Implementing Cisco Express Forwarding

Implementing Cisco Express Forwarding

Cisco Express Forwarding (CEF) is an advanced, Layer 3 IP switching technology. CEF optimizes network performance and scalability for networks with large and dynamic traffic patterns, such as the Internet, on networks characterized by intensive web-based applications, or interactive sessions. CEF is an inherent feature and the users need not perform any configuration to enable it. If required, the users can change the default route purge delay and static routes.

Components

Cisco IOS XR software CEF always operates in CEF mode with two distinct components:

- Forwarding Information Base (FIB) database: The protocol-dependent FIB process maintains the forwarding tables for IPv4 and IPv6 unicast in the route processor . The FIB on each node processes Routing Information Base (RIB) updates, performing route resolution and maintaining FIB tables independently in the route processor . FIB tables on each node can be slightly different.
- Adjacency table—a protocol-independent adjacency information base (AIB)

  Adjacency FIB entries are maintained only on a local node, and adjacency entries linked to FIB entries could be different.

CEF is a primary IP packet-forwarding database for Cisco IOS XR software. CEF is responsible for the following functions:

- Software switching path
- Maintaining forwarding table and adjacency tables (which are maintained by the AIB) for software and hardware forwarding engines

The following features are supported for CEF on Cisco IOS XR software:

- Bundle interface support
- Multipath support
- Route consistency
- High availability features such as packaging, restartability, and Out of Resource (OOR) handling
- OSPFv2 SPF prefix prioritization
- BGP attributes download

CEF Benefits

- Improved performance—CEF is less CPU-intensive than fast-switching route caching. More CPU processing power can be dedicated to Layer 3 services such as quality of service (QoS) and encryption.
- Scalability—CEF offers full switching capacity at each line card.
- Resilience—CEF offers an unprecedented level of switching consistency and stability in large dynamic networks. In dynamic networks, fast-switched cache entries are frequently invalidated due to routing changes. These changes can cause traffic to be process switched using the routing table, rather than fast switched using the route cache. Because the Forwarding Information Base (FIB) lookup table contains all known routes that exist in the routing table, it eliminates route cache maintenance and the fast-switch or process-switch forwarding scenario. CEF can switch traffic more efficiently than typical demand caching schemes.

The following CEF forwarding tables are maintained in Cisco IOS XR software:

- IPv4 CEF database—Stores IPv4 Unicast routes for forwarding IPv4 unicast packets
- IPv6 CEF database—Stores IPv6 Unicast routes for forwarding IPv6 unicast packets
- MPLS LFD database—Stores MPLS Label table for forwarding MPLS packets

Verifying CEF

To view the details of the IPv4 or IPv6 CEF tables, use the following commands:

- show cef {*ipv4 address*| *ipv6 address*} hardware egress

Displays the IPv4 or IPv6 CEF table. The next hop and forwarding interface are displayed for each prefix. The output of the **show cef** command varies by location.

```
Router# show cef 37.37.37.37/32 hardware egress
37.37.37.37/32, version 46, internal 0x1000001 0x0 (ptr 0x8b0477f8) [1], 0x0 (0x0), 0x0
(0x0)
local adjacency 1.0.0.2
 Prefix Len 32, traffic index 0, precedence n/a, priority 3
   via 1.0.0.2/32, TenGigE0/0/0/0, 5 dependencies, weight 0, class 0 [flags 0x0]
    path-idx 0 NHID 0x0 [0x8b43bb30 0x0]
    next hop 1.0.0.2/32
    local adjacency
 Show-data Print at RPLC
 LEAF - HAL pd context :
 sub-type : IPV4, ecd_marked:0, has_collapsed_ldi:0
 collapse_bwalk_required:0, ecdv2_marked:0
Leaf H/W Result:
```

```
  VRF            Net Address                    Mask                  Status
    0            37.37.37.37         255.255.255.255              Programmed

TX H/W Result for NP:0 (index: 0x18739 (BE)):
 Next Hop Data
 Next Hop Valid:         YES
 Next Hop Index:         100153
 --More-- RP/0/RP0/CPU0:Aug  6 16:38:59.302 : vic_1[180]: %PLATFORM-VIC-4-SIGNAL : Interface
 HundredGigE0/0/1/3, Detected Signal failure
RP/0/RP0/CPU0:Aug  6 16:38:59.312 : ifmgr[172]: %PKT_INFRA-LINK-3-UPDOWN : Interface
HundredGigE0/0/1/3, changed state to Up
 Egress Next Hop IF:     100045
 Hw Next Hop Intf:       6
 HW Port:                1
 Next Hop Flags:         COMPLETE-->This indicates that the router can forward traffic.
 Next Hop MAC:           4055.395f.46b7

NHINDEX H/W Result for NP:0 (index: 0 (BE)):
NhIndex is NOT required on this platform

NHINDEX STATS: pkts 0, bytes 0 (all NPs combined, no stats)

RX H/W Result on NP:0 [Adj ptr:0x40 (BE)]:
Rx-Adj is NOT required on this platform
```

