

Chapter 20

Internet Group Management Protocol (IGMP)

enable | disable

Name

`enable` | `disable` - enables or disables IGMP on this interface or list of interfaces

Syntax

```
enable ; | disable ;
```

Parameters

none

Description

The **enable** statement causes IGMP to be spoken over the interfaces named in the **interface** statement in which the **enable** statement is contained. The **disable** statement specifies that IGMP messages received on any of the interfaces named in the **interface** statement must be ignored.

Defaults

```
enable ;
```

Context

igmp interface statement

Examples

Example 1

The following configuration disables IGMP on all fxp interfaces except for fxp2.

```
igmp on {  
    interface fxp { disable; } ;  
    interface fxp2 { enable; } ;  
};
```

Example 2

The default behavior is that if one or more **interface** statements appear within an **igmp on** or **igmp yes** clause, then IGMP runs on the interfaces named in the **interface** statements. Therefore, the following syntax provides the same configuration as in example 1.

```
igmp on {  
    interface fxp2 { enable; } ;  
};
```

Example 3

The absence of the enable or disable statement implies that the named interfaces should be enabled; thus, the configuration in example 2 can be reduced even further as follows:

```
igmp on {  
    interface fxp2;  
};
```

Example 4

The following example will disable IGMP on all interfaces, not only those mentioned in the **interface** statements below.

```
igmp off {  
    interface fxp2 fxp3;  
};
```

See Also

"Chapter 24 Internet Group Management Protocol (IGMP)" on page 119 in *Configuring GateD*

igmp on on page 529

igmp

Name

igmp - enables or disables the IGMP protocol

Syntax

```
igmp (on | off) [{igmp_statements}];
```

Parameters

on - turns on the IGMP protocol

off - turns off the IGMP protocol

igmp_statements - any of the various commands and options that are available (in this section)

Description

The **igmp** statement specifies whether or not the IGMP protocol is to run on any interfaces. If **igmp yes** or **igmp on** is specified, then IGMP will run on all of the interfaces named in the **interface** statements, contained within the **igmp** statement, subject to the other options contained within the **interface** statements. If no **interface** statements are present, then IGMP will run on all broadcast and multicast-capable or point-to-point interfaces. If **igmp no** or **igmp off** is specified, then IGMP will not run on any interfaces, regardless of the presence of **interface** statements contained within the **igmp** statement.

This statement enables or disables the IGMP protocol. If enabled, IGMP will default to enabling all interfaces that are both broadcast and multicast-capable or point-to-point. These interfaces are identified by the **IFF_BROADCAST** and **IFF_MULTICAST** interface flags. IGMP must be enabled before one of the IP Multicast routing protocols is enabled.

Defaults

```
igmp off
```

Context

global

Examples

Example 1

The following configuration enables IGMP on all interfaces.

```
igmp on ;
```

Example 2

The following configuration disables IGMP on all interfaces.

```
igmp off;
```

Example 3

The following configuration enables IGMP on interface fxp2 only.

```
igmp on {  
    interface fxp2 { enable; };  
};
```

See Also

"Chapter 24 Internet Group Management Protocol (IGMP)" on page 119 in *Configuring GateD*

enable | **disable** on page 527

interface on page 531

interface

Name

interface - selects the interfaces on which the IGMP protocol will be run

Syntax

```
interface interface_list
```

Parameters

interface_list - one or more interface names, including wildcard names (names without a number) and names that can specify more than one interface or address, or the token **all** for all interfaces

Description

The **interface** statement specifies the interfaces on which IGMP will run.

Defaults

If no **interface** statements appear within the **igmp** statement, then the default behavior is to run IGMP on all broadcast and multicast-capable or point-to-point interfaces. If at least one **interface** statement is present, then IGMP will run only on the named interfaces, subject to the options contained within the **interface** statements.

