IPv6 Neighbor Discovery

This chapter contains the following sections:

- Neighbor Discovery, on page 1
- Configuring IPv6 Neighbor Discovery on a Bridge Domain, on page 2
- Configuring IPv6 Neighbor Discovery on a Layer 3 Interface, on page 5
- Configuring IPv6 Neighbor Discovery Duplicate Address Detection , on page 10

Neighbor Discovery

The IPv6 Neighbor Discovery (ND) protocol is responsible for the address auto configuration of nodes, discovery of other nodes on the link, determining the link-layer addresses of other nodes, duplicate address detection, finding available routers and DNS servers, address prefix discovery, and maintaining reachability information about the paths to other active neighbor nodes.

ND-specific Neighbor Solicitation or Neighbor Advertisement (NS or NA) and Router Solicitation or Router Advertisement (RS or RA) packet types are supported on all ACI fabric Layer 3 interfaces, including physical, Layer 3 sub interface, and SVI (external and pervasive). Up to APIC release 3.1(1x), RS/RA packets are used for auto configuration for all Layer 3 interfaces but are only configurable for pervasive SVIs.

Starting with APIC release 3.1(2x), RS/RA packets are used for auto configuration and are configurable on Layer 3 interfaces including routed interface, Layer 3 sub interface, and SVI (external and pervasive).

ACI bridge domain ND always operates in flood mode; unicast mode is not supported.

The ACI fabric ND support includes the following:

- Interface policies (`nd:IfPol`) control ND timers and behavior for NS/NA messages.
- ND prefix policies (`nd:PfxPol`) control RA messages.
- Configuration of IPv6 subnets for ND (fv:Subnet).
- ND interface policies for external networks.
- Configurable ND subnets for external networks, and arbitrary subnet configurations for pervasive bridge domains are not supported.

Configuration options include the following:

- Adjacencies

  - Configurable Static Adjacencies: (<vrf, L3Iface, ipv6 address> --> mac address)
  - Dynamic Adjacencies: Learned via exchange of NS/NA packets

- Per Interface
  - Control of ND packets (NS/NA)
    - Neighbor Solicitation Interval
    - Neighbor Solicitation Retry count
  - Control of RA packets
    - Suppress RA
    - Suppress RA MTU
    - RA Interval, RA Interval minimum, Retransmit time
- Per Prefix (advertised in RAs) control
  - Lifetime, preferred lifetime
  - Prefix Control (auto configuration, on link)
- Neighbor Discovery Duplicate Address Detection (DAD)

Configuring IPv6 Neighbor Discovery on a Bridge Domain

Creating the Tenant, VRF, and Bridge Domain with IPv6 Neighbor Discovery on the Bridge Domain Using the REST API

Procedure

Create a tenant, VRF, bridge domain with a neighbor discovery interface policy and a neighbor discovery prefix policy.

Example:

```
<fvTenant descr="" dn="uni/tn-ExampleCorp" name="ExampleCorp" ownerKey="" ownerTag="">
    <ndIfPol name="NDPol001" ctrl="managed-cfg”  descr="" hopLimit="64" mtu="1500"
nsIntvl="1000" nsRetries=“3" ownerKey="" ownerTag="" raIntvl="600" raLifetime="1800"
reachableTime="0" retransTimer="0"/>
    <fvCtx descr="" knwMcastAct="permit" name="pvn1" ownerKey="" ownerTag=""
pcEnfPref="enforced">
    </fvCtx>
   <fvBD arpFlood="no" descr="" mac="00:22:BD:F8:19:FF" multiDstPktAct="bd-flood" name="bd1"
 ownerKey="" ownerTag="" unicastRoute="yes" unkMacUcastAct="proxy" unkMcastAct="flood">
        <fvRsBDToNdP tnNdIfPolName="NDPol001"/>
        <fvRsCtx tnFvCtxName="pvn1"/>
        <fvSubnet ctrl="nd" descr="" ip="34::1/64" name="" preferred="no" scope="private">
```

```
            <fvRsNdPfxPol tnNdPfxPolName="NDPfxPol001"/>
        </fvSubnet>
        <fvSubnet ctrl="nd" descr="" ip="33::1/64" name="" preferred="no" scope="private">
            <fvRsNdPfxPol tnNdPfxPolName="NDPfxPol002"/>
        </fvSubnet>
    </fvBD>
    <ndPfxPol ctrl="auto-cfg,on-link" descr="" lifetime="1000" name="NDPfxPol001" ownerKey=""
 ownerTag="" prefLifetime="1000"/>
     <ndPfxPol ctrl="auto-cfg,on-link" descr="" lifetime="4294967295" name="NDPfxPol002"
ownerKey="" ownerTag="" prefLifetime="4294967295"/>
</fvTenant>
```

