Chapter 5
Interface Statements

aliases-nexthop

Name

`aliases-nexthop` - specifies the address that GateD will install as the next hop for the route associated with this interface

Syntax

```
aliases-nexthop ( primary | lowestip ) ;
```

ALSO

```
aliases-nh ( primary | lowestip ) ;
```

Parameters

`primary` - The primary interface address (default) will be installed as the next hop for the route associated with this interface. This is the recommended setting for all interfaces and is the default.

`lowestip` - The address with the lowest IP address will be installed as the next hop for the route associated with this interface.

Description

`aliases-nexthop` specifies which address GateD will install as the next hop for the route associated with an interface. GateD allows the use of aliases on interfaces: More than one logical interface can exist for each physical interface on the machine.

Typically, you create these logical interfaces using the ifconfig(1) UNIX command. Two options in the **`interfaces`** command affect the operation of GateD with respect to aliases.

1. **`aliases-nh ( lowestip | primary )`**
2. **`interface`** *`interface-name`* **`alias primary`** *`address`* **`mask`** *`mask`*

The configuration information in the **`interfaces`** command directly affects the behavior of the protocols when aliases are configured. When used with the **`options`** command, **`aliases-nh`** specifies the default behavior. When used with the **`interface`** command, **`aliases-nh`** indicates the default for aliases of the physical interface(s) specified.

When configured with **`aliases-nh primary`**, which is the default, GateD chooses a primary address on each subnet that is configured on the interface. The primary chosen by GateD is based on the order in which the addresses are read from the kernel. For example, consider a machine with one physical interface, le0, with five logical addresses:

```
le0: flags=1000843 <UP, BROADCAST, RUNNING, MULTICAST, IPv4> mtu 1500
     inet 172.16.0.178 netmask ffff0000 broadcast 172.16.255.255
     inet 172.16.0.179 netmask ffff0000 broadcast 172.16.255.255
     inet 12.1.1.2 netmask ff000000 broadcast 12.255.255.255
     inet 12.1.1.1 netmask ff000000 broadcast 12.255.255.255
     inet 192.168.10.1 netmask ffffff00 broadcast 192.168.10.255
```

In this case, GateD will mark the following interfaces as primary addresses:

172.16.0.178 for subnet 172.16.0.0/16

12.1.1.2 for subnet 12.0.0.0/8

192.168.10.1 for subnet 192.168.10.0/24

The flags for the interface can be seen in the gii **`show interfaces`** command, in the trace file after an interface scan, or in the GateD dump file. (See “GateD Interactive Interface” in *Operating GateD* for more information about gii.)

When configured as above, the protocols use the primary address for operation. Attempting to use a logical address that has not been marked as primary will lead to undesired results (to change the primary addresses, see below).

When using a physical interface name in the configuration file, some protocols will attempt to operate on all primary addresses on that interface. Here is an example OSPF statement:

```
ospf yes {
    backbone {
        interface le0 cost 1;
    }
}
```

When configured this way, OSPF will run over the three primary addresses shown above. In the case where there are no neighbors on some of the interfaces, stub links will be announced to these networks. See “Chapter 12 Open Shortest Path First (OSPF)” on page 45 in *Configuring GateD* for more information about OSPF.

To mark primary addresses for a subnet in the configuration file, use the **`alias primary`** option. GateD will allow only one primary address to be configured for each subnet on the interface; attempting to configure more than one will result in a parse error. Note that in addition to the interface address, the mask must be specified.

For interface routes the next hop for a direct subnet will be the primary address.

Using **`aliases-nh lowestip`**

Versions of GateD prior to 8.0 defaulted to using the lowest IP of an interface for all protocol operations. This feature has been left in place for compatibility. Note that aliases are not really supported with this option; the only valid logical interface is the interface with the numerically lowest IP address.

When configured to use **`lowestip`**, GateD will install routes to direct nets with a next hop of the lowest IP address for that network configured on the machine. We recommend that operators avoid using this option.

