# Chapter 19
# Multicast Source Discovery Protocol (MSDP)

## connect-retry-period

### Name

**connect-retry-period -** specifies the number of seconds to wait after a failed connection attempt before trying again to establish a connection to an MSDP peer

### Syntax

**connect-retry-period** *sec* **;**

### Parameters

*sec* **-** an integer between 1 and 65535, inclusive

### Description

MSDP peering sessions are explicitly configured and the side of the session with the lower IP address initiates the session. The **connect-retry-period** statement specifies the number of seconds to wait after a failed connection attempt, before attempting another connection.

### Defaults

**connect-retry-period 30;**

### Context

**msdp** statement

### Examples

The following configuration sets the **connect-retry-period** to 15 seconds.

```
msdp yes {
    peer 192.0.2.1 192.0.2.2;
    connect-retry-period 15;
};
```

## See Also

"Multicast Source Discovery Protocol (MSDP)" on page 115 in *Configuring GateD*

`msdp` on page 515

`peer` on page 517

## default-rpf-peer

### Name

**default-rpf-peer -** specifies the MSDP peer from which Source-Active messages will be accepted, regardless of the RP address they contain

### Syntax

**default-rpf-peer** *peer_ip;*

### Parameters

*peer_ip* **-** the IPv4 address of the peer from which Source-Active messages will be accepted, regardless of the RP address they contain

### Description

This statement allows one to specify that Source-Active messages received from a certain peer should be accepted regardless of the RP address contained in the message and regardless of other topology information.

### Defaults

The default is to not configure a default peer of any kind.

### Context

**msdp** statement

### Examples

In the following configuration, a **default-rpf-peer** is configured such that Source-Active messages from 192.0.2.3 will be accepted regardless of other topology information.

```
msdp yes {
    peer 192.0.2.1 192.0.2.2;
    peer 192.0.2.1 192.0.2.3;
    default-rpf-peer 192.0.2.3;
};
```

### See Also

"Multicast Source Discovery Protocol (MSDP)" on page 115 in *Configuring GateD*

**msdp** on page 515

**peer** on page 517

**static-rpf-peer** on page 525

## **keepalive-period**

### Name

**keepalive-period -** specifies the frequency with which MSPD KeepAlive messages are transmitted to each peer

### Syntax

**keepalive-period** *sec* **;**

### Parameters

*sec* **-** an integer between 75 and 65535, inclusive

### Description

MSDP KeepAlive messages must be periodically sent to all MSDP peers. The **keepalive-period** specifies the number of seconds between subsequent KeepAlive messages.

### Defaults

**keepalive-period 75;**

### Context

**msdp** statement

### Examples

The following configuration configures the **keepalive-period** as 90 seconds.

```
msdp yes {
    peer 192.0.2.1 192.0.2.2;
    keepalive-period 90;
};
```

### See Also

"Multicast Source Discovery Protocol (MSDP)" on page 115 in *Configuring GateD*

**msdp** on page 515

**peer** on page 517

**peer-holdtime** on page 519

## **msdp**

## Name

**msdp -** enables or disables the MSDP protocol

## Syntax

**msdp ( on | off)**

## Parameters

**on, off -** "**yes**" and "**no**" are interpreted as "**on**" and "**off**" respectively.

## Description

The **msdp** statement enables or disables the MSDP protocol. If the **msdp** statement is not specified, MSDP will not run.

## Defaults

**msdp off**

## Context

global

## Examples

### Example 1

The following configuration specifies an MSDP peering session with 192.0.2.1 and 192.0.2.2 as the local and the remote end, respectively.

```
msdp yes {
    peer 192.0.2.1 192.0.2.2;
};
```

### Example 2

In the following configuration, the MSDP component is disabled.

```
msdp off {
    peer 192.0.2.1 192.0.2.2;
};
```

## See Also

"Multicast Source Discovery Protocol (MSDP)" on page 115 in *Configuring GateD*

**peer** on page 517

## msdp-draft-6-compat

### Name

`msdp-draft-6-compat -` specifies that this router should speak MSDP as specified in draft-ietf-msdp-spec-06

### Syntax

`msdp-draft-6-compat;`

### Parameters

### Description

This statement specifies that the router should speak MSDP as specified in draft-ietf-msdp-spec-06.

