

9 Other GateD Protocols

Along with the protocols discussed earlier in this manual (RIP, OSPF, and BGP), there are several statements that affect the operation of GateD as if they were protocols. This chapter covers four types of statements that can be added to the **gated.conf** file that affect the performance of GateD when viewing or adding routes to its routing table. The statements are:

Kernel Statements. These commands allow you to modify the performance of the GateD kernel (operating system). For information on the kernel statement, see *The Kernel Interface* on page 9-2.

Static Statements. These commands allow you to create and configure static routes. For information on the static statement, see *Static Routes* on page 9-6.

Router Discovery Statements. These commands allow you to employ the server and client functions of router discovery. For information on the router discovery statement, see *The Router Discovery Protocol* on page 9-9.

ICMP Statements. These commands allow you to enable ICMP tracing options. For information on the ICMP statement, see *The Internet Control Message Protocol (ICMP)* on page 9-14.

The Kernel Interface

While the kernel interface is not technically a routing protocol, it has many characteristics of one. The routes GateD chooses to install in the kernel forwarding table are those that will actually be used by the kernel to forward packets.

A kernel is the central module of an operating system. It is the part of the operating system that loads first and remains in the main memory. Typically, the kernel is responsible for memory management, process and task management, and disk management.

The operations GateD uses to update the typical kernel forwarding table take a measurable amount of time. This is not usually a problem for older routing protocols, such as RIP, that are not time critical. Newer routing protocols, such as OSPF and BGP, have stricter timing requirements and are often used to process large numbers of routes. The speed of the kernel interface becomes critical when these newer protocols are used.

To prevent GateD from locking up for significant periods of time, routing is done in batches. The size of these batches can be controlled by the using the parameters described in the kernel statement.

◆ Note ◆

The kernel parameters are used to fine tune the kernel performance. Under normal circumstances, the default kernel settings should handle most network traffic.

During normal shutdown processing, GateD deletes all the routes it has installed in the kernel forwarding table. In larger networks, it may be advantageous to instruct GateD to retain its routes. This can greatly reduce the time it takes to recover from a GateD restart.

The Kernel Statement

The following shows the full kernel command statement. These statements are entered in the **gated.conf** file. Each part of the statement is described in the sections following this full statement.

```
kernel {  
    options  
        [ nochange ]  
        [ noflushatexit ]  
    ;  
    routes number;  
    flash  
        [ limit number ]  
        [ type interface | interior | all ]  
    ;  
    background  
        [ limit number ]  
        [ priority flash | higher | lower ]  
    ;  
    traceoptions trace_options;  
};
```

The syntax conventions used for showing the kernel commands are the same as those described in Chapter 3 of this manual.

options

This set of commands allows you to configure global kernel options. Kernel options affect how GateD structures the kernel routing table. There are four operations that can be performed on routes in the kernel routing table: changes, deletes, adds, and retains. These operations are affected by the options listed:

nochange	This command prevents change operations from being performed. Only deletes and adds are accepted as valid updates.
noflushatexit	During normal shutdown processing, GateD deletes all routes from the kernel forwarding table that do not have a retain indication. This command prevents route deletions at shutdown. Instead, routes are changed and added to make sure that all the routes marked with retain are installed.

routes *number*

On some systems, kernel memory is at a premium. With this parameter, a limit can be placed on the maximum number of routes GateD will install in the kernel. Normally GateD adds, changes, and deletes routes in a specific order: interface, then internal, then external. (For a detailed description of these types, see Chapter 3 of this manual.)

