3 GateD Basics

GateD is a dynamically loadable program that turns your OmniSwitch into a complete, integrated routing environment. GateD supports multiple routing protocols and a routing table database. GateD is a complete IP routing environment, providing route filtering, redistribution, aggregation, and routing policy enforcement.

GateD uses information updates from the various routing protocols to build its routing table database. Once constructed, it imports the data (subject to user configured filters) into a forwarding database, which is used to route data.

GateD allows users to implement policies. Policies are the filtering capabilities available for configuration by the user, and can control where routes are learned from and distributed to. Such policies include protocol, source and destination autonomous systems, source and destination interfaces, previous hop router, and specific destination.

This program is part of the Advanced Routing software package for switching products. It is a *modified* version of the popular and widely-used UNIX routing environment known as "GateD" (for Gate Daemon). GateD is developed, maintained, and licensed by Merit Network Inc.'s GateD Consortium.

Alcatel's support of GateD includes all of the program's core routing services and four of the routing protocols currently supported by GateD: RIP (Versions 1 and 2), OSPF, and BGP4.

GateD Requirements

The following switch features are required for GateD to operate:

- Release 4.1 or greater.
- 16 MB of memory (32 MB recommended).
- **gated.img**.
- **gated.conf** (This file is a text file created by the user with a text editor and then loaded onto the switch).

Protocols Supported

Routing protocols determine the best route to each destination, and they distribute routing information among the systems on a network. The Alcatel implementation of GateD supports the following protocols:

- **Routing Information Protocol (RIP)**. The Routing Information Protocol, Version 1 and Version 2, is the most commonly used interior routing protocol. RIP selects the route with the lowest metric as the best route. The metric is a hop count representing the number of networks through which data must pass in order to reach its destination. The longest path that RIP accepts is 15 hops. If the metric is greater than 15, a destination is considered unreachable and GateD discards the route. RIP assumes that the best route is the one that uses the fewest networks, i.e., the shortest path, not taking into account congestion or delay on route. See Chapter 6 for more specific information on RIP and RIP version 2.

- **Open Shortest Path First (OSPF)**. Open Shortest Path First is a link-state protocol. OSPF maintains a complete map of the routing topology and distributes the map to all switches participating in GateD routing. A designated router is specified to maintain and update the map. OSPF is better suited than RIP for complex networks with many routers. OSPF provides equal cost multipath routing. See Chapter 7 for more specific information on OSPF.

- **Border Gateway Protocol (BGP), version 4**. Border Gateway Protocol is replacing EGP as the exterior routing protocol of choice. BGP exchanges reachability information between autonomous systems, but provides more capabilities than EGP. BGP uses path attributes to provide more information about each route as an aid in selecting the best route. Path attributes may include, for example, administrative preferences based on political, organizational, or security (policy) considerations in the routing decision. BGP supports non-hierarchical topologies and can be used to implement a network structure of equivalent autonomous systems. See Chapter 8 for more specific information on BGP.

The Alcatel implementation of GateD does not support the following protocols at this time:

- IS-IS
- HELLO

Routing Functions

There are several core routing functions that are included in GateD along with the supported protocols. The core routing services include:

- **Route Preference**. Preference is a value GateD uses to order its choice of routes when multiple options are available. Preference can be set based on one network interface over another, one protocol over another, or one remote gateway over another. Each route has only one preference value associated with it, even though preference can be set at many places in the configuration file. The last or most specific preference value set for a route is the value used.

  See Chapter 4 for more specific information about Route Preference.

- **Static Route Configuration**. Static routes can be defined using Static Statements in the configuration file. See Chapter 9 for more specific information.

- **ICMP Router Discovery**. The Internet Control Message Protocol allows for Internet devices to transmit error or test messages, and allows for the implementation of Router Discovery. See Chapter 9 for more specific information.

- **Kernel Interface**. Kernel interface commands that are specified in the configuration file allow you to modify some of the settings of the switch operating system. See Chapter 9 for more specific information.

- **Routing Policy Enforcement (Importing, Exporting, and Filtering)**. The import statement controls which routes received from other systems are used by GateD, and the export statement controls which routes are advertised by GateD to other systems. Like the import statement, the syntax of the export statement varies slightly per protocol. Filtering allows you to exclude specific routes or groups of routes from receiving or sending information. Routes are filtered by specifying configuration language that will match a certain set of routes by destination, or by destination and mask. See Chapter 10 for more specific information.

