Chapter 9
Intermediate System to Intermediate System (IS-IS)

area

Name

area - configures the IS-IS area of the router

Syntax

area *D.D.D.D* ;

area *HH.HHHH.HHHH.HHHH.HHHH.HHHH* ;

Parameters

D - a decimal integer between 0 and 255. The **area** *D.D.D.D* form of the command is supplied to be interchangeable with the **area** command in OSPF. (See “OSPF” on page 45 in *Configuring GateD*.) This form of **area** takes as its argument a dotted-quad (IPv4-like) address. This form will set the area to be 4 octets as given by the four decimal numbers in the dotted-quad argument.

H - a hexadecimal digit between 0 and F. The more general form of **area** uses hexadecimal digits and is variable in length. The area will be as long as the given argument dictates. Each pair of hexadecimal digits is a single octet, and thus an even number of hex-digits must be given.

Description

area sets the area of the router. In OSI, each router (IS) has a network entity title (NET) assigned to it. The NET uniquely identifies the IS throughout the routing domain and determines which area the router is in.

A NET is divided into the area portion, then the system ID, and completed by a single 0 octet (byte). The system ID is always (by convention) 6 octets. Thus, the area portion is variable and can be from 1 to 13 octets.

Some routers require that the user specify the area and system ID by setting the NET directly. This mechanism is for the most part outdated because most organizations do not actually run an OSI routing domain; it has been simplified to resemble other current routing protocols, such as OSPF.

Note: This option can appear up to three times.

Default

```
area 00.0000.0000.0000 ;
```

Context

isis statement

Examples

Example 1

The following example configures the router to be in area 0.0.0.3 (i.e., 3 octets of 0 followed by 1 of 3).

```
isis on {
    area 0.0.0.3;
};
```

Example 2

This example configures the router's area using the hexadecimal form; its length is 5 octets and the final octet has the value 31.

```
isis on {
    area 00.0000.001F;
};
```

See Also

"Chapter 13 Intermediate System to Intermediate System (IS-IS)" on page 55 of *Configuring GateD*

isis on page 185

systemid on page 222

area auth

Name

area auth - configures the area password(s)

Syntax

```
area auth simple key ;
area auth md5 key key ;
area auth {
    [ simple key ; ]
    [ md5 key key ; ]
    [ md5 key key {
        [ start-accept YYYY/MM/DD HH:MM [.ss] ; ]
        [ stop-accept YYYY/MM/DD HH:MM [.ss] ; ]
        [ start-generate YYYY/MM/DD HH:MM [.ss] ; ]
        [ stop-generate YYYY/MM/DD HH:MM [.ss] ; ]
    } ; ]
} ;
```

Parameters

key - a password of length 1 to 255, given inside double quotes

Description

IS-IS authentication is divided up into three sections: hello authentication, level 1 authentication and level 2 authentication.

area auth configures the level 1 authentication. All non-hello level 1 packets are authenticated using the **area auth** *key*. This includes level 1 link state packets (LSPs), level 1 partial sequence number (PSN) packets, and level 1 complete sequence numbers (CSN) packets.

All level 1 routers in the area should be configured with the same key. If multiple keys are specified, the first key in the list is used to authenticate transmitted packets and all the keys in the list are used to authenticate received packets. Many older (and non-standard) implementations do not actually authenticate the PSN and CSN packets, thus another command (**require-snp-auth** – see page 203) is provided to enable or disable SNP authentication. For interoperability reasons, SNP authentication is off by default.

The MD5 time ranges may be used to roll over one key to another. The time ranges are a convenience that allow the router to automatically roll over the key. A sufficient window of time should be used in which routers are accepting both keys to allow a smooth transition. The expiration of a generate time does not cause level 1 LSP to be regenerated; therefore, a sufficient window of time should be allocated to allow for normal refreshing of the LSP, which would cause the new authentication to be used.

Default

no authentication

Context

isis statement

Examples

The following example configures the area password to be “my-area-password”.

```
isis on {
    area auth simple “my-area-password” ;
};
```

See Also

auth on page 157

“Chapter 13 Intermediate System to Intermediate System (IS-IS)” on page 55 of *Configuring GateD*

domain auth on page 166

isis on page 185

require-snp-auth on page 203

auth

Name

auth - configures the interface password(s)

Syntax

```
auth simple key [ level ( 1 | 2 | 1 and 2 ) ] ;
auth md5 key key ;
auth {
    [ simple key ; ]
    [ md5 key key ; ]
    [ md5 key key  {
        [ start-accept YYYY/MM/DD HH:MM [.ss] ; ]
        [ stop-accept YYYY/MM/DD HH:MM [.ss] ; ]
        [ start-generate YYYY/MM/DD HH:MM [.ss] ; ]
        [ stop-generate YYYY/MM/DD HH:MM [.ss] ; ]
    }; ]
[ level ( 1 | 2 | 1 and 2 ) ] ;
}
```

Parameters

key - a password of length 1 to 255, given inside double quotes

Description

IS-IS authentication is divided into three sections: hello authentication, level 1 authentication, and level 2 authentication.

auth configures the hello authentication. All hello packets are authenticated using the **auth** *key*. The level subcommands allow separate level 1 and level 2 hello authentication keys to be specified.

All routers at a given level on a link should be configured with the same key. If multiple keys are specified, the first key in the list is used to authenticate transmitted packets and all the keys in the list are used to authenticate received packets.

The MD5 time ranges may be used to roll-over one key to another. The time ranges are a convenience that allow the router to automatically roll over the key. A sufficient window of time should be used where routers are accepting both keys to allow a smooth transition.

Default

no authentication

Context

isis interface statement

Examples

Example 1

The following example configures the interface eth0's password to “my-intf-password” for level 1 and 2.

```
isis on {
    interface eth0 {
        auth simple “my-intf-password” ;
    };
};
```

Example 2

This example configures eth0's level 1 interface password to be “my-level-1-password” and the level 2 interface password to be “my-level-2-password”.

```
isis on {
    interface eth0 {
        auth simple “my-level-1-password” level 1;
        auth simple “my-level-2-password” level 2;
    };
};
```

Example 3

This example enables md5 authentication on interface “eth0” for level 1 and 2.

```
isis on {
    interface eth0 {
        auth md5 key “my-key” level 1 and 2;
    };
};
```

Example 4

This example enables key rollover on interface “eth0”.

```
isis on {
    interface eth0 {
        auth md5 key “key-1” {
            start-generate 2001/10/1 12:00 ;
            stop-generate 2001/10/2 12:00 ;
            start-accept 2001/10/1 12:00 ;
            stop-accept 2001/10/3 12:00 ;
        };
        auth md5 key “key-2” {
```

```
                start-generate 2001/10/1 12:00 ;
                stop-generate 2002/10/2 12:00 ;
                start-accept 2001/10/1 12:00 ;
                stop-accept 2002/10/3 12:00 ;
            };
        };
    };
```

See Also

area auth on page 155

"Chapter 13 Intermediate System to Intermediate System (IS-IS)" on page 55 of *Configuring GateD*

domain auth on page 166

interface on page 184

isis on page 185

config-time

Name

config-time - used to set the ISO IS config time in the operating system kernel

Syntax

config-time *seconds***;**

Parameters

seconds - the value of the config time

Description

This value is used to set the ISO IS config time in the operating system kernel. This requires ISO socket support in the operating system.

