Implementing Static Routes

Static routes are user-defined routes that cause packets moving between a source and a destination to take a specified path. Static routes can be important if the software cannot build a route to a particular destination. They are useful for specifying a gateway of last resort to which all unroutable packets are sent.

References for Static Routes, on page 7 provides additional conceptual information on static routes.

> **Note** Currently, only default VRF is supported.

This module describes how to implement static routes.

Restrictions for Implementing Static Routes

These restrictions apply while implementing Static Routes:

- Currently, only default VRF is supported.

- Static routing to an indirect next hop, (any prefix learnt through the RIB and may be more specific over the AIB), that is part of a local subnet requires configuring static routes in the global table indicating the egress interfaces as next hop. To avoid forward drop, configure static routes in the global table indicating the next-hop IP address to be the next hop.

- Generally, a route is learnt from the AIB in the global table and is installed in the FIB. However, this behavior will not be replicated to leaked prefixes. This could lead to inconsistencies in forwarding behavior.

Configure Static Route

Static routes are entirely user configurable and can point to a next-hop interface, next-hop IP address, or both. In the software, if an interface was specified, then the static route is installed in the Routing Information Base (RIB) if the interface is reachable. If an interface was not specified, the route is installed if the next-hop address is reachable. The only exception to this configuration is when a static route is configured with the permanent attribute, in which case it is installed in RIB regardless of reachability.

Note Currently, only default VRF is supported

.

This task explains how to configure a static route.

SUMMARY STEPS

1. **configure**
2. **router static**
3. **vrf** *vrf-name*
4. **address-family** { **ipv4** | **ipv6** } { **unicast** | **multicast** }
5. *prefix mask* [**vrf** *vrf-name*] { *ip-address* | *interface-type interface-instance* } [*distance*] [**description** *text*] [**tag** *tag*] [**permanent**]
6. Use the **commit** or **end** command.

DETAILED STEPS

Step 1 **configure**

Example:

```
RP/0/RP0/CPU0:router# configure
```

Enters mode.

Step 2 **router static**

Example:

```
RP/0/RP0/CPU0:router(config)# router static
```

Enters static route configuration mode.

Step 3 **vrf** *vrf-name*

Example:

```
RP/0/RP0/CPU0:router(config-static)# vrf vrf_A
```

(Optional) Enters VRF configuration mode.

If a VRF is not specified, the static route is configured under the default VRF.

Step 4   **address-family** { **ipv4** | **ipv6** } { **unicast** | **multicast** }

Example:

```
RP/0/RP0/CPU0:router(config-static-vrf)# address-family ipv4 unicast
```

Enters address family mode.

Step 5   *prefix mask* [**vrf** *vrf-name*] { *ip-address* | *interface-type* *interface-instance* } [*distance*] [**description** *text*] [**tag** *tag*] [**permanent**]

Example:

```
RP/0/RP0/CPU0:router(config-static-vrf-afi)# 10.0.0.0/8 172.20.16.6 110
```

Configures an administrative distance of 110.

- This example shows how to route packets for network 10.0.0.0 through to a next hop at 172.20.16.6 if dynamic information with administrative distance less than 110 is not available.

Step 6   Use the **commit** or **end** command.

commit —Saves the configuration changes and remains within the configuration session.

end —Prompts user to take one of these actions:

- **Yes** — Saves configuration changes and exits the configuration session.

- **No** —Exits the configuration session without committing the configuration changes.

- **Cancel** —Remains in the configuration session, without committing the configuration changes.

A default static route is often used in simple router topologies. In the following example, a route is configured with an administrative distance of 110.

```
configure
 router static
  address-family ipv4 unicast
  0.0.0.0/0 2.6.0.1 110
  end
```

Floating Static Routes

Floating static routes are static routes that are used to back up dynamic routes learned through configured routing protocols. A floating static route is configured with a higher administrative distance than the dynamic routing protocol it is backing up. As a result, the dynamic route learned through the routing protocol is always preferred to the floating static route. If the dynamic route learned through the routing protocol is lost, the floating static route is used in its place.