- **Route Aggregation**. This is a process of combining multiple subnet addresses into a single address in order to reduce the amount of information exchanged between switches. See Chapter 11 for more specific information.

Configuration File

The configuration file for GateD is a text file that is loaded onto the switch along with the **gated.img** file. The file must be called **gated.conf**. This file is created by the user and controls most of the operations of GateD. There are specific commands that must be entered, in a specific format, in order for GateD to use the file. For more information on the GateD configuration file, see *The GateD Configuration File* on page 3-9.

UI commands

While most of GateD configuration is done using the **gated.conf** file, there are several UI commands available on the switch that allow you to view the configuration parameters and statistics. Most of these commands are detailed in the chapters specifically highlighting protocols (i.e., OSPF, BGP, and RIP). For information on the UI commands for these protocols, see Chapter 6 for RIP and RIP version 2, Chapter 7 for OSPF, and Chapter 8 for BGP.

For information on global UI commands, see Chapter 4.

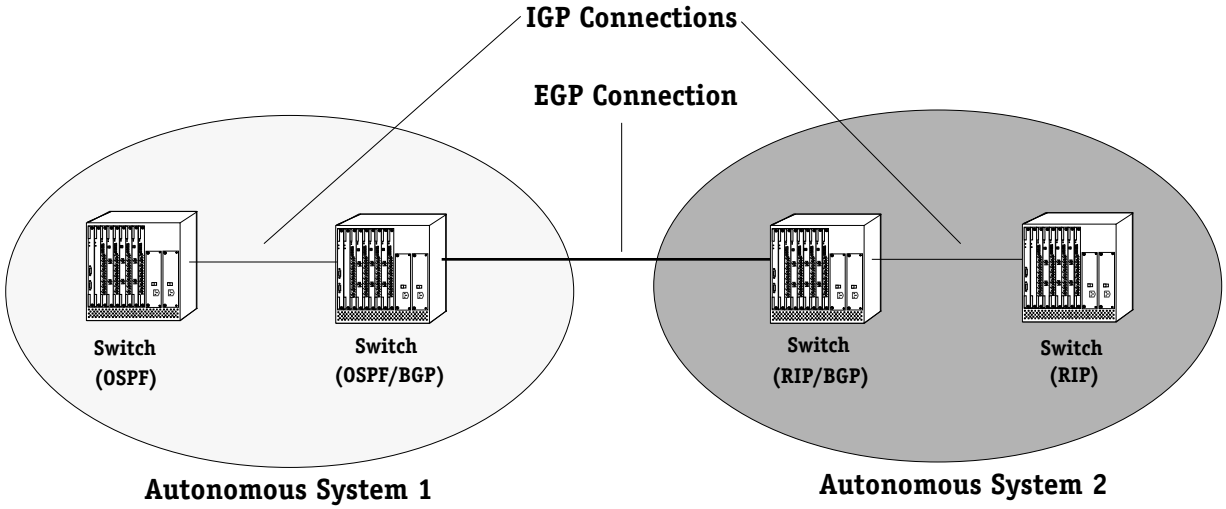
Routing Terminology

The use of the protocols in GateD assumes some general knowledge of routing. This section provides a summary of some terminology required to understand the GateD protocols and their implementation.

Most routing protocols rely on the concept of a routing domain, or autonomous system. An *autonomous system* (also referred to simply as AS) is a grouping of routers, workstations, and other network devices that are administrated by a single entity. Any router within an autonomous system can derive the path and determine the reachability of any workstation or other network device in the same autonomous system.

Some routing protocols specialize in routing data among routers and workstations *within* an autonomous system. These protocols are known as *Interior Gateway Protocols*, or IGPs. IGPs cannot route data outside the boundary of a routing domain, or autonomous system. These protocols, which include RIP and OSPF, can take data as far as the edge of an autonomous system. The Alcatel implementation of GateD supports RIP and OSPF.

Other routing protocols specialize in routing data *between* autonomous systems. These protocols are known as *Exterior Gateway Protocols*, or EGPs. One of the earliest implementations of an EGP was a protocol of the same name, the Exterior Gateway Protocol. Another example of an EGP is the Border Gateway Protocol (BGP). The Alcatel implementation of GateD supports BGP, version 4.

