Chapter 17
Protocol Independent Multicast (PIM)

assert-holdtime

Name

assert-holdtime - specifies the number of seconds that Assert state should be maintained in the absence of a refreshing Assert message

Syntax

assert-holdtime *sec* **;**

Parameters

sec - an integer between 1 and 2^32 - 1 (4,294,967,295), inclusive

Description

When a PIM router receives an Assert message, it modifies the outgoing interface list for a (*,G) or (S,G) entry, as specified by the message. The lifetime of this modification is specified by the **assert-holdtime** statement. If another Assert message does not refresh the Assert state before the lifetime expires, then the outgoing interface list reverts to its previous state. The **assert-holdtime** statement can appear outside of a **sparse** clause, or it can appear within an **interface** statement. In the former case, the holdtime applies to all configured interfaces. In the latter case, the holdtime applies only to the interface(s) to which the **interface** statement refers. An **assert-holdtime** statement associated with an interface overrides any **assert-holdtime** statement that may appear outside of the **sparse** statements.

Defaults

assert-holdtime 180 ;

Context

pim statement

pim-sm (sparse) **interface** statement

Examples

Example 1

The following **pim** statement configures a PIM-SM component, "sm0" with an Assert holdtime of 20 seconds. This holdtime applies to all the interfaces configured for the component.

```
pim yes {
    assert-holdtime 20;
    sparse "sm0" {
        interface fxp0 fxp1;
    };
};
```

Example 2

The following **pim** statement configures a PIM-SM component, “sm0”, with interface fxp1 having an Assert holdtime of 45 seconds. Interface fxp0 will be configured with the default Assert holdtime of 20 seconds, specified by the **assert-holdtime** statement appearing outside of the **sparse** statement.

```
pim yes {
    assert-holdtime 20;
    sparse "sm0" {
        interface fxp0 ;
        fxp1 {
            assert-holdtime 45;
        }
    };
};
```

See Also

pim statement on page 414

boundary

Name

boundary - specifies that the associated interface is at a PIM domain boundary

Syntax

```
boundary ;
```

Parameters

none

Description

boundary specifies that the indicated interface(s) is (are) at a PIM domain boundary. PIM-SM Bootstrap Router (BSR) and Candidate-RP-Advertisement messages will not be sent or accepted over the associated interface(s). PIM Join/Prune messages however, are still exchanged. **boundary** makes it possible for adjacent but administratively separate PIM domains to be connected via MSDP, and thus for multicast group members in one domain to learn of and receive traffic from sources in another.

Defaults

no boundary specified for an interface

Context

pim-sm (sparse) **interface** statement

Examples

The following **pim** statement configures a PIM-SM component, "sm0", with a boundary on interface fxp1 and no boundary on interface fxp0.

```
pim yes {
    assert-holdtime 20;
    sparse "sm0" {
        interface fxp0;
        interface fxp1 {
            boundary;
        };
    };
};
```

See Also

pim statement on page 414

bsr

Name

bsr - specifies that this router should act as a Candidate Bootstrap Router (CBSR)

Syntax

bsr ((*address* **| on) [priority** *pri* **]) | (off)**

Parameters

address - *address* must be a valid IPv4 address, host name, or interface name associated with one of the interfaces configured within the **sparse** statement in which the **bsr** statement appears.

pri - If a BSR priority is specified using the **priority** keyword, then *pri* must be an integer in the range 0 to 255, inclusive.

Description

GateD provides two mutually exclusive methods for RP set distribution: statically configured RPs and BSR. The BSR method is compliant with the mechanism described in draft-ietf-pim-sm-v2-new-01. The **bsr** statement enables the BSR method of RP set distribution.

A PIM BSR is responsible for distributing RP and group address information to its PIM domain. Multiple routers in a PIM-SM domain may be configured as CBSRs, and the PIM-SM protocol provides an election mechanism for selecting a BSR from the candidate pool. If **bsr off** is specified, or if the **bsr** statement is omitted, then this router is not eligible to become a BSR.

PIM-SM BSR messages contain a BSR address. If **bsr on** or **bsr yes** is specified, then the BSR address is chosen from one of the interfaces configured for the PIM-SM component. If one wishes to specify the exact address to be used as the BSR address, then **bsr** *address* should be used, where *address* is the desired IPv4 address.

PIM-SM BSR messages also contain a BSR priority, which can be used to bias the BSR election process. CBSRs with higher priorities are preferred. If two CBSRs have the same priority, then the CBSR with the larger IP address is preferred. The optional **priority** *pri* statement contained within the **bsr** statement specifies this CBSR's priority. If the **priority** *pri* statement is omitted, then the priority defaults to 0.

Defaults

bsr no ;

Context

pim-sm (sparse) statement

Examples

Example 1

The following **pim** statement configures a PIM-SM component, "sm0", containing interfaces 192.168.10.2 and 192.168.22.1. The component will announce itself as a CBSR with an interface chosen from one of the two configured interfaces. The advertised CBSR priority will be 0.

```
pim yes {
    sparse "sm0" {
        bsr on;
        interface 192.168.10.2 192.168.22.1;
    };
};
```

Example 2

The following **pim** statement configures a PIM-SM component, "sm0", containing interfaces 192.168.10.2 and 192.168.22.1. The component will announce itself as a CBSR with address 192.168.10.2. The advertised CBSR priority will be 0.

```
pim yes {
    sparse "sm0" {
        bsr 192.168.10.2;
        interface 192.168.10.2 192.168.22.1;
    };
};
```

Example 3

The following **pim** statement configures a PIM-SM component, "sm0", containing interfaces 192.168.10.2 and 192.168.22.1. The component will announce itself as a CBSR with address 192.168.10.2. The advertised CBSR priority will be 10.

```
pim yes {
    sparse "sm0" {
        bsr 192.168.10.2 {
            priority 10;
        };
        interface 192.168.10.2  192.168.22.1;
    };
};
```

See Also

pim statement on page 414

crp statement on page 388

static-rp statement on page 425

bsr-holdtime

Name

bsr-holdtime - specifies the time after which the elected Bootstrap Router (BSR) will be assumed unreachable when bootstrap messages are not received from it

Syntax

bsr-holdtime *secs* **;**

Parameters

secs - an integer between 1 and 2^32 - 1 (4,294,967,295), inclusive

Description

bsr-holdtime specifies the time after which the elected BSR will be assumed unreachable when bootstrap messages have not been received from it. The recommended value for this parameter is 2 * **bsr-period** + 10. The BSR mechanism implemented by GateD is described in draft-ietf-pim-sm-v2-new-01.