Defaults

```
aliases-nexthop primary ;
```

Context

`interfaces interface` statement

`interfaces options` statement

Examples

```
interfaces { options alias-nexthop lowestip ; } ;
```

See Also

“Chapter 7 Interface Statement” on page 23 of *Configuring GateD*

`interfaces` on page 30

`options` on page 36

alias primary

Name

`alias primary` - specifies primary addresses for an interface

Syntax

`alias primary` *`address`* **`mask`** *`mask ;`*

Parameters

`address` - the interface's primary network address

`mask` - a mask specifying a subnet

Description

`alias primary` is used to override the default selection of the primary logical interfaces on a given physical interface. There is exactly one primary interface for each subnet defined on a given physical interface. By default, GateD selects the first logical interface read from the kernel for each subnet defined on the physical interface as a primary interface. **`alias primary`** provides a way of specifying which of the logical interfaces that share the same subnet is to be selected as primary for that subnet.

Defaults

A default primary address is selected by GateD for each subnetwork defined on a physical interface. The default primary addresses are the addresses of the first interfaces read from the kernel for each subnet on a physical interface.

Context

`interfaces interface` statement

Examples

```
interfaces {
    interface ppp1 alias primary 142.77.34.184 mask 255.255.192.0;
};
```

See Also

`aliases-nexthop` on page 15

"Chapter 7 Interface Statement" on page 23 of *Configuring GateD*

`interfaces` on page 30

AS

Name

AS - specifies the autonomous system (AS) used to create an AS path associated with the route created from the definition of this interface

Syntax

AS *autonomoussystem ;*

Parameters

autonomoussystem - AS numbers are currently in the range 1 to 65535, inclusive.

Description

AS specifies the autonomous system that will be used to create an AS path associated with the route created from the definition of this interface. The autonomous system number of the router running GateD is specified in the global AS statement. The autonomous system numbers of BGP's peers is specified in BGP configuration. An interface defines a route, known as an "interface route". Interface routes are also known as direct routes. Specifying an AS number on the interface clause will cause the direct route corresponding to the interface to be generated with a non-empty AS path. If this route is then exported into BGP, update messages advertising this route will include the specified AS in their AS path. Normally, the interface AS number is not set.

Defaults

By default, no AS number is associated with an interface.

Context

interfaces interface statements

Examples

```
interfaces {
    interface ppp1
    AS 1439;
};
```

See Also

bgp statement on page 232

“Chapter 7 Interface Statement” on page 23 of *Configuring GateD*

interfaces on page 30

blackhole

Name

`blackhole` - causes the local address of this interface to be used as the next hop for blackhole routes.

Syntax

`blackhole ;`

Parameters

none

Description

`blackhole` specifies that this interface is the blackhole pseudo-interface supported by some older kernels. This interface must be of type "loopback" and should discard any packets sent to it without sending back "unreachable messages". This option is archaic and should no longer be necessary. Modern kernels recognize the RTF_BLACKHOLE flag associated with blackhole routes and directly perform the desired action of silently discarding packets sent to those routes.

Defaults

`blackhole` is disabled by default.

Context

`interfaces interface` statement

Examples

```
interfaces {
    interface bh0
    blackhole;
} ;
```

See Also

"Chapter 7 Interface Statement" on page 23 of *Configuring GateD*

"Chapter 19 Static Routes" on page 101 of *Configuring GateD*

`reject` on page 40

broadcast

Name

`broadcast` - defines the interface as broadcast capable

Syntax

`broadcast` *`address`* **`;`**

Parameters

`address` - the broadcast address

Description

`broadcast` defines the interface as broadcast capable (for example, Ethernet or Token Ring) and specifies the broadcast address. The broadcast address is typically the subnet address followed by all 1s. It is an error to specify a broadcast address on a point-to-point link.

Defaults

If the interface exists, the default broadcast address is that of the interface.

Context

`interfaces define` statement

Examples

```
interfaces {
    define subnet local 192.168.13.129
        netmask 255.255.255.252
        broadcast 192.168.13.131 ;
} ;
```

See Also

"Chapter 7 Interface Statement" on page 23 of *Configuring GateD*

`interfaces` on page 30

define

Name

`define`

Syntax

`define (subnet | p2p) local` *`address`* **`[`** *`options`* **`] ;`**

Parameters

`subnet` - defines a subnet type interface

`p2p` - defines a point-to-point link type interface

`address` - the local address to be used for the interface

`options` - options for this interface

Description

The **`define`** statement is used to alter the existing interface configuration, and to provide a default configuration should the specified interface(s) be UNIX configured, using ifconfig(1). **`define`** may specify interfaces that might not be present when GateD is started, enabling the configuration file to refer to them when **`strictinterfaces`** is defined.