If this parameter is specified and the route limit is reached, GateD does two scans of the router list instead. On the first scan it deletes routes, deletes all changed routes, and turns the queued changes into adds. It then rescans the list adding routes until it hits the limit again. This process favors internal routes over external routes.

flash

When routes change, the process of notifying the protocols is called a flash update. The kernel forwarding table interface is the first to be notified. Normally a maximum of 20 interface routes may be processed during one flash update. This command allows tuning of these parameters. The following are valid parameters of the **flash** command:

limit <i>number</i>	Specifies the maximum number of routes that may be processed during one flash update. The default is 20 . A value of -1 will cause all pending route changes of the specified type to be processed during the flash update.
type interface interior all	Specifies the type of routes that will be processed during a flash update. Interface specifies that only interface routes will be installed during a flash update. Interior specifies that interface and interior routes are installed. All specifies the inclusion of interface, interior, and exterior routes. The default is interface .

background

Since only interface routes are normally installed during a flash update, the remaining routes are processed in batches in the background (when no routing protocol traffic is being received). Normally, 120 routes are installed at a time, and background processing (or kernel updates) is done at a lower priority than flash updates. The following parameters are valid modifiers of the background command:

limit <i>number</i>	Specifies the number of routes that are processed during one batch. The default is 120.
priority flash higher lower	Specifies the priority for processing batches of kernel updates in relationship to the flash update processing. The default is lower , which means that flash updates are processed first. To process kernel updates at the same priority as flash updates, specify flash ; to process them at a higher priority, use higher .

traceoptions

The following tracing options can be specified for kernel routes in the configuration file.

remnants	Routes read from the kernel when GateD starts.
request	Requests by GateD to add, delete, or change routes in the kernel forwarding table.

The following packet tracing options can also be specified (these options may be modified with **detail**, **send** and **recv**; for details on trace option modifiers, see Chapter 12 of this manual):

remnants	Routes read from the kernel when GateD starts.
request	Requests by GateD to Add/Delete/Change routes in the kernel forwarding table.
routes	Routes exchanged with the kernel, including Add/Delete/Change messages, and Add/Delete/Change messages received from other processes.

Kernel Statement Example

The following is an example of a kernel statement:

```
kernel {  
    options noflushatexit ;  
    routes 30 ;  
    flash limit 20 ;  
    background priority higher ;  
};
```

In the above example:

- The **noflushatexit** modifier specifies that GateD routes are not removed from the routing table when GateD is restarted.
- The **routes** statement specifies that only 30 routes are installed in the kernel memory (to conserve memory space).
- The **flash** statement with the **limit** modifier specifies that only 20 routes are processed during a flash update. The flash update modifies the routing list as required.
- The **background** statement with the **priority higher** modifier specifies that background (kernel) updates take priority over flash updates.

Static Routes

Static statements define the static routes used by GateD. A static route is a permanent route with specifically fixed endpoints. A single static statement can specify any number of routes. The static statement occurs after protocol statements and before control statements in the **gated.conf** file. Any number of static statements may be specified, each containing any number of static route definitions. These routes can be overridden by routes with better preference values.

Static Routes Creation (GateD vs. **aisr** command)

You can add static routes to a switch configuration through the GateD **static** statement or through the switch's **aisr** command. Both types of routes will be seen and advertised by GateD. However, routes added through the **aisr** command will appear in GateD as *kernel* routes rather than as static routes. This occurrence may have an affect on the use of GateD policies. For this reason, it is recommended that you use the GateD **static** statement to create static routes that will be used on a permanent basis. Use the **aisr** command to create static routes for short-term needs, such as troubleshooting.

The Static Statement

The following pages show the full static route command statement. These statements are entered in the **gated.conf** file. Each part of the statement is described in sections following this full statement.

```
static {
  ( host host ) | default | ( network [ ( mask mask ) | ( masklen number ) ] ) gateway
  gateway_list
  [ interface interface_list ]
  [ preference preference ]
  [ retain ]
  [ reject ]
  [ blackhole ]
  [ noinstall ] ;
  ( network [ ( mask mask ) | ( masklen number ) ] ) interface interface
  [ preference preference ]
  [ retain ]
  [ reject ]
  [ blackhole ]
  [ noinstall ] ;
};
```

The syntax conventions used for showing the static route commands are the same as those described in Chapter 3 of this manual.

```
( host host ) | default | ( network [ ( mask mask ) | ( masklen number ) ] ) gateway
gateway_list
```

This is the most general form of the static statement. It defines a static route through one or more gateways. The syntax of this statement always contains a starting point (expressed above as **host** *host* | **default** | *network* [**mask** *mask* | **masklen** *number*]) and a gateway (expressed above as **gateway** *gateway_list*).

