and 193.2.1.46 are included in all the router advertisements. Also, the addresses have a preference of 50.

RDP Client

The RDP client runs on hosts, listening for router advertisements over the all-hosts multicast address `224.0.0.1` (if it supports IP multicasting) or on the physical interface's broadcast address (if the host does not support multicasting). When a host starts up or is reconfigured, it sends certain router solicitations requesting advertisements. When it sends the solicitations, it sends them to the all-routers multicast address `224.0.0.2` or to the interface's broadcast address (if multicasting is not supported).

When the RDP client receives a router advertisement, the host installs a default route to each of the addresses listed in the advertisement. If the advertisement has a preference of `ineligible` (that is, the addresses in the advertisement are not eligible to be the default route for any hosts), or if the addresses are not on an attached physical interface, the route is marked as unusable but is still retained. If the preference is usable, then that route is among the routes considered. The route with the highest preference is used. If more than one route with the same preference is received, the one with the lowest IP address is used. The default routes are not exportable to other protocols.

If an RDP client receives a router advertisement with a zero lifetime (that is, the addresses in the advertisement are no longer valid), the host deletes all the routes with next-hop addresses learned from that router.

The host also deletes any routes learned from ICMP `redirects` pointing to the invalid addresses. Also, if a router advertisement is not received before the addresses listed by the host becomes invalid (that is, before its lifetime expires), the routes learned from that router are deleted by the host.

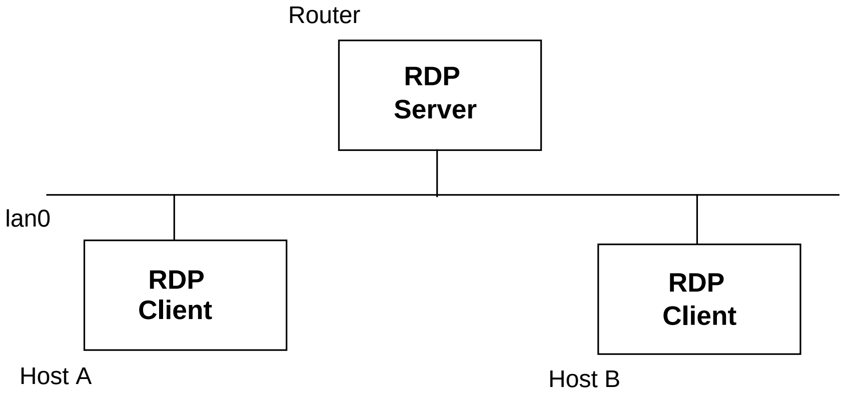
An example of the `routerdiscovery client` statement is as follows:

```
routerdiscovery client yes {
    preference 50 ;
    interface lan0
    broadcast ;
} ;
```

In the example, the client is enabled on physical interface `lan0`, and the default routes are given a preference of 50.

A simple example of an RDP server and two RDP clients is shown in Figure 3-14.

Figure 3-14 **RDP Server and Clients Example**



Customizing Routes

gated maintains the routing table in user space, and synchronizes this table with the kernel routing table. This section describes statements for setting up customized routes in the Static class of the gated configuration file, /etc/gated.conf. You can use these statements to specify default routers, static routes, passive interfaces, and routing metrics for interfaces.

Specifying a Default Router

A static route provides a specific destination for network packets. The static route is a network address or a host address through a router. This route is installed in the kernel's routing table. The following is an example of a static route for the default route:

```
static {
    default gateway 15.13.114.196 retain ;
    } ;
```

The retain qualifier ensures that the entry is not deleted when gated exists.

Installing Static Routes

The static statement specifies a router or an interface in the kernel routing tables. The following is an example of a static route:

```
static {
    193.2.1.32 mask 0xfffffff0 gateway 193.2.1.30
    preference 8 retain ;
    } ;
```

If you specify an export statement for the default route, the route is passed on to other routers. If you specify only the static statement and not an export statement, then the default route is not passed on as a route to other routers. This is considered a passive default route, and is used only by the host where this gated is running. The retain clause retains the route in the kernel even after gated shuts down.

