Chapter 16
Distance Vector Multicast Routing Protocol (DVMRP)

defaultmetric

Name

defaultmetric - specifies the default value for the interface metric

Syntax

defaultmetric *metric* **;**

Parameters

metric - numeric value, ranging from 1 to 32

Description

defaultmetric specifies the metric applied to any interface that does not explicitly have a metric set with the **dvmrp interface**, **metric** parameter.

Default

defaultmetric 1 ;

Context

dvmrp statement

Examples

This example configures two interfaces eth0 and eth1. eth0 has an explicit metric given in its **interface** statement. eth1 does not, so the default metric value of 10 will apply to eth1, but not to eth0.

```
dvmrp yes {
    interface eth0 {
        metric 5;
    };
    interface eth1;
    defaultmetric 10;
};
```

See Also

"Chapter 20 Distance Vector Multicast Routing Protocol (DVMRP)" on page 103 of *Configuring GateD*

metric on page 368

disable

Name

disable - disables DVMRP on this interface

Syntax

```
disable ;
```

Parameters

none

Description

disable explicitly disables the interfaces provided in the interface list.

Default

```
enable ;
```

Context

dvmrp interface statement

Examples

Enable all interfaces of type eth, but specifically disable eth0.

```
dvmrp on {
    interface eth;
    interface eth0 {
        disable;
    };
};
```

See Also

"Chapter 20 Distance Vector Multicast Routing Protocol (DVMRP)" on page 103 of *Configuring GateD*

enable on page 366

routing-only on page 373

dvmrp

Name

dvmrp - enables or disables the DVMRP protocol

Syntax

dvmrp off ;

dvmrp routing-only {*dvmrp parameters***} ;**

dvmrp on {*dvmrp parameters***} ;**

Parameters

dvmrp parameters

Description

If enabled, DVMRP will default to enabling all interfaces that are multicast capable. **dvmrp routing-only** specifies that DVMRP will be used only to propagate the multicast RIB, but that it will not be used for tree construction. **dvmrp routing-only** may be used to let DVMRP carry the multicast RIB, which is then used by PIM-SM. *dvmrp parameters* include all the parameters in this section.

Default

dvmrp off ;

Context

global

Examples

Example 1

This explicitly turns DVMRP off. However, because DVMRP is off by default, it is unnecessary.

```
dvmrp off;
```

Example 2

This turns DVMRP on and configures all multicast-capable interfaces.

```
dvmrp on;
```

Example 3

This turns DVMRP on and configures only two interfaces.

```
dvmrp on {
    interface eth0 eth1;
};
```

Example 4

This turns on DVMRP routing only on all multicast-capable interfaces.

```
dvmrp routing-only;
```

Example 5

This turns on DVMRP routing on only two interfaces.

```
dvmrp routing-only {
    interface eth0 eth1;
};
```

See Also

bgp on page 227

"Chapter 20 Distance Vector Multicast Routing Protocol (DVMRP)" on page 103 of *Configuring GateD*

export on page 623

import on page 605

isis on page 153

ospf on page 95

enable

Name

enable - enables DVMRP on this interface

Syntax

```
enable ;
```

Parameters

none

Description

enable explicitly enables the interfaces provided in the interface list.

Default

```
enable ;
```

Context

dvmrp interface statement

Examples

Disable all interfaces of type eth, but specifically enable eth0.

```
dvmrp on {
    interface eth {
        disable;
    };
    interface eth0 {
        enable;
    };
};
```

See Also

"Chapter 20 Distance Vector Multicast Routing Protocol (DVMRP)" on page 103 of *Configuring GateD*

disable on page 363

routing-only on page 373

interface

Name

interface - explicitly configures one or more interfaces for DVMRP

Syntax

interface *interface_list* **{** *dvmrp interface parameters* **};**

Parameters

interface_list - a list of one or more interface names, interface wildcards, or IP addresses

Description

interface configures interfaces for use by DVMRP. If no interfaces are explicitly configured, then all multicast-capable interfaces are implicitly enabled; however, if even one interface is explicitly configured, then no interfaces are implicitly enabled.

Default

```
interface all;
```

Context

dvmrp statement

Examples

All multicast-capable interfaces are implicitly enabled.

```
dvmrp on;
```

Explicitly configure two interfaces, enabling the first and disabling the second.

```
dvmrp on {
    interface eth0 {
        enable;
    };
    interface 192.168.0.1 {
        disable;
    };
};
```

See Also

"Chapter 20 Distance Vector Multicast Routing Protocol (DVMRP)" on page 103 of *Configuring GateD*

dvmrp on page 364

metric

Name

metric - explicitly sets a metric for an interface

Syntax

metric *metric* **;**

Parameters

metric - an integer value ranging from 1 to 32, inclusive

Description

metric sets the metric to be used on the interface. This value is added to all routes learned from neighbors through this interface.

Default

metric 1;

Context

dvmrp interface statement

Examples

```
dvmrp on {
    interface eth0 {
        metric 3;
    };
    interface eth1 {
        metric 5;
    };
};
```

See Also

"Chapter 20 Distance Vector Multicast Routing Protocol (DVMRP)" on page 103 of *Configuring GateD*

defaultmetric on page 361

interface on page 367

nodvmrpout

Name

nodvmrpout - tells DVMRP to just listen on an interface

Syntax

```
nodvmrpout ;
```

Parameters

none

Description

nodvmrpout disables DVMRP as a speaker on an interface, although it will listen and accept routes on the interface.

