

## Chapter 19

# DVMRP Configuration Guidelines

To configure the Distance Vector Multicast Routing Protocol (DVMRP), include the `dvmrp` statement:

```
dvmrp {
  disable;
  export [ policy-names ];
  import [ policy-names ];
  interface interface-name {
    disable;
    hold-time seconds;
    metric metric;
    mode (forwarding | unicast-routing);
  }
  rib-group group-name;
  traceoptions {
    file name <replace> <size size> <files number> <no-stamp>
      <(world-readable | no-world-readable)>;
    flag flag <flag-modifier> <disable>;
  }
}
```

You can include this statement at the following hierarchy levels:

- [edit protocols]
- [edit logical-routers *logical-router-name* protocols]

For an overview of logical routers and a detailed example of logical router configuration, see the logical routers chapter of the *JUNOS Feature Guide*.

By default, DVMRP is disabled.

This chapter describes the following tasks for configuring DVMRP:

- Minimum DVMRP Configuration on page 152
- Creating Routing Tables for DVMRP Routes on page 153
- Enabling DVMRP on page 153
- Modifying the DVMRP Hold-Time Period on page 154
- Modifying the Metric Value on page 154

- Disabling DVMRP on an Interface on page 155
- Configuring DVMRP Routing Policy on page 155
- Configuring DVMRP Routing Modes on page 156
- Tracing DVMRP Protocol Traffic on page 156
- Configuration Examples on page 157

## Minimum DVMRP Configuration

---

To enable DVMRP on an interface, include at least the following statements in the configuration. All other DVMRP configuration statements are optional.

```

routing-options {
  interface-routes {
    rib-group group-name1;
  }
  rib-groups {
    group-name1 {
      import-rib [ inet.0 inet.2 ];
    }
    group-name2 {
      import-rib inet.2;
      export-rib inet.2;
    }
  }
}
protocols {
  dvmrp {
    rib-group group-name2;
    interface interface-name;
  }
}

```

You can include these statements at the following hierarchy levels:

- [edit]
- [edit logical-routers *logical-router-name*]

The port of a DVMRP router can be either a physical interface to a directly attached subnetwork or a tunnel interface to another multicast-capable area of the Multicast Backbone (MBone). All interfaces can be configured with a metric specifying cost for receiving packets on a given port. The default metric is 1.

## Creating Routing Tables for DVMRP Routes

---

DVMRP needs to access route information from the unicast routing table, `inet.0`, and from a separate routing table that is reserved for DVMRP. You need to create the routing table for DVMRP and to create groups of routing tables so that the routing protocol process imports and exports routes properly. We recommend that you use routing table `inet.2` for DVMRP routing information.

To create the necessary routing tables and routing table groups for DVMRP, include the following statements:

```
routing-options {
  interface-routes {
    rib-group group-name1;
  }
  rib-groups {
    group-name1 {
      import-rib [ inet.0 inet.2 ];
    }
    group-name2 {
      import-rib inet.2;
      export-rib inet.2;
    }
  }
}
```

You can include these statements at the following hierarchy levels:

- [edit]
- [edit logical-routers *logical-router-name*]

To associate the routing tables with DVMRP, include the `rib-group` statement at the [edit protocols `dvmrp`] hierarchy level, as described in “Enabling DVMRP” on page 153.

## Enabling DVMRP

---

To enable DVMRP on the router, include the following statements:

```
dvmrp {
  interface interface-name;
  rib-group group-name;
  traceoptions;
}
```

You can include these statements at the following hierarchy levels:

- [edit protocols]
- [edit logical-routers *logical-router-name* protocols]

The `rib-group` statement selects a routing table group. DVMRP exports routes from this group and imports routes to this group. The `rib-group` statement associates with DVMRP the routing table group that imports and exports routes into the specified routing table group. This is a group you defined with the `rib-groups` statement at the `[edit routing-options]` hierarchy level.

You must specify the interface or interfaces on which to enable DVMRP. Specify the full interface name, including the physical and logical address components. To configure all interfaces, specify the interface name `all`. For details about specifying interfaces, see the *JUNOS Network Interfaces Configuration Guide*.