Setting Interface States

gated times out routes that pass through interfaces not receiving any RIP, OSPF, or BGP packets. The passive clause in the interface statement of the Static class prevents gated from changing the preference of a route to the interface if routing information is not received for the interface. HP recommends that you use the passive clause for all interfaces.

Specifying Tracing Options

Trace options specify the desired level of tracing output from gated. Tracing output provides useful system information for setting up a node on the network. Use trace options to set up a node and to send a certain type of tracing to a log file. You can specify tracing in the following ways:

- In a protocol statement in the /etc/gated.conf configuration file.
- In the Trace class of the /etc/gated.conf configuration file.
- On the command line with the -t option when starting gated.

Trace information is appended to the trace file unless you specify replace. Command-line options are useful for tracing events in gated before the configuration file is read.

NOTE In gated 3.5.9, the two Trace class statements (tracefile and traceoptions) are combined into one traceoptions statement. Therefore, the tracefile statement is eliminated. For details about the new syntax, type man 4 gated.conf at the HP-UX prompt.

Table 3-2 shows the gated.conf global trace options related to protocols.

Table 3-2 Protocol-Related Global Trace Options for gated Configuration Files

Option	Effect
state	Traces the state machine transitions in the protocols.
normal	Traces the normal protocol events (abnormal protocol events are always traced).
policy	Traces the application of protocol and user-specified policies to routes that are imported and exported.
task	Traces the system interface and processing associated with this protocol or peer.

Table 3-2 Protocol-Related Global Trace Options for gated Configuration Files (Continued)

Option	Effect
`timer`	Traces the timer usage by this protocol or peer.
`route`	Traces all routing table changes for routes installed by this protocol or peer.
`general`	A combination of `normal` and `route`.
`all`	Enables all tracing options.

Some of the options specified in Table 3-2 do not apply to all of the protocols. For more information on options applicable for each protocol and for the different trace options available within the configuration file, type `man 4 gated.conf` at the HP-UX prompt. See “Troubleshooting gated” on page 101 for more information on tracing options.

Specifying Route Preference

gated maintains a routing table that consists of the route information learned from OSPF and from other active routing protocols, such as RIP or EGP. You can also configure static routes in the /etc/gated.conf file with one or more static clauses (see “Installing Static Routes” on page 90 for more information).

The gated routing pool can therefore contain multiple routes to a single destination. When multiple routes exist, the route chosen by gated is determined by the following (in order of precedence):

- The preference value associated with the route. The preference value is a number in the range from 0 (most preferred) to 255 (least preferred). Routes from different sources have different default preference values. For example, OSPF routes within a given AS have a preference value of 10. Table 3-3 shows the default preference values of various types of routes.
- If multiple routes use the same protocol and have the same preference value, the route with the lowest metric or cost is chosen.
- If metric or cost is the same, the router with the lowest IP address is chosen.

Table 3-3 Default Preference Values of Routes

Route Type	Preference	/etc/gated.config Configuration
Interface routes	0	Can be changed with interface statement in Interface class.
OSPF inter- and intra-areas	10	Cannot be changed.
Internal default	20	Generated by BGP or EGP when routing information is learned from a peer.
ICMP Redirect	30	Can be changed with redirect statement in Protocol class.
SNMP	50	Can be changed in SNMP statement in Protocol class.

Table 3-3 **Default Preference Values of Routes (Continued)**

Route Type	Preference	/etc/gated.config Configuration
Static routes	60	Can be changed in static statement in Static class.
RIP	100	Can be changed with import statement in Control class.
Point-to-point interface	110	Can be changed with interface statement in Interface class.
“Down” interface	120	Can be changed with interface statement in Interface class.
OSPF ASE	150	Can be changed in defaults statement in OSPF protocol definition and with import statement in Control class.
BGP	170	Can be changed with import statement in Control class.
EGP	200	Can be changed with import statement in Control class.
Kernel remnant	254	These are the static routes that are retained in the kernel after gated is stopped. Preference value cannot be configured.