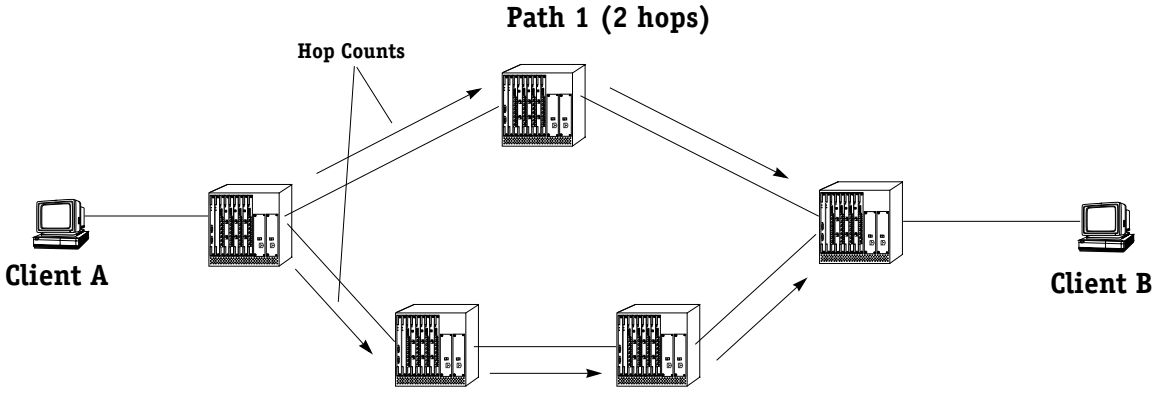


Simple GateD Routing Network

Routing is designed to get data from one place in a network to another. Networks today are complex systems with hundreds, even thousands of possible destinations. Routing protocols use methods that can find ways through this complex web of possible pathways. Besides IGPs and EGPs, routing protocols are categorized by the way in which they find paths through complex networks.

One method for deriving paths through a network is called the distance vector algorithm. Routers using a *distance vector protocol* use routing tables to determine the best route to every destination in a network.

The routing database entries for a distance vector protocol, such as RIP, identify the next router along a path to which data should be sent. Each link along the path is referred to as a hop and is assigned a cost. Cost is used to quantify the distance (time delay, hops, or other obstacles data encounters along a given link) between a sender and its destination. When the costs of each hop are summed, the value is referred to as a *metric*. Routers use metric information to decide the best path to a destination.



Distance Vector Protocol

In the above example, when Client A sends data to Client B, the router connected to Client A calculates the distance to Client B based on the number of router hops. This number, or metric, is used to determine which path is chosen to send data. In this case, Path 1 would be chosen as it shows the lowest aggregated metric to Client B.

In a sense, each router receives an update containing the information for a group of routers, and accepts this information as accurate.

♦ Note ♦

Sometimes a longer path is chosen instead of a path with less hops, if the shorter path has a hop that has been assigned an inflated hop count. This usually occurs if there is a specific area of the network with known delays. In this situation, a longer path (more hops) could be faster than a shorter path (less hops, but more delays).

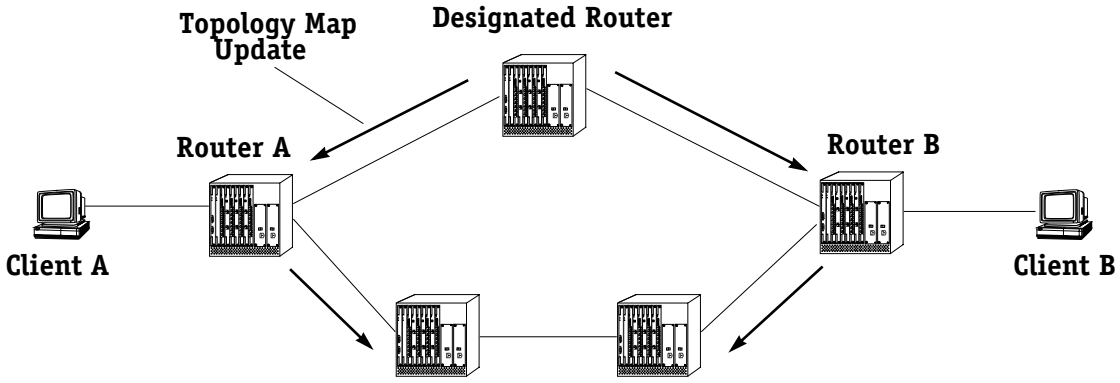
The table below shows the metrics and their usage for the RIP, OSPF, and BGP protocols.