`subnet` implies a broadcast or NBMA interface and defines a subnet (and refers to all interfaces on the subnet). When an interface comes up on this subnet, its configuration will be merged with the options present in the **`define`** clause.

`p2p` defines a point-to-point link. The configuration of a point-to-point interface will be merged with the options present in the **`define`** clause.

Possible **`define`** *`options`* are:

`broadcast` *`address`* - **`broadcast`** defines the interface as broadcast capable (for example, Ethernet or Token Ring) and specifies the broadcast address. The broadcast address is typically the subnet address followed by all 1s. It is an error to specify a broadcast address on a point-to-point link.

`remote` *`address`* - The remote address specifies the remote address of the logical point-to-point link. It is an error to specify a remote address if **`subnet`** is specified.

`tunnel` *`encapsulation_protocol`* - **`tunnel`** defines the encapsulation protocol to use for a point-to-point tunnel. Currently, only IPIP encapsulation (RFC 2003) is recognized. (IPIP encapsulation is used by multicast tunnels supported by kernels.) RFC 2003 is available at: http://ietf.org/rfc/rfc2003.txt

`netmask` *`address`* - **`netmask`** defines the netmask of a logical subnet. It is an error to specify a netmask address on a point-to-point link.

`[no]multicast` - **`[no]multicast`** explicitly enables/disables multicast on the interface. By default, IPIP tunnels are multicast, nounicast interfaces.

`[no]unicast` - **`[no]unicast`** explicitly enables/disables unicast routing on the interface. By default, IPIP tunnels are multicast, nounicast interfaces.

Defaults

If the interface exists, defaults come from it. An interface not defined as broadcast or point-to-point is assumed to be NBMA, such as an X.25 network. Remaining defaults are currently not part of the API specification.

Context

`interfaces` statement

Examples

Example 1

```
interfaces {
    interface ppp2;
    define p2p local 176.144.13.44 remote 176.144.12.92 nomulticast;
};
```

Example 2

This **`define`** configures a multicast-only IP-in-IP tunnel usable by routing protocols for the multicast RIB. (See Chapter 1, Section 8 “Multiple RIBs” on page 33 of *Configuring GateD* for more information about multicast RIBs.) Note that the keywords **`multicast nounicast`** here are redundant with the defaults for **`tunnel ipip`**. In fact, the standard multicast kernel cannot support any other combination.

```
interfaces {
    define p2p local 198.108.60.89 remote
        141.213.10.41 multicast nounicast
        tunnel ipip;
};
```

Example 3

This **`define`** tells GateD to treat the interface with the local address 192.168.12.114 as a subnet (192.168.12/24), even if it's actually a point-to-point link. (This does, however, require that the actual remote point-to-point address fall within the configured subnet prefix.)

```
interfaces {
    define subnet local 192.168.12.114 netmask
        255.255.255.0;
};
```

Example 4

This **`define`** shows how a /30 may be implemented in the **`define`** statement. The **`define`** tells GateD to treat the interface with a local address of 192.168.13.129 as a subnet type interface with a netmask of 255.255.255.252 and a broadcast address of 192.168.13.131.

```
    interfaces {
        define subnet local 192.168.13.129 netmask
            255.255.255.252
            broadcast 192.168.13.131;
    };
```

Example 5

This **`define`** tells GateD to treat the interface with the local address 192.168.13.114 as a point-to-point link to 192.168.13.116, even if it is not actually a point-to-point link. (If it is actually a subnet, this requires that the configured remote point-to-point address fall within the actual subnet prefix.)

```
    interfaces {
        define p2p local 192.168.13.114 remote
             192.168.13.116;
    };
```

See Also

"Chapter 7 Interface Statement" on page 23 of *Configuring GateD*

`interfaces` on page 30

disable

Name

`disable` - indicates that GateD will disable operation on this interface even if the kernel reports it as up

Syntax

```
disable ;
```

Parameters

none

Description

This option indicates that GateD shall disable operation on this interface even if the kernel reports it as up.

Defaults

Interfaces are enabled by default.

Context

interfaces interface statement

Examples

```
interfaces {
    interface all enable ;
    interface ex0 disable ;
} ;
```

See Also

“Chapter 7 Interface Statement” on page 23 of *Configuring GateD*

enable on page 27

interfaces on page 30

down preference

Name

`down preference` - sets the preference for routes to this interface when GateD does not believe it to be functioning properly, but the kernel does not indicate that it is down

Syntax

`down preference` *`downpreference`* **`;`**

Parameters

`downpreference` - Preferences are in the range 0 to 255, inclusive, with 0 being the highest preference a route can have.