You can define preference in the /etc/gated.conf file configuration file in the following instances:

- In the static route definition in the Static class. This preference definition sets the preference for static routes. (See “Customizing Routes” on page 90 for more information.) If this option is not set, the preference values for static routes is 60.
- In the interface statement options in the Interface class. This preference definition sets the preference for routes to this interface. If this option is not set, the preference value is 0. For more information, type man 4 gated.conf at the HP-UX prompt.

- In a `defaults` statement in the OSPF protocol configuration. This preference definition specifies the preference value of ASE routes that are imported into OSPF. See “AS External Routes (AS Boundary Routers Only)” on page 80 for more information. ASE routes are imported into OSPF with a default preference of 150.
- In an `import` statement in the Control class of the `/etc/gated.conf` file. This preference definition overrides any preference defined in the `defaults` section of the OSPF protocol configuration. See “AS External Routes (AS Boundary Routers Only)” on page 80 and “Importing and Exporting Routes” on page 97 for more information.

Importing and Exporting Routes

You can propagate routes from one routing protocol to another using the `import` and `export` control statements. Routes are imported into a `gated` forwarding table and exported out to the routing protocols.

For more information on `import` and `export` statements, type `man 4 gated.conf` at the HP-UX prompt.

The import Statement

`import` statements restrict or control routes that are imported to the `gated` forwarding table. When routes are imported to the `gated` forwarding table, you can export them to the routing protocols. You can use `import` statements to perform the following tasks:

- Prevent routes from being imported into the `gated` forwarding table by using a `restrict` clause.
- Assign a preference value to use when comparing a route with other routes from other protocols. The route with the lowest preference is installed in the `gated` forwarding table. The individual protocols configure the default preferences.

The format of the `import` statement varies depending on the protocol from which you are importing routes.

With OSPF, you can apply `import` statements only to OSPF ASE routes. All OSPF intra-area and inter-area routes are imported into the `gated` forwarding table with an assigned preference of 10.

The export Statement

`export` statements determine the routes that are exported from the `gated` forwarding table to the routing protocols. You can also restrict the routes that are exported and assign metrics (values used for route selection) to them. These metrics are applied only after the routes are exported.

The format of the `export` statement varies depending on the original protocol used to build the routes that you are exporting, and the protocol to which you are exporting routes.

Examples of import and export Statements

The following import statement imports a BGP route for network 195.1.1 to the gated forwarding table with a preference of 15:

```
import proto bgp as 1 {
      195.1.1 mask 0xffffff00 preference 15;
   };
```

The following export statement exports to OSPF the ASE route that was imported to the gated forwarding table in the previous example. The route was originally built by BGP and the destination of the route is network 195.1.1.

```
export proto ospfase type 1 /* Export an ASE route to OSPF */
       proto bgp as 1 {/*route came from BGP and AS 1*/
       195.1.1 ; /* the route is to network 195.1.1 */
     };
   };
```

Starting gated

To start gated, complete the following steps:

1. Set the environment variable GATED to 1 in the file /etc/rc.config.d/netconf to start gated automatically upon system startup.
2. Reboot your system, or issue the following command to run the gated startup script:

   ```
   /sbin/init.d/gated start
   ```

You can also start gated by running the command gdc start. The following message appears to indicate that gated has started:

```
gated started, pid 29777
```

where 29777 is the process ID (pid) of the gated process. You can specify the command-line arguments for starting gated with the GATED_ARGS environment variable in the file /etc/rc.config.d/netconf. Table 3-4 lists the commonly used command-line options for gated.