Default

off

Context

dvmrp interface statement

Examples

DVMRP will normally listen and report on all interfaces except eth1.

```
dvmrp on {
    interface eth {
        enable ;
    } ;
    interface eth2 {
        enable ;
        nodvmrpout ;
    };
};
```

See Also

"Chapter 20 Distance Vector Multicast Routing Protocol (DVMRP)" on page 103 of *Configuring GateD*

dvmrp on page 364

noretransmit

Name

noretransmit - specifies to refrain from resending DVMRP prune packets

Syntax

```
noretransmit ;
```

Parameters

none

Description

noretransmit configures GateD to not perform the exponential backoff prune retransmission. After the transmission of the first prune, no additional prunes will be transmitted on reception of data until the prune lifetime has expired.

Default

off

Context

dvmrp interface statement

Examples

```
dvmrp on {
    interface eth0;
    interface eth1 {
        noretransmit;
    };
};
```

See Also

"Chapter 20 Distance Vector Multicast Routing Protocol (DVMRP)" on page 103 of *Configuring GateD*

dvmrp on page 364

preference

Name

preference - sets the value that GateD uses for DVMRP routes in the active route selection process

Syntax

preference *pref* **;**

Parameters

pref - an integer value ranging from 0 to 255, inclusive

Description

preference sets the value that GateD uses for DVMRP routes in the active route selection process. A route from a protocol with a lower preference value is selected over a route from a protocol with a higher preference value.

Default

preference 70 ;

Context

dvmrp statement

Examples

```
dvmrp on {
    preference 10;
};
```

See Also

"Chapter 20 Distance Vector Multicast Routing Protocol (DVMRP)" on page 103 of *Configuring GateD*

dvmrp on page 364

prune-lifetime

Name

prune-lifetime - specifies the maximum default lifetime of prunes in seconds

Syntax

prune-lifetime *time*;

Parameters

time - number of seconds as an integer value ranging from 1 to 2,147,483,648, inclusive

Description

prune-lifetime configures the maximum value to be placed into a prune message. The actual lifetime value is the minimum of all the downstream prunes for the source and a randomized value that falls between one-half the prune lifetime and the prune lifetime. The value is in seconds.

Default

prune-lifetime 7200;

Context

dvmrp statement

Examples

Set the maximum prune lifetime to 1 hour.

```
dvmrp on {
    prune-lifetime 3600;
};
```

See Also

"Chapter 20 Distance Vector Multicast Routing Protocol (DVMRP)" on page 103 of *Configuring GateD*

dvmrp on page 364

routing-only

Name

routing-only - specifies to do DVMRP route exchange on this interface only

Syntax

```
routing-only ;
```

Parameters

none

Description

routing-only configures the interfaces provided in the interface list to perform only DVMRP route exchange. The DVMRP multicast delivery tree-building operations will not be performed on this interface.

Default

```
enable ;
```

Context

dvmrp statement

dvmrp interface statement

Examples

```
dvmrp on {
    interface eth0;
    interface eth1 {
        routing-only;
    };
};
```

See Also

"Chapter 20 Distance Vector Multicast Routing Protocol (DVMRP)" on page 103 of *Configuring GateD*

disable on page 363

enable on page 366

traceoptions

Name

traceoptions - specifies the tracing options for DVMRP

Syntax

traceoptions *trace_options* **;**

Parameters

packets - Trace all DVMRP packets.

probe - Trace DVMRP probe packets.

report - Trace DVMRP route report packets.

mapper - Trace DVMRP neighbor and neighbor2 packets.

prune - Trace DVMRP prune packets.

graft - Trace DVMRP graft and graft ack packets.

detail - must be specified before **send** or **recv**. 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.

Description

Packet tracing options can be modified with **detail**, **send**, or **recv**. **packets** traces all DVMRP packets. **probe** traces all DVMRP router probe packets. **report** traces all DVMRP route report packets. **mapper** traces all DVMRP neighbor and neighbor 2 packets. **prune** traces all DVMRP prune packets. **graft** traces all DVMRP graft and graft ack packets.

Default

DVMRP not traced by default

Context

dvmrp statement

Examples

Example 1

Trace all packets sent.

```
dvmrp on {
    traceoptions packets send;
};
```

Example 2

Trace all probes received.

```
dvmrp on {
    traceoptions probe recv;
};
```

Example 3

Trace all prunes and graft/graft ack packets.

```
dvmrp on {
    traceoptions prune graft;
};
```

See Also

"Chapter 20 Distance Vector Multicast Routing Protocol (DVMRP)" on page 103 of *Configuring GateD*

dvmrp on page 364

tunnel-compat

Name

tunnel-compat - configures GateD to perform old-style DVMRP tunnel encapsulation

Syntax

tunnel-compat ;

Parameters

none

Description

tunnel-compat configures GateD to perform old-style DVMRP tunnel encapsulation. In old-style tunnel encapsulation, DVMRP control messages are not ip-ip encapsulated, but merely unicasted to the tunnel endpoint.

Default

off

Context

dvmrp interface statement

Examples

Example 1

This configuration configures a DVMRP tunnel between 10.1.25.13 and 10.1.16.4.

```
# Simple draft-10 compliant tunnel
interfaces {
        define p2p local 10.1.25.13 remote 10.1.16.4 tunnel ipip;
};
dvmrp on {
        interface 10.1.16.4;
};
static {
        10.1.16.0 masklen 24 gw 10.1.25.14;
};
```

Example 2

```
# Simple mrouted compatible tunnel
interfaces {
        define p2p local 10.1.25.13 remote 10.1.16.4 tunnel ipip;
```

```
};
dvmrp on {
        interface 10.1.16.4 {
                tunnel-compat;
        };
};
static {
         10.1.16.0 masklen 24 gw 10.1.25.14;
};
```

See Also

“Chapter 20 Distance Vector Multicast Routing Protocol (DVMRP)” on page 103 of *Configuring GateD*

dvmrp on page 364