



Managing Layer 3 Networking

This chapter contains the following sections:

- [Configuring External Connectivity Using a Layer 3 Out, on page 1](#)
- [Configuring a Tenant Layer 3 Outside Network Connection Overview, on page 1](#)
- [Configuring Layer 3 Outside for Tenant Networks Using the REST API, on page 2](#)
- [Configuring BGP Max Path, on page 5](#)
- [Configuring AS Path Prepend, on page 5](#)
- [Configuring BFD, on page 6](#)

Configuring External Connectivity Using a Layer 3 Out

This section provides a step-by-step configuration required for the ACI fabric to connect to an external routed network through L3Outs and MP-BGP route reflectors.

This example uses Open Shortest Path First (OSPF) as the routing protocol in an L3Out under the 'mgmt' tenant.

Configuring a Tenant Layer 3 Outside Network Connection Overview

This topic provides a typical example of how to configure a Layer 3 Outside for tenant networks when using Cisco APIC.



Note Cisco ACI does not support IP fragmentation. Therefore, when you configure Layer 3 Outside (L3Out) connections to external routers, or Multi-Pod connections through an Inter-Pod Network (IPN), it is recommended that the interface MTU is set appropriately on both ends of a link. On some platforms, such as Cisco ACI, Cisco NX-OS, and Cisco IOS, the configurable MTU value does not take into account the Ethernet headers (matching IP MTU, and excluding the 14-18 Ethernet header size), while other platforms, such as IOS-XR, include the Ethernet header in the configured MTU value. A configured value of 9000 results in a max IP packet size of 9000 bytes in Cisco ACI, Cisco NX-OS, and Cisco IOS, but results in a max IP packet size of 8986 bytes for an IOS-XR untagged interface.

For the appropriate MTU values for each platform, see the relevant configuration guides.

We highly recommend that you test the MTU using CLI-based commands. For example, on the Cisco NX-OS CLI, use a command such as `ping 1.1.1.1 df-bit packet-size 9000 source-interface ethernet 1/1`.

Configuring Layer 3 Outside for Tenant Networks Using the REST API

The external routed network that is configured in the example can also be extended to support both IPv4 and IPv6. Both IPv4 and IPv6 routes can be advertised to and learned from the external routed network. To configure an L3Out for a tenant network, send a post with XML such as the example.

This example is broken into steps for clarity. For a merged example, see [REST API Example: L3Out](#).

Before you begin

- Configure the node, port, functional profile, AEP, and Layer 3 domain.
- Create the external routed domain and associate it to the interface for the L3Out.
- Configure a BGP route reflector policy to propagate the routes within the fabric.

For an XML example of these prerequisites, see [REST API Example: L3Out Prerequisites](#).

Step 1 Configure the tenant, VRF, and bridge domain.

This example configures tenant `t1` with VRF `v1` and bridge domain `bd1`. The tenant, VRF, and BD are not yet deployed.

Example:

```
<fvTenant name="t1">
  <fvCtx name="v1"/>
  <fvBD name="bd1">
    <fvRsCtx tnFvCtxName="v1"/>
    <fvSubnet ip="44.44.44.1/24" scope="public"/>
    <fvRsBDToOut tnL3extOutName="l3out1"/>
  </fvBD/>
</fvTenant>
```

Step 2 Configure an application profile and application EPG.

This example configures application profile `app1` (on node 101), EPG `epg1`, and associates the EPG with `bd1` and the contract `httpCtrct`, as the consumer.

Example:

```
<fvAp name="app1">
  <fvAEPg name="epg1">
    <fvRsDomAtt instrImedcy="immediate" tDn="uni/phys-dom1"/>
    <fvRsBd tnFvBDName="bd1" />
    <fvRsPathAtt encap="vlan-2011" instrImedcy="immediate" mode="regular"
tDn="topology/pod-1/paths-101/pathep-[eth1/3]"/>
    <fvRsCons tnVzBrCPName="httpCtrct"/>
  </fvAEPg>
</fvAp>
```

Step 3 Configure the node and interface.

This example configures VRF `v1` on node 103 (the border leaf switch), with the node profile, `nodep1`, and router ID `11.11.11.103`. It also configures interface `eth1/3` as a routed interface (Layer 3 port), with IP address `12.12.12.1/24` and Layer 3 domain `dom1`.

Example:

```
<l3extOut name="l3out1">
  <l3extRsEctx tnFvCtxName="v1"/>
  <l3extLNodeP name="nodep1">
    <l3extRsNodeL3OutAtt rtrId="11.11.11.103" tDn="topology/pod-1/node-103"/>
    <l3extLIIfP name="ifp1"/>
    <l3extRsPathL3OutAtt addr="12.12.12.3/24" ifInstT="l3-port"
tDn="topology/pod-1/paths-103/pathep-[eth1/3]"/>
    </l3extLIIfP>
  </l3extLNodeP>
  <l3extRsL3DomAtt tDn="uni/l3dom-dom1"/>
</l3extOut>
```

Step 4 Configure the routing protocol.