Default

config-time 60;

Context

isis statement

Examples

```
isis on {
        config-time 20;
        interface ex0 cost 1;
};
```

See Also

"Chapter 13 Intermediate System to Intermediate System (IS-IS)" on page 55 of *Configuring GateD*

isis on page 185

csn-interval

Name

csn-interval - configures the Complete Sequence Number (CSN) interval for the interface

Syntax

csn-interval *intervalnum* **[level (1 | 2 | 1 and 2)] ;**

Parameters

intervalnum - a number of seconds between 1 and 100

Description

The Designated Intermediate System (DIS) on a broadcast link periodically multicasts CSN packets to the link. The CSN packets summarize the current link state packet (LSP) database as it appears on the DIS. This allows the other routers on the link to request updates or to know that they should send updates. This method keeps all the routers on the link synchronized (for example, having the same database).

Every *intervalnum* seconds, the DIS will multicast as many CSN packets as are required to completely describe its LSP database. Setting *intervalnum* lower can decrease convergence time at the expense of more link usage. Setting *intervalnum* higher will decrease link usage at the expense of convergence time. The level subcommands allow separate *intervalnum* intervals for each level at which the interface operates. If no level is specified, **level 1 and 2** is assumed.

This command is typically ignored on point-to-point links because CSNPs are transmitted only upon establishment of the adjacency. In order for CSN to be transmitted periodically on point-to-point links, according to the **csn-interval** parameter, the **periodic-csn** command must be used.

Default

csn-interval 10 level 1 and 2;

Context

isis interface statement

Examples

Example 1

The following example sets the CSN interval to 20 seconds for all levels.

```
isis on {
    interface eth0 {
        csn-interval 20;
    }
```

```
};
```

Example 2

This example sets the level 1 CSN interval to 15 seconds, and sets the level 2 CSN interval to 20 seconds.

```
isis on {
    interface eth0 {
        csn-interval 15 level 1;
        csn-interval 20 level 2;
    }
};
```

See Also

"Chapter 13 Intermediate System to Intermediate System (IS-IS)" on page 55 of *Configuring GateD*

interface on page 184

isis on page 185

priority on page 200

periodic-csn on page 202

disable

Name

disable - disables IS-IS on an interface

Syntax

```
disable ;
```

Parameters

none

Description

If **disable** is specified in the **interface** statement, the interface will not be considered a part of the IS-IS routing domain.

Default

not applicable

Context

isis interface statement

Examples

The following example disables interface eth0.

```
isis on {
    interface eth0 {
       disable;
    };
};
```

See Also

"Chapter 13 Intermediate System to Intermediate System (IS-IS)" on page 55 of *Configuring GateD*

enable on page 169

interface on page 184

isis on page 185

dis-hello-interval

Name

dis-hello-interval - configures the hello interval for a designated intermediate system (DIS) interface

Syntax

dis-hello-interval *intervalnum* **[level (1 | 2 | 1 and 2)] ;**

Parameters

intervalnum - a number of seconds between 1 and 100

Description

When a router is elected DIS for a link, the router switches to a different hello interval than in the non-DIS case. **dis-hello-interval** allows you to set the new interval length to use when becoming DIS. The level subcommand allows separate intervals for each level.

The **dis-hello-interval** is used to calculate the hold time announced in hello packets as follows:

hold time = **dis-hello-interval** * **hello-multiplier**

Each hello packet contains a hold time. The hold time informs the receiving routers how long to wait without seeing another hello from the sending router before considering the sending router down.

Default

dis-hello-interval 3 level 1 and 2;

Context

isis interface statement

Examples

Example 1

The following example sets the DIS hello interval on interface eth0 to 5 seconds.

```
isis on {
    interface eth0 {
        dis-hello-interval 5;
    }
};
```

Example 2

This example sets per-level DIS hello intervals for interface eth0. For level 1, the interval is set to 4; for level 2, the interval is set to 6.

```
isis on {
    interface eth0 {
        dis-interval 4 level 1;
        dis-hello-interval 6 level 2;
    }
};
```

See Also

"Chapter 13 Intermediate System to Intermediate System (IS-IS)" on page 55 of *Configuring GateD*

`hello-interval` statement on page 176

`hello-multiplier` statement on page 178

`interface` statement on page 184

`isis` statement on page 185

`priority` on page 200

domain auth

Name

domain auth - configures the domain password(s)

Syntax

```
domain auth simple key ;
domain auth md5 key key ;
domain auth {
    [ simple key ; ]
    [ md5 key key ; ]
    [ md5 key key {
        [ start-accept YYYY/MM/DD HH:MM [.ss] ; ]
        [ stop-accept YYYY/MM/DD HH:MM [.ss] ; ]
        [ start-generate YYYY/MM/DD HH:MM [.ss] ; ]
        [ stop-generate YYYY/MM/DD HH:MM [.ss] ; ]
    } ; ]
} ;
```

Parameters

key - a password of length 1 to 255, given inside double quotes

Description

IS-IS authentication is divided into three sections: hello authentication, level 1 authentication, and level 2 authentication.

domain auth configures the level 2 authentication. All non-hello level 2 packets are authenticated using the **domainauth** *key*. This includes level 2 LSPs, level 2 partial sequence numbers (PSN) packets, and level 2 complete sequence numbers (CSN) packets. All level 2 routers in the routing domain should be configured with the same key. If multiple keys are specified, the first key in the list is used to authenticate transmitted packets, and all the keys in the list are used to authenticate received packets. Many older (and non-standard) implementations do not actually authenticate the PSN and CSN packets; thus, another command (**require-snp-auth** – see page 203) is provided to enable or disable SNP authentication. For interoperability reasons, SNP authentication is off by default.

Default

no authentication

Context

isis statement

Examples

The following example configures the domain password to be “my-domain-password”.

```
isis on {
    area auth simple “my-domain-password” ;
};
```

See Also

area auth on page 155

auth on page 155

“Chapter 13 Intermediate System to Intermediate System (IS-IS)” on page 55 of *Configuring GateD*

isis on page 185

require-snp-auth on page 203

domain-wide

Name

`domain-wide` - enables or disables domain-wide extension

Syntax

`domain-wide ( on | off ) ;`

Parameters

none

Description

The **`domain-wide`** statement enables support for redistributing routes from level 2 down into level 1. This method can be used to achieve more optimal routing. Normally (with **`domain-wide off`**), level 1 routers send packets destined for networks not within their area to the closest level 2 router. If multiple level 2 routers are active for the area, the closest level 2 router may not actually be the closest in overall path cost.

A level 2 router can filter some of the prefixes advertised into the level 1 area through the use of the **`summary-filter`** statement (see page 216).

All level 2 routers must be configured the same way, with domain-wide routing either **`on`** or **`off`**. If all level 2 routers are not configured the same way, forwarding loops may arise.

This functionality is described in RFC 2966.

Default

`domain-wide off;`

Context

`isis` statement

Examples

The following example configures the router to advertise level 2 routes into its level 1 area.

```
isis on {
    domain-wide on;
};
```

See Also

"Chapter 13 Intermediate System to Intermediate System (IS-IS)" on page 55 of *Configuring GateD*

`isis` on page 185

`summary-filter` on page 216

enable

Name

enable - enables IS-IS on an interface

Syntax

enable ;

Parameters

none

Description

If **enable** is specified in the **interface** statement, the interface will be considered a part of the IS-IS routing domain. **enable** is present for completeness. (**enable** is the opposite of **disable**.)