Context

igmp statement

Examples

Example 1

The following configuration specifies that IGMP should run on interface fxp2 only.

```
igmp on {  
    interface fxp2 {enable;} ;  
};
```

Example 2

The following configuration specifies that IGMP should run on all interfaces whose name begins with "fxp".

```
igmp on {  
    interface fxp {enable;} ;  
}
```

See Also

“Chapter 24 Internet Group Management Protocol (IGMP)” on page 119 in *Configuring GateD*

igmp on page 529

last-mem-query-intvl

Name

last-mem-query-intvl - taken together with the value specified in the **robustness** and **max-response-time** statements, **last-mem-query-intvl** specifies the number of seconds to wait for an IGMP Group Membership Report before concluding that no members are present for the indicated group.

Syntax

```
last-mem-query-intvl sec ;
```

Parameters

sec - an integer between the **max-response-time** + 1 and 25, inclusive.

Description

When an IGMP router receives an IGMP Group Leave message on interface *i*, it sends several IGMP Group-Specific Queries. If no Membership Report message is received before the Max Response time of the last Group-Specific Query elapses, then the router concludes that the indicated group has no members reachable via *i*. The **last-mem-query-intvl** statement specifies the amount of time between transmission of subsequent Group-Specific Query messages. The Max Response Time field of each message is set to the value specified in the **max-response-time** statement. The value specified by the **last-mem-query-intvl** statement corresponds to "Last Member Query Interval" as defined in the IGMPv2 protocol specification. The **robustness** statement determines the total number of Group-Specific Queries that are sent.

The **last-mem-query-intvl** statement can appear inside the **igmp** statement but outside of all **interface** statements, as well as within an **interface** statement. If it appears outside of all **interface** statements, then it specifies the value for all interfaces. If the **last-mem-query-intvl** statement appears inside of an **interface** statement, then it specifies the value for all interfaces named in the **interface** statement, overriding any **last-mem-query-intvl** statement that may appear elsewhere.

Defaults

```
last-mem-query-intvl 1 ;
```

Context

igmp statement

igmp interface statement

Examples

Example 1

The following configuration sets the Last Memory Query Interval to five seconds for all interfaces on which IGMP is running. The default value of the **robustness** statement is two and the default value of the **max-response-time** statement is 10 seconds. Therefore,

after receiving a Group Leave message for Group G on fxp1, GateD will send at most two Group-Specific Query messages, one every 5 seconds, each with the Max Response Time field set to 10 seconds. If no Group Membership Report for G is received within 15 seconds (5+10=15), then the router will conclude that there are no members of G reachable via fxp1.

```
igmp on {  
    interface fxp;  
    last-mem-query-intvl 5;  
};
```

Example 2

This will configure fxp2 with a Last Member Query Interval of 3 seconds; fxp3 will have a Last Member Query Interval of 4 seconds; all the rest of the fxp interfaces will have a Last Member Query Interval of 5 seconds.

```
igmp on {  
    interface fxp {  
        enable;  
    };  
    interface fxp2 {  
        enable;  
        last-mem-query-intvl 3;  
    };  
    interface fxp1 {  
        enable;  
        last-mem-query-intvl 4;  
    }  
    last-mem-query-intvl 5;  
};
```

Example 3

This will configure the default Last Member Query Interval to 5 seconds for all interfaces on which IGMP is running. On fxp1, this value is overridden to 4 seconds, and on fxp2, it is overridden to 3 seconds.

```
igmp on {  
    last-mem-query-intvl 5;  
    interface fxp {  
        enable;  
    };  
    interface fxp1 {  
        enable;  
        last-mem-query-intvl 4;  
    }  
};
```

```
};  
interface fxp2 {  
    enable;  
    last-mem-query-intvl 3;  
};  
}
```

See Also

"Chapter 24 Internet Group Management Protocol (IGMP)" on page 119 in *Configuring GateD*

max-response-time on page 536

robustness on page 541

max-response-time

Name

max-response-time - Taken together with the value specified in the **last-mem-query-intvl** and robustness statements, **max-response-time** specifies the number of seconds to wait for an IGMP Group Membership Report before concluding that no members are present for the indicated group.

Syntax

```
max-response-time sec ;
```

Parameters

sec - The amount of time, in seconds, that a router will wait to hear a membership report before removing the group dependency. This is an integer between 1 and MIN(25, Query Interval) inclusive.