Note If you have a public subnet when you configure the routed outside, you must associate the bridge domain with the outside configuration.

Configuring a Tenant, VRF, and Bridge Domain with IPv6 Neighbor Discovery on the Bridge Domain Using the NX-OS Style CLI

Procedure

Step 1 Configure an IPv6 neighbor discovery interface policy and assign it to a bridge domain:

a) Create an IPv6 neighbor discovery interface policy:

Example:

```
apic1(config)# tenant ExampleCorp
apic1(config-tenant)# template ipv6 nd policy NDPol001
apic1(config-tenant-template-ipv6-nd)# ipv6 nd mtu 1500
```

b) Create a VRF and bridge domain:

Example:

```
apic1(config-tenant)# vrf context pvn1
apic1(config-tenant-vrf)# exit
apic1(config-tenant)# bridge-domain bd1
apic1(config-tenant-bd)# vrf member pvn1
apic1(config-tenant-bd)# exit
```

c) Assign an IPv6 neighbor discovery policy to the bridge domain:

Example:

```
apic1(config-tenant)# interface bridge-domain bd1
apic1(config-tenant-interface)# ipv6 nd policy NDPol001
apic1(config-tenant-interface)#exit
```

Step 2 Configure an IPV6 bridge domain subnet and neighbor discovery prefix policy on the subnet:

Example:

```
apic1(config-tenant)# interface bridge-domain bd1
```

```
apic1(config-tenant-interface)# ipv6 address 34::1/64
apic1(config-tenant-interface)# ipv6 address 33::1/64
apic1(config-tenant-interface)# ipv6 nd prefix 34::1/64 1000 1000
apic1(config-tenant-interface)# ipv6 nd prefix 33::1/64 4294967295 4294967295
```

Creating the Tenant, VRF, and Bridge Domain with IPv6 Neighbor Discovery on the Bridge Domain Using the GUI

This task shows how to create a tenant, a VRF, and a bridge domain (BD) within which two different types of Neighbor Discovery (ND) policies are created. They are ND interface policy and ND prefix policy. While ND interface policies are deployed under BDs, ND prefix policies are deployed for individual subnets. Each BD can have its own ND interface policy . The ND interface policy is deployed on all IPv6 interfaces by default. In Cisco APIC, there is already an ND interface default policy available to use. If desired, you can create a custom ND interface policy to use instead. The ND prefix policy is on a subnet level. Every BD can have multiple subnets, and each subnet can have a different ND prefix policy.

Procedure

Step 1 On the menu bar, click **TENANT** > **Add Tenant**.

Step 2 In the **Create Tenant** dialog box, perform the following tasks:

a) in the **Name** field, enter a name.
b) Click the **Security Domains +** icon to open the **Create Security Domain** dialog box.
c) In the **Name** field, enter a name for the security domain. Click **Submit**.
d) In the **Create Tenant** dialog box, check the check box for the security domain that you created, and click **Submit**.

Step 3 In the **Navigation** pane, expand **Tenant-name** > **Networking**. In the **Work** pane, drag the **VRF** icon to the canvas to open the **Create VRF** dialog box, and perform the following actions:

a) In the **Name** field, enter a name.
b) Click **Submit** to complete the **VRF** configuration.

Step 4 In the **Networking** area, drag the **BD** icon to the canvas while connecting it to the **VRF** icon. In the **Create Bridge Domain** dialog box that displays, perform the following actions:

a) In the **Name** field, enter a name.
b) Click the **L3 Configurations** tab, and expand **Subnets** to open the **Create Subnet** dialog box, enter the subnet mask in the **Gateway IP** field.