Default

enable ;

Context

isis interface statement

See Also

"Chapter 13 Intermediate System to Intermediate System (IS-IS)" on page 55 of *Configuring GateD*

disable statement on page 163

interface statement on page 184

isis statement on page 185

es-config-time

Name

es-config-time - used to set the ISO ES config time in the operating system kernel

Syntax

es-config-time *seconds***;**

Parameters

seconds - the value of the ES config time

Description

This value is used to set the ISO ES config time in the operating system kernel. This requires ISO socket support in the operating system.

Default

es-config-time 60;

Context

isis statement

Examples

```
isis on {
    es-config-time 60;
    interface ex0 cost 1;
};
```

See Also

"Chapter 13 Intermediate System to Intermediate System (IS-IS)" on page 55 of *Configuring GateD*

isis statement on page 185

export-defaults

Name

export-defaults

Syntax

export-defaults metric-type (internal | external);

export-defaults metric (*metricnum* **| inherit);**

export-defaults level (1 | 2);

Parameters

metric-type [internal | external] - default metric type for exported reachability

metric *metricnum* - default metric for exported reachability, an integer between 1 and max_metric

metric inherit - default metric for exported reachability

level [1 | 2] - default level into which to export exported reachability

Description

This command sets the defaults for exportation of exported reachability. The values configured in the export clause may override what is given here.

export-defaults metric-type configures the default metric type that routes from other protocols receive when they are exported into IS-IS. The default metric type is used only when the user does not specify the metric type in the **export** statement. IS-IS has two distinct metric types, **internal** and **external**. Routes with internal metrics are preferred over routes with external metrics at the same level. Interface metrics advertised by the IS are considered **internal**. If the exported route will have a metric compatible with the interface metrics, it should be exported with metric type **internal**. If the metric is not compatible with interface metrics, it should be exported with metric type **external**.

In any case, routes originating in level 1 are always preferred over level 2, and routes originating in level 2 are always preferred over routes being summarized back into level 1. Note that the same route originated in IS-IS (i.e., an interface route) and exported into IS-IS with internal metric will be directly compared. The exported version could be preferred to the originated version if the cost is smaller.

export-defaults metric configures the default metric that routes from other protocols receive when they are exported into IS-IS. The default metric is used only when the user does not specify the metric in the **export** statement. The **inherit** form of the statement causes the source protocol's **metric** to be used. It is assumed that the metrics are compatible in meaning. If the source protocol's **metric** exceeds *max_metric*, the exported metric will be set to *max_metric*, where *max_metric* is 63 if **rfc1195-metrics** is enabled; otherwise, it is 2^{24}-1 (or 4,261,412,864).

The **export-defaults level** statement configures the default level into which routes from other protocols are injected. The default level is used only when the user does not specify the level on the **export** statement. Exporting routes into level 1 is not strictly standard; however, most implementations of IS-IS support it.

Default

export-defaults level 2; (unless gated is configured as level 1 only)

export-defaults metric inherit;

export-defaults metric-type internal;

Context

isis statement

Examples

Example 1

```
 isis on {
     export-defaults level 2;
     export-defaults metric 1;
     export-defaults metric-type external;
     interface ex0 cost 1;
 };
```

Example 2

The following example will cause all static routes to be exported into IS-IS at level 1.

```
isis on {
    export-defaults level 1;
};

export proto isis {
    proto static {
        all;
    };
};
```

Example 3

The following example sets the default metric for routes exported into IS-IS to 10.

```
isis on {
    export-defaults metric 10;
};
```

Example 4

The following example sets the default metric type for routes exported into IS-IS to **external**.

```
isis on {
```

```
    export-defaults metric-type external;
};
```

See Also

"Chapter 13 Intermediate System to Intermediate System (IS-IS)" on page 55 of *Configuring GateD*

`export` on page 623

`extended-metrics` on page 174

`isis` statement on page 185

"Chapter 32 Route Exportation" on page 145 of *Configuring GateD*

`rfc1195-metrics` on page 205

extended-metrics

Name

extended-metrics - configures whether the router should originate extended (wide) metrics in the LSPs

Syntax

extended-metrics (on | off) ;

Parameters

on - enables the transmission and processing of extended metrics

off - disables the transmission and processing of extended metrics

Description

The **extended-metrics** statement configures whether or not the router shall originate and process (used in the SPF) the “Extended IP Reachability TLV” and the “Extended IS Reachability TLV.” These new TLVs (Type Length Value) allow for metrics larger than 63. Enabling extended-metrics causes GateD to originate these options in our Link State Packets (LSP) and process them in the Shortest Path First (SPF) calculation.

The setting of this option is independent of the setting of the **rfc1195-metrics** option. If both types of metrics are enabled, the lowest cost path is used in the SPF calculation.

Default

extended-metrics off ;

Context

isis statement

Examples

The following example sets GateD to originate and process extended metrics.

```
isis on {
    extended-metrics on ;
};
```

See Also

“Chapter 13 Intermediate System to Intermediate System (IS-IS)” on page 55 of *Configuring GateD*

isis on page 185

rfc1195-metrics on page 205

external preference

Name

external preference - configures the router preference for IS-IS external routes

Syntax

external preference *preferencenum* **;**

Parameters

preferencenum - an integer between 0 and 255

Description

The **external preference** statement sets the preference to assign to any route received in IS-IS that is marked as **external**. A route is marked as **external** when it is exported into IS-IS from another protocol via **rfc1195-metrics** with external reachability.

Each protocol on the router assigns a preference to the routes it makes available to the router. When more than one route to a specific destination exists on the router (for example, from **isis** and **bgp**) the route with the lower preference value is used by the router.

For more information on preference, see **preference** on page 199.

Default

external preference 151 ;

Context

isis statement

Examples

The following example sets the **external preference** to 140.

```
isis on {
    external preference 140;
};
```

See Also

"Chapter 3 Preferences and Route Selection" on page 11 of *Configuring GateD*

"Chapter 13 Intermediate System to Intermediate System (IS-IS)" on page 55 of *Configuring GateD*

"Chapter 32 Route Exportation" on page 145 of *Configuring GateD*

preference on page 199

rfc1195-metrics on page 205

hello-interval

Name

hello-interval - configures the hello interval for a non-DIS interface

Syntax

hello-interval *intervalnum* [**level (1 | 2 | 1 and 2)**] **;**

Parameters

intervalnum - a number of seconds between 1 and 100

Description

The **hello-interval** statement configures the rate at which hello packets are sent out an interface. IS-IS uses hello packets to learn about other routers on the link. Every *intervalnum* seconds, the router will multicast a hello to the link, announcing both itself and all the other routers it knows on the link. When the router receives a hello, it adds the sending router to its known list of routers on the link. Through this exchange, both routers learn of each other and acknowledge this fact. Until this exchange and acknowledgement happens, packets will not be forwarded between the routers.

Each hello packet also contains a hold time. The hold time informs the receiving router how long to wait without seeing another hello from the sending router before considering the sending router down. The announced hold time is derived from the router's **hello-interval** and the **hello-multiplier** as follows:

hold time = **hello-interval** * **hello-multiplier**

Setting the **hello-interval** smaller will decrease the amount of time before a router's presence (or lack thereof) is noticed, but will increase the link usage. Setting the **hello-interval** larger will decrease the link usage, but will increase the amount of time required to notice the router is up or down.

Through the use of the **level** keyword, you can configure separate intervals for each level.