Note	By default, static routes have smaller administrative distances than dynamic routes, so static routes are preferred to dynamic routes.

Configure Floating Static Route

This task explains how to configure a floating static route.

SUMMARY STEPS

1. **configure**
2. **router static**
3. **vrf** *vrf-name*
4. **address-family** { **ipv4** | **ipv6** } { **unicast** | **multicast** }
5. *prefix mask* [**vrf** *vrf-name*] { *ip-address* | *interface-type interface-instance* } [*distance*] [**description** *text*] [**tag** *tag*] [**permanent**]
6. Use the **commit** or **end** command.

DETAILED STEPS

Step 1 **configure**

Example:

```
RP/0/RP0/CPU0:router# configure
```

Enters mode.

Step 2 **router static**

Example:

```
RP/0/RP0/CPU0:router(config)# router static
```

Enters static route configuration mode.

Step 3 **vrf** *vrf-name*

Example:

```
RP/0/RP0/CPU0:router(config-static)# vrf vrf_A
```

(Optional) Enters VRF configuration mode.

If a VRF is not specified, the static route is configured under the default VRF.

Step 4 **address-family** { **ipv4** | **ipv6** } { **unicast** | **multicast** }

Example:

```
RP/0/RP0/CPU0:router(config-static-vrf)# address-family ipv6 unicast
```

Enters address family mode.

Step 5   *prefix mask* [**vrf** *vrf-name*] { *ip-address* | *interface-type* *interface-instance* } [*distance*] [**description** *text*] [**tag** *tag*] [**permanent**]

Example:

```
RP/0/RP0/CPU0:router(config-static-vrf-afi)# 2001:0DB8::/32 2001:0DB8:3000::1 201
```

Configures an administrative distance of 201.

Step 6   Use the **commit** or **end** command.

commit —Saves the configuration changes and remains within the configuration session.

end —Prompts user to take one of these actions:

- **Yes** — Saves configuration changes and exits the configuration session.

- **No** —Exits the configuration session without committing the configuration changes.

- **Cancel** —Remains in the configuration session, without committing the configuration changes.

A floating static route is often used to provide a backup path if connectivity fails. In the following example, a route is configured with an administrative distance of 201.

```
configure
 router static
  address-family ipv6 unicast
  2001:0DB8::/32 2001:0DB8:3000::1 201
  end
```

Change Maximum Number of Allowable Static Routes

This task explains how to change the maximum number of allowable static routes.

Before you begin

Note   The number of static routes that can be configured on a router for a given address family is limited by default to 4000. The limit can be raised or lowered using the **maximum path** command. Note that if you use the **maximum path** command to reduce the configured maximum allowed number of static routes for a given address family below the number of static routes currently configured, the change is rejected. In addition, understand the following behavior: If you commit a batch of routes that would, when grouped, push the number of static routes configured above the maximum allowed, the first *n* routes in the batch are accepted. The number previously configured is accepted, and the remainder are rejected. The *n* argument is the difference between the maximum number allowed and number previously configured.

SUMMARY STEPS

1. **configure**

2. **router static**
3. **maximum path** { **ipv4** | **ipv6** } *value*
4. Use the **commit** or **end** command.

DETAILED STEPS

Step 1 **configure**

Example:

```
RP/0/RP0/CPU0:router# configure
```

Enters mode.

Step 2 **router static**

Example:

```
RP/0/RP0/CPU0:router(config)# router static
```

Enters static route configuration mode.

Step 3 **maximum path** { **ipv4** | **ipv6** } *value*

Example:

```
RP/0/RP0/CPU0:router(config-static)# maximum path ipv4 10000
```

Changes the maximum number of allowable static routes.

- Specify IPv4 or IPv6 address prefixes.

- Specify the maximum number of static routes for the given address family. The range is from 1 to 140000.

- This example sets the maximum number of static IPv4 routes to 10000.

Step 4 Use the **commit** or **end** command.

commit —Saves the configuration changes and remains within the configuration session.

end —Prompts user to take one of these actions:

- **Yes** — Saves configuration changes and exits the configuration session.