Protocol	Metric Represents	Range	Unreachable
RIP	Router Hops	0-15	16
OSPF	Cost of Path	0 and up	Delete
BGP	Cost of Path	0-65534	65535

The range reflects the variable difference in the metric, with the lower numbers taking precedence over higher numbers. When a destination is unknown or restricted, the value in the **Unreachable** column is the metric value.

Other protocols, such as OSPF, are referred to as *link state* protocols. Each router in a system using a link state protocol has a database, or map, of the entire local subnet topology. A specific router, the *designated router*, is set up as the root, and maintains the "official" map of the network. All routers have exactly the same database, and packets are sent out periodically to update routers on topology changes and to make sure all databases are in synchronization. The status of each router and its interfaces are included in the global topology database.

From this topology database, routers use a shortest-path algorithm to determine the best router through a network.



Link State Protocol

In the diagram above, the designated router maintains the topology map of the network, and periodically floods it to all switches. When data is sent from Client A to Client B, Router A constructs a forwarding table based on the most recent topology map (sent and maintained by the designated router) and sends the data along the shortest network path.

Configuration Overview for GateD

The following section provides a general overview of the steps required to get GateD up and running on your switch.

Step 1. Create the Configuration File

This step will require more time and thought than all the other steps combined. GateD requires a configuration file to be installed on the switch. This configuration file (**gated.conf**) contains all the parameters for your GateD routing environment. These configuration parameters are described in the chapters that follow.

This chapter contains some very basic configuration file examples. These simple examples can be found in *GateD Quick Start* on page 3-15. In addition, there are several rules and conventions to follow when setting up the **gated.conf** file. These rules are described in *The GateD Configuration File* on page 3-9.

More explicit information on configuring protocol options can be found in Chapter 6 for RIP and RIP version 2, Chapter 7 for OSPF, and Chapter 8 for BGP.

Step 2. Load the Configuration File on the Switch

GateD will not operate if the configuration file, **gated.conf**, is not loaded on the switch. Load the **gated .conf** file into the flash memory on the switch via FTP, ZMODEM, or the Ethernet management port. Refer to the chapter titled "Installing Switch Software" in your switch manual for instruction on loading image files on the switch.

Step 3. Install GateD on the Switch

The main GateD image file is called **gated.img**. It must be installed in the flash memory on your switch before GateD can be run. In addition, **gated.img** will dynamically load *once it sees a gated.conf file on the switch*. You must be sure to have the configuration file, **gated.conf**, on your switch or **gated.img** will not operate. Refer to chapter titled "Installing Switch Software" in your switch manual for instruction on loading image files on the switch.

Step 4. Modify the mpm.cmd File (optional)

GateD functions without any modifications to the **mpm.cmd** file. However, if you want to enable trace options, you will need to add a line to this file. There are three possible options for the added command:

use_gated=0	This command tells the switch not to run GateD even if the **gated.img** and **gated.conf** files are detected and valid.
use_gated=1	This command tells the switch to run GateD without enabling tracing options (it is equivalent to having none of these commands added to the **mpm.cmd** file).
use_gated=2	This command tells the switch to run GateD and enable trace options.

For information on how to add lines to the **mpm.cmd** file, see the chapter titled "Managing Files" of your switch manual.

Step 5. Reboot the Switch

Once you have completed the steps above, you need to reboot the switch. GateD will be activated when the switch resumes operation. For information on rebooting a switch, see the chapter titled “Switch Security” in your switch manual.

The GateD Configuration File

The configuration file is the most crucial component of GateD. It allows you to configure the operational parameters of GateD, and to tailor its performance to suit your network environment.

Without a valid configuration file, GateD will not run. GateD looks for the configuration file during startup; if no configuration file is detected, GateD is not initialized. The configuration file must be titled **gated.conf** (case sensitive). The configuration file can be created using any text editor, such as Notepad (Windows) or VI (UNIX).

♦ Important Note ♦

If the configuration file is created in a DOS or Windows environment, there may be problems with the GateD parser when it reads the file. Since GateD is UNIX based software, it may not recognized DOS and Windows end-of-line characters, causing the configuration file to be rejected as invalid. This will shut down GateD operations. If possible, the configuration file should be created in a UNIX platform. There are several programs that will translate text files from DOS to UNIX if this is not possible.