Default

hello-interval 10 level 1 and 2 ;

Context

isis interface statement

Examples

Example 1

The following example sets the hello interval on interface eth0 to 5 seconds.

```
isis on {
    interface eth0 {
        hello-interval 5;
```

```
    }
};
```

Example 2

This example sets per-level hello intervals for interface eth0. For level 1, the interval is set to 5; for level 2, the interval is set to 8.

```
isis on {
    interface eth0 {
        hello-interval 5 level 1;
        hello-interval 8 level 2;
    }
};
```

See Also

"Chapter 13 Intermediate System to Intermediate System (IS-IS)" on page 55 of *Configuring GateD*

dis-hello-interval on page 164

hello-multiplier on page 178

interface on page 184

isis on page 185

hello-multiplier

Name

hello-multiplier - configures the hello multiplier for an interface

Syntax

hello-multiplier *multipliernum* [**level (1 | 2 | 1 and 2)**] **;**

Parameters

multipliernum - a number between 1 and 100

Description

hello-multiplier sets the hello multiplier for an interface. The hello multiplier is used to calculate the hold time announced in hello packets as follows:

hold time = **hello-interval** * **hello-multiplier**

Each hello packet contains a hold time. The hold time informs the receiving routers how long to wait without seeing another hello from the sending router before considering the sending router down.

The hello multiplier can also be considered the number of hello packets allowed to be missing before a router will be considered down. For example, if the **hello-multiplier** is 3, then other routers must miss three consecutive hello packets before they consider this router down. Setting the **hello-multiplier** to a smaller number will decrease the amount of time before other routers notice that this router is down, but will increase the chances that a router is falsely considered down (e.g., if packets are being dropped).

Setting the hello interval to a larger number will increase the amount of time before other routers notice that this router is down, but will decrease chances that the router is falsely considered down (for example, if packets are being dropped).

Through the use of the **level** keyword, you can configure separate multipliers for each level.

Default

hello-multiplier 3 level 1 and 2 ;

Context

isis interface statement

Examples

Example 1

The following example sets the hello multiplier on interface eth0 to 5.

```
isis on {
    interface eth0 {
```

```
        hello-multiplier 5;
    }
};
```

Example 2

This example sets per-level hello multipliers for interface eth0. For level 1, the interval is set to 4; and for level 2 the interval is set to 5.

```
isis on {
    interface eth0 {
        hello-multiplier 4 level 1;
        hello-multiplier 5 level 2;
    }
};
```

See Also

"Chapter 13 Intermediate System to Intermediate System (IS-IS)" on page 55 of *Configuring GateD*

dis-hello-interval on page 164

hello-interval on page 176

interface on page 184

isis on page 185

hold-time

Name

hold-time - used to set the ISO ES hold time in the operating system kernel

Syntax

hold-time *seconds***;**

Parameters

seconds - the value of the ISO ES hold time

Description

This value is used to set the ISO ES hold time in the operating system kernel. This requires ISO socket support in the operating system.

Default

hold-time 120;

Context

isis statement

Examples

```
isis on {
    hold-time 120;
    interface ex0 cost 1;
};
```

See Also

"Chapter 13 Intermediate System to Intermediate System (IS-IS)" on page 55 of *Configuring GateD*

isis on page 185

hostname

Name

hostname - configures the RFC 2763 hostname

Syntax

hostname "*name*" ;

Parameters

name - the hostname for this IS

Description

hostname sets the value of the "Dynamic Hostname" option described in RFC 2763. This name will be used in both level 1 and level 2 Link State Packets if configured. The maximum length is 255 bytes.

Default

none

Context

isis statement

Examples

The following example configures the host name to be "foorouter".

```
isis on {
    hostname "foorouter";
};
```

See Also

"Chapter 13 Intermediate System to Intermediate System (IS-IS)" on page 55 of *Configuring GateD*

isis statement on page 185

inet

Name

inet - configures IS-IS to route IPv4

Syntax

```
inet (on | off) ;
```

Parameters

none

Description

inet is used to specify whether or not IS-IS can exchange IPv4 reachability. **on** configures it to use IPv4; **off** configures it to not use IPv4.

Default

```
inet on ;
```

Context

isis statement

Examples

The following example configures IS-IS to route IPv4:

```
isis on {
    inet on ;
};
```

See Also

"Chapter 13 Intermediate System to Intermediate System (IS-IS)" on page 55 of *Configuring GateD*

isis on page 185

inet6

Name

inet6 - configures IS-IS to route IPv6

Syntax

inet6 (on | off) ;

Parameters

none

Description

inet6 is used to specify whether or not IS-IS can exchange IPv6 reachability. **on** configures it to use IPv6; **off** configures it to not use IPv6.

Note: IPv6 support must be present in GateD and in the underlying OS for this option to parse.

Default

inet6 off ;

Context

isis statement

Examples

The following example configures IS-IS to route IPv6:

```
isis on {
    inet6 on ;
};
```

See Also

"Chapter 13 Intermediate System to Intermediate System (IS-IS)" on page 55 of *Configuring GateD*

isis on page 185

interface

Name

interface - configures IS-IS to run on the given interface

Syntax

interface *interface_name* **[{** *isis-interface-statements* **}] ;**

Parameters

interface_name - an interface name

isis-interface-statements - IS-IS interface-specific configuration statements

Description

interface is used to configure **isis** on an interface. Interface-specific configuration is accomplished through the use of the substatements: **auth**, **csn-interval**, **disable**, **dis-hello-interval**, **enable**, **encap**, **hello-interval**, **hello-multiplier**, **lsp-interval**, **level** (interface), **max-burst**, **mesh-group**, **metric**, **passive**, **periodic-csn**, **priority**, **retransmit-interval**.

If the **interface** *ifname* is enabled, it will be announced as reachable in the IS-IS routing domain.

Default

interface all;

Context

isis statement

Examples

The following example configures IS-IS to run on eth0 and ser0.

```
isis on {
    interface eth0;
    interface ser0;
};
```

See Also

"Chapter 13 Intermediate System to Intermediate System (IS-IS)" on page 55 of *Configuring GateD*

disable on page 163

enable on page 169

isis statement on page 185

isis

Name

isis - configures IS-IS

Syntax

isis on [{ [*isis-statements* **] }] ;**

isis off [{ [*isis-statements* **] }] ;**

Parameters

isis-statements - **isis**-specific configuration statements, which are described throughout this chapter

Description

The **isis** statement is used to enable (**on**) or disable (**off**) IS-IS on the router. IS-IS-specific options can be configured through specific **isis** statements, which are described throughout this chapter.

Default

isis off;

Context

global

Examples

Example 1

The following configures IS-IS to run on all broadcast and IS-IS capable p2p interfaces using the default configuration.

```
isis on {
    interface all;
};
```

Example 2

In this example, IS-IS is enabled and configured to run on eth0 and fxp1.

```
isis on {
    interface eth0;
    interface fxp1;
};
```

See Also

"Chapter 13 Intermediate System to Intermediate System (IS-IS)" on page 55 of *Configuring GateD*

level

Name

`level` - configures the level(s) of the router or interface

Syntax

`level ( 1 | 2 | 1 and 2 ) ;`

Parameters

none

Description

The global level statement sets the levels at which this router is willing to operate. A level 1 router will communicate only with other level 1 routers in its area. A level 2 router will communicate with any other level 2 router, regardless of which area the other router is in. A level 1 and 2 router will communicate with other level 1 routers in its own area, and with any level 2 router, regardless of which area the other router is in. The **`interface level`** statement sets the levels at which this interface is willing to operate. The **`interface level`** must be less than or equal to the **`router level`**.

Level 2 routers form the layer above the level 1 areas that interconnect those areas. Level 2 must be fully connected. That is, you must not have two or more disconnected sets of level 2 routers.