Note: There is a tying of **bsr-period** and **bsr-holdtime**. Configuring one away from default configures the other, unless it too is explicitly configured.

Defaults

```
bsr-holdtime 130 ;
```

Context

pim-sm (sparse) statement

Examples

The following **pim** statement configures a PIM-SM component, "sm0", containing interfaces 192.168.10.2 and 192.168.22.1. Once a BSR is elected, the BSR's holdtime will be 110 seconds. (The component itself is not configured to be a CBSR.)

```
pim yes {
    sparse "sm0" {
        bsr-holdtime 110;
        interface 192.168.10.2 192.168.22.1;
    };
};
```

See Also

bsr-period statement on page 386

pim statement on page 414

bsr statement on page 382

bsr-period

Name

bsr-period - specifies the interval between originating bootstrap messages and should be equal to 60 seconds

Syntax

bsr-period *secs* **;**

Parameters

secs - an integer between 1 and 2^32 - 1 (4,294,967,295), inclusive

Description

If a router is acting as the elected BSR for a PIM-SM domain, **bsr-period** specifies the number of seconds the router should wait between successive bootstrap message transmissions.

Note: There is a tying of **bsr-period** and **bsr-holdtime**. Configuring one away from default configures the other, unless it too is explicitly configured.

Defaults

bsr-period 60 ;

Context

pim-sm (bsr) statement

Examples

The following **pim** statement configures a PIM-SM component, "sm0", containing interfaces 192.168.10.2 and 192.168.22.1. The component will announce itself as a CBSR with address 192.168.10.2. If this router is elected to be the BSR for the PIM-SM domain, it will generate a BSR message every 45 seconds.

```
pim yes {
    sparse "sm0" {
        bsr 192.168.10.2 {
            priority 10;
            bsr-period 45;
        };
        interface 192.168.10.2 192.168.22.1;
    };
};
```

See Also

bsr-holdtime statement on page 385

pim statement on page 414

bsr statement on page 382

crp

Name

crp - specifies that this router should act as a Candidate Rendezvous Point (CRP)

Syntax

crp (*address* **| on | yes)**

Parameters

address - the IPv4 address, host name, or interface name to advertise in C-RP-Adv messages as the CRP address

Description

The PIM protocol allows for multiple routers to volunteer as the Rendezvous Point (RP) for a given multicast group. Such volunteers are called CRPs. If **crp off** or **crp no** is specified, or if the **crp** clause is omitted entirely, then this PIM router is not a CRP.

PIM Candidate-RP-Advertisement messages contain the address of the CRP. If **crp on** or **crp yes** is specified, then the CRP address is chosen from the list of addresses configured for the PIM-SM component. If you want to specify the exact address to be used as the CRP address, you must use **crp** *address* where address is the desired IPv4 address, host name, or interface name.

If the **crp** statement is specified without naming group addresses within curly braces, (for example, **crp on**) then the router will be a CRP for the group address range, 224/4. If group addresses (or address ranges) are named within curly braces, then the router will be a CRP only for the named addresses/ranges.

A router is chosen as the RP for a multicast group from the set of CRPs via a well-known hash function. A CRP's suitability for a given multicast group may be biased with a priority. When choosing an RP for a group from the set of CRPs, the hash function is computed for each member of the set of CRPs with the lowest priority for the group. The CRP yielding the highest hash value is selected as the RP for the group.

(Note the difference between CRP and BSR priorities. For BSR priorities, higher values are better.)

The **crp** statement allows one to specify an optional default priority as well as a priority associated with individual group addresses. Both priorities must be integers between 0 and 255, inclusive.

Defaults

crp off ;

Context

pim-sm (sparse) statement

Examples

Example 1

In the following example, a single PIM-SM component, "sm0" is specified. The router will act as a CRP with the CRP address being chosen from one of the two interfaces configured within the **sparse** statement. Since no group ranges are specified, the router will advertise itself as a CRP for groups in the range 224/4 (for example, all multicast addresses). Finally, it will advertise itself with a priority of 2.

```
pim yes {
    sparse "sm0" {
        crp on {
            priority 2;
        };
        interface 192.168.10.2 192.168.22.1;
    };
};
```

Example 2

In the following example, a single PIM-SM component, "sm0" is specified. The router will act as a CRP with the CRP address being chosen from one of the two interfaces configured within the **sparse** statement. The router will advertise itself as a CRP for groups 224.1.2.3 and 224.1.2.4. The advertisement for group 224.1.3.4 will contain a priority of 1, whereas the advertisement of 224.1.2.4 will contain a priority of 2.

```
pim yes {
    sparse "sm0" {
        crp on {
            priority 2;
            group {
                224.1.2.3 priority 1;
                224.1.2.4;
            } ;
        };
        interface 192.168.10.2 192.168.22.1;
    };
};
```

Example 3

Group ranges can be specified within the curly braces of the **crp** statement via the **group-address mask** *mask* or **group-address masklen** *length* statements. In addition, an ASCII network name can be specified with the **host** keyword. These features are illustrated below. The router will advertise itself as a CRP with priority 2 for all groups in the range 224.0.1.0 to 224.0.1.3, as well as group 224.0.1.1 (ntp.mcast.net).

```
pim yes {
    sparse "sm0" {
        crp on {
            priority 2;
            224.0.1.1 masklen 30;
            host ntp.mcast.net;
        };
        interface 192.168.10.2 192.168.22.1;
    };
};
```

See Also

pim statement on page 414

bsr statement on page 382

static-rp statement on page 425

crp-adv-period

Name

crp-adv-period - sets the interval at which a Candidate Rendezvous Point (CRP) will send CRP Advertisement (Adv) messages to the Bootstrap Router (BSR)

Syntax

crp-adv-period *crp-adv-periodsecs* **;**

Parameters

crp-adv-periodsecs - an integer between 0 and 65535, inclusive

Description

When using the BSR mechanism to distribute RP set information throughout a PIM-SM domain, CRPs must periodically send C-RP-Adv messages to the BSR. The **crp-adv-period** statement specifies the time, in seconds, between successive C-RP-Adv messages.