**NOTE:** If you have configured Protocol Independent Multicast (PIM) on the interface, you can configure DVMRP in unicast-routing mode only. You cannot configure PIM and DVMRP in forwarding mode at the same time.

---

## Modifying the DVMRP Hold-Time Period

The DVMRP hold-time period is the amount of time that a neighbor should consider the sending router (this router) to be operative (up). The default hold-time period is 35 seconds.

To modify the hold-time value for the local router, include the `hold-time` statement:

```
hold-time seconds;
```

You can include this statement at the following hierarchy levels:

- `[edit protocols dvmrp interface interface-name]`
- `[edit logical-routers logical-router-name protocols dvmrp interface interface-name]`

The hold-time period can range from 1 through 255 seconds.

---

## Modifying the Metric Value

For each source network reported, a route metric is associated with the unicast route being reported. The metric is the sum of the interface metrics between the router originating the report and the source network. A metric of 32 marks the source network as unreachable, thus limiting the breadth of the DVMRP network and placing an upper bound on the DVMRP convergence time.

By default, a metric value of 1 is associated with each DVMRP route. To modify the metric value, include the `metric` statement:

```
metric metric;
```

You can include this statement at the following hierarchy levels:

- [edit protocols dvmrp interface *interface-name*]
- [edit logical-routers *logical-router-name* protocols dvmrp interface *interface-name*]

The metric can range from 1 through 31.

## Disabling DVMRP on an Interface

---

To disable DVMRP on an interface, include the `disable` statement:

```
disable;
```

You can include this statement at the following hierarchy levels:

- [edit protocols dvmrp interface *interface-name*]
- [edit logical-routers *logical-router-name* protocols dvmrp interface *interface-name*]

## Configuring DVMRP Routing Policy

---

All routing protocols use the routing table to store the routes that they learn and to determine which routes they should advertise in their protocol packets. Routing policy allows you to control which routes the routing protocols store in and retrieve from the routing table.

When configuring DVMRP routing policy, you can apply routing policies. To do this, include the `import` and `export` statements at the [edit protocols dvmrp] or [edit logical-routers *logical-router-name* protocols dvmrp] hierarchy level.

To apply policies to routes imported into the routing table from DVMRP, include the `import` statement, listing the names of one or more policy filters to be evaluated. If you specify more than one policy, they are evaluated in the order specified, from first to last, and the first matching policy is applied to the route. If no match is found, DVMRP shares with the routing table only those routes that were learned from DVMRP routers.

```
import [ policy-names ];
```

You can include this statement at the following hierarchy levels:

- [edit protocols dvmrp]
- [edit logical-routers *logical-router-name* protocols dvmrp]

To apply policies to routes exported from the routing table into DVMRP, include the `export` statement, listing the names of one or more policies to be evaluated. If you specify more than one policy, they are evaluated in the order specified, from first to last, and the first matching policy is applied to the route. If no match is found, the routing table exports into DVMRP only the routes that it learned from DVMRP and direct routes.

```
export [ policy-names ];
```

You can include this statement at the following hierarchy levels:

- [edit protocols dvmrp]
- [edit logical-routers *logical-router-name* protocols dvmrp]

## Configuring DVMRP Routing Modes

---

You can configure DVMRP for either forwarding or unicast routing mode. In forwarding mode, DVMRP operates its protocol normally (for example, it does the routing as well as multicast data forwarding). In unicast routing mode, you can use DVMRP for unicast routing only; to forward multicast data, enable PIM on that interface. To configure the mode, include the `mode` statement.

To configure DVMRP for multicast forwarding, include the `mode forwarding` statement:

```
mode forwarding;
```

To configure DVMRP for unicast routing, include the `mode unicast-routing` statement:

```
mode unicast-routing;
```

You can include these statements at the following hierarchy levels:

- [edit protocols dvmrp interface *interface-name*]
- [edit logical-routers *logical-router-name* protocols dvmrp interface *interface-name*]

The default mode is forwarding.