Description

Sets the preference for routes to this interface when GateD does not believe it to be functioning properly, but the kernel does not indicate that it is down. See **`passive`** on page 38 for a description of “not functioning properly”.

Defaults

`down preference 120;`

Context

`interfaces interface` statement

Examples

```
interfaces {
    interface all down preference 150;
} ;
```

See Also

“Chapter 7 Interface Statement” on page 23 of *Configuring GateD*

`interfaces` on page 30

`passive` on page 38

`preference` on page 39

enable

Name

`enable` - indicates that GateD will enable this interface if it is reported as up by the kernel

Syntax

`enable ;`

Parameters

none

Description

This option exists for symmetry with the **`disable`** option. When present, it indicates that this interface shall be enabled by GateD if it is reported as up by the kernel.

Defaults

Interfaces are enabled by default.

Context

`interfaces interface` statement

Examples

```
interfaces {
    interface all disable ;
    interface ex0 enable ;
}
```

See Also

"Chapter 7 Interface Statement" on page 23 of *Configuring GateD*

`disable` on page 25

`interfaces` on page 30

interface

Name

`interface` - sets or modifies interface characteristics for the specified interfaces

Syntax

`interface` *`interface_list`* **`[`** *`options`* **`] ;`**

Parameters:

`interface_list` - one or more of the following:

`all` - all available interfaces

`wildcard` - **`interface`** *`wildcard`* refers to all the interfaces of the same type. UNIX interfaces consist of the name of the device driver, such as ie, and a unit number, such as 0, 5, or 22. References to the name contain only alphabetic characters and match any interfaces that have the same alphabetic part. For example, ie on a Sun would refer to all Interlan Ethernet interfaces, and le would refer to all Lance Ethernet interfaces. However, ie would not match iex0.

`name` - **`interface`** *`name`* refers to a specific interface, usually one physical interface. This name is specified as an alphabetic part followed by a numeric part. **`interface`** *`name`* will match one specific interface. Be aware that on many systems, more than one protocol (i.e., IP) address can be on a given physical interface. For example, ef1 will match an interface named ef1, but not an interface named ef10.

`address` - **`interface`** *`address`* matches one specific interface. The reference can be by protocol address (for example, 10.0.0.51), or by symbolic hostname (for example, nic.ddn.mil). Note that a symbolic hostname reference is valid only when it resolves to only one address. Use of symbolic hostnames is not recommended.

This address specifies the unique address of the interface referred to. The unique address is the local address for all but point-to-point interfaces and is the remote address for point-to-point interfaces.

`local` *`address`* - **`local`** *`address`* is the same as *`address`* except that the local address (not remote) is matched.

`remote` *`address`* - **`remote`** *`address`* is the same as *`address`* except that the remote address (not local) is matched.

`options` - one or more of the following:

`preference` - specifies the preference of the interface when up

`down preference` - specifies the preference of the interface when down

`enable` - specifies that the interface is to be used by GateD

`disable` - specifies that the interface is not used by GateD

`passive` - specifies that the interface is not to be marked as down due to inactivity

`simplex` - indicates that the interface does not receive its own multicast packets

`reject` - specifies that the interface is to be used by the kernel for reject routes

`blackhole` - specifies that the interface is to be used by the kernel for blackhole routes

AS *autonomoussystem* - associates the value of *autonomoussystem* with the route created for this interface

alias primary *address* **mask** *mask* - specifies the primary address for this interface

aliases-nexthop (primary | lowestip) - specifies how to select the next hop for this interface

Description

Define a set of interfaces. Multiple wildcards, names and addresses may be specified. It is an error if the interface list is empty.

If many interface lists with more than one parameter are present in the configuration file, these parameters are collected at run-time to create the specific parameter list for a given interface. If the same parameter is specified on more than one list, the parameter with the most specific interface is used.