Step 5 In the **Subnet Control** field, ensure that the **ND RA Prefix** check box is checked.

Step 6 In the **ND Prefix policy** field drop-down list, click **Create ND RA Prefix Policy**.

Note There is already a default policy available that will be deployed on all IPv6 interfaces. Alternatively, you can create an ND prefix policy to use as shown in this example. By default, the IPv6 gateway subnets are advertised as ND prefixes in the ND RA messages. A user can choose to not advertise the subnet in ND RA messages by un-checking the ND RA prefix check box.

Step 7 In the **Create ND RA Prefix Policy** dialog box, perform the following actions:

a) In the **Name** field, enter the name for the prefix policy.

Note For a given subnet there can only be one prefix policy. It is possible for each subnet to have a different prefix policy, although subnets can use a common prefix policy.

b) In the **Controller State** field, check the desired check boxes.
c) In the **Valid Prefix Lifetime** field, choose the desired value for how long you want the prefix to be valid.
d) In the **Preferred Prefix Lifetime** field, choose a desired value. Click **OK**.

Note An ND prefix policy is created and attached to the specific subnet.

Step 8 In the **ND policy** field drop-down list, click **Create ND Interface Policy** and perform the following tasks:

a) In the **Name** field, enter a name for the policy.
b) Click **Submit**.

Step 9 Click **OK** to complete the bridge domain configuration.

Similarly you can create additional subnets with different prefix policies as required.

A subnet with an IPv6 address is created under the BD and an ND prefix policy has been associated with it.

Configuring IPv6 Neighbor Discovery on a Layer 3 Interface

Guidelines and Limitations

The following guidelines and limitations apply to Neighbor Discovery Router Advertisement (ND RA) Prefixes for Layer 3 Interfaces:

- An ND RA configuration applies only to IPv6 Prefixes. Any attempt to configure an ND policy on IPv4 Prefixes will fail to apply.

Configuring an IPv6 Neighbor Discovery Interface Policy with RA on a Layer 3 Interface Using the GUI

Note The steps here show how to associate an IPv6 neighbor discovery interface policy with a Layer 3 interface. The specific example shows how to configure using the non-VPC interface.

Before you begin

- The tenant, VRF, BD are created.
- The L3Out is created under External Routed Networks.

Procedure

Step 1 In the **Navigation** pane, navigate to the appropriate external routed network under the appropriate Tenant.

Step 2 Under **External Routed Networks**, expand > **Logical Node Profiles** > ***Logical Node Profile_name*** > **Logical Interface Profiles**.

Step 3 Double-click the appropriate **Logical Interface Profile**, and in the **Work** pane, click **Policy** > **Routed Interfaces**.

Note If you do not have a Logical Interface Profile created, you can create a profile here.

Step 4 In the **Routed Interface** dialog box, perform the following actions:

a) In the **ND RA Prefix** field, check the check box to enable ND RA prefix for the interface.

When enabled, the routed interface is available for auto configuration.

Also, the **ND RA Prefix Policy** field is displayed.

b) In the **ND RA Prefix Policy** field, from the drop-down list, choose the appropriate policy.

c) Choose other values on the screen as desired. Click **Submit**.

Note **When you configure using a VPC interface, you must enable the ND RA prefix for both side A and side B as both are members in the VPC configuration**. In the **Work** Pane, in the **Logical Interface Profile** screen, click the **SVI** tab. Under **Properties**, check the check boxes to enable the **ND RA Prefix** for both Side A and Side B. Choose the identical **ND RA Prefix Policy** for Side A and Side B.

Configuring an IPv6 Neighbor Discovery Interface Policy with RA on a Layer 3 Interface Using the REST API

Procedure

Configure an IPv6 neighbor discovery interface policy and associate it with a Layer 3 interface:

The following example displays the configuration in a non-VPC set up.

