Nexus 7000: M3 Unicast Forwarding

M3 Unicast Forwarding

This document provides debugging steps for layer 3 multicast forwarding paths.

It covers debugging packet flow for Layer 3 unicast forwarding from Layer3 to Layer 3 interfaces

URIB - (Unicast Routing Information Base) is responsible to cache routes added by routing protocols - EIGRP, OSPF, BGP (each of which runs as separate process) and sends route-updates to UFDM (Unicast Forwarding Distribution Manager) to program the hardware unicast forwarding tables.

Check the (VPN, IP_DA) programming in PI

show ip route

IP Route Table for VRF "default"
'*' denotes best ucast next-hop
'***' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

2.1.1.0/24, ubest/mbest: 1/0, attached  
*via 2.1.1.1, Eth2/1, [0/0], 23:50:07, direct  
2.1.1.1/32, ubest/mbest: 1/0, attached  
*via 2.1.1.1, Eth2/1, [0/0], 23:50:07, local  
2.2.2.0/24, ubest/mbest: 1/0, attached  
*via 2.2.2.2, Eth2/2, [0/0], 00:32:07, direct  
2.2.2.2/32, ubest/mbest: 1/0, attached  
*via 2.2.2.2, Eth2/2, [0/0], 00:32:07, local
Check the (VPN, IP_DA) programming in IPFIB-PD to make sure that the routes are correctly programmed.

module-2#show system internal forwarding route detail

This is the Local interface /32 route pointing to the SUP

2.1.1.1/32, sup-eth1
Dev: 0, Idx: 0xf750a, Prio: 0x0, RPF Flags: S, DGT: VPN: 1
RPF_intf_5: sup-eth1 (0x7fe1)
AdjIdx: 0x104, LIFB: 0, LIF: sup-eth1 (0x7fe1), DI: 0x00
DMAC: 0000.0000.0000 SMAC: 0000.0000.0000

This is the Connected Next Hop Programming pointing to Eth2/1

2.1.1.2/32, Ethernet2/1
Dev: 0, Idx: 0xf7284, Prio: 0x0, RPF Flags: VS, DGT: VPN: 1
RPF_intf_5: Ethernet2/1 (0x1000)
AdjIdx: 0x112, LIFB: 0, LIF: Ethernet2/1 (0x1000), DI: 0xb99
DMAC: 0000.087e.7b43 SMAC: 0516.e000.0000

Check ELTM Info for ingress interface (ethernet 1/1).

module-2# show system internal eltmc info interface ethernet 2/1

IFTMC Detailed info for Interface Ethernet2/1
CR_flags = INTF LIF (0x3), LTL = 2969 (0xb99), (S 0x0, P 0x0)
IF_INDEX = 0x1a080000, LIF = 16441 (0x4039), SDB LIF = 16441 (0x4039)
IF_INDEX = 0x1a080000, VDC LIF = 4096 (0x1000), SDB LIF = 4096 (0x1000)

State = UP
port_layer: 0x3 port_mode: 0x0
local_port: 0x1 ldb_port_prop.flags: 0x0
ldb_sharing: 0x0 ilm_sharing: 0x0
ldb_port_prop.dsm: 0x0 ldb_port_prop.dnl: 0x0
parent_pc_ifindex: 0x0 fb_if: 0x0

Interface Features:
bd: 2 (0x2)
admin_port_mode: 0x0
ipv4_en: 0x1 ipv4_mcast_en: 0x0
df_mask: 0x0 ipsq_en: 0x0
v4_table_id: 0x1 non_ipv6_vpn: 0x1
v6_table_id: 0x80000001 v6_vpn: 0x1
use_table_id: 0x0 use_vpn: 0x0
icmp_redirect: 0x1 ipv6_redirect: 0x1
v4_same_if_check: 0x1 v6_same_if_check: 0x1
mtu_index: 0x1 new_mtu_index: 0x1
mtu: 0x5dc port_trust: 0x0
v4_rpf_mode: 0x0 v6_rpf_mode: 0x0
intf_type: 0x0 intf_type_flags: 0x0
sub_type: 0x0 port_type: 0x0
per_pkt_ls_en: 0x0 default_vsi_encap: 0x0
si_vlan: 0x0 dvif: 0x0
fcoe_mac_ip_chk: 0x0 fp_core_bcast_learn: 0x0
mpls_en: 0x0 frr_down: 0x0
egress_vsl_drop: 0x0 mcec_flag: 0x0
module-2# show hardware internal forwarding l3 inst 0 table lif_map_tbl_entry index 0x1000