Defaults

none

Context

interfaces statement

Examples

```
interfaces { interface en ex1 disable ; } ;
```

See Also

"Chapter 7 Interface Statement" on page 23 of *Configuring GateD*

interfaces on page 30

preference on page 39

down preference on page 26

enable on page 27

disable on page 25

passive on page 38

simplex on page 44

reject on page 40

blackhole on page 20

AS on page 19

alias primary on page 18

aliases-nexthop on page 15

interfaces

Name

interfaces

Syntax

```
interfaces {
    [ options
        [ strictinterfaces ]
        [ scaninterval time ]
        [ aliases-nexthop ( primary | lowestip ) ] ; ]
    [ interface interface_list
        [ preference interfacepreference ]
        [ down preference downpreference ]
        [ enable ]
        [ disable ]
        [ passive ]
        [ simplex ]
        [ reject ]
        [ blackhole ]
        [ AS autonomoussystem ]
        [ alias primary address mask mask ]
        [ aliases-nexthop ( primary | lowestip ) ] ; ]
    [ define ( subnet | p2p ) local address
        [ broadcast address ]
        [ remote address ]
        [ tunnel encapsulation_protocol ]
        [ netmask mask ]
        [ multicast | nomulticast ]
        [ unicast | nounicast ] ; ]
};
```

Parameters

options

interface

define

Description

An interface is the connection between a router and one of its attached networks. A physical interface may be specified by interface name, by IP address, or by domain name. Multiple levels of reference in the configuration language allow interfaces to be identified using wildcard or interface type name. The *`interface_list`* is a list of one or more interface names, including wildcard names (names without a number) and names that may specify more than one interface or address, or the token "all" for all interfaces.

The *BSD 4.3* and later networking implementations allow four types of interfaces. Some implementations allow multiple protocol addresses per physical interface. These implementations are mostly based on *BSD 4.3 Reno* or later.

Loopback

Loopback must have the address of 127.0.0.1 or ::1. Packets sent to loopback are sent back to the originator. This interface is also used as a catch-all interface for implementing other features, such as reject and blackhole routes.

Although a netmask is reported on this interface, it is ignored. Assign an additional address to this interface that is the same as the OSPF or BGP router ID to allow routing to a system based on the router ID that will work if some interfaces are down.

Broadcast

Broadcast is a multi-access interface capable of a physical level broadcast, such as Ethernet, Token Ring and FDDI. This interface has an associated subnet mask and broadcast address. The interface route to a broadcast network will be a route to the complete subnet.

Point-to-point

Point-to-point is a tunnel to another host, usually on some sort of serial link. This interface has a local address and a remote address. Although it may be possible to specify multiple addresses for a point-to-point interface, there does not seem to be a useful reason for doing so. The remote address must be unique among all the interface addresses on a given router.

The local address may be shared among many point-to-point and up to one non-point-to-point interface. Point-to-point is technically a form of the router ID method for addressless links. This technique conserves subnets because none are required when using it.

If a subnet mask is specified on a point-to-point interface, it is only used by RIP version 1 to determine which subnets may be propagated to the router on the other side of this interface.

Non-broadcast multi-access (NBMA)

NBMA is multi-access, but not capable of broadcast. An example of this would be frame relay and X.25. This type of interface has a local address and a subnet mask. For consistency, GateD ensures that there is a route available to each IP interface that is configured and up. Normally this is done by the "ifconfig" command that configures the interface.

To ensure consistency, GateD installs a route in the kernel's FIB for the address of each IP interface that is configured and up.

For point-to-point interfaces, GateD installs some special routes. If the local address on one or more point-to-point interfaces is not shared with a non-point-to-point interface, GateD installs a route to the local address pointing at the loopback interface with a preference of 110. This ensures that packets originating on this host destined for this local address are handled locally.

OSPF prefers to route packets for the local interface across the point-to-point link where they will be returned by the router on the remote end. This is used to verify operation of the link. Because OSPF installs routes with a preference of 10, these routes will override the route installed with a preference of 110.

If the local address of one or more point-to-point interfaces is shared with a non-point-to-point interface, GateD installs a route to the local with a preference of 0 that will not be installed in the forwarding table. This is to prevent protocols like OSPF from routing packets to this address across a serial interface when this system could be functioning as a host. When the status of an interface changes, GateD notifies all the protocols, which take the appropriate action. GateD assumes that interfaces that are not marked UP do not exist.

GateD ignores any interfaces that have invalid data for the local, remote, or broadcast addresses, or the subnet mask. Invalid data includes zeros in any field. GateD will also ignore any point-to-point interface that has the same local and remote addresses. GateD assumes that the interface is in some sort of loopback test mode.