- **No** —Exits the configuration session without committing the configuration changes.

- **Cancel** —Remains in the configuration session, without committing the configuration changes.

Configuring a static route to point at interface null 0 may be used for discarding traffic to a particular prefix. For example, if it is required to discard all traffic to prefix 2001:0DB8:42:1/64, the following static route would be defined:

```
configure
```

```
router static
 address-family ipv6 unicast
 2001:0DB8:42:1::/64 null 0
 end
```

Default VRF

A static route is always associated with a VPN routing and forwarding (VRF) instance. The VRF can be the default VRF or a specified VRF. Specifying a VRF, using the **vrf** *vrf-name* command, allows you to enter VRF configuration mode for a specific VRF where you can configure a static route. If a VRF is not specified, a default VRF static route is configured.

Note
An IPv4 or IPv6 static VRF route is the same as a static route configured for the default VRF. The IPv4 and IPV6 address families are supported in each VRF.

References for Static Routes

The following topics provide additional conceptual information on static routes:

Static Route Functional Overview

Networking devices forward packets using route information that is either manually configured or dynamically learned using a routing protocol. Static routes are manually configured and define an explicit path between two networking devices. Unlike a dynamic routing protocol, static routes are not automatically updated and must be manually reconfigured if the network topology changes. The benefits of using static routes include security and resource efficiency. Static routes use less bandwidth than dynamic routing protocols, and no CPU cycles are used to calculate and communicate routes. The main disadvantage to using static routes is the lack of automatic reconfiguration if the network topology changes.

Static routes can be redistributed into dynamic routing protocols, but routes generated by dynamic routing protocols cannot be redistributed into the static routing table. No algorithm exists to prevent the configuration of routing loops that use static routes.

Static routes are useful for smaller networks with only one path to an outside network and to provide security for a larger network for certain types of traffic or links to other networks that need more control. In general, most networks use dynamic routing protocols to communicate between networking devices but may have one or two static routes configured for special cases.

Default Administrative Distance

Static routes have a default administrative distance of 1. A low number indicates a preferred route. By default, static routes are preferred to routes learned by routing protocols. Therefore, you can configure an administrative distance with a static route if you want the static route to be overridden by dynamic routes. For example, you could have routes installed by the Open Shortest Path First (OSPF) protocol with an administrative distance of 120. To have a static route that would be overridden by an OSPF dynamic route, specify an administrative distance greater than 120.

Directly Connected Routes

The routing table considers the static routes that point to an interface as "directly connected." Directly connected networks are advertised by IGP routing protocols if a corresponding **interface** command is contained under the router configuration stanza of that protocol.

In directly attached static routes, only the output interface is specified. The destination is assumed to be directly attached to this interface, so the packet destination is used as the next hop address. The following example shows how to specify that all destinations with address prefix 2001:0DB8::/32 are directly reachable through interface TenGigE 0/0/0/0:

```
RP/0/RP0/CPU0:router(config)# router static
RP/0/RP0/CPU0:router(config-static)# address-family ipv6 unicast
RP/0/RP0/CPU0:router(config-static-afi)# 2001:0DB8::/32 TenGigE 0/0/0/0
```

Directly attached static routes are candidates for insertion in the routing table only if they refer to a valid interface; that is, an interface that is both up and has IPv4 or IPv6 enabled on it.

Floating Static Routes

Floating static routes are static routes that are used to back up dynamic routes learned through configured routing protocols. A floating static route is configured with a higher administrative distance than the dynamic routing protocol it is backing up. As a result, the dynamic route learned through the routing protocol is always preferred to the floating static route. If the dynamic route learned through the routing protocol is lost, the floating static route is used in its place.

Note By default, static routes have smaller administrative distances than dynamic routes, so static routes are preferred to dynamic routes.