Description

The **max-response-time** statement specifies in seconds the value to include in the Max Response Time field of IGMP Group-Specific Query messages. When an IGMP router receives an IGMP Group Leave message on interface *i*, it sends several IGMP Group-Specific Queries. If no Membership Report message is received before the Max Response Time of the last Group-Specific Query elapses, then the router concludes that the indicated group has no members reachable via *i*. The **last-mem-query-intvl** statement specifies the time in seconds between subsequent transmissions of Group-Specific Query messages. The **robustness** statement determines the total number of Group-Specific Queries that are sent.

The argument to **max-response-time** can be any integer between one and either 25 or the Query Interval, whichever is lower. The Query Interval value is set with the **query-interval** statement.

The **max-response-time** statement can appear inside the **igmp** statement, but outside of all **interface** statements, as well as within an **interface** statement. If it appears outside of all **interface** statements, then it specifies the value for all interfaces. If the **max-response-time** statement appears inside of an **interface** statement, then it specifies the value for all interfaces named in the **interface** statement, overriding any **max-response-time** statement that may appear elsewhere.

Defaults

```
max-response-time 10 ;
```

Context

igmp statement

igmp interface statement

Examples

Example 1

The following configuration sets Max Response Time to 220 seconds for all interfaces. The default value of the `last-mem-query-intvl` is one second. Therefore, after receiving a Group Leave message for group G on fxp1, GateD will send at most two Group-Specific Query messages, one every second, each with a Max Response Time set to 220 seconds. If no Group Membership Report for G is received within $1+220=221$ seconds, then the router will conclude that there are no members of group G reachable via fxp1.

Set the max-response time for all interfaces to 220 seconds.

```
igmp on {  
    interface all {  
        enable;  
    };  
    max-response-time 220;  
}
```

Example 2

Set the max-response time to 220 seconds for all interfaces on which IGMP is running. On fxp1, this value is overridden to 30 seconds.

```
igmp on {  
    interface all {  
        enable;  
    };  
    max-response-time 220;  
    interface fxp1 {  
        max-response-time 30;  
    };  
}
```

See Also

"Chapter 24 Internet Group Management Protocol (IGMP)" on page 119 in *Configuring GateD*

`last-mem-query-intvl` on page 533

`robustness` on page 541

nosend

Name

nosend - allows the interface to receive, but prevents it from sending, any IGMP packets

Syntax

```
nosend ;
```

Parameters

none

Description

nosend allows the interface to receive, but prevents it from sending, any IGMP packets. **nosend** permits the construction of a kernel Virtual InterFace (VIF) for the configured interface and, thus, allows GateD to subdue kernel upcalls (for example, IGMP NOCACHE upcalls) in some (older) kernels.

Defaults

disabled

Context

igmp interface statement

Examples

This example configures IGMP to run all fxp interfaces, but prevents the sending of IGMP messages.

```
igmp on {  
    interface fxp {  
        enable;  
    };  
    interface fxp2 {  
        enable;  
        nosend;  
    };  
}
```

See Also

"Chapter 24 Internet Group Management Protocol (IGMP)" on page 119 in *Configuring GateD*

igmp on page 529

query-interval

Name

query-interval - specifies the time between the transmission of General Query messages in seconds

Syntax

```
query-interval sec ;
```

Parameters

sec - an integer between 1 and 65535 inclusive

Description

The Designated Querier on a subnet periodically sends General Query messages to verify the state of group memberships. The frequency with which these messages are sent is specified with the **query-interval** statement.

The **query-interval** statement can appear inside the **igmp** statement, but outside of all **interface** statements, as well as within an **interface** statement. If it appears outside of all **interface** statements, then it specifies the value for all interfaces. If the **query-interval** statement appears inside of an **interface** statement, then it specifies the value for all interfaces named in the **interface** statement, overriding any **query-interval** statement that may appear elsewhere.