When traffic in an area must reach another area, it is forwarded to a level 2 router. (How this occurs depends on the **`domain-wide`**, **`summary-originate`**, and **`summary-filter`** statements.) By default, a level 1 router will forward all traffic not destined for its own area to the nearest level 2 router.

If the level 2 router does not know how to reach the area for which traffic is destined, the traffic will be dropped. This is why all level 2 routers must be connected.

Default

`level 1 and 2;`

Context

`isis` statement

`isis interface` statement

Examples

Example 1

The following example configures the router to partake only in level 1 routing.

```
isis on {
    level 1;
};
```

Example 2

The following example configures the eth0 interface to partake only in level 1 routing.

```
isis on {
    interface eth0 {
    level 1;
    };
};
```

See Also

"Chapter 13 Intermediate System to Intermediate System (IS-IS)" on page 55 of *Configuring GateD*

interface statement on page 184

isis statement on page 185

lsp-interval

Name

lsp-interval - time between successive transmission of a Link State Packet (LSP)

Syntax

lsp-interval *msintervalnum* ;

Parameters

msintervalnum - a number in milliseconds from 1 to 3000

Description

The **lsp-interval** statement configures the average time in milliseconds to wait after sending an LSP or Complete Sequence Number (CSN) packet on a broadcast link before sending another LSP or CSN packet on the same link. The actual time between transmissions is determined by the combination of the **lsp-interval** statement and the **max-burst** statement. The router is allowed to burst up to **max-burst** consecutive packets back-to-back on the link, provided that over any 1 second interval, the average time between transmissions was *msintervalnum*.

Default

```
lsp-interval 33;
```

Context

isis interface statement

Examples

The following example sets the lsp interval to 300ms.

```
isis on {
    interface ser0 {
        lsp-interval 300;
    };
};
```

See Also

"Chapter 13 Intermediate System to Intermediate System (IS-IS)" on page 55 of *Configuring GateD*

interface statement on page 184

isis statement on page 185

max-burst statement on page 190

max-burst

Name

max-burst - configures the maximum number of link state packets (LSPs) to burst

Syntax

max-burst *burstnum* ;

Parameters

burstnum - a number of LSPs from 1 to 10

Description

The **max-burst** count configures the number of consecutive back-to-back transmissions of LSPs or complete sequence number (CSN) packets on a given link. The router must on average not transmit more than one LSP or CSN packet per **lsp-interval**. CPU usage can be lowered by allowing the router to actually transmit the packets in bursts. Setting *burstnum* higher will allow the router more CPU time for other actions, but may cause packets to be dropped (e.g., if other routers cannot handle the back-to-back traffic).

Default

max-burst 5;

Context

isis interface statement

Examples

Example 1

The following example configures the maximum burst of 10 on interface eth0.

```
isis on {
    interface eth0 {
       max-burst 10;
    };
};
```

Example 2

This example effectively disables bursting by restricting it to 1 on interface eth0.

```
isis on {
    interface eth0 {
        max-burst 1;
    };
};
```

See Also

"Chapter 13 Intermediate System to Intermediate System (IS-IS)" on page 55 of *Configuring GateD*

`interface` statement on page 184

`isis` statement on page 185

`lsp-interval` statement on page 189

mesh-blocked

Name

mesh-blocked - prevents the flooding of Link State Packets (LSPs) on an interface

Syntax

mesh-blocked ;

Parameters

none

Description

The **mesh-blocked** statement prevents LSPs from being flooded out on an interface at certain times. It is most useful for pruning flooding topologies in a full mesh network topology.

Default

none

Context

isis interface statement

Examples

The following example prevents LSPs from being flooded out the eth0 interface.

```
isis on {
    interface eth0 {
        mesh-blocked;
    };
};
```

See Also

“Chapter 13 Intermediate System to Intermediate System (IS-IS)” on page 55 of *Configuring GateD*

interface statement on page 184

isis statement on page 185

mesh-group on page 193

mesh-group

Name

mesh-group - configures the mesh group number for an interface

Syntax

mesh-group *meshnum* **;**

Parameters

meshnum - the mesh group number for this interface

Description

mesh-group implements an extension to the IS-IS protocol described in RFC 2973 which allows an IS to track a "mesh group" number that prevents redundant flooding when a full mesh of circuits are used among a group of ISs. Use of this option may be combined with "**mesh-blocked**" on other circuits to gain optimal flooding topologies. This may be any 32-bit value including 0.

Default

none

Context

isis interface statement

Examples

The following example sets the mesh group of interface eth0 to be 1.

```
isis on {
    interface eth0 {
        mesh-group 1;
    };
};
```

See Also

"Chapter 13 Intermediate System to Intermediate System (IS-IS)" on page 55 of *Configuring GateD*

interface statement on page 184

isis statement on page 185

mesh-blocked statement on page 192

metric

Name

metric - sets the outbound cost of the IS-IS interface at the given level

Syntax

metric *metricnum* **[level (1 | 2 | 1 and 2)] ;**

Parameters

metricnum - a number from 1 to *max_metric. max_metric* is 63 if **rfc1195-metrics** is on; otherwise it is 2^{24}-1 (or 4,261,412,864).

Description

The **metric** statement sets the interface outbound metric to *metricnum*. This value is advertised throughout the IS-IS routing domain as the cost to send a packet out the interface, and is used to calculate the shortest path between routers.

Each interface is capable of having up to two unique metric values, one for **level 1** and one for **level 2**. If a level is not specified, it defaults to setting both the **level 1 and level 2** metric to the indicated value *metricnum*.

Default

metric 10 level 1 and 2;

Context

isis interface statement

Examples

Example 1

The following will set interface eth0's metric to 4.

```
isis on {
    interface eth0 {
        metric 4;
    };
};
```

Example 2

This example sets the **level 1** metric of eth0 to 1 and the **level 2** metric to 5.

```
isis on {
    interface eth0 {
        metric 1 level 1;
```

```
        metric 5 level 2;
    };
};
```

See Also

"Chapter 13 Intermediate System to Intermediate System (IS-IS)" on page 55 of *Configuring GateD*

extended-metrics on page 174

interface statement on page 184

isis statement on page 185

rfc1195-metrics on page 205

overload-bit

Name

overload-bit - administratively set the overload bit

Syntax

overload-bit (on | off) ;

Parameters

none

Description

The **overload-bit** statement configures whether the router announces itself as overloaded to the IS-IS routing domain. If the **overload-bit** is **on**, then no traffic will be directed at the router for forwarding; however, traffic originated by (or received at) the router will still be forwarded properly. This can be useful in testing configurations or monitoring IS-IS operation without actually affecting it. It can also be used to temporarily disable the router for diagnostics.

Default

overload-bit off;

Context

isis statement

Examples

The following example configures the router to announce itself as being overloaded.

```
isis on {
    overload-bit on;
};
```

See Also

"Chapter 13 Intermediate System to Intermediate System (IS-IS)" on page 55 of *Configuring GateD*

isis statement on page 185

passive

Name

passive - configures an interface to be passive

Syntax

```
passive ( on | off ) ;
```

Parameters

none

Description

The **passive** statement configures whether the interface actively participates in IS-IS on the given link. If **passive** is set to **on**, the link will be advertised in the router's link state packet (LSP) and thus enable routing to that link through the router; however, the router will not actually run IS-IS on the link. This has the effect of allowing traffic destined to the link's networks to use this router but not allowing traffic destined for other networks to use this router and link to reach those networks.