Use the VDC LIF from the output above

lif:  16441  ASIC lif should map to Global LIF

module-2# show hardware internal forwarding l3 inst 0 table ilm index 0x1000

Use the VDC LIF from the output above

non_ipv6 vpn:  VPN ID for the VRF that this interface belongs to
v6_vpn:  1
ipv4_en:  1  ASIC lif should have ipv4 enabled
v4_sgt_prio:  3
v4_dgt_prio:  1
cpp_en:  1
acl_en:  3
qos_en:  1
label_b:  8189
label_a:  8189
module-2#

Make sure that the FIB TCAM, DRAM, Adjacency, RIT and RIT2ADJ are programmed correctly for the route.

ddebug forwarding spl fib-tcam inst 0 lookup ipv4-ucast ipv4_address 2.1.1.2 vpn 1

Use this command to get the TCAM Index

index: 0xf7284 priority: 0  gives the TCAM_HIT Index

module-2# show hardware internal forwarding l3 inst 0 table rslt_ucastr index 0xf7284

Use this command to get the DRAM Index

ecmp_grp_idx:  0x112  Gives the RIT Index, provides the rewrite information, which provides DMAC and SMAC.
ecmp_grp_idx_vld:  0

If it’s a ECMP or a VOBJ then the ecmp_grp_index gives ecmp_grp_idx_vld is 1 and ecmp_grp_idx points to the index in the ECMP Group Table
module-2# show hardware internal forwarding l3 inst 0 table rit index 0x112 detail

Use the RIT idx from the output above

<-snip->
l2_smac_ptr:  1
rbid_code:  0
rbid:  0
suppress_ml2:  0
iptomac:  0
lsmtomac:  0
ext_sel:  0  If this is 0, then l2_smac_ptr is an index into the SMAC TBL
findex:  0

module-2# show hardware internal forwarding l2 inst 0 table smac_tbl index 1

Use this command to get the Source Mac Table

+-------------------------------------------------------------------------------
| Source MAC table (logical layout) for F4 L2FWD driver |
| Inst 0; port(s) 1-24                                           |
| INDEX         VALUE                   |
| DEC  /  HEX       HEX                 |
|--------------    --------                |
| 1 /     1    0000516 e0000000          |
+-------------------------------------------------------------------------------

module-2# show hardware internal forwarding l3 inst 0 table rw2adj_map_tbl_adjptr index 0x112

Use the  RIT index to get the RIT2ADJ Ptr

adjptr:  132  Each RIT is mapped to an ADJ via the RIT2ADJ mapping table

module-2# show hardware internal forwarding l3 inst 0 table adj index 132

Use the RIT2ADJ Ptr from the above output to get the egress_lif

same_if_mask_sel 0x0
ingress_lif_segmentid_sel 0x0
format 0x0
fc_iod_drop 0x0
mcast_cpp_lif 0x0
ad_age 0x1
l3_enable 0x1
ad_trig 0x0
valid 0x1
rdt 0x1
peer_id_sel 0x0
no_intra_split_horizon 0x0
egress_lif 0x1000
ri 0x3
LIF to DI:

Get the IF_Index from LIF

```
module-2# show system internal iftmc hardware lif brief | grep 0x1000
```

Use the egress LIF go get the interface index from the LIF

```
0x1a080000 0x4039 0x1000 0x1000
```

Using the IF_index get the DI

```
module-2# show system internal iftmc info interface brief | grep 0x1a080000
```

Use the interface index to get the DI

```
Eth2/1 0x1a080000 0x4039 0xb99 INTF LIF UP L3
```