### Defaults

The default is to speak MSDP as specified in draft-ietf-msdp-spec-12.

### Context

`msdp` statement

### Examples

The following configuration states that this router should speak MSDP as specified in draft-ietf-msdp-spec-06.

```
msdp yes{
    msdp-draft-6-compat;
    peer 192.0.2.1 192.0.2.2;
};
```

### See Also

"Multicast Source Discovery Protocol (MSDP)" on page 115 in *Configuring GateD*

`msdp` on page 515

`peer` on page 517

## peer

### Name

`peer` - specifies the local and remote ends of an MSDP peering session

### Syntax

`peer` *local_host remote_host* `mesh` *id* `peeras` *asnum* `;`

### Parameters

*local_host*, *remote_host* - the IPv4 addresses of the local and remote ends of the desired MSDP peering session

`mesh` *id* - This subclause specifies that the indicated peer is a member of a mesh group number ID. The parameter, *id*, is an integer between 1 and 65535, inclusive.

`peeras` *asnum* - This subclause indicates the AS number of the remote end. The parameter, *asnum*, is an integer between 1 and 65535, inclusive.

### Description

Each desired MSDP peering session is explicitly configured using a `peer` statement. Peers can be configured into groups called "mesh groups", the purpose of which is to reduce flooding of SA-Advertisement messages. If the router receives an SA-Advertisement message from a peer in mesh group n, then it must not forward the message to other peers that are members of mesh group n. If the router receives an SA-Advertisement from a peer that is not in any mesh group at all, then the SA-Advertisement is forwarded to all other peers.

### Defaults

The default is for there to be no peering sessions configured. If a `peer` statement is specified without the `mesh` subclause, then peer is not configured as a member of a mesh group. If a `peer` statement is specified without the `peeras` subclause, then the AS number of the remote end cannot be known.

### Context

`msdp` statement

### Examples

In the following configuration, 192.0.2.1 and 192.0.2.2 are the peers of an MSDP peering session belonging to mesh group 1. The AS number for the remote end of the peer is 1.

```
msdp yes {
    peer 192.0.2.1 192.0.2.2 mesh 1 peeras 1 ;
};
```

### See Also

"Multicast Source Discovery Protocol (MSDP)" on page 115 in *Configuring GateD*

`msdp` on page 515

## peer-holdtime

### Name

**peer-holdtime -** specifies the number of seconds that a peer session should be maintained in the absence of a KeepAlive message from the peer

### Syntax

**peer-holdtime** *sec* **;**

### Parameters

*sec* **-** an integer between 3 and 65535, inclusive

### Description

MSDP peers periodically transmit KeepAlive messages as an indication that the peer is still functioning correctly. In the absence of other control messages, if a KeepAlive message is not received within a certain period of time, then the session with the peer will be torn down. This statement specifies the number of seconds that a peer session will be maintained in the absence of any MSDP control messages.

### Defaults

**peer-holdtime 90 ;**

### Context

**msdp** statement

### Examples

The following configuration configures the **peer-holdtime** as 15 seconds.

```
msdp yes{
    peer 192.0.2.1 192.0.2.2;
    peer-holdtime 15;
};
```

### See Also

"Multicast Source Discovery Protocol (MSDP)" on page 115 in *Configuring GateD*

**msdp** on page 515

**peer** on page 517

**keepalive-period** on page 514

## pim-filter

### Name

**pim-filter -** specifies an (S,G) pair which, if learned from the PIM-SM routing fabric, must not be announced in Source-Active or Source-Active-Request-Reply messages.

### Syntax

**pim-filter** *source_ip* **masklen** *length group_ip* **masklen** *length***;**

### Parameters

*source_ip, length* **-** a source address prefix which, when combined with the group address prefix, specifies a range of (S,G) pairs that must be filtered.