- show cef {ipv4| ipv6} summary

Displays a summary of the IPv4 or IPv6 CEF table.

```
Router#show cef ipv4 summary
Router ID is 0.0.0.0
IP CEF with switching (Table Version 0) for node0_RP0_CPU0
  Load balancing: L4
  Tableid 0xe0000000 (0x8a2d7380), Vrfid 0x60000000, Vrid 0x20000000, Flags 0x1019
  Vrfname default, Refcount 88
  39 routes, 0 protected, 0 reresolve, 0 unresolved (0 old, 0 new), 9048 bytes
  11 rib, 0 lsd, 17:0 aib, 0 internal, 8 interface, 4 special, 1 default routes
  39 load sharing elements, 32176 bytes, 2 references
  2 shared load sharing elements, 1600 bytes
  37 exclusive load sharing elements, 30576 bytes
  0 route delete cache elements
  24 local route bufs received, 0 remote route bufs received,  0 mix bufs received
  11 local routes, 0 remote routes
  26 total local route updates processed
  0 total remote route updates processed
  0 pkts pre-routed to cust card
  0 pkts pre-routed to rp card
  0 pkts received from core card
  0 CEF route update drops, 2 revisions of existing leaves
  0 CEF route update drops due to version mis-match
  Resolution Timer: 15s
  0 prefixes modified in place
  0 deleted stale prefixes
  0 prefixes with label imposition, 0 prefixes with label information
  0 LISP EID prefixes, 0 merged, via 0 rlocs
 20 next hops
  0 incomplete next hops
 0 PD backwalks on LDIs with backup path
```

- show cef { ipv4 address| ipv6 address } detail

Displays the details of the IPv4 or IPv6 CEF table.

```
Router#show cef 1.0.0.2 detail
1.0.0.2/32, version 0, internal 0x1020001 0x0 (ptr 0x8a35d1a8) [1], 0x0 (0x0), 0x0 (0x0)
 Updated Aug  3 11:00:37.829
 local adjacency 1.0.0.2
 Prefix Len 32, traffic index 0, Adjacency-prefix, precedence n/a, priority 15
  gateway array (0x8a08af70) reference count 1, flags 0x4000, source aib-hi (3), 0 backups
                [1 type 3 flags 0x48401 (0x8a1d9e18) ext 0x0 (0x0)]
  LW-LDI[type=0, refc=0, ptr=0x0, sh-ldi=0x0]
  gateway array update type-time 1 Aug  3 11:00:37.829
 LDI Update time Aug  3 11:00:37.829
   via 1.0.0.2/32, TenGigE0/0/0/0, 3 dependencies, weight 0, class 0 [flags 0x0]
    path-idx 0 NHID 0x0 [0x8b7df0a0 0x0]
    next hop 1.0.0.2/32
    local adjacency
    Load distribution: 0 (refcount 1)
    Hash  OK  Interface                 Address
    0     Y   TenGigE0/0/0/0            1.0.0.2
```