For example, to create a static route for host 1.1.1.1 through gateway 2.2.2.2, you would enter the following:

```
host 1.1.1.1 gateway 2.2.2.2
```

Static routes are installed when one or more of the gateways listed are available on directly attached interfaces.

Parameters for static routes are:

interface <i>interface_list</i>	When this parameter is specified, gateways are only considered valid when they are on one of the listed interfaces.
preference <i>preference</i>	This option selects the preference of this static route. The preference controls how this route competes with routes from other protocols. The default preference is 60 . For more information on preference, see Chapter 4 of this manual.
retain	Normally GateD removes all routes except interface routes from the kernel forwarding table during a graceful shutdown. The retain option may be used to prevent specific static routes from being removed. This is useful to insure that some routing is available when GateD is not running.
reject	Instead of forwarding a packet like a normal route, reject routes cause packets to be dropped and unreachable messages to be sent to the packet originators. Specifying this option causes this route to be installed as a reject route.
blackhole	A blackhole route is the same as a reject route except that unreachable messages are not sent.
noinstall	Normally the route with the lowest preference is installed in the kernel forwarding table and is the route exported to other protocols. When noinstall is specified on a route, the route will not be installed in the kernel forwarding table when it is active, but it will still be eligible to be exported to other protocols.

```
( network [ ( mask mask ) | ( masklen number ) ] ) interface interface
```

This form defines a static interface route that is used for primitive support of multiple network addresses on one interface. The **preference**, **retain**, **reject**, **blackhole**, and **noinstall** options are the only options for this type of static route. Descriptions for these options are given above.

Static Statement Example

The following is an example of a static statement:

```
static {  
    host 1.1.1.1 gateway 198.5.3.0 preference 50 retain;  
};
```

In the example above, a static route to host 1.1.1.1 is created that uses gateway 198.5.3.0. This route has a preference of 50, and is retained when GateD is restarted rather than being flushed with all other GateD routes.

The Router Discovery Protocol

The Router Discovery Protocol is an IETF standard protocol (RFC 1256) used to inform hosts of the existence of routers. It is intended to be used instead of having hosts wiretap routing protocols such as RIP. It is used in place of, or in addition to, statically configured default routes in hosts.

The protocol is split into two portions, the server portion which runs on routers, and the client portion which runs on hosts. GateD treats these two portions much like two separate protocols, only one of which may be enabled at a time.

The Router Discovery Server

The Router Discovery Server runs on routers and announces their existence to hosts. It does this by periodically multicasting or broadcasting a router advertisement to each interface on which the server is enabled. These router advertisements contain a list of all the router's addresses on a given interface and the preference of each address for use as the default router on that interface.

Initially these router advertisements occur every few seconds, then decrease in frequency to every few minutes. In addition, a host may send a router solicitation to which the router will respond with a unicast router advertisement (unless a multicast or broadcast advertisement is due momentarily).

Each router advertisement contains an advertisement lifetime field indicating how long the advertised addresses are valid. A lifetime of zero indicates that one or more addresses are no longer valid.

On systems supporting IP multicasting, the router advertisements are sent by default to the all-hosts multicast address 224.0.0.1. When router advertisements are sent to the all-hosts multicast address, or an interface is configured for the limited-broadcast address 255.255.255.255, all IP addresses configured on the physical interface are included in the router advertisement. When the router advertisements are sent to a network or subnet broadcast, only the address associated with that network or subnet is included.

