ERICSSON

Configuring DVSR

SYSTEM ADMINISTRATOR GUIDE

19/1543-CRA 119 1170/1-V1 Uen E

Contents

1 Overview

This document provides an overview of dynamically verified static routing (DVSR), describes the tasks and commands used to configure, monitor, troubleshoot, and administer DVSR features through the SmartEdge router , and provides DVSR configuration examples.

This document applies to both the Ericsson SmartEdge® and SM family routers. However, the software that applies to the SM family of systems is a subset of the SmartEdge OS; some of the functionality described in this document may not apply to SM family routers.

For information specific to the SM family chassis, including line cards, refer to the SM family chassis documentation.

For specific information about the differences between the SmartEdge and SM family routers, refer to the Technical Product Description *SM Family of Systems* (part number 5/221 02-CRA 119 1170/1) in the **`Product Overview`** folder of this Customer Product Information library.

DVSR is a semidynamic and semistatic routing protocol used mainly for making edge routing decisions.

The SmartEdge router supports DVSR as a unique edge routing feature in addition to static routing and regular Interior Gateway Protocols (IGPs), such as Intermediate System-to-Intermediate System (IS-IS), Open Shortest Path First (OSPF), and Routing Information Protocol (RIP). DVSR is similar to normal static routing. The main difference is that the DVSR's next hop, or some other relevant host IP address, is dynamically verified by this protocol before the prefix can be injected into the local routing table. In many ISP networks, using static routing without proper next-hop checks results in blackholing of network traffic.

Static routes are often used on edge routers; however, with this additional dynamic host address verification, it can be safely used in some cases where static routing is not considered to be appropriate.

The DVSR routes can be redistributed into Border Gateway Protocol (BGP) or IGPs. A number of mechanisms can be used to redistribute specific DVSR routes; for example:

- Use the **`redistribute`** command (in BGP, IS-IS, OSPF, or RIP router configuration mode) to redistribute all the DVSR routes into a dynamic routing protocol.

- Use the **`route map`** command to either match the route type of DVSR, or to match the route tag. A route tag can be defined in a DVSR profile to cover all the DVSR routes associated with the profile, or it can be explicitly specified using the **`ip route`** command (in context configuration mode).

There are many applications where DVSR can be applied, including the following applications:

- Anycast routing

  Some ISPs use anycast routing to offer load sharing services for their Domain Name System (DNS), HTTP, File Transfer Protocol (FTP), and mail relay services. DVSR provides simple way to announce the routes of the services for the servers that are up.

- Customer access and multi-homing

  With the use of DVSR, the status of remote access connections can be verified, and static routes can be removed from the router if the remote connection is not available. It can also ease the burden on customers to run BGP on their sites for the purpose of multi-homing.

- Using dynamic routing to back up static routing

  Static routing is often used to back up dynamic routing. With DVSR, dynamic routing can be used to back up static routing; for example, DVSR routes can be temporarily set up to alleviate link congestion. When those DVSR routes fail, dynamic routing takes over, which avoids blackholing of traffic.

- Load sharing on multiple LAN circuits

  Unlike some point-to-point circuits, LAN or virtual permanent virtual circuits (PVCs) do not always offer a mechanism to learn the next-hop status, which means that using normal static routing is not appropriate in such cases; however, DVSR can be safely used.

- Suppressing summary routes in the case of IGP area partition.

  When multiple area border routers announce the same summary routes, and if there is an intra-area network partition, traffic into that area may be blackholed. With DVSR, the area border routers can detect the area partition status, and suppress the summary route announcements.

2 Configuration and Operations Tasks

To configure DVSR, perform the tasks described in the sections that follow.

2.1 Configuring a DVSR Profile

To configure a DVSR profile, perform the tasks described in Table 1. Enter all commands in DVSR profile configuration mode, unless otherwise noted.

Note: In this section, the command syntax in the task tables displays only the root command.

Table 1 Configure a DVSR Profile

Task	Root Command	Notes
Create a DVSR profile and enter DVSR profile configuration mode.	*dvsr-profile*	Enter this command in context configuration mode. If no DVSR parameters are set, the profile uses default values for the DVSR parameters. All DVSR routes must reference an existing DVSR profile.
Configure the distance value for a DVSR profile.	*distance (DVSR profiles)*	You can also define the distance value when configuring a DVSR route. In that case, the defined DVSR route distance overwrites the distance specified in the DVSR profile.
Configure the packet source IP address value for the DVSR profile.	*source-address*	—
Configure the route tag value for the DVSR profile.	*tag*	You can also define the route tag value when configuring a DVSR route. In that case, the specified DVSR route tag value overwrites the value in the DVSR profile.
Configure the TTL value for the DVSR profile.	*ttl*	—
Configure verify-set values for a DVSR profile.	*verify-set*	—

2.2 Performing DVSR Operations

To manage DVSR functions, perform the appropriate tasks described in Table 2. Enter the **`show`** command (in any mode); enter the **`clear`** and **`debug`** commands in exec mode.

Table 2 DVSR Operations Tasks

Task	**Root Command**
Clear all DVSR statistics in the DVSR summary table.	*clear dvsr statistics*
Enable the generation of DVSR debug messages.	*debug static dvsr*
Display information about all DVSR routes.	*show dvsr*

3 Configuration Examples

The sections that follow provide DVSR configuration examples.