## Tracing DVMRP Protocol Traffic

---

To trace DVMRP protocol traffic, you can specify options in the global `traceoptions` statement at the [edit routing-options] or [edit logical-routers *logical-router-name* routing-options] hierarchy level. Options applied at the routing options level trace all packets, and options applied at the protocol level trace only DVMRP traffic.

You can specify DVMRP-specific options by including the `traceoptions` statement:

```
traceoptions {
  file name <replace> <size size> <files number> <no-stamp>
    <(world-readable | no-world-readable)>;
  flag flag <flag-modifier> <disable>;
}
```

You can include this statement at the following hierarchy levels:

- [edit protocols dvmrp]
- [edit logical-routers *logical-router-name* protocols dvmrp]

You can specify the following DVMRP-specific options in the DVMRP `traceoptions` statement:

- `all`—Trace everything.
- `general`—Trace general events.
- `graft`—Trace graft messages.
- `neighbor`—Trace neighbor probe messages.
- `normal`—Trace normal events.
- `packets`—Trace all DVMRP packets.
- `poison`—Trace poison-route-reverse packets.
- `policy`—Trace policy processing.
- `probe`—Trace probe packets.
- `prune`—Trace prune messages.
- `report`—Trace DVMRP route report packets.
- `route`—Trace routing information.
- `state`—Trace state transitions.
- `task`—Trace routing protocol task processing.
- `timer`—Trace routing protocol timer processing.

For general information about tracing and global tracing options, see the *JUNOS Routing Protocols Configuration Guide*.

## Configuration Examples

---

This section contains the following DVMRP configuration examples:

- Example: Tracing DVMRP Protocol Traffic on page 158
- Example: Configuring DVMRP on page 158
- Example: Configuring DVMRP to Announce Unicast Routes on page 159

**Example: Tracing DVMRP Protocol Traffic**

Trace only unusual or abnormal operations to the file `routing-log`, and trace detailed information about all DVMRP messages to the file `dvmrp-log`:

```
[edit]
routing-options {
  traceoptions {
    file routing-log;
  }
}
protocols {
  dvmrp {
    traceoptions {
      file dvmrp-log;
      flag packets;
    }
    interface so-0/0/0;
  }
}
```

**Example: Configuring DVMRP**

Configure DVMRP on the router:

```
[edit]
routing-options {
  interface-routes {
    rib-group ifrg;
  }
  rib-groups {
    ifrg {
      import-rib [ inet.0 inet.2 ];
    }
    dvmrp-rib {
      import-rib inet.2;
      export-rib inet.2;
    }
  }
}
protocols {
  sap;
  dvmrp {
    rib-group dvmrp-rib;
    traceoptions {
      flag normal;
      flag state;
    }
    interface ip-f/p/0.0 {
      hold-time 130;
    }
  }
}
```

**Example: Configuring DVMRP to Announce Unicast Routes**

In this example, DVMRP is used to announce unicast routes used solely for multicast reverse-path forwarding (RPF). Include the `mode unicast-routing` statement at the `[edit protocols dvmrp interface]` hierarchy level. Redistribute static routes by including the `static` statement at the `[edit routing-options]` hierarchy level to export the routes to all DVMRP neighbors.

```
[edit]
routing-options {
  rib inet.2 {
    static {
      route 0.0.0.0/0 discard;
    }
  }
  rib-groups {
    pim-rg {
      import-rib inet.2;
    }
    dvmrp-rg {
      export-rib inet.2;
      import-rib inet.2;
    }
  }
}
protocols {
  dvmrp {
    rib-group inet dvmrp-rg;
    export dvmrp-export;
    interface all {
      mode unicast-routing;
    }
  }
  pim {
    rib-group inet pim-rg;
    interface all;
  }
}
policy-options {
  policy-statement dvmrp-export {
    term 10 {
      from {
        protocol static;
        route-filter 0.0.0.0/0 exact;
      }
      then accept;
    }
  }
}
```