Defaults

crp-adv-period 60 ;

Context

pim-sm (crp) statement

Examples

The following **pim** statement configures a PIM-SM component, "sm0", containing interfaces 192.168.10.2 and 192.168.22.1. The component will announce itself as a CRP with address 192.168.10.2 by sending a C-RP-Adv message every 20 seconds.

```
pim yes {
    sparse "sm0" {
        crp 192.168.10.2 {
            crp-adv-period 20;
        };
        interface 192.168.10.2 192.168.22.1;
    };
};
```

See Also

pim statement on page 414

crp statement on page 388

crp-holdtime

Name

crp-holdtime - specifies the holdtime, in seconds, advertised in Candidate Rendezvous Point Advertisement (C-RP-Adv) messages

Syntax

crp-holdtime *secs* **;**

Parameters

secs - an integer between 0 and 65535, inclusive

Description

For CRPs, **crp-holdtime** specifies the holdtime advertised in C-RP-Adv messages, and is used by the Bootstrap Router (BSR) to time out RPs. The recommended value for this parameter is 2.5 * [**crp-adv-period**].

Defaults

crp-holdtime 150 ;

Context

pim-sm (crp) statement

Examples

The following **pim** statement configures a PIM-SM component, "sm0", containing interfaces 192.168.10.2 and 192.168.22.1. The component will announce itself as a CRP with address 192.168.10.2. The holdtime advertised in its C-RP-Adv message will be 30 seconds.

```
pim yes {
    sparse "sm0" {
        crp 192.168.10.2 {
            crp-holdtime 30;
        };
        interface 192.168.10.2 192.168.22.1;
    };
};
```

See Also

pim statement on page 414

crp statement on page 388

dr-switch-immediate

Name

dr-switch-immediate - causes a Designated Router (DR) to initiate a switch to the Shortest Path Tree (SPT) for (S,G) upon receipt of the first data packet from source S

Syntax

```
dr-switch-immediate ;
```

Parameters

none

Description

The PIM-SM protocol allows a Rendezvous Point (RP) or a DR to switch from receiving data from a source S sent to a group G via the RP tree, to receiving data via the SP tree. Two methods are available within GateD for deciding when an SP tree switch should be initiated. One of these methods is to initiate a switch to the SP tree for an (S,G) pair upon receipt of the first data packet from S addressed to G. The **dr-switch-immediate** statement causes a DR to initiate an SP tree switch upon receipt of the first packet from S addressed to G.

If the **dr-switch-immediate** statement does not appear anywhere within the **sparse** statement, then a switch to the SP tree is initiated when the traffic rate exceeds a threshold.

Defaults

The default is to switch to the SP tree when the traffic rate exceeds a threshold.

Context

pim-sm (sparse) statement

Examples

The following **pim** statement configures a PIM-SM component, “sm0”. If the router is the DR for local receivers, then the first time this router receives data from a new source, S, addressed to group G, it will initiate an SPT switch for (S,G). If the data arrive before the router is the DR for local receivers, then the switch will be initiated only when the router becomes the DR for local receivers.

```
pim yes {
    sparse “sm0” {
        interfaces fxp0 fxp1 ;
        dr-switch-immediate ;
    }
}
```

See Also

threshold statement on page 426

threshold-rp statement on page 430

threshold-dr statement on page 428

pim statement on page 414

rp-switch-immediate statement on page 422

group

Name

group - specifies a set of multicast groups and optional priorities for which the router will volunteer to be a Candidate Rendezvous Point (CRP)

Syntax

group { *group-information* **} ;**

Parameters

group-information - the groups for which the router is volunteering to be a CRP

Description

When using the BSR method of RP set distribution, CRPs will periodically send C-RP-Adv messages to the BSR. These messages specify a set of groups for which the CRP is volunteering to be an RP. In addition, a priority is associated with each set of groups and is used to decide which CRP will actually serve as the RP for a given group. The **group** statement optionally appears inside the **crp** statement and allows you to configure the groups for which the router is volunteering to be an RP, as well as the priorities associated with these groups.

Defaults

If the **group** statement is omitted from the **crp,** then the router will send C-RP-Adv messages indicating that it will serve as an RP for all multicast groups. The priority associated with a set of groups is determined by the **priority** statement.

Context

pim-sm (crp) statement

Examples

Example 1

In the following example, a single PIM-SM component, "sm0" is specified. The router will act as a CRP with the CRP address being chosen from one of the two interfaces configured within the **sparse** statement. Since no group ranges are specified, the router will advertise itself as a CRP for groups in the range 224/4 (for example, all multicast addresses). Finally, it will advertise itself with a priority of 2.

```
pim yes {
    sparse "sm0" {
        crp on {
            priority 2;
        };
        interface 192.168.10.2 192.168.22.1;
```

```
    };
};
```

Example 2

In the following example, a single PIM-SM component, "sm0", is specified. The router will act as a CRP with the CRP address being chosen from one of the two interfaces configured within the **sparse** statement. The router will advertise itself as a CRP for groups 224.1.2.3 and 224.1.2.4. The advertisement for group 224.1.3.4 will contain a priority of 1, whereas the advertisement of 224.1.2.4 will contain a priority of 2.

```
pim yes {
    sparse "sm0" {
        crp on {
            priority 2;
            group {
                224.1.2.3 priority 1;
                224.1.2.4;
            } ;
        };
        interface 192.168.10.2 192.168.22.1;
    };
};
```

Example 3

Group ranges can be specified within the curly braces of the **crp** statement via the *group-address* **mask** *mask* or *group-address* **masklen** *length* statements. In addition, an ASCII network name can be specified with the **host** keyword. These features are illustrated below. The router will advertise itself as a CRP with priority 2 for all groups in the range 224.0.1.0 to 224.0.1.3, as well as group 224.0.1.1 (ntp.mcast.net).

```
pim yes {
    sparse "sm0" {
        crp on {
            priority 2;
            224.0.1.0 masklen 30;
            host ntp.mcast.net;
        };
        interface 192.168.10.2 192.168.22.1;
    };
};
```

See Also

crp statement on page 388

hello-holdtime

Name

hello-holdtime - specifies how long neighbors should wait for Hello messages before expiring the sender's neighbor state

Syntax

hello-holdtime *sec* **;**

Parameters

sec - an integer between 0 to 65535, inclusive

Description

PIM Hello messages contain a holdtime specifying how long neighbors must wait for Hello messages before expiring the sender's neighbor state. **hello-holdtime** specifies the hold-time, in seconds, to advertise in Hello messages.