Fully Specified Static Routes

In a fully specified static route, both the output interface and next hop are specified. This form of static route is used when the output interface is multiaccess and it is necessary to explicitly identify the next hop. The

next hop must be directly attached to the specified output interface. The following example shows a definition of a fully specified static route:

```
RP/0/RP0/CPU0:router(config)# router static
RP/0/RP0/CPU0:router(config-static)# address-family ipv6 unicast
RP/0/RP0/CPU0:router(config-static-afi)# 2001:0DB8::/32 TenGigE 0/0/0/0 2001:0DB8:3000::1
```

A fully specified route is valid (that is, a candidate for insertion into the routing table) when the specified interface, IPv4 or IPv6, is enabled and up.

Recursive Static Routes

In a recursive static route, only the next hop is specified. The output interface is derived from the next hop. The following example shows how to specify that all destinations with address prefix 2001:0DB8::/32 are reachable through the host with address 2001:0DB8:3000::1:

```
RP/0/RP0/CPU0:router(config)# router static
RP/0/RP0/CPU0:router(config-static)# address-family ipv6 unicast
RP/0/RP0/CPU0:router(config-static-afi)# 2001:0DB8::/32 2001:0DB8:3000::1
```

A recursive static route is valid (that is, it is a candidate for insertion in the routing table) only when the specified next hop resolves, either directly or indirectly, to a valid output interface, provided the route does not self-recurse, and the recursion depth does not exceed the maximum IPv6 forwarding recursion depth.

A route self-recurses if it is itself used to resolve its own next hop. If a static route becomes self-recursive, RIB sends a notification to static routes to withdraw the recursive route.

Assuming a BGP route 2001:0DB8:3000::0/16 with next hop of 2001:0DB8::0104, the following static route would not be inserted into the IPv6 RIB because the BGP route next hop resolves through the static route and the static route resolves through the BGP route making it self-recursive:

```
RP/0/RP0/CPU0:router(config)# router static
RP/0/RP0/CPU0:router(config-static)# address-family ipv6 unicast
RP/0/RP0/CPU0:router(config-static-afi)# 001:0DB8::/32 2001:0DB8:3000::1
```

This static route is not inserted into the IPv6 routing table because it is self-recursive. The next hop of the static route, 2001:0DB8:3000:1, resolves through the BGP route 2001:0DB8:3000:0/16, which is itself a recursive route (that is, it only specifies a next hop). The next hop of the BGP route, 2001:0DB8::0104, resolves through the static route. Therefore, the static route would be used to resolve its own next hop.

It is not normally useful to manually configure a self-recursive static route, although it is not prohibited. However, a recursive static route that has been inserted in the routing table may become self-recursive as a result of some transient change in the network learned through a dynamic routing protocol. If this occurs, the fact that the static route has become self-recursive will be detected and it will be removed from the routing table, although not from the configuration. A subsequent network change may cause the static route to no longer be self-recursive, in which case it is re-inserted in the routing table.

Dynamic ECMP

The dynamic ECMP (equal-cost multi-path) for IGP (Interior Gateway Protocol) prefixes feature supports dynamic selection of ECMP paths ranging from 1 to 64 IGP paths. ECMP for non-recursive prefixes is dynamic. This feature enables loadbalancing support in hardware among egress links.

The dynamic ECMP (equal-cost multi-path) for IGP (Interior Gateway Protocol) prefixes feature supports dynamic selection of ECMP paths ranging from 1 to 64 IGP paths. ECMP for non-recursive prefixes is dynamic.

This feature enables loadbalancing support in hardware among egress links.

IPv4 Multicast Static Routes

IP multicast static routes (mroutes) allow you to have multicast paths diverge from the unicast paths. When using Protocol Independent Multicast (PIM), the router expects to receive packets on the same interface where it sends unicast packets back to the source. This expectation is beneficial if your multicast and unicast topologies are congruent. However, you might want unicast packets to take one path and multicast packets to take another.

The most common reason for using separate unicast and multicast paths is tunneling. When a path between a source and a destination does not support multicast routing, configuring two routers with a GRE tunnel between them is the solution. In the figure below, each unicast router (UR) supports unicast packets only; each multicast router (MR) supports multicast packets.