This example configures BGP as the primary routing protocol, with a BGP peer with the IP address, `15.15.15.2` and ASN `100`.

Example:

```
<l3extOut name="l3out1">
  <l3extLNodeP name="nodep1">
    <bgpPeerP addr="15.15.15.2">
      <bgpAsP asn="100"/>
    </bgpPeerP>
  </l3extLNodeP>
  <bgpExtP/>
</l3extOut>
```

Step 5 Configure the connectivity routing protocol.

This example configures OSPF as the communication protocol, with regular area ID `0.0.0.0`.

Example:

```
<l3extOut name="l3out1">
  <ospfExtP areaId="0.0.0.0" areaType="regular"/>
  <l3extLNodeP name="nodep1">
    <l3extLIIfP name="ifp1">
      <ospfIIfP/>
    </l3extLIIfP>
  </l3extLNodeP>
</l3extOut>
```

Step 6 Configure the external EPG.

This example configures the network 20.20.20.0/24 as external network `extnw1`. It also associates `extnw1` with the route control profile `rp1` and the contract `httpCtrct`, as the provider.

Example:

```
<l3extOut name="l3out1">
  <l3extInstP name="extnw1">
    <l3extSubnet ip="20.20.20.0/24" scope="import-security"/>
    <fvRsProv tnVzBrCPName="httpCtrct"/>
  </l3extInstP>
</l3extOut>
```

Step 7 Optional. Configure a route map.

This example configures a route map for the BGP peer in the outbound direction. The route map is applied for routes that match a destination of 200.3.2.0/24. Also, on a successful match (if the route matches this range) the route AS PATH attribute is updated to 200 and 100.

Example:

```
<fvTenant name="t1">
  <rtctrlSubjP name="match-rule1">
    <rtctrlMatchRtDest ip="200.3.2.0/24"/>
  </rtctrlSubjP>
  <l3extOut name="l3out1">
    <rtctrlProfile name="rp1">
      <rtctrlCtxP name="ctxp1" action="permit" order="0">
        <rtctrlScope>
          <rtctrlRsScopeToAttrP tnRtctrlAttrPName="attrp1"/>
        </rtctrlScope>
        <rtctrlRsCtxPToSubjP tnRtctrlSubjPName="match-rule1"/>
      </rtctrlCtxP>
    </rtctrlProfile>
    <l3extInstP name="extnw1">
      <l3extSubnet ip="20.20.20.0/24" scope="import-security"/>
      <l3extRsInstPToProfile direction='export' tnRtctrlProfileName="rp1"/>
      <fvRsProv tnVzBrCPName="httpCtrct"/>
    </l3extInstP>
  </l3extOut>
</fvTenant>
```

Step 8 This example creates filters and contracts to enable the EPGs to communicate. The external EPG and the application EPG are already associated with the contract `httpCtrct` as provider and consumer respectively. The scope of the contract (where it is applied) can be within the application profile, the tenant, the VRF, or it can be used globally (throughout the fabric). In this example, the scope is the VRF (`context`).**Example:**

```
<vzFilter name="http-filter">
  <vzEntry name="http-e" etherT="ip" prot="tcp"/>
</vzFilter>
<vzBrCP name="httpCtrct" scope="context">
  <vzSubj name="subj1">
    <vzRsSubjFiltAtt tnVzFilterName="http-filter"/>
  </vzSubj>
</vzBrCP>
```

Step 9 Configure Advertise Host Routes.**Example:**

```
"<fvBD dn="uni/tn-t1/BD-b100" hostBasedRouting="yes"/>"
```

Configuring BGP Max Path

The following feature enables you to add the maximum number of paths to the route table to enable equal cost, multipath load balancing.

Configuring BGP Max Path Using the REST API

This following example provides information on how to configure the BGP Max Path feature using the REST API:

```
<fvTenant descr="" dn="uni/tn-t1" name="t1">
  <fvCtx name="v1">
    <fvRsCtxToBgpCtxAfPol af="ipv4-ucast" tnBgpCtxAfPolName="bgpCtxPol1"/>
  </fvCtx>
  <bgpCtxAfPol name="bgpCtxPol1" maxEcmp="8" maxEcmpIbgp="4"/>
</fvTenant>
```

Configuring AS Path Prepend

A BGP peer can influence the best-path selection by a remote peer by increasing the length of the AS-Path attribute. AS-Path Prepend provides a mechanism that can be used to increase the length of the AS-Path attribute by prepending a specified number of AS numbers to it.

AS-Path prepending can only be applied in the outbound direction using route-maps. AS Path prepending does not work in iBGP sessions.