- show adjacency detail

Displays detailed adjacency information, including Layer 2 information for each interface. The output of the show adjacency command varies by location.

```
Router#show adjacency detail
------------------------------------------------------------------------------
0/RP0/CPU0
------------------------------------------------------------------------------
Interface                  Address                 Version  Refcount Protocol
Te0/0/0/29                  (interface)                 30        1(    0)
                           (interface entry)
                           mtu: 1500, flags 1 4
                           0 packets, 0 bytes
Te0/0/0/77                  (interface)                 78        1(    0)
                           (interface entry)
                           mtu: 1500, flags 1 4
                           0 packets, 0 bytes
Te0/0/0/20                  (interface)                 21        1(    0)
                           (interface entry)
                           mtu: 1500, flags 1 4
                           0 packets, 0 bytes
Te0/0/0/33                  (interface)                 34        1(    0)
                           (interface entry)
                           mtu: 1500, flags 1 4
                           0 packets, 0 bytes
Hu0/0/1/3                   (interface)                 84        1(    0)
                           (interface entry)
                           mtu: 1500, flags 1 4
                           88 packets, 3696 bytes
Te0/0/0/24                  (interface)                 25        1(    0)
```

Unicast Reverse Path Forwarding

Configuration of Unicast IPv4 and IPv6 Reverse Path Forwarding (uRPF) enables a router to verify the reachability of the source address in packets being forwarded. Configuring uRPF, both strict and loose modes, helps to mitigate problems caused by the introduction of spoofed IP source addresses into a network. Configuration of uRPF discards IP packets that lack a verifiable IP source address after a reverse lookup in the CEF table.

When **strict uRPF** is enabled, the source address of the packet is checked in the FIB. If the packet is received on the same interface that would be used to forward the traffic to the source of the packet, the packet passes

the check and is further processed. Otherwise, the packet is dropped. Configure strict uRPF only where there is natural or configured symmetry. Internal interfaces are likely to have a routing asymmetry, that is, multiple routes to the source of a packet. Therefore, you should not implement strict uRPF on interfaces that are internal to the network.

Implementation of strict mode uRPF requires maintenance of a uRPF interfaces list for the prefixes. The list contains only the interfaces configured with strict mode uRPF. The interfaces are provided by the prefix path. The uRPF interface list is shared among the prefixes wherever possible.

When **loose uRPF** is enabled, the source address of the packet is checked in the FIB. If the source address exists and matches a valid forwarding entry, the packet passes the check and is further processed. Otherwise, the packet is dropped.

Note The behavior of strict uRPF varies slightly on the basis of platforms, the number of recursion levels, and the number of paths in Equal-Cost Multipath (ECMP) scenarios. A platform may switch to loose uRPF check for some or all prefixes, even though strict uRPF is configured. For example, if ECMP Path is eight or more, strict mode is converted to loose mode.

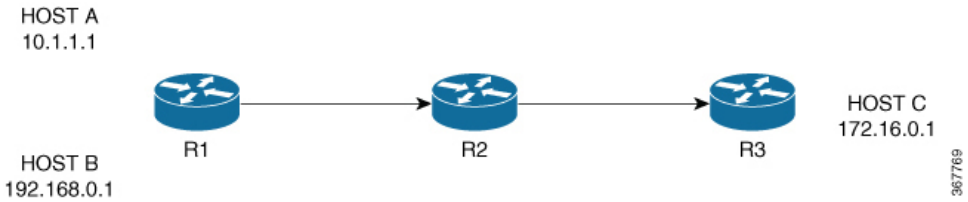
Loose and strict uRPF supports two options: **allow self-ping** and **allow default**. The **allow self-ping** option allows the source of the packet to ping itself. The **allow default** option allows the lookup result to match a default routing entry. When the **allow default** option is enabled with the strict mode of the uRPF, the packet is processed further only if it arrives through the default interface.

Restrictions

Consider the following restrictions when you configure uRPF:

- Global configuration followed by cold reload is required to enable or disable uRPF on the router.
- Configuration of uRPF per interface enables or disables uRPF mode in the hardware of the corresponding interface.
- Configuration of uRPF reduces the route scale to half. Half of the routing table is used for destination lookup and the other half is used for source lookup of received packets.
- Unicast RPF allows packets with 0.0.0.0 source addresses and 255.255.255.255 destination addresses to pass so that Bootstrap Protocol and Dynamic Host Configuration Protocol (DHCP) functions properly.
- Unicast RPF allows packets whose destination IP address is not a unicast address.
- The behavior of strict uRPF varies slightly on the basis of platforms, the number of recursion levels, and the number of paths in Equal-Cost Multipath (ECMP) scenarios. A platform may switch to loose uRPF check for some or all prefixes, even though strict uRPF is configured.
- The **allow self-ping** option is the default option in uRPF configuration for both strict and loose mode. You cannot disable the **allow-self-ping** option. The **allow-default** option needs to be configured specifically per interface.