3.1 Basic DVSR

To enable DVSR, or to announce DVSR routes, you must first define a DVSR profile. DVSR routes may have different requirement, thus more than one DVSR profile can be configured. Optionally, each DVSR route can specify parameters to overwrite profile definitions.

The following example shows one DVSR profile, and one DVSR route, using all default parameters. The DVSR profile **abc-web** is configured with a prefix of **10.10.0.0/16**, and with a next hop of **10.1.1.1**. The DVSR verify host is the next hop of the prefix, which is **10.1.1.1**. As long as the 10.1.1.1 host address is up, the prefix **10.10.0.0/16** is injected into the local routing table as a static route with a DVSR subtype:

```
[local]Redback(config)#context local

[local]Redback(config-ctx)#dvsr-profile abc-web

[local]Redback(config-dvsr)#exit

[local]Redback(config-ctx)#ip route 10.10.0.0/16 10.1.1.1 dvsr abc-web
```

3.2 DVSR in Anycast Application

Figure 1 illustrates a network topology where a DVSR-enabled edge router, **Router A**, shares a LAN with two workstations in a **webfarm**.

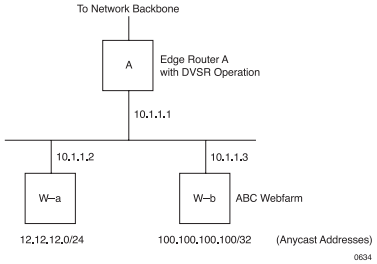

Figure 1 Basic Anycast Network Topology

The **W-a** and **W-b** workstations serve applications with IP subnets of **12.12.12.0/24** and **100.100.100.100/32** as anycast addresses. (Somewhere else, other workstations also serve the same anycast addresses.) Edge **Router**

A should announce those two anycast addresses only if workstations **W-a** and **W-b** are up. The anycast routes are redistributed into BGP.

The DVSR configuration for edge router **A** is as follows:

```
[local]Redback(config)#context local
[local]Redback(config-ctx)#dvsr-profile abc-webfarm
[local]Redback(config-dvsr)#ttl 2
[local]Redback(config-dvsr)#verify-set 30 timeout-multiplier 4 min-success 3
[local]Redback(config-dvsr)#exit
[local]Redback(config-ctx)#ip route 12.12.12.0/24 10.1.1.2 dvsr abc-webfarm
[local]Redback(config-ctx)#ip route 100.100.100.100/32 10.1.1.3 dvsr abc-webfarm
[local]Redback(config-ctx)#router bgp 65000
[local]Redback(config-bgp)#address-family ipv4 unicast
[local]Redback(config-addrfamily)#redistribute static dvsr
```

3.3 DVSR in Customer Multihoming Application

Figure 2 illustrates that an ISP has a customer network multihomed into edge router **A** and edge router **B**. The customer network has IP subnets 12.12.12.0/24, 12.12.25.0/23, and 158.10.10.0/24.

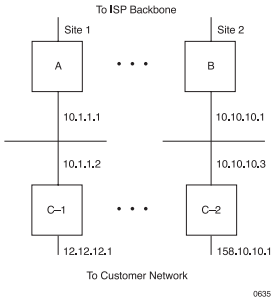

Figure 2 Basic Customer Multihoming Network Topology

Routers **C-1** and **C-2** do not run BGP, or any other dynamic routing protocol. DVSR is used in this case to inject customer routes into the backbone. If router **C-1** or **C-2** fails, or if customer internal links fail, routers **A** or **B** withdraws the DVSR routes, thus avoiding the blackholing of traffic towards the customer network.

The DVSR configuration for edge router **A** is as follows:

```
[local]Redback(config)#context local
[local]Redback(config-ctx)#dvsr-profile multi-home-c
[local]Redback(config-dvsr)#ttl 3
[local]Redback(config-dvsr)#tag 123
[local]Redback(config-dvsr)#exit
[local]Redback(config-ctx)#ip route 12.12.12.1/32 10.1.1.2
[local]Redback(config-ctx)#ip route 12.12.12.0/24 10.1.1.2 dvsr multi-home-c 12.12.12.1
[local]Redback(config-ctx)#ip route 12.12.25.0/23 10.1.1.2 dvsr multi-home-c 12.12.12.1
[local]Redback(config-ctx)#ip route 158.10.10.0/24 10.1.1.2 dvsr multi-home-c 12.12.12.1
[local]Redback(config-ctx)#router isis ip-backbone
[local]Redback(config-isis)#redistribute static dvsr
```

The DVSR configuration for edge router **B** is as follows:

```
[local]Redback(config)#context local
[local]Redback(config-ctx)#dvsr-profile multi-home-c
[local]Redback(config-dvsr)#ttl 3
[local]Redback(config-dvsr)#tag 123
[local]Redback(config-dvsr)#exit
[local]Redback(config-ctx)#ip route 158.10.10.1/32 10.10.10.3
[local]Redback(config-ctx)#ip route 12.12.12.0/24 10.10.10.3 dvsr multi-home-c 158.10.10.1
[local]Redback(config-ctx)#ip route 12.12.25.0/23 10.10.10.3 dvsr multi-home-c 158.10.10.1
[local]Redback(config-ctx)#ip route 158.10.10.0/24 10.10.10.3 dvsr multi-home-c 158.10.10.1
[local]Redback(config-ctx)#router isis ip-backbone
[local]Redback(config-isis)#redistribute static dvsr
```