The Router Discovery Server Statement

The following is the full router discovery server command statement. These statements are entered in the **gated.conf** file. Each part of the statement is described in the sections following this full statement.

```
routerdiscovery server yes | no | on | off [ {  
    traceoptions trace_option;  
    interface interface_list  
        [ minadvinterval time ] |  
        [ maxadvinterval time ] |  
        [ lifetime time ]  
    ;  
    address interface_list  
        [ advertise ] | [ ignore ] |  
        [ broadcast ] | [ multicast ] |  
        [ ineligible ] | [ preference preference ]  
    ;  
} ] ;
```

The syntax conventions used for showing the router discovery commands are the same as those described in Chapter 3 of this manual.

traceoptions *trace_option*

Specifies the tracing options for Router Discovery. The router discovery client and server support the **state** trace flag, which traces various protocol occurrences.

The router discovery client and server do not directly support any packet tracing options. Tracing of router discovery packets is enabled using the ICMP Statement. See *The Internet Control Message Protocol (ICMP)* on page 9-14.

interface *interface_list*

Specifies the parameters that apply to physical interfaces. Note a slight difference in convention from the rest of GateD; interface specifies a list of physical interfaces (such as le0, ef0 and en1), while address specifies a list of IP addresses.

One or more of the following parameters must be provided for the interface:

maxadvinterval *time*

The maximum time allowed between sending broadcast or multicast Router Advertisements from the interface. Must be no less than 4 and no more than 30:00 (30 minutes or 1800 seconds). The default is 10:00 (10 minutes or 600 seconds).

minadvinterval *time*

The minimum time allowed between sending unsolicited broadcast or multicast Router Advertisements from the interface. Must be no less than 3 seconds and no greater than maxadvinterval. The default is 0.75 * maxadvinterval.

lifetime *time*

The lifetime of addresses in a Router Advertisement. Must be no less than maxadvinterval and no greater than 2:30:00 (two hours, thirty minutes or 9000 seconds). The default is 3 * maxadvinterval.

address *interface_list*

Specifies the parameters that apply to the specified set of addresses on this physical interface. Note a slight difference in convention from the rest of GateD; interface specifies a list of physical interfaces (such as le0, ef0 and en1), while address specifies a list of IP addresses.

One or more of the following parameters must be provided for the address:

advertise	Specifies that the listed address(es) should be included in router advertisements. This is the default.
ignore	Specifies that the listed address(es) should not be included in router advertisements.
broadcast	Specifies that the given address(es) should be included in a broadcast router advertisement because this system does not support IP multicasting, or some hosts on attached network do not support IP multicasting. It is possible to mix addresses on a physical interface such that some are included in a broadcast router advertisement and some are included in a multicast router advertisement. Mixing routes is the default if the router does not support IP multicasting.
multicast	Specifies that the given address(es) should only be included in a multicast router advertisement. If the system does not support IP multicasting the address(es) will not be included. If the system supports IP multicasting, the default is to include the address(es) in a multicast router advertisement. If the given interface does not support IP multicasting the address(es) will be included in a broadcast router advertisement.
ineligible	Specifies that the given address(es) will be assigned a preference of hex 80000000; this means that the address is not eligible to be the default route for any hosts. This is useful when the address(es) should not be used as a default route, but are given as the next hop in an ICMP redirect. This allows the hosts to verify that the given addresses are up and available.
preference <i>preference</i>	The degree of preference of the address(es) as a default router address, relative to other router addresses on the same subnet. This is a 32-bit, signed, two's-complement integer, with higher values meaning more preferable. Note that hex 80000000 may only be specified as ineligible. The default is 0.

Router Discovery Server Statement Example

The following is an example of a Router Discovery Server Statement:

```
routerdiscovery server yes {
    address 1.1.1.1 ignore;
};
```

In the above example:

- The **routerdiscovery** statement activates the router discovery protocol. Note that the token **server** is required to distinguish it from the client portion of the software.
- Per the **ignore** option, address 1.1.1.1 should not be sent out with router advertisements.

The Router Discovery Client

A host listens for router advertisements via the all-hosts multicast address (224.0.0.2) if IP multicasting is available and enabled, or on the interface's broadcast address if multicasting is not available or disabled. When starting up, or when reconfiguring, a host may send a few router solicitations to the all-routers multicast address or the interface's broadcast address.