Default

```
passive off;
```

Context

isis interface statement

Examples

The following example configures interface eth0 to be **passive**.

```
isis on {
    interface eth0 {
        passive on;
    };
};
```

See Also

"Chapter 13 Intermediate System to Intermediate System (IS-IS)" on page 55 of *Configuring GateD*

interface statement on page 184

isis statement on page 185

periodic-csn

Name

periodic-csn - configures periodic Complete Sequence Number (CSN) flooding on point-to-point interfaces

Syntax

```
periodic-csn ( on | off ) ;
```

Parameters

none

Description

In typical configurations, it is not necessary or desirable to send CSNs at regular intervals on point-to-point links. These packets are normally transmitted only upon initial establishment of the adjacency. This option forces CSN packets to be sent on an interface, subject to the normal CSN control parameters such as **csn-interval**, **max-burst**, and **lsp-interval**. This option makes sense only on point-to-point links. To block CSN packet flooding on LAN interfaces, see the **passive** option.

Default

```
periodic-csn off;
```

Context

isis interface statement

Examples

The following example forces periodic CSN flooding on interface gif0.

```
isis on {
    interface gif0 {
        periodic-csn on;
    };
};
```

See Also

"Chapter 13 Intermediate System to Intermediate System (IS-IS)" on page 55 of *Configuring GateD*

interface statement on page 184

isis statement on page 185

mesh-blocked statement on page 192

mesh-group statement on page 193

passive statement on page 197

preference

Name

preference - configures the router preference for IS-IS internal routes

Syntax

preference *preferencenum* **;**

Parameters

preferencenum - a number from 0 to 255

Description

The **preference** statement sets the **preference** to assign to any route received in IS-IS that is marked as internal. All networks in the IS-IS routing domain are marked as internal. If **rfc1195-metrics** is on, then routes exported into IS-IS from other protocols are marked as external and will not be affected by this statement, even those marked by an internal metric type.

Each protocol on the router assigns a **preference** to the routes it makes available to the router. When more than one route to a specific destination exists on the router (for example, from **isis** and **rip**) the route with the lower **preference** value is used by the router.

Default

preference 11;

Context

isis statement

Examples

The following example sets the **preference** assigned to internal IS-IS routes to 5.

```
isis on {
    preference 5;
};
```

See Also

"Chapter 13 Intermediate System to Intermediate System (IS-IS)" on page 55 of *Configuring GateD*

export on page 623

external preference on page 175

priority

Name

priority - configures the priority of an interface for Designated Intermediate System (DIS) election

Syntax

priority *prioritynum* **[level (1 | 2 | 1 and 2)] ;**

Parameters

prioritynum - a number from 0 to 127

Description

The **priority** statement configures the priority of this router to become the DIS for the given interface's link. The router with the highest **priority** on a link will be elected DIS. If more than two routers have the same highest priority, the interface's MAC address breaks the tie (highest winning).

The DIS for a link is ultimately responsible for making sure all other routers on the same link have the same link state database. It is therefore important that the router elected DIS be stable. The DIS will be required to process more IS-IS routing packets, and should thus be capable of doing so.

Through the use of the **level** keyword, one can configure separate priorities for each level.

Default

priority 64 level 1 and 2;

Context

isis interface statement

Examples

The following example configures interface eth0's priority to be 0, thus making it as unlikely as possible to be elected DIS.

```
isis on {
    interface eth0 {
        priority 0;
    };
};
```

See Also

"Chapter 13 Intermediate System to Intermediate System (IS-IS)" on page 55 of *Configuring GateD*

csn-interval statement on page 161

dis-hello-interval statement on page 164

isis statement on page 185

psn-interval

Name

psn-interval - configures the rate of Partial Sequence Number (PSN) transmission

Syntax

psn-interval *intervalnum* **;**

Parameters

intervalnum - a number of seconds from 1 to 20

Description

The **psn-interval** statement sets the rate at which PSN packets are multicast from the IS on broadcast interfaces.

On point-to-point links, PSN packets are used to acknowledge receipt of Link State Packets (LSPs.) If an LSP is received, but a PSN acknowledgement is not sent, the LSP will be retransmitted by the sending router. We recommend that the **psn-interval** be set smaller than **retransmit-interval**.

On broadcast links, non-DIS circuits use PSN packets to request newer LSPs that have been advertised by the DIS in CSN packets. See **csn-interval** for more description of CSN packets.

Default

psn-interval 2;

Context

isis statement

Examples

The following example configures the **psn-interval** to be 1 second.

```
isis on {
    psn-interval 1;
};
```

See Also

"Chapter 13 Intermediate System to Intermediate System (IS-IS)" on page 55 of *Configuring GateD*

csn-interval on page 161

isis statement on page 185

retransmit-interval on page 204

require-snp-auth

Name

require-snp-auth - determines whether SNP authentication is required

Syntax

require-snp-auth (on | off) ;

Parameters

none

Description

require-snp-auth sets whether Complete Sequence Number (CSN) and Partial Sequence Number (PSN) packets should be required to contain authentication.

The IS-IS standard requires that all IS-IS packets contain authentication (if the router is configured with authentication); however, some routers do not conform to this standard regarding CSN and PSN packets.

The default is to not require authentication of CSN and PSN packets, so that the router will interoperate with these other vendors.

Default

require-snp-auth off;

Context

isis statement

Examples

The following example enables requiring authentication of CSN and PSN packets.

```
isis on {
    require-snp-auth on;
};
```

See Also

area-auth on page 155

“Chapter 13 Intermediate System to Intermediate System (IS-IS)” on page 55 of *Configuring GateD*

domain-auth on page 166

isis statement on page 185

retransmit-interval

Name

retransmit-interval - time before retransmitting an LSP on a point-to-point interface

Syntax

retransmit-interval *intervalnum* **;**

Parameters

intervalnum - a number of seconds from 1 to 100

Description

The **retransmit-interval** sets the amount of time to wait between successive transmissions of the same LSP on a point-to-point link. Setting this value higher will avoid unnecessary retransmission, but will slow convergence (synchronization) time between routers. Setting this value lower will cause convergence time to decrease, but may cause unnecessary retransmission (e.g., the other router has received the LSP initially but not had a chance to acknowledge the fact). This number should be longer than the **psn-interval**.

Default

retransmit-interval 5;

Context

isis interface statement

Examples

The following example configures the retransmit interval to be 10 seconds.

```
isis on {
    interface ser0 {
        retransmit-interval 10;
    };
};
```

See Also

"Chapter 13 Intermediate System to Intermediate System (IS-IS)" on page 55 of *Configuring GateD*

interface statement on page 184

isis statement on page 185

psn-interval on page 202

rfc1195-metrics

Name

`rfc1195-metrics` - configure whether the router should originate metrics in the normal IS Neighbors Type Length Value (TLV)

Syntax

`rfc1195-metrics ( on | off )` ;

Parameters

none

Description

There are two IS-IS options that may be used to originate reachability to neighbors. The option specified in RFC 1195 allows a metric no larger than 63. This parameter indicates whether the router originates metrics using this type of option.

Default

`rfc1195-metrics on` ;

Context

`isis` statement

Examples

```
isis on {
     rfc1195-metrics on;
     interface all {
         enable;
     };
 };
```

See Also

"Chapter 13 Intermediate System to Intermediate System (IS-IS)" on page 55 of *Configuring GateD*

`extended-metrics` statement on page 174

`isis` statement on page 185

ribs

Name

ribs - configures the Routing Information Bases (RIBs) into which IS-IS installs routes

Syntax

```
ribs ( unicast | unicast  multicast ) ;
```

Parameters

none

Description

ribs configures the RIBs into which IS-IS should install its. If the router has multicast capability, IS-IS can be configured to install routes in the multicast RIB as well as the unicast RIB. IS-IS cannot be configured to install routes into only the multicast RIB. The multicast RIB is used by protocols such as **pim**.