Table 3-4 Command Line Options for gated

Flag	Effect
-t	When used alone, -t causes gated to log all error messages and route changes. It turns on the general trace option automatically. When -t is followed by one or more trace options, only those options are turned on. (See “Specifying Tracing Options” on page 92 for more information.) Multiple trace options are separated by commas. The -t flag must immediately precede the other flags.
-C	Specifies that the configuration file will be parsed for syntax errors. gated exits with a status of 1 if there are any errors and 0 (zero) if there are no errors.
-c	Specifies that the configuration file will be parsed for syntax errors. A dump file /var/tmp/gated_dump is created if there are no errors. Only the trace option general is logged. See “Specifying Tracing Options” on page 92 for a detailed description of all the tracing options.

Table 3-4 Command Line Options for gated (Continued)

Flag	Effect
-n	Specifies that gated will not modify the kernel's routing tables.

For more information about the command-line options, type `man 1M gated` at the HP-UX prompt.

Verifying That gated Is Running

Issue the following command to determine if gated is running:

```
/usr/bin/ps -ef | /usr/bin/grep gated
```

This command reports the process identification (PID), current time, and the command invoked (gated). Following is an example output:

```
daemon   4484   1  0  Feb 18     ?      0:00 gated
```

You can also check if gated is running by issuing the following command:

```
gdc running
```

The following message appears to indicate that gated is running:

```
gdc is running  (pid 1312)
```

If gated is not running, the following message appears:

```
gated is not running
```

Troubleshooting gated

This section describes the following techniques for troubleshooting gated and some common problems encountered with gated operation:

- “Checking for Syntax Errors in the Configuration File” on page 101
- “Tracing gated Activity” on page 101
- “Operational User Interface for gated – gdc” on page 102
- “The gated Routing Table” on page 103
- “The ripquery Tool” on page 103
- “The ospf_monitor Tool” on page 103
- “Common Problems” on page 104

Checking for Syntax Errors in the Configuration File

After creating or modifying a gated configuration file, you must start gated from the command line with the -C option. This option parses the the configuration file for syntax errors.

Tracing gated Activity

gated prints information about its activity in the form of tracing output. This information includes routes that gated reads, adds, and deletes from the kernel routing table, as well as packets sent and received.

You can specify tracing either with the gated -t command-line option or with the traceoptions statement in the /etc/gated.conf file. Using any of the following combinations, you can determine where the tracing output is printed and whether tracing is performed:

- If you specify trace options and a trace file, tracing output is printed to the log file.
- If you specify trace options but do not specify a trace file, tracing output is printed on the display where gated was started.
- If you specify a trace file but do not specify any trace options, no tracing takes place.

NOTE

In gated 3.5.9, the two statements in the Trace class (tracefile and traceoptions) are combined into one traceoptions statement. Therefore, the tracefile statement is eliminated. For details about the new syntax, type man 4 gated.conf at the HP-UX prompt.

After tracing is started to a file, you can rotate the trace file. Receipt of a SIGUSR1 signal stops gated from tracing and closes the trace file. The trace file can then be moved out of the way. To send a SIGUSR1 signal to gated, issue one of the following commands:

```
/usr/bin/kill -SIGUSR1 pid
```

or

```
/usr/bin/kill -USR1 pid
```

where *pid* is the gated's process ID, determined by invoking the command ps -ef | grep gated.

A subsequent SIGUSR1 signal starts tracing again to the same trace file. If the trace options are changed before tracing is started again, the new options are used.

NOTE

You cannot use the SIGUSR1 signal if you did not previously specify tracing to a file while starting gated.

Operational User Interface for gated – gdc

gdc provides a user-oriented interface for the operation of gated. It provides the following functions:

- Starting and stopping gated.
- Delivery of signals to manipulate gated.
- Maintenance and checking the gated configuration file for syntax.
- Production and removal of state dumps and core dumps.

gdc determines the state of gated and produces a reliable exit status during errors, which is useful in shell scripts that manipulate gated. The syslogd facility is used to log all the commands and error messages generated during gdc operation. gated uses these log messages to provide an audit trail of operations performed on the daemon.