When a router advertisement with non-zero lifetime is received, the host installs a default route to each of the advertised addresses. If the preference is ineligible, or the address is not on an attached interface, the route is marked unusable but retained. If the preference is usable, the metric is set as a function of the preference such that the route with the best preference is used. If more than one address with the same preference is received, the one with the lowest IP address will be used. These default routes are not exportable to other protocols.

When a router advertisement with a zero lifetime is received, the host deletes all routes with next-hop addresses learned from that router. In addition, any routes learned from ICMP redirects pointing to these addresses will be deleted. The same will happen when a router advertisement is not received to refresh these routes before the lifetime expires.

The Router Discovery Client Statement

The following shows the full router discovery client command statement. These statements are entered in the **gated.conf** file. Each part of the statement is described in the sections following this full statement.

```
routerdiscovery client yes | no | on | off [ {  
    traceoptions trace_options;  
    preference preference;  
    interface interface_list  
        [ enable ] | [ disable ]  
        [ broadcast ] | [ multicast ]  
        [ quiet ] | [ solicit ]  
    }  
};
```

The syntax conventions used for showing the router discovery commands are the same as those described in Chapter 3 of this manual.

traceoptions *trace_options*

Specifies the tracing options for router discovery. The router discovery client and server support the **state** trace flag which traces various protocol occurrences.

The router discovery client and server do not directly support any packet tracing options. Tracing of router discovery packets is enabled using the ICMP statement. See *The Internet Control Message Protocol (ICMP)* on page 9-14.

preference *preference*

Specifies the preference of all router discovery default routes. The default is 55.

interface *interface_list*

Specifies the parameters that apply to physical interfaces. Note a slight difference in convention from the rest of GateD; interface specifies just physical interfaces (such as le0, ef0 and en1). The router discovery client has no parameters that apply only to interface addresses.

enable	Specifies that router discovery should be performed on the specified interface(s). This is the default.
disable	Specifies that router discovery should not be performed on the specified interface(s).
multicast	Specifies that router solicitations should be multicast on the specified interface(s). If IP multicast is not available on this host and interface, no solicitation will be performed. The default is to multicast router solicitations if the host and interface support it, otherwise router solicitations are broadcast.
quiet	Specifies that no router solicitations will be sent on this interface, even though router discovery will be performed.
solicit	Specifies that initial router solicitations will be sent on this interface. This is the default.

Router Discovery Client Statement Example

The following is an example of a Router Discovery Client Statement:

```
routerdiscovery client yes {
    preference 50 ;
    interface 1.1.1.1 quiet;
};
```

In the above example:

- The **routerdiscovery** statement activates the router discovery protocol. Note that the token **client** is required to distinguish it from the server portion of the software.
- Routes learned via the router discovery client are assigned a preference of 50.
- No router solicitations are to be sent on the specified interface per the **quiet** option.

The Internet Control Message Protocol (ICMP)

GateD listens to Internet Control Message Protocol (ICMP) messages received by the system. Currently, ICMP reports on the condition of the network by providing information on tests and errors.

The processing of ICMP redirect messages is handled by the **redirect** statement. The only reason to specify the ICMP statement is to be able to trace the ICMP messages that GateD receives.

The ICMP Statement

The following is the full ICMP command statement. These statements are entered in the **gated.conf** file. Each part of the statement is described in the sections following this full statement.

```
icmp {  
    traceoptions trace_options;  
};
```

The syntax conventions used for showing the ICMP commands are the same as those described in Chapter 3 of this manual.

traceoptions *trace_options* ;

Specifies the tracing options for ICMP. Packet tracing options (which may be modified with **detail** and **recv**; for details on trace option modifiers, see Chapter 12 of this manual):

packets	All ICMP packets received.
redirect	Only ICMP redirect packets received.
routerdiscovery	Only ICMP router discovery packets received.
info	Only ICMP informational packets received, which include mask request/response, info request/response, echo request/response, and timestamp request/response.
error	Only ICMP error packets received, which include time exceeded, parameter problem, unreachable, and source quench.