Default

```
ribs unicast;
```

Context

isis statement

Examples

The following example configures the router to install IS-IS routes in both the unicast and multicast RIBs.

```
isis on {
    ribs unicast multicast;
};
```

See Also

"Chapter 13 Intermediate System to Intermediate System (IS-IS)" on page 55 of *Configuring GateD*

"Chapter 25 Multicast Statement" on page 123 of the *Configuring GateD Guide*

isis statement on page 185

multicast statement on page 545

spf-interval

Name

spf-interval - configures the wait time before calculating the routing table

Syntax

spf-interval *intervalnum* **;**

Parameters

intervalnum - a number of seconds from 1 to 60

Description

spf-interval sets the amount of time to wait in seconds before running the routing table calculation after it first determines that it needs to do so.

When new Link State Packets (LSPs) are received or originated by the router, the routing table needs to be recalculated. It would be inefficient to run the routing table calculation every time a new LSP was received or originated, because many LSPs may be being received or originated quickly. The **spf-interval** guarantees that the calculation will not occur more often than *intervalnum* seconds.

When a new LSP is received or originated, a timer is set to expire in *intervalnum* seconds. When it expires, the routing table is recalculated. During this wait time, any new LSPs received or originated will also be processed in the forthcoming calculation.

Setting this value higher will decrease the amount of CPU time the router spends calculating the routing table but will slow down the convergence time. Setting this value lower will increase convergence time at the cost of CPU usage.

Default

spf-interval 2;

Context

isis statement

Examples

The following example configures the spf interval to be 5 seconds.

```
isis on {
    spf-interval 5;
};
```

See Also

"Chapter 13 Intermediate System to Intermediate System (IS-IS)" on page 55 of *Configuring GateD*

isis statement on page 185

start-accept

Name

start-accept - configures the time at which an MD5 key shall start to be accepted by this router

Syntax

start-accept *YYYY/MM/DD HH*:*MM* **[.***ss***] ;**

Parameters

YYYY/MM/DD HH:*MM* **[.***ss***]** *;* - a date/time specification

Description

This command configures the time at which an MD5 key shall start to be accepted by this router.

Default

The default is to always accept the key.

Context

isis domain auth md5 statement

isis area auth md5 statement

isis interface auth md5 statement

Examples

```
isis on {
    area auth {
        md5 key "foo" {
            start-accept 2001/12/1 10:15.22;
        };
    };
};
```

See Also

"Chapter 13 Intermediate System to Intermediate System (IS-IS)" on page 55 of *Configuring GateD*

isis statement on page 185

auth on page 157

area auth on page 160

domain auth on page 166

start-generate

Name

start-generate - configures the time at which an MD5 key shall start to be generated by this router

Syntax

start-generate *YYYY/MM/DD HH*:*MM* **[.***ss***] ;**

Parameters

YYYY/MM/DD HH:*MM* **[.***ss***] ;** - a date/time specification

Description

This command configures the time at which an MD5 key shall start to be generated by this router. Note that the expiration of a start-generate time does not cause automatic regeneration of the Link State Packets (LSPs); the new authentication will be originated when the LSPs are originated for some other reason. At longest, this is one refresh interval.

Default

The default is to always generate the key. When more than one key is configured for a time range, configuration file order breaks ties.

Context

isis domain auth md5 statement

isis area auth md5 statement

isis interface auth md5 statement

Examples

```
isis on {
    area auth {
        md5 key "foo" {
            start-generate 2001/12/1 10:15.22;
        };
    };
};
```

See Also

"Chapter 13 Intermediate System to Intermediate System (IS-IS)" on page 55 of *Configuring GateD*

isis statement on page 185

auth on page 157

area auth on page 160

domain auth on page 166

start-accept on page 208

stop-accept on page 212

stop-generate on page 214

stop-accept

Name

stop-accept - configures the time at which an MD5 key shall stop being accepted by this router

Syntax

stop-accept *YYYY/MM/DD HH*:*MM* **[.***ss***] ;**

Parameters

YYYY/MM/DD HH:*MM* **[.***ss***] ;** - a date/time specification

Description

This command configures the time at which an MD5 key shall stop being accepted by this router.

Default

The default is to always accept the key.

Context

isis domain auth md5 statement

isis area auth md5 statement

isis interface auth md5 statement

Examples

```
isis on {
    area auth {
        md5 key "foo" {
            stop-accept 2001/12/1 10:15.22;
        };
    };
};
```

See Also

"Chapter 13 Intermediate System to Intermediate System (IS-IS)" on page 55 of *Configuring GateD*

isis statement on page 185

auth on page 157

area auth on page 160

domain auth on page 166

start-accept on page 208

start-generate on page 210

stop-generate on page 214

stop-generate

Name

stop-generate - configures the time at which an MD5 key shall cease to be generated by this router

Syntax

stop-generate *YYYY/MM/DD HH*:*MM* **[.***ss***] ;**

Parameters

YYYY/MM/DD HH:*MM* **[.***ss***] ;** - a date/time specification

Description

This command configures the time at which an MD5 key shall cease to be generated by this router.

Default

The default is to always generate the key.

Context

isis domain auth md5 statement

isis area auth md5 statement

isis interface auth md5 statement

Examples

```
isis on {
    area auth {
        md5 key "foo" {
            stop-generate 2001/12/1 10:15.22;
        };
    };
};
```

See Also

"Chapter 13 Intermediate System to Intermediate System (IS-IS)" on page 55 of *Configuring GateD*

isis statement on page 185

auth on page 157

area auth on page 160

domain auth on page 166

`start-accept` on page 208

`start-generate` on page 210

`stop-accept` on page 212

summary-filter

Name

summary-filter - configures filtering of downward summarization

Syntax

```
[ summary-filter [ inet ] {
    [ ipv4-network mask ipv4-netmask ; ]
    [ ipv4-network masklen ipv4-masklen ; ]
} ; ]
[ summary-filter inet6  {
    [ ipv6-network mask ipv6-netmask ; ]
    [ ipv6-network masklen ipv6-masklen ; ]
} ; ]
```

Parameters

inet - specifies that the **summary-filter** statement applies to IPv4

inet6 - specifies that the **summary-filter** statement applies to IPv6

ipv4-network - an IPv4 address in dotted-quad format

ipv4-netmask - an IPv4 **mask** in dotted-quad format

ipv4-masklen - a number from 0 to 32

ipv6-network - an IPv6 address in colon-separated format

ipv6-netmask - an IPv6 **mask** in colon-separated format

ipv6-masklen - a number from 0 to 128

Description

summary-filter configures filters to block advertisement of networks from level 2 into level 1. It is used only when **domain-wide** is configured to be **on**.

If a level 1 and 2 router is configured with **domain-wide on**, each level 2 route is by default advertised down into level 1. This advertisement may be restricted by using **summary-filter**. If a level 2 route that would normally be announced down into level 1 matches an address and mask pair in the **summary-filter** statement, that announcement is blocked.