Figure 1: uRPF Topology



In Figure 1, uRPF is enabled on Router R2 and R2 has the following FIB table entries:

- 10.1.1.0/24 [110/3] via 10.25.24.1, 2d08h, HundredGigE0/0/0/24
- 12.1.1.0/24 [110/3] via 10.25.24.1, 2d08h, HundredGigE0/0/0/25
- 14.0.5.0/24 [110/3] via 20.0.0.2, 2d08h, HundredGigE0/0/0/1

R2 allows the packet from Host 1 to be routed because Host 1 subnet is available in R2's FIB table. However, R2 does not allow Host 3 to be routed because Host 3 subnet is not available in R2's FIB table. If strict uRPF is enabled on R2, then the source address 10.1.1.1/24 should be reachable through the same interface from which it is received. If loose uRPF is enabled on R2, then it is not mandatory that the source address 10.1.1.1/24 be reachable through the same interface from which it is received. The only criteria for a packet to be forwarded is that the host address should be present in R2's FIB table.

Configure Unicast Reverse Path Forwarding

Configuring Unicast Reverse Path Forwarding (uRPF) enables a router to verify the reachability of the source address of packets being forwarded. If the source IP address is not valid, the packet is discarded. This capability can limit the appearance of spoofed addresses in a network.

To configure uRPF on a hardware module, use the following steps:

1. Configure a hardware module in the global configuration mode and enable uRPF.
2. Reboot the router.
3. Configure an interface.
4. Configure IPv4 or IPv6 uRPF through strict or loose mode under the interface.

Configuration

Use the following configuration to configure uRPF in strict mode:

```
/* Configure a hardware module in the global configuration mode and enable uRPF. */
Router# configure
Router(config)# hw-module urpf enable
Mon Sep 17 09:34:00.945 UTC
In order to activate/deactivate urpf, you must manually reboot the box.
Router(config)# commit


/* Reboot the router manually. */

/* Configure an interface. */
```

```
Router(config)# interface BVI1
Router(config-ipv6-acl)# description BVI INTERFACE
Router(config-ipv6-acl)# commit

/* Configure IPv4 or IPv6 uRPF through strict or loose mode under the interface. */
Router(config)# ipv4 address 11.1.1.1 255.255.255.0
Router(config-pifib-policer-global)# ipv4 verify unicast source reachable-via rx
Router(config-pifib-policer-global)# commit
```

Use the following configuration to configure uRPF in loose mode:

```
/* Configure a hardware module in the global configuration mode and enable uRPF. */
Router# configure
Router(config)# hw-module urpf enable
Mon Sep 17 09:34:00.945 UTC
In order to activate/deactivate urpf, you must manually reboot the box.
Router(config)# commit


/* Reboot the router manually. */

/* Configure an interface. */
Router(config)# interface BVI1
Router(config-ipv6-acl)# description BVI INTERFACE
Router(config-ipv6-acl)# commit

/* Configure IPv4 or IPv6 uRPF through strict or loose mode under the interface. */
Router(config)# ipv4 address 11.1.1.1 255.255.255.0
Router(config-pifib-policer-global)# ipv4 verify unicast source reachable-via any
Router(config-pifib-policer-global)# commit
```

Verification

Use the following command to check the uRPF status:

```
Router#show prm server  profile-status software
Software PROFILE STATUS VARIABLES
**************************************
BGP-3107 LB             : 0
LB Seed                 : 0x65c5208d
LB Seed Cfg             : NO
LB Sub-Sel-Offset Cfg   : NO
LB SubSel               : Hash Field B1
LB Offset               : 13
SR-PE Mode              : 0
Current URPF            : YES
Configured URPF         : YES
Current HW Profile      : SP-Profile
Configured HW Profile   : SP-Profile
**************************************
```

Per-Flow Load Balancing

The system inherently supports the 5-tuple hash algorithm. Load balancing describes the functionality in a router that distributes packets across multiple links based on Layer 3 (network layer) and Layer 4 (transport layer) routing information. If the router discovers multiple paths to a destination, the routing table is updated with multiple entries for that destination.