The **hello-holdtime** statement can appear both outside of a **sparse** statement and within an **interface** statement. If it appears outside of a **sparse** statement, then the specified value will be used for Hello messages transmitted via all interfaces configured into PIM components. If it appears inside of an **interface** statement, then the specified value overrides any previous specifications for the associated interfaces.

Defaults

hello-holdtime 105 ;

Context

pim statement

pim (sparse) **interface** statement

Examples

Example 1

The following **pim** statement configures a PIM-SM component, "sm0", containing interfaces fxp0 and fxp1. Each Hello message will contain a holdtime of 45 seconds.

```
pim yes {
    hello-holdtime 45;
    sparse "sm0" {
        interface fxp0 fxp1;
    };
};
```

Example 2

The following **pim** statement configures a PIM-SM component, “sm0”, containing interfaces fxp0 and fxp1. Hello messages sent via interface fxp0 will contain a holdtime of 45 seconds, while those sent via interface fxp1 will contain a holdtime of 20 seconds.

```
pim yes {
    hello-holdtime 45;
    sparse "sm0" {
        interface fxp0;      # Hellos will have holdtime of 45 secs
        interface fxp1 {     # Hellos will have a holdtime of 20 secs
            hello-holdtime 20;
        };
    };
};
```

See Also

hello-interval statement on page 401

pim statement on page 414

hello-interval

Name

hello-interval - specifies the frequency with which Hello messages are sent

Syntax

hello-interval *sec* **;**

Parameters

sec - an integer between 0 and 65535, inclusive

Description

PIM routers periodically multicast Hello messages on each network to which they are connected to alert other routers to the presence of the sender. The hello-interval parameter specifies the time, in seconds, between successive Hello messages.

The **hello-interval** statement can appear both outside of a **sparse** statement and within an **interface** statement. If it appears outside of a **sparse** statement, then the value specifies the time between Hello messages sent on all interfaces configured into PIM components. If it appears inside of an **interface** statement, then the specified value overrides any previous specifications for the associated interfaces.

Defaults

hello-interval 30 ;

Context

pim statement

pim-sm (sparse) **interface** statement

Examples

Example 1

The following **pim** statement configures a PIM-SM component, "sm0", containing interfaces fxp0 and fxp1. Hello messages will be sent via interfaces fxp0 and fxp1 every 35 seconds.

```
pim yes {
    hello-interval 35;
    sparse "sm0" {
        interface fxp0 fxp1;
    };
};
```

Example 2

The following **pim** statement configures a PIM-SM component, "sm0", containing interfaces fxp0 and fxp1. Hello messages will be sent via interface fxp0 every 45 seconds, while those sent via interface fxp1 will be sent every 20 seconds.

```
pim yes {
    hello-interval 45;
    sparse "sm0" {
        interface fxp0;    # Hellos will be sent every 45 secs
        interface fxp1 {   # Hellos will be sent every 20 secs
            hello-interval 20;
        };
    };
};
```

See Also

hello-holdtime statement page 397

pim statement on page 414

hello-priority

Name

hello-priority - specifies the priority used to determine Designated Forwarder (DF) and include in PIM Hello messages

Syntax

hello-priority *pri* **;**

Parameters

pri - an integer between 1 and 2^32 - 1 (4,294,967,295), inclusive

Description

PIM Hello messages may contain a priority field that is used to elect a DF on a shared network. All Hello messages originated by GateD contain such a priority. DFs are responsible for encapsulating multicast data from local sources into PIM-SM Register messages and for unicasting them to the Rendezvous Point. The router with the highest priority wins the DF election. In the case of a tie, the router with the highest IP address wins.

If at least one neighbor on the network does not use Hello priorities, then election of a DF is carried out using only IP addresses, where the highest address wins.

The **hello-priority** parameter can appear both outside of a **sparse** statement and within an **interface** statement. If it appears outside of a **sparse** statement, then the value specifies the priority of Hello messages sent on all interfaces configured into PIM components. If it appears inside of an interface statement, then the specified value overrides any previous specifications for the associated interfaces.

Defaults

hello-priority 1 ;

Context

pim statement

pim-sm (sparse) **interface** statement

Examples

Example 1

The following **pim** statement configures a PIM-SM component, "sm0", containing interfaces fxp0 and fxp1. Hello messages sent via interfaces fxp0 and fxp1 will contain a priority of 2.

```
pim yes {
    hello-priority 2;
    sparse "sm0" {
        interface fxp0 fxp1;
    };
```

```
};
```

Example 2

The following **pim** statement configures a PIM-SM component, "sm0", containing interfaces fxp0 and fxp1. Hello messages sent via interface fxp0 will have a priority of 10, while those sent via interface fxp1 will have a priority of 2.

```
pim yes {
    hello-priority 10;
    sparse "sm0" {
        interface fxp0;        # Hello priority of 10
        interface fxp1 {       # Hello priority of 2
            hello-priority 2;
        };
    };
};
```

See Also

pim statement on page 414

interface

Name

interface - specifies the interfaces associated with a PIM-SM component and over which PIM-SM will be spoken

Syntax

interface *interface-list* **[{** *pim-sm_interface_parameters* **}] ;**

Parameters

interface-list - the list of interfaces, IP addresses, host names, or interface names with which the component should be associated and over which PIM-SM will be spoken

pim-sm_interface_parameters - refers to the PIM-SM specific parameters, described in this document, that may be configured at the granularity of an interface

Description

This statement specifies the list of interfaces associated with a PIM-SM component and over which PIM-SM will be spoken. An interface does not need to be up (or even exist) to be configured. When the interface becomes available, PIM-SM will begin running on the interface. If the **interface** statement is omitted, then PIM will not be spoken at all.

Defaults

```
interface  all {disable;};
```

Context

pim-sm (sparse) statement

Examples

The following **pim** statement configures a PIM-SM component, "sm0", containing interfaces fxp0, fxp1 and fxp2. Interfaces fxp0 and fxp1 are configured with the **pim** statement-specified Hello priority of 10. Interface fxp2 is configured with an interface-specific Hello priority of 2.

```
pim yes {
    hello-priority 10;
    sparse "sm0" {
        interface fxp0 fxp1;    # Hello priority of 10
        interface fxp2 {        # Hello priority of 2
            hello-priority 2;
        };
    };
};
```

See Also

pim statement on page 414

jp-holdtime

Name

jp-holdtime - specifies the Join/Prune holdtime that is advertised in PIM Join/Prune messages

Syntax

jp-holdtime *sec ;*

Parameters

sec - an integer between 0 and 65535, inclusive

Description

This statement specifies the holdtime that is advertised in PIM Join/Prune messages. Receivers must wait at least this long after receiving a Join/Prune message before deleting the Join/Prune state associated with the advertiser. The recommended value is 3.5 * **jp-interval**.