Defaults

interfaces configured by the UNIX ifconfig(1) command

Context

global

Examples

```
interfaces { options strictinterfaces scaninterval 10:0 ; } ;
interfaces {
      options aliases-nexthop lowestip ;
      interface all down preference 150;
      interface en2 preference 32 down preference 37 passive simplex AS
            18432 alias primary 7.4.17.76 ;
      define p2p local 176.144.13.44 remote 176.144.12.92 nomulticast;
} ;
```

See Also

“Chapter 7 Interface Statement” on page 23 of *Configuring GateD*

`export` on page 623

`import` on page 605

`options` on **page 36**

`interface` on page 28

`define` on page 22

multicast | nomulticast

Name

`multicast | nomulticast` - explicitly enables/disables multicast on a defined interface

Syntax

`multicast | nomulticast`

Parameters

none

Description

explicitly enables/disables multicast on a defined interface. By default, IPIP tunnels are multicast, nounicast interfaces.

Defaults

`multicast`

Context

`interfaces define` statement

Examples

```
interfaces {
    define p2p local 176.144.13.44
        remote 176.144.13.92
        nomulticast ;
} ;
```

See Also

“Chapter 7 Interface Statement” on page 23 of *Configuring GateD*

`interfaces` on page 30

netmask

Name

`netmask` - defines the netmask of a logical subnet

Syntax

`netmask` *`mask`*

Parameters

`mask` - the network mask for the subnet of the interface being defined

Description

`netmask` defines the netmask of a logical subnet. It is an error to specify a netmask address on a point-to-point link.

Defaults

`netmask` defaults to the host mask for point-to-point interfaces, and cannot be changed. For non-point-to-point, the netmask defaults to that of the existing interface with the same local address, if any.

Context

`interfaces define` statement

Examples

```
    interfaces {
        define subnet local 192.168.13.114
        netmask 255.255.255.0 ;
    } ;
```

See Also

“Chapter 7 Interface Statement” on page 23 of *Configuring GateD*

`interfaces` on page 30

options

Name

`options` - specifies interface related global options

Syntax

```
options [ strictinterfaces ]
    [ scaninterval time ]
    [ aliases-nexthop ( primary | lowestip ) ];
```

Parameters

`strictinterfaces` - **`strictinterfaces`** indicates that it is a fatal error to refer to an interface in the configuration file that is not present when GateD is started and not listed in a define statement. Without **`strictinterfaces`**, a warning message will be issued, but GateD will continue.

`scaninterval` *`time`* - **`scaninterval`** specifies how often GateD scans the kernel interface list for changes. The default is every 15 seconds on systems without a routing socket, and 60 seconds on systems that pass interface status changes through the routing socket. *`time`* must be in the range of 15-3600 seconds. Note that GateD will also scan the interface list on receipt of a SIGUSR2. The time may be specified as integer seconds, or mintues:seconds, or hours:minutes:seconds. For example:

`30` = 30 seconds

`1:30` = 1 minute, 30 seconds, or 90 seconds

`1:0:0` = 1 hour (or 3600 seconds)

For operating systems with a routing socket, a *`time`* of zero disables periodic interface scans. Systems that utilize a routing socket that do not prevent against loss of data on the socket may result in a FIB that is inconsistent with GateD's routing table. A **`scaninterval`** of zero is highly discouraged on systems with a "lossy" routing socket.

`aliases-nexthop ( primary | lowestip )` - **`aliases-nexthop`** specifies which address GateD will install as the next hop for interface routes. If **`primary`** is used, the primary interface address (default) will be installed. If **`lowestip`** is used, the address with the lowest IP address will be installed. This **`options`** value sets a global parameter that may be overridden for particular interfaces using the **`interface`** option of the same name. **`primary`** is the recommended setting.

Description

`options` enables configuration of three global options related to interfaces. The options are **`strictinterfaces`**, **`scaninterval`**, and **`aliases-nexthop`**.

Defaults

`strictinterfaces` is disabled. **`scaninterval`** defaults to 15 seconds on systems without a routing socket and 60 seconds otherwise. **`aliases-nexthop`** defaults to the primary interface address.