For more information, type man 1M gated at the HP-UX prompt.

The gated Routing Table

Sending gated a SIGINT signal causes gated to write information about interface configurations, task information, and routing tables to the /var/tmp/gated_dump file.

The ripquery Tool

You can use the /usr/sbin/ripquery support tool to query gated for RIP routing information. ripquery sends two types of commands: a POLL command and a RIP request command. gated responds to a POLL command by listing all routes learned from RIP that are in its routing table. This does not include the interface routes for the local network or routes from other protocols that are announced through RIP. When gated receives a RIP request command on a interface, it announces routes via RIP on that interface. This includes routes from other protocols that are imported by gated on the node.

You can use ripquery with the -p option to query other non-gated RIP routers. With this option, ripquery initially sends POLL commands and then, if there is no response, sends RIP request commands. The default query (POLL commands) sent by ripquery may not be supported by all RIP routers. For more information, type man 1M ripquery at the HP-UX prompt.

The ospf_monitor Tool

You can use the /usr/sbin/ospf_monitor support tool to query OSPF routers for information on OSPF routing tables, interfaces, and neighbors, as well as data on AS external databases, link-state databases, error logs, and input/output statistics. Running the ospf_monitor command displays a prompt that allows you to enter interactive commands. For details on using this tool, type man 1M ospf_monitor at the HP-UX prompt.

Common Problems

This section discusses the common problems experienced during gated operation.

Problem 1: gated does not act as expected.

First, check the syslogd output for any syntax errors that may have been flagged.

To detect incorrect configuration commands, use gated tracing. Following are two sample configurations, along with the trace files generated by gated. The node used has three interfaces: lan0, lan1, and lan2. In the configuration files, lan0, lan1, and lan3 are specified. In the first configuration shown, the strictintfs option is specified for the interfaces to ensure that gated exits when the error is detected.

Interface Configuration With strictintfs Option Specified The following configuration references a non-existent interface. The options strictintfs line in the interfaces statement ensures that all the configured interfaces exit before gated starts.

```
traceoptions "tt" general;
interfaces {
   options strictintfs ;
   interface lan0 lan1 lan3 passive ;
} ;
rip yes ;
```

The following is the tracing output that is produced when gated is started with this configuration:

```
trace_on: Tracing to "/tt" started

Tracing flags enabled: general


parse: conf.tt:4 Interface not found at ’lan3’


parse_parse: 2 parse errors


Exit gated[15941] version @(#)Revision: 1.0 based on Cornell G
ateD R3_5Beta_3
```

Interface Configuration Without strictintfs Option Specified The following configuration references a non-existent interface, but does not include the strictintfs option:

```
traceoptions "tt" general;
interfaces {
   interface lan0 lan1 lan3 passive ;
} ;
rip yes ;
```

The following is the tracing output that is produced when gated is started with this configuration:

```
trace_on: Tracing to "/tt" started

Tracing flags enabled: general

inet_routerid_notify: Router ID: 15.13.119.134
```