The **jp-holdtime** statement can appear both outside of a **sparse** statement and within an **interface** statement. If it appears outside of a **sparse** statement, then the value specifies the holdtime of Join/Prune messages sent on all interfaces configured into PIM components. If it appears inside of an **interface** statement, then the specified value overrides any previous specifications for the associated interfaces.

Note: There is a tying of **jp-holdtime** and **jp-interval**. If the **jp-interval** is configured and the **jp-holdtime** is not, then the **jp-holdtime** is automatically set to 3.5 * **jp-interval**.

Defaults

jp-holdtime 210 ;

Context

pim statement

pim-sm (sparse) **interface** statement

Examples

The following **pim** statement configures a PIM-SM component, "sm0", containing interfaces fxp0 and fxp1. Join/Prune messages sent via interfaces fxp0 and fxp1 will contain a holdtime of 30.

```
pim yes {
    jp-holdtime 30;
    sparse "sm0" {
        interface fxp0 fxp1;  # J/P Holdtime of 30
    };
};
```

See Also

jp-interval statement on page 407

pim statement on page 414

jp-interval

Name

jp-interval - specifies the frequency with which Join/Prune messages are sent

Syntax

jp-interval *sec* **;**

Parameters

sec - an integer between 0 and 65535, inclusive

Description

The Join/Prune state on an upstream neighbor must be refreshed by periodic Join/Prune messages. This parameter specifies the number of seconds between successive Join/Prune messages sent to upstream neighbors to maintain the neighbor's Join/Prune state.

The **jp-interval** statement can appear both outside of a **sparse** statement and within an **interface** statement. If it appears outside of a **sparse** statement, then the value specifies the frequency with which periodic Join/Prune messages are sent on all interfaces configured into PIM components. If it appears inside of an **interface** statement, then the specified value overrides any previous specifications for the associated interfaces.

Note: There is a tying of **jp-holdtime** and **jp-interval**. If the **jp-interval** is configured and the **jp-holdtime** is not, then the **jp-holdtime** is automatically set to 3.5 * **jp-interval**.

Defaults

```
jp-interval 60 ;
```

Context

pim statement

pim-sm (sparse) **interface** statement

Examples

Example 1

The following **pim** statement configures a PIM-SM component, "sm0", containing interfaces fxp0 and fxp1. Periodic Join/Prune messages are sent to neighbors on interfaces fxp0 and fxp1 every 30 seconds.

```
pim yes {
    jp-interval 30;
    sparse "sm0" {
        interface fxp0 fxp1;  # J/P's sent every 30 secs
    };
```

```
};
```

Example 2

The following **pim** statement configures a PIM-SM component, "sm0", containing interfaces fxp0 and fxp1. Periodic Join/Prune messages sent via interface fxp0 will be sent every 30 seconds, while those sent via interface fxp1 will be sent every 10 seconds.

```
pim yes {
    jp-interval 30;
    sparse "sm0" {
        interface fxp0;       # J/P Holdtime of 10
        interface fxp1 {
            jp-interval 10;
        };
    };
};
```

See Also

jp-holdtime statement on page 405

pim statement on page 414

mrt-period

Name

mrt-period - specifies the number of seconds to wait between examinations of a PIM component's multicast routing table (MRT)

Syntax

mrt-period *sec* **;**

Parameters

sec - an integer between 1 and 3600, inclusive

Description

A PIM component's MRT is examined periodically in order to remove entries that have been marked for deletion. **mrt-period** specifies the number of seconds to wait between examinations. This can be a computationally expensive operation if the number of entries is large.

Defaults

```
mrt-period 15 ;
```

Context

pim (sparse) statement

Examples

The following **pim** statement configures a PIM-SM component, "sm0". The MRT for this component will be examined every 30 seconds.

```
pim yes {
    mrt-period 30;
    sparse "sm0" {
        interface fxp0 fxp1;
    };
};
```

See Also

mrt-spt-mult statement on page 410

mrt-stale-mult statement on page 412

pim statement on page 414

mrt-spt-mult

Name

mrt-spt-mult - together with the **mrt-period**, specifies the interval at which the data rate threshold for all S,G entries will be checked for a possible switch to the SP tree

Syntax

mrt-spt-mult *m* **;**

Parameters

m - an integer between 1 and 100, inclusive

Description

The PIM-SM protocol allows a Rendezvous Point (RP) or a Designated Router (DR) to switch from receiving data from a source S sent to a group G via the RP tree, to receiving data from the SP tree. Two methods are available within GateD for deciding when an SPT switch should be initiated. One of these methods involves setting a threshold, in average bytes per second. If the data rate of traffic received from S addressed to G exceeds this threshold, then a switch is initiated. The **mrt-spt-mult** statement specifies the interval over which the average bytes per second calculation is made. The interval is expressed as a multiple of the **mrt-period**, defined by the **mrt-period** statement.

Defaults

mrt-spt-mult 14 ;

Context

pim (sparse) statement

Examples

The following **pim** statement configures a PIM-SM component, "sm0". The MRT for this component will be examined every 30 seconds. If this router is an RP or DR for group G, then all (S,G) entries on the RP tree, where S can be any source, will be examined every 60 seconds to determine whether an SPT switch should be initiated for S.

```
pim yes {
    mrt-period 30;
    sparse "sm0" {
        mrt-spt-mult 2;
        interface fxp0 fxp1;
    };
};
```

See Also

mrt-period statement on page 409

rp-switch-immediate statement on page 422

dr-switch-immediate statement on page 393

pim statement on page 414

threshold statement on page 426

threshold-rp statement on page 430

threshold-dr statement on page 428

mrt-stale-mult

Name

mrt-stale-mult - together with the **mrt-period** statement, specifies the minimum number of seconds that a source may be silent before its corresponding (S,G) entry can be timed out

Syntax

mrt-stale-mult *m* **;**

Parameters

m - an integer between 1 and 100, inclusive

Description

When a source stops sending to a group, the corresponding (S,G) entry is said to have become "stale" and is now a candidate for deletion from the PIM multicast forwarding table (MRT). The PIM-SM protocol defines a Keep-Alive Timer (KAT) for each (S,G) entry, which is reset by the arrival of data from source S addressed to group G. If the KAT ever expires, and if other conditions are met, then the entry can be deleted. The **mrt-stale-mult** statement, together with the **mrt-period** statement, specifies the minimum number of seconds that a source must be silent before the entry is considered stale. The value, *m*, specified in the **mrt-stale-mult** statement indicates that a source must be silent for *m* * **mrt-period** in order to be declared stale.