It is recommended that configuration files be stored on a disk or hard drive as well as on the switch. This way, when configuration changes are needed, you can simply edit the stored file, upload the new version of the file to your switch, and restart GateD to implement the changes.

GateD reads its configuration file (**gated.conf**) to learn which type of routing it is to perform. The configuration file is made up of *configuration statements*, which are groups of related commands. There are eight categories of configuration statements. Configuration statements must be entered into the configuration file in a specific order for GateD to understand the file and implement its contents. The eight statement categories are shown in the correct order, with brief descriptions, in the following table:

Statement Order for gate.conf File

Configuration Statement	Description
Options	Sets global parameters for GateD.
Interface	Sets parameters for a router interface, which is used to link a router to the GateD network.
Definition	Sets general configuration options used in more than one protocol.
Protocol	Sets the parameters for configuration protocols (RIP, OSPF, BGP).
Static	Sets static routes.
Control	Sets parameters for importing and exporting routes.
Aggregate	Sets parameters for creating a more general route from several specific routes.
Trace	Sets parameters for trace options. This is the only configuration statement that can be placed anywhere in the configuration file.

A more detailed version of the statements and their order is described in *Statement Order* on page 3-13.

Statement Syntax

In addition to the order of the configuration statements, the commands in each statement group must follow a specific syntax, which is based loosely on standard UNIX conventions.

The GateD configuration file (**gated.conf**) consists of a sequence of statements, each terminated by a semicolon (**;**). These statements are composed of *tokens* that are separated by white space, which can be any combination of blank spaces, tabs, or blank lines. Often tokens are accompanied by some kind of value that qualifies the token.

The following line is an example of a *statement*:

```
rip on;
```

The token in this case is **rip**, with **on** being a parameter value that activates RIP functions.

Many statements have further suboptions that are noted by placing them in braces (**{}**). Braces must always be ordered in pairs. For example:

```
rip on {
    interface 1.1.1.1;
};
```

Often there is more than one suboption to configure. Multiple suboptions are ordered sequentially. For example:

```
routerid 1.1.1.1
ospf on {
        backbone {
                interface 1.1.1.1 {
                                priority 1;
                };
        };
};
```

Note the semicolon is placed after each statement, whether in braces or not.

This structure makes it easier to identify the various parts of the configuration file, as well as their associations with each other and with the specific protocols.

♦ Note ♦

It is not required that statements be on separate lines, or that braces be split between lines. This structure is a convention that is easily recognized by most users. A configuration file can be set up as a continuous line with no breaks, assuming that the configuration statement order and the correct use of braces and semicolons is followed.

Comments can also be added to the configuration file. Comments may be specified in one of two forms. The first form begins with a pound sign (**#**) and runs to the end of the line. The other form, the C-programming style, starts with a slash-asterisk (*/**) and continues until it reaches an asterisk-slash (***/*). Comments are not viewed as statements by GateD and are solely used for information. For example:

```
rip on {
    interface 1.1.1.1;
};

#This router is in building A
```

The router location comment after the pound sign is ignored by GateD.

Syntax Description Conventions

For the rest of this chapter and the subsequent chapters relating to GateD, generic examples of configuration files are shown, listing all tokens and suboptions along with any value required to modify the token or suboption. Many times, using one token precludes using another, or certain parameters are optional. A syntax convention for displaying generic examples has been established and is detailed below.

- Tokens and special characters that GateD expects to be *entered exactly* are shown using **bold type**.
- Required parameters that *must* accompany the token are shown using *italics type*.
- Optional keywords, and their associated parameters, are shown inside square brackets (**[** and **]**). The brackets are for clarity only; they are not added to the configuration file.
- The vertical bar character (**|**) separates the individual entries in a series of optional parameters and indicates that only one of these optional parameters may be used.
- The parentheses characters (**(** and **)**) are used to group keywords and parameters for added clarity. The parentheses are for clarity only; they are not added to the configuration file.

For example, consider the following syntax description:

[backbone | (area *area* **)]**

The two tokens are **backbone** and **area**. The square brackets indicate that both of these tokens are optional. The vertical bar between the two tokens indicates that either one of them may be specified, but not both of them. The second instance of the word “area” is shown in italics (*area*), as it is a required, but variable, parameter that must be provided when the token **area** is specified. The statement in the file could look like the following:

```
area 1
```

Sometimes required parameters (which are denoted by the *italics* type) will refer to a group of choices. Such parameters are shown as compound words connected by an underscore character. Examples of such parameter groups are: *interface_list* and *trace_options*.