*group_ip, length* **-** a group address prefix which, when combined with the source address prefix, specifies a range of (S,G) pairs that must be filtered.

### Description

The **pim-filter** statement allows one to specify a set of (S,G) pairs which, if learned from the PIM-SM routing fabric, must not be announced in Source-Active messages. Note that (S,G) pairs covered by the filter will still be added to the MSDP SA cache, but will not appear in Source-Active or Source-Active-Request-Reply messages.

### Defaults

The default is to not filter any (S,G) pairs learned from PIM-SM.

### Context

**msdp** statement

### Examples

In the following configuration, a **pim-filter** is configured such that any (S,G) pairs learned from PIM-SM fabric that match (*, 227.1.1.1.1) will be placed in the SA Cache, but will not be announced in Source-Active messages or Source-Active-Request-Reply messages.

```
msdp yes{
    peer 192.0.2.1 192.0.2.2;
    pim-filter 0.0.0.0 masklen 0 227.1.1.1 masklen 32;
};
```

### See Also

"Multicast Source Discovery Protocol (MSDP)" on page 115 in *Configuring GateD*

**msdp** on page 515

**peer** on page 517

**sa-filter** on page 522

## sa-cache-timeout

### Name

`sa-cache-timeout -` specifies the number of seconds that an entry in the SA Cache will be maintained in the absence of a refreshing SA-Advertisement message

### Syntax

`sa-cache-timeout` *sec*`;`

### Parameters

*sec* `-` an integer between 90 and 65535, inclusive

### Description

An entry is created in the MSDP SA Cache when an (S,G) pair is learned of via an MSDP SA-Advertisement message. The `sa-cache-timeout` statement specifies the number of seconds that the entry will remain in the cache unless another SA-Advertisement for the pair is received.

### Defaults

`sa-cache-timeout 210;`

### Context

`msdp` statement

### Examples

In the following configuration, the `sa-cache-timeout` is configured as 250 seconds.

```
msdp yes{

    peer 192.0.2.1 192.0.2.2;

    sa-cache-timeout 250;

};
```

### See Also

"Multicast Source Discovery Protocol (MSDP)" on page 115 in *Configuring GateD*

`msdp` on page 515

`peer` on page 517

## sa-filter

### Name

**sa-filter -** specifies an (S,G) pair which, if received in a Source-Active message, must not be forwarded to any MSDP peers.

### Syntax

**sa-filter [ import | export]** *source_ip* **masklen** *length group_ip* **masklen** *length* **;**

### Parameters

*source_ip, length* **-** A source address prefix which, when combined with the group address prefix, specifies a range of (S,G) pairs that must be filtered. The *source_ip* is an IPv4 address in dotted-quad notation.

*group_ip, length* **-** A group address prefix which, when combined with the source address prefix, specifies a range of (S,G) pairs that must be filtered. The *group_ip* is an IPv4 class D address in dotted-quad notation.

**import -** an indication that matching (S,G) pairs must not be communicated to associated PIM-SM components

**export -** an indication that matching (S,G) pairs must not be forwarded to MSDP peers, or appear in Source-Active-Request-Reply messages

### Description

The **sa-filter** statement allows one to specify filters for (S,G) pairs received in Source-Active TLVs received from peers. If the **import** keyword is present, then associated PIM-SM components will not be informed of (S,G) pairs matching the filter. If the **export** keyword is present and the **import** keyword is absent, then matching (S,G) pairs will be used by the configured PIM-SM components, but will not be forwarded to MSDP peers in Source-Active messages or in Source-Active-Request-Reply messages.

### Defaults

The default is to not filter any (S,G) pairs.

### Context

**msdp** statement

### Examples

#### Example 1

The following example configures an SA filter such that the associated PIM-SM components will not be informed of any (S,G) pairs received in a Source-Active TLVs that match (*, 227.1.1.1).