Defaults

mrt-stale-mult 14 ;

Context

pim (sparse) statement

Examples

The following **pim** statement configures a PIM-SM component, "sm0". The MRT for this component will be examined every 30 seconds. The first sampling could take place immediately after seeing the last packet, and the second sampling at the next interval. In this example, it takes a minimum of 30 seconds and a maximum of 60 seconds to be considered stale.

```
pim yes {
    mrt-period 30;
    sparse "sm0" {
        mrt-stale-mult 2;
        interface fxp0 fxp1;
    };
};
```

See Also

mrt-spt-mult statement on page 410

mrt-period statement on page 409

pim statement on page 414

pim

Name

pim - enables or disables the PIM protocol

Syntax

```
pim ( on | off )
```

Parameters

on or **off**

Description

The **pim** statement enables or disables the PIM protocol. If the **pim** statement is not specified, PIM will not run. All interfaces that will run PIM must be multicast capable and specified within the **sparse** statement in order to determine the mode and PIM component with which the interface will be associated.

Defaults

```
pim off ;
```

Context

global

Examples

Example 1

The following **pim** statement configures a PIM-SM component, "sm0".

```
pim yes {
    sparse "sm0" {
        interface fxp0 fxp1;
    };
};
```

Example 2

Assume that GateD was running with the **pim** statement given in Example 1, and you want to disable the "sm0" component but maintain the configuration information in the configuration file. Simply change **yes** to **no**, as in the example below, and reconfigure GateD.

```
pim no {
    sparse "sm0" {
        interface fxp0 fxp1;
    };
};
```

See Also

"Protocol Independent Multicast (PIM)" on page 105 of the *Configuring GateD Guide*

priority

Name

priority - specifies a priority to be associated with the groups for which a router is volunteering to be a Candidate Rendezvous Point (CRP)

Syntax

priority *pri* **;**

Parameters

pri - an integer between 0 and 255, inclusive

Description

When using the BSR method of RP set distribution, CRPs will periodically send C-RP-Adv messages to the BSR. These messages specify a set of groups for which the CRP is volunteering to be an RP. In addition, a **priority** is associated with each set of groups and is used to decide which CRP will actually serve as the RP for a given group. The **priority** statement optionally appears inside the **crp** statement and allows you to configure the groups for which the router is volunteering to be an RP, as well as the priorities associated with these groups. The **priority** statement can appear inside of the **group** statement, in which case the **priority** is associated with a specific group range. It can also appear outside of the **group** statement, in which case the indicated **priority** applies to all group ranges inside the **group** statement that have no explicitly associated priorities.

Defaults

If the **priority** statement is omitted from the **crp** statement, then the router's priority defaults to 0 for groups for which it volunteers to be an RP.

Context

pim-sm (crp) statement

pim-sm (group) statement

Examples

Example 1

In the following example, a single PIM-SM component, "sm0", is specified. The router will act as a CRP with the CRP address being chosen from one of the two interfaces configured within the **sparse** statement. Since no group ranges are specified, the router will advertise itself as a CRP for groups in the range 224/4 (for example, all multicast addresses). Finally, it will advertise itself with a priority of 2.

```
pim yes {
    sparse "sm0" {
        crp on {
            priority 2;
```

```
        };
        interface 192.168.10.2 192.168.22.1;
    };
};
```

Example 2

In the following example, a single PIM-SM component, "sm0" is specified. The router will act as a CRP with the CRP address being chosen from one of the two interfaces configured within the **sparse** statement. The router will advertise itself as a CRP for groups 224.1.2.3 and 224.1.2.4. The advertisement for group 224.1.3.4 will contain a priority of 1, whereas the advertisement of 224.1.2.4 will contain a priority of 0.

```
pim yes {
    sparse "sm0" {
        crp on {
            priority 2;
            group {
                224.1.2.3 priority 1;
                224.1.2.4;
            } ;
        };
        interface 192.168.10.2 192.168.22.1;
    };
};
```

Example 3

Group ranges can be specified within the curly braces of the **crp** statement via the *group-address* **mask** *mask* or *group-address* **masklen** *length* statements. In addition, an ASCII network name can be specified with the **host** keyword. These features are illustrated below. The router will advertise itself as a CRP with priority 2 for all groups in the range 224.0.1.0 to 224.0.1.3, as well as group 224.0.1.1 (ntp.mcast.net).

```
pim yes {
    sparse "sm0" {
        crp on {
            priority 2;
            224.0.1.0 masklen 30;
            host ntp.mcast.net;
        };
        interface 192.168.10.2 192.168.22.1;
    };
};
```

See Also

crp statement on page 388

group statement on page 395

probe-period

Name

probe-period - specifies the number of seconds prior to the RegisterStop timer expiry to send a null Register message to the Rendezvous Point (RP)

Syntax

probe-period *secs* **;**

Parameters

secs - an integer between 0 and **reg-sup-timeout**, inclusive

Description

When PIM null Register messages are used, **probe-period** specifies the number of seconds prior to the RegisterStop timer expiry to send a null Register message to the RP. If a PIM RegisterStop message is received from the RP before the RegisterStop timer expires, the RegisterStop timer is reset, and the sending of encapsulating Register messages is delayed.