Statement Order

The following sections describe the configuration statements and list the major suboptions for each group. The list below shows the required ordering of GateD's configuration statements. If you enter any statement out of order an error will occur when GateD reads its configuration file. The one exception to this rule are trace statements, which can appear anywhere in the file.

1. **Options statement**. The options group allows you to set global GateD parameters. A detailed explanation of the options commands is provided in Chapter 4 of this manual. None of the options statements are required.

2. **Interfaces statement**. The interfaces group is used to control the way GateD views and handles interfaces. An interface is the connection between a router and one of its attached networks. A physical interface is specified by an an IP address. Interface options are described in detail in Chapter 5 of this manual. The following are suboptions to this statement:

   - **options**. Allows you to set specific global options for an interface.
   - **interface** *interface_list*. Allows you to set specific options for a listed interface.
   - **define** *address*. Allows you to specify interfaces that might not be present when GateD is started so that they can be referenced in the configuration file.

3. **Definition statement**. These options allow you to configure general configuration statements that refer to at least two protocols in GateD. Definition options are described in detail in Chapter 4 of this manual. The following are suboptions to this statement:

   - **autonomoussystem** *autonomous system*. Defines the Autonomous System (AS) number. The AS number is used in OSPF and BGP for specifying an area.
   - **routerid** *host*. Defines the originating router identification, which is used in BGP and OSPF. The default address is the first address encountered by the GateD protocol.
   - **martians**. Defines a list of destination addresses for which all routing information is ignored.

4. **Protocol statement**. The protocols group allows you to set configuration options for the protocols supported by Alcatel's version of GateD, as well as the router discovery and kernel interface commands. Protocol options are described in detail in Chapters 6, 7, and 8 of this manual. Router discovery and kernel interface options are described in Chapter 9. The following are suboptions to this statement:

   - **rip**. Enables the RIP1 and/or RIP2 protocols.
   - **ospf**. Enables the OSPF protocol.
   - **bgp**. Enables the BGP protocol.
   - **router discovery**. Enables the Router Discovery protocol.
   - **kernel**. Configures the options for the kernel interface.

5. **Static statement**. The static group allows you to set configuration options for static routes. Static statements are defined in detail in Chapter 9 of this manual.

6. **Control statement**. The control group allows you to configure options for importing and exporting routes, and for route filtering. Import statements define the way GateD protocols handle routing updates it receives, export statements define the way GateD protocols handle routing updates it sends, and route filtering statements define which interfaces are restricted or accepted for learning routing information. Control options are described in detail in Chapter 10 of this manual. The following are suboptions to this statement:
   - **route importation**. Allows control over what routes are added to the routing table.
   - **route exportation**. Allows control over what routes are advertised to other systems.
7. **Aggregate statement**. The aggregate group allows you to control the advertising of a single or several specific routes as a generic route. Aggregate options are described in detail in Chapter 11 of this manual. The following are suboptions to this statement:
   - **aggregate**. Creates a general route for a single or several specific routes.
   - **generate**. Creates a route based on specified conditions. Often referred to as the "route of last resort."
8. **Trace statement**. Trace statements control tracing options. GateD's tracing options may be configured at many levels. Tracing options include the control options and global and protocol-specific tracing options. Unless overridden, tracing options from the next higher level are inherited by lower levels. Tracing options are described in detail in Chapter 12 of this manual.

GateD Quick Start

This section provides some very simple examples of GateD use. They are intended to get you up and running with one of the protocols supported by Alcatel's version of GateD. Be aware that these files are intended only to activate GateD and enable the routing protocols. Additional configuration will probably be required on live networks.

For more details on configuration file information, see *The GateD Configuration File* on page 3-9. For more information on specific protocols, see Chapter 6 for RIP and RIP version 2, Chapter 7 for OSPF, and Chapter 8 for BGP.

Enable RIP, Version 1

To enable the Routing Information Protocol (RIP), version 1 for a specific interface, you need to configure a **gated.conf** file with the following 2 lines:

```
rip yes {
   interface 1.1.1.1;
 };
```

For this example, an IP address of 1.1.1.1 is used to specify a specific interface that RIP operates on. The value of this line should be an IP address. You can specify more than one value by adding more interface lines, as shown:

```
rip yes {
   interface 1.1.1.1 ;
   interface 1.1.1.2 ;
   interface 1.1.1.3 ;
 };
```

If your network includes multiple switches, and you want all interfaces on those switches to support RIP, you may want to change the interface line of RIP to **all**, rather than a specific IP address. This will, however, increase the memory usage for the routing table as RIP will keep track of all ports that have been configured to use RIP.