Context

`interfaces` statements

Examples

Example 1

Enable strict interfaces.

```
interfaces { options strictinterfaces ; } ;
```

Example 2

Set scan interval to 5 minutes.

```
interfaces { options scaninterval 300 ; } ;
```

Example 3

Use lowest IP address as the next hop for alias interface routes.

```
interfaces { options aliases-nexthop lowestip ; } ;
```

Example 4

For operating systems with a routing socket, a scan-interval of zero disables periodic interface scans.

```
interfaces {
    options scaninterval 0 ;
};
```

See Also

"Chapter 7 Interface Statement" on page 23 of *Configuring GateD*

`interfaces` on page 30

passive

Name

`passive`

Syntax

`passive ;`

Parameters

none

Description

`passive` prevents GateD from changing the preference of the route to this interface if it is not believed to be functioning properly due to lack of received routing information. A mechanism called aging is implemented by protocol modules to detect a likely malfunctioning interface. Protocols using an interface maintain timestamps indicating when a message was last received over the interface. The timestamps are typically updated in response to received "hello" messages. If a protocol message is not received on the interface within an "aging period" (currently, 300 seconds), the interface is marked down. An interface being marked down causes the "interface route" preference to be changed to "down preference".

Defaults

This "aging mechanism" is enabled only if the GateD configure option "**`--enable-interfacer_aging`**" is configured. Otherwise, aging is disabled. Also note that "aging" does not take place if no protocols are using the interface.

Context

`interfaces` statement

Examples

```
interfaces {
    interface all
    passive;
};
```

See Also

“Chapter 7 Interface Statement” on page 23 of *Configuring GateD*

`down preference` on page 26

`interfaces` on page 30

`preference` on page 39

preference

Name

`preference` - used to select the best route when multiple routes exist for the same destination

Syntax

`preference` *`preference`* **`;`**

Parameters

`preference` - Preferences are in the range 0 to 255 inclusive, with 0 being the highest preference a route may have.

Description

Multiple routes (prefix, next hop) may exist for the same destination. When multiple routes exist for the same destination, the route's preference is used to select the best route. For interface routes, **`preference`** sets the preference for routes to this interface when it is up and appears to be functioning properly.

Defaults

`interface` routes default to:

```
preference 0;
```

`static` routes default to:

```
preference 60;
```

Context

`interfaces interface` statement

Examples

```
interfaces {
    interface all preference 14;
};
```

See Also:

"Chapter 7 Interface Statement" on page 23 of *Configuring GateD*

"Chapter 19 Static Routes" on page 101 of *Configuring GateD*

`down preference` on page 26

`interface` on page 28

`interfaces` on page 30

reject

Name

`reject` - causes the local address of this interface to be used as the next hop for reject routes.

Syntax

`reject ;`

Parameters

none

Description

`reject` specifies that this interface is the reject pseudo-interface supported by some older kernels. This interface must be of type “loopback” and should discard any packets sent to it, sending back an “unreachable message”. This option is archaic and should no longer be necessary. Modern kernels recognize the RTF_REJECT flag associated with reject routes and directly perform the desired action of discarding packets sent to those routes and returning an “unreachable message” to the originator of the packet.

Defaults

`reject` is disabled by default.

Context

`interfaces interface` statement

Examples

```
interfaces {
    interface rj0 reject;
} ;
```

See Also

“Chapter 7 Interface Statement” on page 23 of *Configuring GateD*

“Chapter 19 Static Routes” on page 101 of *Configuring GateD*

`blackhole` on page 20

remote

Name

`remote` - specifies the remote address of a logical point-to-point link

Syntax

`remote` *`address`*

Parameters

`address` - the host address of the remote system on a point-to-point interface

Description

The **`remote`** *`address`* specifies the remote address of the logical point-to-point link. It is an error to specify a remote address if **`subnet`** is specified as the defined interface type.

Defaults

none

Context

`interfaces define` statement

Examples

```
interfaces {
    define p2p local 176.144.13.44
        remote 176.144.13.92
        nomulticast ;
} ;
```

See Also

"Chapter 7 Interface Statement" on page 23 of *Configuring GateD*

`interfaces` on page 30

scaninterval

Name

`scaninterval` - specifies how often GateD scans the kernel interface list for changes

Syntax

`scaninterval` *`time`*

Parameters

`time` - is a time from 0 to 3600 seconds and can be specified as:

`seconds`

`minutes``:`*`seconds`*

`hours``:`*`minutes`*`:`*`seconds`*

Description

`scaninterval` specifies how often GateD scans the kernel interface list for changes. The default is every 15 seconds on systems that don't have a routing socket, and 60 seconds on systems that pass interface status changes through the routing socket. *`time`* must be in the range of 15-3600 seconds. Note that GateD will also scan the interface list on receipt of a SIGUSR2. The time may be specified as integer seconds, or mintues:seconds, or hours:minutes:seconds. For example:

30 = 30 seconds

1:30 = 1 minute, 30 seconds, or 90 seconds

1:0:0 = 1 hour (3600 seconds)

For operating systems with a routing socket, a time of zero disables periodic interface scans. Systems that utilize a routing socket that do not prevent against loss of data on the socket may result in a FIB that is inconsistent with GateD's routing table. A scaninterval of zero is highly discouraged on systems with a "lossy" routing socket.

Defaults

`scaninterval 15` for most systems

`scaninterval 60` on systems that pass interface status changes through the routing socket

Context

`interfaces options` statement

Examples

Set scan interval to 5 minutes.

```
interfaces { options scaninterval 5:0 ; } ;
```

See Also

"Chapter 7 Interface Statement" on page 23 of *Configuring GateD*

`interfaces` on page 30

simplex

Name

`simplex` - defines an interface as unable to hear its own multicast packets

Syntax

```
simplex ;
```

Parameters

none

Description

`simplex` defines an interface as unable to hear its own multicast packets. Some systems define an interface as simplex with the IFF_SIMPLEX flag. On others, it needs to be specified in the configuration file. On simplex interfaces, a received multicast packet with a source address belonging to the interface on which the packet was received is assumed to have been looped back in software and is not used as an indication that the interface is functioning properly. This flag is valid on non-broadcast interfaces.

Defaults

If the interface has been configured by the utility ifconfig(1), then the default is based on the kernel IFF_SIMPLEX flag; otherwise, the simplex flag is off by default.

Context

`interfaces interface` statement

Examples

```
interfaces {
    interface en simplex;
};
```

See Also

“Chapter 7 Interface Statement” on page 23 of *Configuring GateD*

`interfaces` on page 30

strictinterfaces

Name

`strictinterfaces` - indicates that it is a fatal error to refer to an interface in the configuration file that is not present when GateD is started and not listed in a **`define`** statement

Syntax

`strictinterfaces`

Parameters

none

Description

`strictinterfaces` indicates that it is a fatal error to refer to an interface in the configuration file that is not present when GateD is started and not listed in a **`define`** statement. Without **`strictinterfaces`**, a warning message will be issued, but GateD will continue.

Defaults

`strictinterfaces` defaults to off.

Context

`interfaces options` statement

Examples

```
interfaces { options strictinterfaces ; } ;
```

See Also

“Chapter 7 Interface Statement” on page 23 of *Configuring GateD*

`interfaces` on page 30

tunnel

Name

`tunnel` - defines the encapsulation protocol to use for a point-to-point tunnel

Syntax

`tunnel` *encapsulation_protocol*

Parameters

encapsulation_protocol - Currently, the only valid protocol is **`ipip`**.

Description

`tunnel` defines the encapsulation protocol to use for a point-to-point tunnel. Currently only IPIP encapsulation (RFC 2003) is recognized. (IPIP encapsulation is used by multicast tunnels supported by kernels.) RFC 2003 is available at:

http://www.ietf.org/rfc/rfc2003.txt

Defaults

A defined interface is not a tunnel by default.

Context

`interfaces define` statement

Examples

```
interfaces {
    define p2p local 198.108.60.89 remote
    141.213.10.41 multicast nounicast
    tunnel ipip;
};
```

See Also

"Chapter 7 Interface Statement" on page 23 of *Configuring GateD*

`interfaces` on page 30

unicast | nounicast

Name

`unicast | nounicast` - explicitly enables/disables unicast on the interface

Syntax

`unicast | nounicast`

Parameters

none

Description

explicitly enables/disables unicast on the interface. By default, IPIP tunnels are multicast, nounicast interfaces. Setting this flag causes a route for this interface to not be installed in the kernel.

Defaults

`unicast` is the default for all defined interfaces.

Context

`interfaces define` statement

Examples

```
interfaces {
    define p2p local 198.108.60.89 remote
    141.213.10.41 multicast nounicast
    tunnel ipip;
};
```

See Also

“Chapter 7 Interface Statement” on page 23 of *Configuring GateD*

`interfaces` on page 30