Defaults

probe-period 5 ;

Context

pim-sm (sparse) statement

Examples

The following **pim** statement configures a PIM-SM component, "sm0". If this router is a DR for a group G, and has received a RegisterStop message from the RP for source S, then it will send a probe (a null Register message) 10 seconds prior to the expiration of the Register Stop timer.

```
pim yes {
    probe-period 10;
    sparse "sm0" {
        interface fxp0 fxp1;
    };
};
```

See Also

reg-sup-timeout statement on page 420

pim statement on page 414

reg-sup-timeout

Name

reg-sup-timeout - specifies the number of seconds between receiving a PIM RegisterStop message and allowing Register messages encapsulating multicast data to again be sent

Syntax

reg-sup-timeout *secs* **;**

Parameters

secs - an integer between 1 and 3600, inclusive

Description

When a router receives a RegisterStop message from a Rendezvous Point (RP) for an (S,G) pair, it must stop sending multicast data encapsulated in Register messages for some period of time. Such a router is said to be "register-suppressed" for the (S,G) pair. This statement specifies the number of seconds for which the router remains register-suppressed. A lower value means that the RP receives more frequent bursts of encapsulated multicast data, while a higher value means a longer join latency for new receivers. (Note that if null Registers are sent **probe-period** seconds before the timeout, then Register bursts are prevented, and **reg-sup-timeout** may then be lowered to decrease join latency.)

Defaults

```
reg-sup-timeout 60 ;
```

Context

pim-sm (sparse) statement

Examples

The following **pim** statement configures a PIM-SM component, "sm0". If this router is a Designated Router for a group G, and has received a RegisterStop message from the RP for an (S,G) pair, then it will wait 45 seconds before once again encapsulating data from S to group G in Register messages.

```
pim yes {
    sparse "sm0" {
        reg-sup-timeout 45;
        interface fxp0 fxp1;
    };
};
```

See Also

probe-period statement on page 418

pim statement on page 414

rp-switch-immediate

Name

rp-switch-immediate - causes a Rendezvous Point (RP) to initiate a switch to the Shortest Path (SP) tree for (S,G) upon receipt of the first Register message encapsulating data from source S

Syntax

```
rp-switch-immediate ;
```

Parameters

none

Description

The PIM-SM protocol allows an RP or a Designated Router (DR) to switch from receiving data from a source S sent to a group G via the RP tree, to receiving data from the SP tree. Two methods are available within GateD for deciding when an SP tree switch should be initiated. One of these methods is to initiate a switch to the SP tree for an (S,G) pair upon receipt of the first Register message containing data from S addressed to group G.

If the **rp-switch-immediate** statement does not appear anywhere within the **sparse** statement, then an active RP will initiate a switch to the SP tree when the traffic rate exceeds a threshold.

Defaults

The default is to switch when the traffic rate exceeds a threshold.

Context

pim-sm (sparse) statement

Examples

The following **pim** statement configures a PIM-SM component, "sm0". If this router is an RP for G, then the first time that the router receives a Register message encapsulating data from S addressed to group G, it will initiate a switch to the SP tree rooted at S.

```
pim yes {
    sparse "sm0" {
        rp-switch-immediate;
        crp 192.168.10.2;
        interface 192.168.10.2 192.168.22.1;
    };
};
```

See Also

threshold statement on page 426

threshold-rp statement on page 430

threshold-dr statement on page 428

pim statement on page 414

dr-switch-immediate statement on page 393

sparse

Name

sparse - configures a PIM-SM component

Syntax

sparse *component_name* **{** *pimsm_statements* **} ;**

Parameters

component_name - a string to identify the PIM-SM component

pimsm_statements - refers to the PIM-SM-specific parameters described in this document that can be configured within the **sparse** statement

Description

The **sparse** statement is used to configure a PIM-SM component. At the time of this writing, GateD supports the configuration of only a single PIM-SM component.

Note: Each **sparse** statement must also contain at least one **interface** statement.

Defaults

The default is to configure no PIM-SM components at all.

Context

pim statement

Examples

The following pim statement configures a PIM-SM component, “sm0”. The component will control the running of PIM-SM over interfaces fxp0 and fxp1.

```
pim yes {
    sparse “sm0” {
        interfaces fxp0 fxp1 ;
    }
}
```

See Also

interface statement on page 403

static-rp

Name

static-rp - lets you statically configure a Rendezvous Point (RP) set

Syntax

static-rp *group-address* **masklen** *length rp-address* **;**

Parameters

group-address - a valid IPv4 group address

length - a mask length between 4 and 32, inclusive

rp-address - a valid IPv4 host address specifying the RP serving the group prefix indicated by the *group-address* and *length* parameters

Description

GateD provides two mutually exclusive methods for RP set distribution: statically configured RPs and Bootstrap Router (BSR). The **static-rp** statement lets you statically configure an RP set. Multiple **static-rp** statements can be used to add elements to the set.

Defaults

The default is to use the BSR method of RP set distribution.

Context

pim-sm statement

Examples

The following **pim** statement configures a PIM-SM component, "sm0". The RP 192.168.10.4 is configured to serve multicast group 224.40.2.1, while 192.168.10.10 is configured to serve all other multicast groups.

```
pim yes {
    sparse "sm0" {
        interfaces fxp0 fxp1 ;
        static-rp 224.40.2.1 masklen 32 192.168.10.4 ;
        static-rp 224.0.0.0 masklen 4 192.168.10.10 ;
    }
}
```

See Also

bsr statement on page 382

crp statement on page 388

threshold

Name

threshold - specifies the threshold, in bytes per second, which, when exceeded for an (S,G) pair, initiates a switch to the Shortest Path (SP) tree

Syntax

threshold *bps* **;**

Parameters

bps - an integer between 1 and 2^32 - 1 (4,294,967,295), inclusive

Description

The PIM-SM protocol allows a Rendezvous Point (RP) or a Designated Router (DR) to switch from receiving data from a source S sent to a group G via the RP tree, to receiving data from the SP tree. Two methods are available within GateD for deciding when an SP tree switch should be initiated. One of these methods involves setting a threshold, in average bytes per second. If data received from S addressed to G exceeds this threshold, then a switch is initiated. The threshold statement specifies this threshold in average bytes per second. The interval over which the average-bytes-per-second calculation is made is specified by the **mrt-period** and **mrt-spt-mult** statements.