Default

no filtering

Context

isis statement

Examples

Example 1

The following example blocks announcement of any level 2 route to 32.0.0.0/8 or more specific (e.g., 32.1.0.0/16) into level 1.

```
isis on {
    domain-wide on;
    summary-filter {
        32.0.0.0 masklen 8;
    };
};

isis on {
    inet6 on;
    summary-filter inet6 {
        fec0:0:0:031f::e masklen 16;
    };
};
```

Example 2

This example blocks all announcements of level 2 routes into level 1.

```
isis on {
    domain-wide on;
    summary-filter {
        0.0.0.0 masklen 0;
    };
};
isis on {
    inet6 on;
    summary-filter inet6 {
        :: masklen 0;
    };
};
```

See Also

"Chapter 13 Intermediate System to Intermediate System (IS-IS)" on page 55 of *Configuring GateD*

domain-wide statement on page 168

isis statement on page 185

summary-originate on page 219

summary-originate

Name

summary-originate - configures summarization for **level 1 and 2** routers

Syntax

```
[ summary-originate [ inet ] {
    [ ipv4-network mask ipv4-netmask metric cost-value ; ]
    [ ipv4-network masklen ipv4-masklen metric cost-value ; ]
} ; ]
[ summary-originate inet6 {
    [ ipv6-network mask ipv6-netmask metric cost-value ; ]
    [ ipv6-network masklen ipv6-masklen metric cost-value ; ]
} ; ]
```

Parameters

inet - specifies that the **summary-originate** statement applies to IPv4

inet6 - specifies that the **summary-originate** statement applies to IPv6

ipv4-network - an IPv4 address in dotted-quad format

ipv4-netmask - an IPv4 mask in dotted-quad format

ipv4-masklen - a number from 0 to 32

cost-value - a number from 1 to *max_metric*

ipv6-network - an IPv6 address in colon-separated format

ipv6-netmask - an IPv6 mask in colon-separated format

ipv6-masklen - a number from 0 to 128

Description

summary-originate controls which reachable IP networks are advertised from level 1 to level 2 by a level 1 and 2 router.

When a level 1 and 2 router has a route in its level 1 area, it by default announces that route into level 2. To reduce the size of the level 2 database (i.e., the number of routes advertised into level 2), a router can configure aggregating **summary-originate** statements.

If a level 1 route exists that matches an address and mask pair in the **summary-originate** statement, the following actions occur:

The level 1 route is not advertised in level 2. If the matching address and mask pair has a configured metric, the address and mask pair is announced into level 2 with that metric; otherwise, the matching address and mask pair are marked **restrict** and no advertisement into level 2 will occur.

Use of the **restrict** keyword can, for example, allow one to re-use private addresses within level 1 areas.

max_metric is 63 if **rfc1195-metrics** is on; otherwise it is 2^{24}-1 (or 4,261,412,864).

Default

no summarization

Context

isis statement

Examples

Example 1

```
isis on {
    summary-originate {
        32.0.0.0 masklen 8 metric 1;
    };
};
```

Example 2

```
isis on {
    summary-originate {
        32.0.0.0 masklen 8 metric 1;
        10.0.0.0 masklen 0 restrict;
    };
};
```

Example 3

```
isis on {
    inet6 on;
    summary-originate inet6 {
        fec0:0:0:031f::e masklen 64 metric 1;
    };
};
```

Example 4

```
isis on {
    inet6 on;
    summary-originate inet6 {
        3ffd:0:0:abcd:F800:: masklen 69 metric 1;
        fec0:0:0:031f::e masklen 16 restrict;
```

```
        };
    };
```

See Also

"Chapter 13 Intermediate System to Intermediate System (IS-IS)" on page 55 of *Configuring GateD*

isis statement on page 185

summary-filter on page 216

systemid

Name

systemid - configures the router's system ID

Syntax

systemid *D.D.D.D* ;

systemid *HHHH.HHHH.HHHH* ;

Parameters

D - a decimal number from 0 to 255

H - a hexidecimal digit from 0 to F

Description

systemid sets the system ID of the router. In OSI, each router (IS) has a network entity title (NET) assigned to it. The NET uniquely identifies the IS throughout the routing domain and determines which area the router is in. A NET is divided into the area portion, then the system ID, and is completed by a single 0 octet (byte). The system ID is always (by convention) 6 octets and the area portion is variable and can be from 1 to 13 octets. Some routers require the user to specify the area and system ID by setting the NET directly.

This mechanism is largely outdated because most users do not actually run the OSI routing domain, and it has thus been simplified to be more like other current routing protocols such as **ospf**. The **systemid** *D.D.D.D* form of the command is supplied to be interchangeable with **router-id**. It takes as its argument a dotted-quad (IPv4-like) address.

This form will set the **systemid** to be two octets of 0 followed by four octets as given by the four decimal numbers in the dotted-quad argument. The more general form of the command utilizes hexadecimal digits and is 6 octets in length. Each pair of hexadecimal digits is a single octet, and thus 12 hex-digits must be given. If a **router-id** is being specified for the router, and this is acceptable as the system ID also, you do not need to specify the **systemid**.

Default

defaults to the configured **router-id** with 2 octets of 0 prepended

Context

isis statement

Examples

Example 1

The following example sets the **systemid** to (in hex format) "0000.0000.0001", or 5 octets of zero followed by 1 octet of 1.

```
isis on {
```

```
    systemid 0.0.0.1;
};
```

Example 2

This example sets the **systemid** to (in hex format) "0000.0000.001F" or 5 octets of zero followed by 1 octet of decimal 31.

```
isis on {
    systemid 0000.0000.001F;
};
```

See Also

area statement on page 153

“Chapter 13 Intermediate System to Intermediate System (IS-IS)” on page 55 of *Configuring GateD*

isis statement on page 185

traceoptions

Name

traceoptions - configures traceoptions specifically for IS-IS

Syntax

traceoptions *std-traceoptions* **[** *isis-traceoptions* **]** ;

traceoptions [*std-traceoptions* **]** *isis-traceoptions* ;

Parameters

std-traceoptions - standard trace options

isis-traceoptions - The following additional options are available: **adjacency, dis-election, db, flood, spf, debug**. The following packet types can be traced: **csn, hello, lsp, packets, psn**.

Description

The **traceoptions** command configures what type of IS-IS specific tracing should occur. For a general discussion of the traceoptions format, see "Chapter 2 Trace Statements and Global Options" on page 3. The specific options available are as follows:

adjacency - Trace changes to the adjacency database.
dis-election - Trace changes in which router is elected DIS.
db - Trace changes to the LSP database.
flood - Trace flooding operations.
spf - Trace the routing table operations.
debug - Trace in much detail all aspects of the protocol's operation.

The packet types that can be traced are as follows:

csn - Complete Sequence Numbers (CSN) packets
hello - IS-IS Hello (IIH) packets
lsp - Link State Packets (LSPs)
packets - all packets
psn - Partial Sequence Numbers (PSNs) packets

If the detail flag is specified, the entire packet will be traced; otherwise, just the fixed header will be traced.

Note: Full tracing (including debugging) can be quite expensive and may affect the overall performance of the router.

Default

traceoptions none;

Context

isis statement

Examples

Example 1

The following example configures the router to trace all adjacency and SPF-related changes.

```
isis on {
    traceoptions adjacency spf;
};
```

Example 2

This example configures the router to trace all received and sent LSPs in full detail (i.e., including variable length fields, or TLVs).

```
isis on {
    traceoptions detail lsp packets;
};
```

See Also

"Chapter 13 Intermediate System to Intermediate System (IS-IS)" on page 55 of *Configuring GateD*

isis statement on page 185

traceoptions on page 3