Example:

```
<fvTenant dn="uni/tn-ExampleCorp" name="ExampleCorp">
  <ndIfPol name="NDPol001" ctrl="managed-cfg" hopLimit="64" mtu="1500" nsIntvl="1000"
nsRetries="3" raIntvl="600" raLifetime="1800" reachableTime="0" retransTimer="0"/>
  <fvCtx name="pvn1" pcEnfPref="enforced">
                    </fvCtx>
  <l3extOut enforceRtctrl="export" name="l3extOut001">
    <l3extRsEctx tnFvCtxName="pvn1"/>
    <l3extLNodeP name="lnodeP001">
      <l3extRsNodeL3OutAtt rtrId="11.11.205.1" rtrIdLoopBack="yes"
tDn="topology/pod-2/node-2011"/>
      <l3extLIfP name="lifP001">
        <l3extRsPathL3OutAtt addr="2001:20:21:22::2/64" ifInstT="l3-port" llAddr="::"
```

```
mac="00:22:BD:F8:19:FF" mode="regular" mtu="inherit"
tDn="topology/pod-2/paths-2011/pathep-[eth1/1]">
          <ndPfxP>
            <ndRsPfxPToNdPfxPol tnNdPfxPolName="NDPfxPol001"/>
          </ndPfxP>
        </l3extRsPathL3OutAtt>
        <l3extRsNdIfPol tnNdIfPolName="NDPol001"/>
      </l3extLIfP>
    </l3extLNodeP>
    <l3extInstP name="instp"/>
   </l3extOut>
  <ndPfxPol ctrl="auto-cfg,on-link" descr="" lifetime="1000" name="NDPfxPol001" ownerKey=""
 ownerTag="" prefLifetime="1000"/>
</fvTenant>
```

Note For VPC ports, ndPfxP must be a child of l3extMember instead of l3extRsNodeL3OutAtt. The following code snippet shows the configuration in a VPC setup.

```
<l3extLNodeP name="lnodeP001">
<l3extRsNodeL3OutAtt rtrId="11.11.205.1" rtrIdLoopBack="yes"
tDn="topology/pod-2/node-2011"/>
<l3extRsNodeL3OutAtt rtrId="12.12.205.1" rtrIdLoopBack="yes"
tDn="topology/pod-2/node-2012"/>
  <l3extLIfP name="lifP002">
    <l3extRsPathL3OutAtt addr="0.0.0.0" encap="vlan-205" ifInstT="ext-svi"
llAddr="::" mac="00:22:BD:F8:19:FF" mode="regular" mtu="inherit"
tDn="topology/pod-2/protpaths-2011-2012/pathep-[vpc7]" >
      <l3extMember addr="2001:20:25:1::1/64" descr="" llAddr="::" name=""
nameAlias="" side="A">
        <ndPfxP >
          <ndRsPfxPToNdPfxPol tnNdPfxPolName="NDPfxPol001"/>
        </ndPfxP>
      </l3extMember>
      <l3extMember addr="2001:20:25:1::2/64" descr="" llAddr="::" name=""
nameAlias="" side="B">
        <ndPfxP >
          <ndRsPfxPToNdPfxPol tnNdPfxPolName="NDPfxPol001"/>
        </ndPfxP>
      </l3extMember>
    </l3extRsPathL3OutAtt>
    <l3extRsNdIfPol tnNdIfPolName="NDPol001"/>    </l3extLIfP>
   </l3extLNodeP>
```

Configuring an IPv6 Neighbor Discovery Interface Policy with RA on a Layer 3 Interface Using the NX-OS Style CLI

This example configures an IPv6 neighbor discovery interface policy, and assigns it to a Layer 3 interface. Next, it configures an IPv6 Layer 3 Out interface, neighbor discovery prefix policy, and associates the neighbor discovery policy to the interface.

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `apic1#` **`configure`**	Enters configuration mode.

Step 2	**tenant** *tenant_name*
	Example:
	`apic1(config)#` **`tenant ExampleCorp`**
	`apic1(config-tenant)#`
		Creates a tenant and enters the tenant mode.

Step 3	**template ipv6 nd policy** *policy_name* **Example:** `apic1(config-tenant)#` **`template ipv6 nd policy NDPol001`**	Creates an IPv6 ND policy.

Step 4	**ipv6 nd mtu** *mtu value*
	Example:
	`apic1(config-tenant-template-ipv6-nd)#` **`ipv6 nd mtu 1500`**
	`apic1(config-tenant-template-ipv6)#` **`exit`**
	`apic1(config-tenant-template)#` **`exit`**
	`apic1(config-tenant)#`
		Assigns an MTU value to the IPv6 ND policy.