Figure 1: Tunnel for Multicast Packets
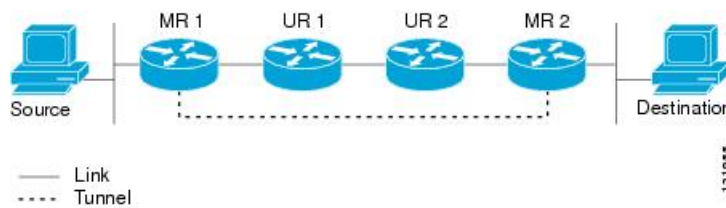
In the figure, the source delivers multicast packets to destination by using MR 1 and MR 2. MR 2 accepts the multicast packet only if it predicts it can reach source over the tunnel. If this situation is true, when the destination sends unicast packets to the source, MR 2 sends them over the tunnel. The check that MR2 can reach the source over the tunnel is a Reverse Path Forwarding (RPF) check, and the static mroute allows the check to be successful when the interface, on which the multicast packet arrives, is not the unicast path back to the source. Sending the packet over the tunnel could be slower than natively sending it through UR 2, UR 1, and MR 1.

A multicast static route allows you to use the configuration in the above figure by configuring a static multicast source. The system uses the configuration information instead of the unicast routing table to route the traffic. Therefore, multicast packets can use the tunnel without having the unicast packets use the tunnel. Static mroutes are local to the router they are configured on and not advertised or redistributed in any way to any other router.

Configure Multicast Static Routes

The following example shows how to configure multiple static routes in IPv4 and IPv6 address family configuration modes:

```
/* Enables a static routing process */
Router(config)# router static

/* Configures the IPv4 address-family for the unicast topology with a destination prefix.
*/
Router(config-static)# address-family ipv4 unicast
Router(config-static-afi)# 10.1.1.0/24 198.51.100.1
```

```
Router(config-static-afi)# 223.255.254.254/32 203.0.113.1
Router(config-static-afi)# exit

/* Configures the IPv4 address-family for the multicast topology with a destination prefix.
 */
Router(config-static)# address-family ipv4 multicast
Router(config-static-afi)# 198.51.100.20/32 209.165.201.0
Router(config-static-afi)# 192.0.2.10/32 209.165.201.0
Router(config-static-afi)# exit

/* Enable the address family IPv4 and IPv6 multicast on the next hop interface. */
Router(config)# interface TenGigE 0/0/0/12
Router(config-if)# address-family ipv4 multicast
Router(config-if)# address-family ipv6 multicast
```

Running Configuration

```
router static
 address-family ipv4 unicast
  10.1.1.0/24 198.51.100.1
  223.255.254.254/32 203.0.113.1
  !
address-family ipv4 multicast
  198.51.100.20/32 209.165.201.0
  192.0.2.10/32 209.165.201.0
  !
interface TenGigE 0/0/0/12
  address-family ipv4 multicast
  address-family ipv6 multicast
```

Verification

Verify the IPv4 multicast routes.

show route ipv4 multicast

```
Codes: C - connected, S - static, R - RIP, B - BGP, (>) - Diversion path
      O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - ISIS, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, su - IS-IS summary null, * - candidate default
       U - per-user static route, o - ODR, L - local, G  - DAGR, l - LISP
       A - access/subscriber, a - Application route
       M - mobile route, r - RPL, t - Traffic Engineering, (!) - FRR Backup path

Gateway of last resort is 10.1.1.20 to network 0.0.0.0

i*L1 0.0.0.0/0 [115/10] via 10.1.1.20, 00:41:12, TenGigE0/0/0/6
C    10.1.1.0/24 is directly connected, 00:41:12, TenGigE0/0/0/0
L    10.1.1.10/32 is directly connected, 00:41:12, TenGigE0/0/0/0
S    172.16.2.10/32 [1/0] via 198.51.100.20, 00:41:12
i L1 172.16.3.1/32 [115/20] via 198.51.100.20, 00:41:12, TenGigE0/0/0/12
i L1 192.0.2.1/24 [115/20] via 198.51.100.20, 00:41:12, TenGigE0/0/0/1
```

Configure Multicast Static Routes