You can also find the results of this same command in the gated_dump file. In the following segment of a gated_dump file, the interface is listed as passive in the interface policy statement toward the end of this example:

```
Interfaces:

lo0     Index 1         Change: <>      State: <Loopback>
        Refcount: 2     Up-down transitions: 0

        127.0.0.1
                Metric: 0       MTU: 4072
                Refcount: 4     Preference: 0   Down: 120
                Change: <>      State: <Up Loopback Multicast>
                Subnet Mask: 255.255.255.255
                proto:  RIP     State: <NoIn NoOut>

lan0    Index 2 Address 802.2 8:0:9:1b:da:1f    Change: <>
  State: <>
        Refcount: 2     Up-down transitions: 0

        15.13.119.134
                Metric: 0       MTU: 1436
                Refcount: 6     Preference: 0   Down: 120
                Change: <>      State: <Up Broadcast Multicast
 NoAge>
                Broadcast Address:   15.13.119.255
                Subnet Number: 15.13.112                Subnet
 Mask: 255.255.248
```

```
lan2    Index 3 Address 802.2 8:0:9:3d:2c:b1    Change: <>
  State: <>
        Refcount: 2     Up-down transitions: 0

        198.1.1.17
                Metric: 0       MTU: 1436
                Refcount: 4     Preference: 0   Down: 120
                Change: <>      State: <Up Broadcast Multicast
>
                Broadcast Address:   198.1.1.255
                Subnet Number: 198.1.1          Subnet Mask: 2
55.255.255

lan1    Index 4 Address 802.2 8:0:9:3d:3c:69    Change: <>
  State: <>
        Refcount: 2     Up-down transitions: 0

        198.2.1.40
                Metric: 0       MTU: 1436
                Refcount: 4     Preference: 0   Down: 120
                Change: <>      State: <Up Broadcast Multicast
 NoAge>
                Broadcast Address:   198.2.1.255
                Subnet Number: 198.2.1          Subnet Mask: 2
55.255.255

Interface policy:
         Interface lan0 lan1 lan3 passive
```

The state recorded in lan2 does not contain the NoAge flag, because the interface was not set to passive in the interface policy statement.

A common mistake is to expect gated to always send out RIP packets when you specify rip yes in a configuration file. gated will be an active RIP participant only if the host is a router (that is, when the host has more than one network interface).

Problem 2: gated deletes routes from the routing table.

gated maintains a complete routing table in user space, and synchronizes this table with the kernel routing table. When gated starts, it reads the entries in the kernel routing table. However, if gated does not get confirmation from its routing protocols (RIP, OSPF, and so on) about a route, it deletes the route from its tables and from the kernel routing table.

Normally, gated deletes the route configured in the /etc/rc.config.d/netconf file. To solve this problem, configure a static default route as described in the section “Installing Static Routes” on page 90.

Another common scenario occurs in networks where all the gateways do not implement the gated routing protocols. In this situation, routes that do not use gated gateways are not confirmed by gated, and gated deletes them unless a static statement is included in /etc/gated.conf:

```
static {
    13.0.0.0 mask 0xff000000 gateway 15.14.14.14 ;
};
```

The static entry in this example ensures that the local system includes a route to network 13.0.0.0 even though the gateway to that network (15.14.14.14) is not running any of the gated protocols.

You can include the restrict clauses in the export statements to restrict these extra routes from being advertised.

Problem 3: gated adds routes that appear to be incorrect.

If gated adds routes that appear to be incorrect, track the original source of the route by completing the following steps:

1. Start by looking at the routing table maintained by gated.
2. Send gated a SIGINT signal, and look at the information output in the /var/tmp/gated_dump file.
3. Look for the entry of the route in question. The entry identifies the protocol over which this route was heard and the first-hop router. The first-hop router is likely to be the immediate source of the information.

Perform one of the following actions:

- If the route was learned over RIP, use /usr/sbin/ripquery to query the first-hop router for the route. That router may claim to have heard the route from another router.
- If the first-hop router is another host running gated, have that host’s gated dump its routing table to find out where it learned about the route.

You may have to repeat this process several times to track down the original source of the route. If you expect the route to go through a different router, turn on `gated` tracing. The tracing tells you which routers are advertising this route and the values attached to those routes.

Problem 4: gated does not add routes that you think it must.

Tracking down this problem is similar to the previous problem (“Problem 3: gated adds routes that appear to be incorrect.” on page 107). You expect one or more routers to advertise the route. Turn on `gated` tracing to verify that `gated` is receiving packets of the type of routing protocol you expect. If these packets do not contain a route you expect to be there, trace packets on the router you expect to advertise the route.

Index

Index