Step 5	**vrf context** *VRF_name*
	Example:
	`apic1(config-tenant)#` **`vrf context pvn1`**
	`apic1(config-tenant-vrf)#` **`exit`**
		Creates a VRF.

Step 6	**l3out** *VRF_name* **Example:** `apic1(config-tenant)#` **`l3out l3extOut001`**	Creates a Layer 3 Out.
Step 7	**vrf member** *VRF_name* **Example:** `apic1(config-tenant-l3out)#` **`vrf member pvn1`**	Associates the VRF with the Layer 3 Out.

	Command or Action	Purpose
	`apic1(config-tenant-l3out)# exit`	

Step 8	**external-l3 epg instp l3out** *l3extOut001*
	Example:
	`apic1(config-tenant)# external-l3 epg instp l3out l3extOut001`
	`apic1(config-tenant-l3ext-epg)# vrf member pvn1`
	`apic1(config-tenant-l3ext-epg)# exit`
		Assigns the Layer 3 Out and the VRF to a Layer 3 interface.

Step 9	**leaf** *2011* **Example:** `apic1(config)# leaf 2011`	Enters the leaf switch mode.

Step 10	**vrf context tenant** *ExampleCorp* **vrf** *pvn1* **l3out** *l3extOut001*
	Example:
	`apic1(config-leaf)# vrf context tenant ExampleCorp vrf pvn1 l3out l3extOut001`
	`apic1(config-leaf-vrf)# exit`
		Associates the VRF to the leaf switch.

Step 11	**int** *eth 1/1*
	Example:
	`apic1(config-leaf)# int eth 1/1`
	`apic1(config-leaf-if)#`
		Enters the interface mode.

Step 12	**vrf member tenant** *ExampleCorp* **vrf** *pvn1* **l3out** *l3extOut001* **Example:** `apic1(config-leaf-if)# vrf member tenant ExampleCorp vrf pvn1 l3out l3extOut001`	Specifies the associated Tenant, VRF, Layer 3 Out in the interface.
Step 13	**ipv6 address** *2001:20:21:22::2/64* **preferred** **Example:** `apic1(config-leaf-if)# ipv6 address 2001:20:21:22::2/64 preferred`	Specifies the primary or preferred IPv6 address.

	Command or Action	Purpose
Step 14	**ipv6 nd prefix** *2001:20:21:22::2/64 1000 1000* **Example:** `apic1(config-leaf-if)# ipv6 nd prefix 2001:20:21:22::2/64 1000 1000`	Configures the IPv6 ND prefix policy under the Layer 3 interface.

Step 15	**inherit ipv6 nd** *NDPol001*
	Example:
	`apic1(config-leaf-if)# inherit ipv6 nd NDPol001`
	`apic1(config-leaf-if)# exit`
	`apic1(config-leaf)# exit`
		Configures the ND policy under the Layer 3 interface.

The configuration is complete.

Configuring IPv6 Neighbor Discovery Duplicate Address Detection

About Neighbor Discovery Duplicate Address Detection

Duplicate Address Detection (DAD) is a process that is used by Neighbor Discovery to detect the duplicated addresses in the network. By default, DAD is enabled for the link-local and global-subnet IPv6 addresses used on the ACI fabric leaf layer 3 interfaces. Optionally, you can disable the DAD process for a IPv6 global-subnet by configuring the knob through the REST API (using the **ipv6Dad="disabled"** setting) or through the GUI. Configure this knob when the same shared secondary address is required to be used across L3Outs on different border leaf switches to provide border leaf redundancy to the external connected devices. Disabling the DAD process in this case will avoid the situation where the DAD considers the same shared secondary address on multiple border leaf switches as duplicates. If you do not disable the DAD process in this case, the shared secondary address might enter into the DUPLICATE DAD state and become unusable.

Configuring Neighbor Discovery Duplicate Address Detection Using the REST API

Procedure

Step 1 Disable the Neighbor Discovery Duplicate Address Detection process for a subnet by changing the value of the ipv6Dad entry for that subnet to **disabled**.