Defaults

threshold 1000 ;

Context

pim-sm (sparse) statement

Examples

The following **pim** statement configures a PIM-SM component, "sm0". If this router is an RP for G, or a DR for the pair (S,G), and if the data from S addressed to G exceeds an average of 10 packets per second, then an SP tree switch will be initiated for the pair. The period over which the average will be calculated will be the **mrt-period** times the **mrt-spt-mult**, 60 seconds.

```
pim yes {
    mrt-period 30;
    sparse "sm0" {
        threshold 10;
        mrt-spt-mult 2;
        interface fxp0 fxp1;
    };
};
```

See Also

mrt-spt-mult statement on page 410

mrt-period statement on page 409

threshold-dr statement on page 428

threshold-rp statement on page 430

pim statement on page 414

rp-switch-immediate statement on page 422

dr-switch-immediate statement on page 393

threshold-dr

Name

threshold-dr - specifies the threshold, in bytes per second, for a Designated Router (DR), which, when exceeded for an (S,G) pair, initiates a switch to the Shortest Path (SP) tree

Syntax

threshold-dr *bps* **;**

Parameters

bps - an integer between 1 and 2^32 - 1 (4,294,967,295), inclusive

Description

The PIM-SM protocol allows a Rendezvous Point (RP) or a DR to switch from receiving data from a source S sent to a group G via the RP tree, to receiving data from the SP tree. Two methods are available within GateD for deciding when an SP tree switch should be initiated. One of these methods involves setting a threshold, in average bytes per second. If data received from S addressed to G exceeds this threshold, then a switch is initiated. The **threshold-dr** statement specifies this threshold in average bytes per second observed by a DR. The **mrt-period** and **mrt-spt-mult** statements specify the interval over which the average-bytes-per-second calculation is made.

Defaults

threshold-dr 1000 ;

Context

pim-sm (sparse) statement

Examples

The following **pim** statement configures a PIM-SM component, "sm0". If this router is an RP for G, then the data from S addressed to G must exceed an average of 10 bytes per second before an SPT switch is initiated. If this router is a DR for the pair (S,G), then the same data must exceed an average of 20 bytes per second before an SPT switch is initiated. The period over which the average will be calculated will be the **mrt-period** times the **mrt-spt-mult**, 60 seconds.

```
pim yes {
    mrt-period 30;
    sparse "sm0" {
        threshold 10;
        threshold-dr 20;
        mrt-spt-mult 2;
        interface fxp0 fxp1;
```

```
    };
};
```

See Also

threshold statement on page 426

mrt-spt-mult statement on page 410

mrt-period statement on page 409

threshold-rp statement on page 430

pim statement on page 414

rp-switch-immediate statement on page 422

dr-switch-immediate statement on page 393

threshold-rp

Name

threshold-rp - specifies the threshold, in bytes per second, for a Rendezvous Point (RP), which, when exceeded for an (S,G) pair, initiates a switch to the Shortest Path (SP) tree

Syntax

threshold-rp *bps* **;**

Parameters

bps - an integer between 1 and 2^32 - 1 (4,294,967,295), inclusive

Description

The PIM-SM protocol allows a RP or a Designated Router (DR) to switch from receiving data from a source S sent to a group G via the RP tree, to receiving data from the SP tree. Two methods are available within GateD for deciding when an SP tree switch should be initiated. One of these methods involves setting a threshold, in average bytes per second. If data received from S addressed to G exceeds this threshold, then a switch is initiated. The threshold-rp statement specifies this threshold in average bytes per second observed by an RP. The **mrt-period** and **mrt-spt-mult** statements specify the interval over which the average-bytes-per-second calculation is made.

Defaults

threshold-rp 1000 ;

Context

pim-sm (sparse) statement

Examples

The following **pim** statement configures a PIM-SM component, "sm0". If this router is an RP for G, then the data from S addressed to G must exceed an average of 100 bytes per second before an SPT switch is initiated. If this router is a DR for the pair (S,G), then the same data must exceed an average of 2000 bytes per second before an SPT switch is initiated. The period over which the average will be calculated will be the **mrt-period** times the **mrt-spt-mult**, 60 seconds.

```
pim yes {
    mrt-period 30;
    sparse "sm0" {
        threshold 2000;
        crp fxp0 {
            threshold-rp 100;
        };
        mrt-spt-mult 2;
```

```
        interface fxp0 fxp1;
    };
};
```

See Also

threshold statement on page 426

mrt-spt-mult statement on page 410

mrt-period statement on page 409

threshold-dr statement on page 428

pim statement on page 414

rp-switch-immediate statement on page 422

dr-switch-immediate statement on page 393

traceoptions

Name

traceoptions - specifies the tracing options for PIM components

Syntax

traceoptions *trace_options* **;**

Parameters

Trace options include:

packets - Trace all types of PIM packets.

assert - Trace PIM Assert packets.

bootstrap - Trace PIM-SM Bootstrap packets.

hello - Trace PIM Hello packets.

jp - Trace PIM Join/Prune packets.

register - Trace PIM-SM Register and RegisterStop packets.

graft - Trace PIM-DM Graft and GraftAck packets.

crp - Trace PIM-SM Candidate-RP-Advertisement packets.

debug - extra trace information of use mainly to developers

Description

traceoptions specifies the tracing options for all PIM components. By default, **traceoptions** is off. If you do specify **traceoptions**, then all global tracing options are inherited.

Defaults

inherited from global **traceoptions**

Context

pim statement

Examples

The following **pim** statement configures a PIM-SM component, "sm0". The component will trace Hello, Register and RegisterStop messages to the log file, /var/tmp/gated.log.

```
pim yes {
    traceoptions "/var/tmp/gated.log" replace register hello;
    sparse "sm0" {
        interface fxp0 fxp1;
    };
};
```

See Also

pim statement on page 414

wholepkt-checksum

Name

wholepkt-checksum - specifies that checksums in Register messages should be calculated over the entire encapsulated data packet, rather than just over the Register message header

Syntax

```
wholepkt-checksum ;
```

Parameters

none

Description

Previous versions of the PIM-SM specification had the checksum of Register messages calculated over the entire message, including encapsulated data. Some versions of Cisco's IOS perform this form of checksumming. The latest version of the specification states that the checksum should be calculated only over the Register message header, not any encapsulated data. This is GateD's default checksum method. Use the **wholepkt-checksum** to specify that checksums in Register messages should be calculated according to the old method.

Defaults

The default is to calculate checksums over the Register message header only.

Context

pim-sm (sparse) statement

Examples

The following **pim** statement configures a PIM-SM component, “sm0”, containing interfaces fxp0 and fxp1. When encapsulating data from local sources in Register messages and sending them to the Rendezvous Point, the message checksum will be calculated over the entire encapsulated packet, rather than just over the Register message header.

```
pim yes {
    sparse “sm0” {
        interface fxp0 fxp1 ;
        wholepkt-checksum ;
    }
}
```

See Also

pim statement on page 414