Per-flow load balancing performs these functions:

- Incoming data traffic is evenly distributed over multiple equal-cost connections.
- Incoming data Data traffic is evenly distributed over multiple equal-cost connections member links within a bundle interface.
- Layer 2 bundle and Layer 3 (network layer) load balancing decisions are taken on IPv4, IPv6, and MPLS flows which are supported for the 5-tuple hash algorithm.
- A 5-tuple hash algorithm provides more granular load balancing and used for load balancing over multiple equal-cost Layer 3 (network layer) paths. The Layer 3 (network layer) path is on a physical interface or on a bundle interface. In addition, load balancing over member links can occur within a Layer 2 bundle interface.
- The 5-tuple load-balance hash calculation contains:
  - Source IP address
  - Destination IP address
  - Router ID
  - Source port
  - Destination port

Configuring Static Route

Routers forward packets using either route information from route table entries that you manually configure or the route information that is calculated using dynamic routing algorithms. Static routes, which define explicit paths between two routers, cannot be automatically updated; you must manually reconfigure static routes when network changes occur. Static routes use less bandwidth than dynamic routes. Use static routes where network traffic is predictable and where the network design is simple. You should not use static routes in large, constantly changing networks because static routes cannot react to network changes. Most networks use dynamic routes to communicate between routers but might have one or two static routes configured for special cases. Static routes are also useful for specifying a gateway of last resort (a default router to which all unroutable packets are sent).

Configuration Example

Create a static route between Router A and B over a TenGigE interface. The destination IP address is 37.37.37.37/32 and the next hop address is 1.0.0.2.


```
Router(config)# router static address-family ipv4 unicast
Router(config-static-afi)# 37.37.37.37/32 tengige 0/0/0/0 1.0.0.2
Router(config-static-afi)# commit
```

Running Configuration

```
Router#show running-config router static address-family ipv4 unicast
router static
 address-family ipv4 unicast
```

```
 37.37.37.37/32 tengigE 0/0/0/0 1.0.0.2
 !
!
```

Verification

Verify that the Next Hop Flags fields indicate COMPLETE for accurate functioning of the configuration.

```
Router#show cef 37.37.37.37/32 hardware egress
37.37.37.37/32, version 46, internal 0x1000001 0x0 (ptr 0x8b0477f8) [1], 0x0 (0x0), 0x0
(0x0)
local adjacency 1.0.0.2
 Prefix Len 32, traffic index 0, precedence n/a, priority 3
   via 1.0.0.2/32, TenGigE0/0/0/0, 5 dependencies, weight 0, class 0 [flags 0x0]
    path-idx 0 NHID 0x0 [0x8b43bb30 0x0]
    next hop 1.0.0.2/32
    local adjacency
 Show-data Print at RPLC
 LEAF - HAL pd context :
 sub-type : IPV4, ecd_marked:0, has_collapsed_ldi:0
 collapse_bwalk_required:0, ecdv2_marked:0
Leaf H/W Result:
  VRF            Net Address                    Mask                  Status
    0            37.37.37.37         255.255.255.255              Programmed

TX H/W Result for NP:0 (index: 0x18739 (BE)):
 Next Hop Data
 Next Hop Valid:         YES
 Next Hop Index:         100153
 --More--  vic_1[180]: %PLATFORM-VIC-4-SIGNAL : Interface HundredGigE0/0/1/3, Detected
Signal failure
: ifmgr[172]: %PKT_INFRA-LINK-3-UPDOWN : Interface HundredGigE0/0/1/3, changed state to Up

 Egress Next Hop IF:     100045
 Hw Next Hop Intf:       6
 HW Port:                1
 Next Hop Flags:         COMPLETE--->This indicates that the router can forward traffic.
 Next Hop MAC:           4055.395f.46b7

NHINDEX H/W Result for NP:0 (index: 0 (BE)):
NhIndex is NOT required on this platform

NHINDEX STATS: pkts 0, bytes 0 (all NPs combined, no stats)

RX H/W Result on NP:0 [Adj ptr:0x40 (BE)]:
Rx-Adj is NOT required on this platform
```

Associated Commands

- show cef