The following example shows how to set the Neighbor Discovery Duplicate Address Detection entry for the 2001:DB8:A::11/64 subnet to **disabled**:

Note In the following REST API example, long single lines of text are broken up with the \ character to improve readability.

Example:

```
         <l3extRsPathL3OutAtt addr="2001:DB8:A::2/64" autostate="enabled" \
          childAction="" descr="" encap="vlan-1035" encapScope="local" \
          ifInstT="ext-svi" ipv6Dad="enabled" llAddr=": :" \
          mac="00:22:BD:F8:19:DD"  mtu="inherit"  \
          rn="rspathL3OutAtt-[topology/pod-1/paths-105/pathep-[eth1/1]]"  \
          status=""  tDn="topology/pod-1/paths-105/pathep-[eth1/1]"  >
             <l3extIp addr="2001:DB8:A::11/64" childAction="" descr="" \
              ipv6Dad="disabled"  name="" nameAlias="" \
              rn="addr-[2001:DB8:A::11/64]" status=""/>

         </l3extRsPathL3OutAtt>
     </l3extLIfP>
</l3extLNodeP>
```

Step 2 Enter the **show ipv6 int** command on the leaf switch to verify that the configuration was pushed out correctly to the leaf switch. For example:

```
swtb23-leaf5# show ipv6 int vrf icmpv6:v1
IPv6 Interface Status for VRF "icmpv6:v1"(9)

vlan2, Interface status: protocol-up/link-up/admin-up, iod: 73
if_mode: ext
  IPv6 address:
    2001:DB8:A::2/64 [VALID] [PREFERRED]
    2001:DB8:A::11/64 [VALID] [dad-disabled]
  IPv6 subnet:  2001:DB8:A::/64
  IPv6 link-local address: fe80::863d:c6ff:fe9f:eb8b/10 (Default) [VALID]
```

Configuring Neighbor Discovery Duplicate Address Detection Using the GUI

Use the procedures in this section to disable the Neighbor Discovery Duplicate Address Detection process for a subnet.

Procedure

Step 1 Navigate to the appropriate page to access the DAD field for that interface. For example:

a) Navigate to **Tenants** > **Tenant** > **Networking** > **External Routed Networks** > ***L3Out*** > **Logical Node Profiles** > ***node*** > **Logical Interface Profiles**, then select the interface that you want to configure.

b) Click on *Routed Sub-interfaces* or *SVI*, then click on the Create (+) button to configure that interface.

Step 2 For this interface, make the following settings for the DAD entries:

- For the primary address, set the value for the DAD entry to **enabled**.
- For the shared secondary address, set the value for the DAD entry to **disabled**. Note that if the secondary address is not shared across border leaf switches, then you do not need to disable the DAD for that address.

Example:

For example, if you were configuring this setting for the SVI interface, you would:

- Set the Side A IPv6 DAD to **enabled**.
- Set the Side B IPv6 DAD to **disabled**.

Example:

As another example, if you were configuring this setting for the routed sub-interface interface, you would:

- In the main Select Routed Sub-Interface page, set the value for IPv6 DAD for the routed sub-interface to **enabled**.
- Click on the Create (+) button on the IPv4 Secondary/IPv6 Additional Addresses area to access the Create Secondary IP Address page, then set the value for IPv6 DAD to **disabled**. Then click on the OK button to apply the changes in this screen.

Step 3 Click on the Submit button to apply your changes.

Step 4 Enter the **show ipv6 int** command on the leaf switch to verify that the configuration was pushed out correctly to the leaf switch. For example:

```
swtb23-leaf5# show ipv6 int vrf icmpv6:v1
IPv6 Interface Status for VRF "icmpv6:v1"(9)

vlan2, Interface status: protocol-up/link-up/admin-up, iod: 73
if_mode: ext
  IPv6 address:
    2001:DB8:A::2/64 [VALID] [PREFERRED]
    2001:DB8:A::11/64 [VALID] [dad-disabled]
  IPv6 subnet:  2001:DB8:A::/64
  IPv6 link-local address: fe80::863d:c6ff:fe9f:eb8b/10 (Default) [VALID]
```