Defaults

```
query-interval 125 ;
```

Context

igmp statement

igmp interface statement

Examples

Example 1

The following configuration sets the query-interval for interface fxp2 to 55 seconds. The query-interval for all remaining interfaces defaults to 125 seconds.

```
igmp yes {  
    interface fxp {  
        enable;  
    };  
    interface fxp2 {  
        enable;  
        query-interval 55;  
    }  
}
```

```
    };  
}
```

Example 2

The following configuration sets the query-interval for fxp2 to 120 seconds and the query-interval for all remaining interfaces to 130 seconds.

```
igmp yes {  
    interface fxp {  
        enable;  
    };  
    interface fxp2 {  
        enable;  
        query-interval 120;  
    };  
    query-interval 130;  
}
```

See Also

“Chapter 24 Internet Group Management Protocol (IGMP)” on page 119 in *Configuring GateD*

igmp on page 529

robustness

Name

robustness - allows for tuning of the IGMP protocol to accommodate a lossy subnet

Syntax

```
robustness value ;
```

Parameters

num - an integer between 2 and 65535 inclusive

Description

The **robustness** statement determines the number of IGMP General Query messages sent when a router first comes up, as well as the number of Group-Specific Query messages sent in response to receiving a Group Leave message. On a given network, IGMP is "robust" to the loss of "robustness - 1" messages. Per the IGMP protocol specification, the robustness value should be greater than 1.

The **robustness** statement can appear inside the **igmp** statement, but outside of all **interface** statements, as well as within an **interface** statement. If it appears outside of all **interface** statements, then it specifies the value for all interfaces. If the **robustness** statement appears inside of an **interface** statement, then it specifies the value for all interfaces named in the **interface** statement, overriding any **robustness** statement that may appear elsewhere.

Defaults

```
robustness 2 ;
```

Context

igmp statement

igmp interface statement

Examples

Example 1

The following configuration sets the robustness for interface fxp2 to three. All other interfaces have the default robustness of two.

```
igmp yes {  
    interface fxp {  
        enable;  
    };  
    interface fxp2 {  
        enable;  
    };  
}
```

```
        robustness 3;
    };
}
```

Example 2

The following configuration sets all the default robustness for all interfaces to three. The robustness for interface fxp2 is overridden to be four.

```
igmp yes {
    interface fxp {
        enable;
    };
    interface fxp2 {
        enable;
        robustness 4;
    };
    robustness 3;
}
```

See Also

“Chapter 24 Internet Group Management Protocol (IGMP)” on page 119 in *Configuring GateD*

`last-mem-query-intvl` on page 533

`max-response-time` on page 536

traceoptions

Name

traceoptions - specifies the tracing options for this group

Syntax

```
traceoptions trace_options ;
```

Parameters

Trace options include:

packets - Trace all IGMP packets.

query - Trace IGMP Membership Query messages.

report - Trace IGMP Membership Report messages.

leave - Trace IGMP Leave messages.

mtrace - Trace Mtrace messages. Mtrace is defined in draft-ietf-idmp-traceroute-ipm-05.txt.

all - Trace all IGMP messages.

Description

traceoptions specifies the tracing options for IGMP. By default, these are inherited from the global trace options.

Default

inherited from global traceoptions

Context

igmp statement

Examples

Example 1

In the following example, all IGMP events will be logged to the file, /var/tmp/igmp.log. This includes timer events and exceptional events, as well as sending and receiving IGMP messages.

```
igmp yes {  
    traceoptions "/var/tmp/igmp.log" all;  
    interface fxp {  
        enable;  
    };  
    interface fxp2 {  
        enable;  
    };  
}
```

```
        robustness 3;
    };
};
```

Example 2

In the following example, only IGMP Membership Report and Membership Query messages that are sent and/or received are logged to the file, /var/tmp/igmp.log.

```
igmp yes {
    traceoptions "/var/tmp/igmp.log" report query;
    interface fxp {
        enable;
    };
    interface fxp2 {
        enable;
        robustness 3;
    };
};
```

Example 3

In the following example, only IGMP Membership Reports that are sent, and Query messages that are received, are logged to the file, /var/tmp/igmp.log.

```
igmp yes {
    traceoptions "/var/tmp/igmp.log" send report recv query;
    interface fxp {
        enable;
    };
    interface fxp2 {
        enable;
        robustness 3;
    };
};
```

See Also

“Chapter 24 Internet Group Management Protocol (IGMP)” on page 119 in *Configuring GateD*

“Mtrace” as defined in draft-ietf-idmp-traceroute-ipm-05.txt

`traceoptions` on page 3