Using **all** rather than specifying an interface has a second consequence. If an interface is configured to run a routing protocol and GateD does not see routing information on that interface, then the routing for that interface will be disabled. When this occurs, the preference for the interface changes to 120 (indicating a low priority and used for all interfaces that are down). The switch will "poison" routes to the network attached to the interface (RIP updates with a metric of 16), and GateD will ultimately stop advertising to the network completely.

For a full description of RIP and the configuration options available for use in the configuration file, see Chapter 6 of this manual.

Enable RIP, Version 2

To enable the Routing Information Protocol (RIP) version 2, you need to configure a **gated.conf** file with the following lines:

```
rip yes {
   interface 1.1.1.1 version 2;
   };
```

For this example, an IP address of 1.1.1.1 is used to specify a specific interface that RIP version 2 operates on. The value of this line should be an IP address. You can specify more than one value by adding more interface lines, as shown:

```
rip yes {
   interface 1.1.1.1 version 2;
   interface 1.1.1.2 version 2;
   interface 1.1.1.3 version 2;
 };
```

If your network includes multiple switches and you want all ports on those switches to support RIP version 2, you may want to change the interface line of RIP version 2 to **all**, rather than a specific IP address. See the section above on RIP for more information about the consequences of enabling all interfaces.

♦ Important Note ♦

RIP version 2 can communicate with RIP version 1, but RIP version 1 cannot communicate with RIP version 2.

For a full description of RIP version 2 and the configuration options available for use in the configuration file, see Chapter 6 of this manual.

Enable OSPF

To enable the Open Shortest Path First (OSPF) protocol, you need to configure a **gated.conf** file with the following lines:

```
routerid 1.1.1.1
ospf yes {
   area 1 {
      interface 1.1.1.1 {
                        priority 1;
      };
   };
};
```

In OSPF, the **routerid**, an area, a priority, and an interface must be specified. The **routerid** identifies the router for the rest of the OSPF network. The priority specifies the eligibility of this router to become either a designated router or a backup designated router. The area number designates which area the machine belongs to.

It is also possible to specify **backbone** instead of an area, if this router is part of the OSPF backbone. (All areas in an OSPF network must be connected to a backbone. See Chapter 7 of this manual for more information on OSPF, areas, and backbones.)

For this example, an IP address of 1.1.1.1 is used to identify a specific interface that OSPF operates on, and the area for this interface is 1. The value of this line should be an IP address. You can specify more than one value by adding more interface lines, as shown:

```
routerid 1.1.1.1
ospf yes {
  area 1 {
     interface 1.1.1.1 {
                        priority 1;
     };
     interface 1.1.1.2 {
                        priority 20;
     };
     interface 1.1.1.3 {
                        priority 20;
     };
  };
};
```

Each OSPF router must be configured into at least one OSPF area. If more than one area is configured, then a backbone must be specified. This is done using the **backbone** keyword in place of the **area** keyword, as shown:

```
routerid 1.1.1.1
ospf yes {
  backbone {
     interface 1.1.1.1 {
                       priority 1;
     };
  };
};
```

If your network includes multiple switches and you want all ports on those switches to support OSPF, you may want to change the interface line of OSPF to **all**, rather than a specific IP address. This will, however, increase the memory usage for the routing table as OSPF will keep track of all ports that have been configure to use OSPF.

For a full description of OSPF and the configuration options available for use in the configuration file, see Chapter 7 of this manual.

Enable BGP

To enable the Border Gateway Protocol (BGP), you need to configure a **gated.conf** file with the following lines:

```
autonomoussystem 30 ;
bgp yes {
   group type external peeras 2 {
      peer 1.1.1.1;
      };
};
```

In BGP, an autonomous system, a group type, and a peer must be specified. The autonomous system (AS) specifies the AS that the machine is a member of, the group type designates the remote Autonomous System (AS) BGP communicates with, and the peer designates the specific machine in the AS.

For this example, an IP address of 1.1.1.1 is used to specify a specific peer that BGP communicates with. The value of this line can be an IP address, a domain name, or an interface ID.

For a full description of BGP and the configuration options available for use in the configuration file, see Chapter 8 of this manual.