**msdp yes {**

```
        peer 192.0.2.1 192.0.2.2;

        peer 192.0.2.1 192.0.2.3;

        sa-filter import 0.0.0.0 masklen 0 227.1.1.1 masklen 32;

    };
```

### Example 2

In the following example, an SA filter is configured such that any (S,G) pairs received in Source-Active TLVs whose source is 192.0.2.4 and group is 227.1.1.1 will be used by the associated PIM-SM component, but will not be forwarded to peers.

```
    msdp yes {

        peer 192.0.2.1 192.0.2.2;

        sa-filter export 192.0.2.4 masklen 32 227.1.1.1 masklen 32;

    };
```

## See Also

"Multicast Source Discovery Protocol (MSDP)" on page 115 in *Configuring GateD*

**msdp** on page 515

**peer** on page 517

**pim-filter** on page 520

## **sa-holddown**

### Name

**sa-holddown -** specifies the number of seconds that must elapse between subsequent SA-Advertisement messages for an (S,G) pair, in order for the SA-Advertisement messages to be forwarded

### Syntax

**sa-holddown** *sec;*

### Parameters

*sec* **-** an integer between 1 and 65535, inclusive

### Description

An entry is created in the MSDP SA Cache when an (S,G) pair is discovered via an MSDP SA-Advertisement message. The message is then forwarded to downstream MSDP peers. In order to mitigate the effects of any SA-Advertisement loops that might develop in the MSDP routing fabric, SA-Advertisements must be rate-limited per (S,G) pair. The **sa-holddown** statement specifies the minimum time that must elapse between forwarding of SA-Advertisement messages containing a given (S,G) pair.

### Defaults

**sa-holddown 30;**

### Context

**msdp** statement

### Examples

The following configuration configures the **sa-holddown** to be 60 seconds.

```
msdp yes{
    peer 192.0.2.1 192.0.2.2;
    sa-holddown 60;
};
```

### See Also

"Multicast Source Discovery Protocol (MSDP)" on page 115 in *Configuring GateD*

**msdp** on page 515

**peer** on page 517

## static-rpf-peer

### Name

`static-rpf-peer` - specifies the MSDP peer from which a Source-Active message for a certain RP will be accepted if none of the other Peer-RPF forwarding rules apply

### Syntax

`static-rpf-peer` *peer_ip rp_addr_ip*`;`

### Parameters

*peer_ip* - the IPv4 address of the peer from which Source-Active messages for RP, *rp_addr_ip*, will be accepted

*rp_addr_ip* - the IPv4 address of the RP which, if contained in a Source-Active message received from *peer_ip*, will be accepted

### Description

This statement allows one to specify that Source-Active messages containing a certain RP address should be accepted from a certain MSDP peer, regardless of other topology information.

### Defaults

The default is to not configure a default peer of any kind.

### Context

`msdp` statement

### Examples

In the following configuration, a `static-rpf-peer` statement specifies that Source-Active messages containing RP 192.0.2.4 from peer 192.0.2.3 will be accepted.

```
msdp yes {
    peer 192.0.2.1 192.0.2.2;
    peer 192.0.2.1 192.0.2.3;
    static-rpf-peer 192.0.2.3 192.0.2.4;
};
```

### See Also

"Multicast Source Discovery Protocol (MSDP)" on page 115 in *Configuring GateD*

`default-rpf-peer` on page 513

`msdp` on page 515

`peer` on page 517

## **traceoptions**

### Name

**traceoptions** - specifies the tracing options for MSDP

### Syntax

**traceoptions** *trace_options* **;**

### Parameters

**detail** - must be specified before **send** or **receive**. Normally, packets are traced in a terse form of one or two lines. When **detail** is specified, a more verbose format provides further detail on the contents of the packet.

**send** or **receive** - limit the tracing to packets sent or received, respectively. If neither is specified, both sent and received packets will be traced.

**keepalive** - Trace all MSDP KeepAlive messages.

**sa-request** - Trace all MSDP SA Request messages.

**sa-reply** - Trace all MSDP SA Reply messages.

**sa** - Trace all MSDP SA messages.

### Description

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

### Default

MSDP not traced by default

### Context

**msdp** statement

### Examples

### See Also

"Multicast Source Discovery Protocol (MSDP)" on page 115 in *Configuring GateD*

**msdp** on page 515