The AS Path Prepend feature enables modification as follows:

Prepend	Appends the specified AS number to the AS path of the route matched by the route map. Note <ul style="list-style-type: none"> You can configure more than one AS number. 4 byte AS numbers are supported. You can prepend a total 32 AS numbers. You must specify the order in which the AS Number is inserted into the AS Path attribute.
Prepend-last-as	Prepends the last AS numbers to the AS path with a range between 1 and 10.

The following table describes the selection criteria for implementation of AS Path Prepend:

Prepend	1	Prepend the specified AS number.
Prepend-last-as	2	Prepend the last AS numbers to the AS path.

DEFAULT	Prepend(1)	Prepend the specified AS number.
---------	------------	----------------------------------

Configuring AS Path Prepend Using the REST API

The following example provides information on how to configure the AS Path Prepend feature using the REST API:

```
<?xml version="1.0" encoding="UTF-8"?>
<fvTenant name="coke">
  <rtctrlAttrP name="attrp1">
    <rtctrlSetASPath criteria="prepend">
      <rtctrlSetASPathASN asn="100" order="1"/>
      <rtctrlSetASPathASN asn="200" order="10"/>
      <rtctrlSetASPathASN asn="300" order="5"/>
    </rtctrlSetASPath/>
    <rtctrlSetASPath criteria="prepend-last-as" lastnum="9" />
  </rtctrlAttrP>

  <l3extOut name="out1">
    <rtctrlProfile name="rp1">
      <rtctrlCtxP name="ctxp1" order="1">
        <rtctrlScope>
          <rtctrlRsScopeToAttrP tnRtctrlAttrPName="attrp1"/>
        </rtctrlScope>
      </rtctrlCtxP>
    </rtctrlProfile>
  </l3extOut>
</fvTenant>
```

Configuring BFD

Configuring BFD Globally Using the REST API

The following REST API shows the global configuration for bidirectional forwarding detection (BFD):

Example:

```
<polUni>
  <infraInfra>
    <bfdIpv4InstPol name="default" echoSrcAddr="1.2.3.4" slowIntvl="1000" minTxIntvl="150"
minRxIntvl="250" detectMult="5" echoRxIntvl="200"/>
    <bfdIpv6InstPol name="default" echoSrcAddr="34::1/64" slowIntvl="1000" minTxIntvl="150"
minRxIntvl="250" detectMult="5" echoRxIntvl="200"/>
  </infraInfra>
</polUni>
```

Configuring BFD Interface Override Using the REST API

The following REST API shows the interface override configuration for bidirectional forwarding detection (BFD):

Example:

```
<fvTenant name="ExampleCorp">
  <bfdIfPol name="bfdIfPol" minTxIntvl="400" minRxIntvl="400" detectMult="5" echoRxIntvl="400"
echoAdminSt="disabled"/>
  <l3extOut name="l3-out">
    <l3extLNodeP name="leaf1">
      <l3extRsNodeL3OutAtt tDn="topology/pod-1/node-101" rtrId="2.2.2.2"/>
      <l3extLIfP name='portIpv4'>
        <l3extRsPathL3OutAtt tDn="topology/pod-1/paths-101/pathep-[eth1/11]" ifInstT='l3-port'
addr="10.0.0.1/24" mtu="1500"/>
        <bfdIfP type="sha1" key="password">
          <bfdRsIfPol tnBfdIfPolName='bfdIfPol' />
        </bfdIfP>
      </l3extLIfP>
    </l3extLNodeP>
  </l3extOut>
</fvTenant>
```

Configuring BFD Consumer Protocols Using the REST API

Step 1 The following example shows the interface configuration for bidirectional forwarding detection (BFD):

Example:

```
<fvTenant name="ExampleCorp">
  <bfdIfPol name="bfdIfPol" minTxIntvl="400" minRxIntvl="400" detectMult="5" echoRxIntvl="400"
echoAdminSt="disabled"/>
  <l3extOut name="l3-out">
    <l3extLNodeP name="leaf1">
      <l3extRsNodeL3OutAtt tDn="topology/pod-1/node-101" rtrId="2.2.2.2"/>
      <l3extLIfP name='portIpv4'>
        <l3extRsPathL3OutAtt tDn="topology/pod-1/paths-101/pathep-[eth1/11]" ifInstT='l3-port'
addr="10.0.0.1/24" mtu="1500"/>
        <bfdIfP type="sha1" key="password">
          <bfdRsIfPol tnBfdIfPolName='bfdIfPol' />
        </bfdIfP>
      </l3extLIfP>
    </l3extLNodeP>
  </l3extOut>
</fvTenant>
```

Step 2 The following example shows the interface configuration for enabling BFD on OSPF and EIGRP:

Example:

BFD on leaf switch

```
<fvTenant name="ExampleCorp">
```

```

    <ospfIfPol name="ospf_intf_pol" cost="10" ctrl="bfd"/>
    <eigrpIfPol ctrl="nh-self,split-horizon,bfd" dn="uni/tn-Coke/eigrpIfPol-eigrp_if_default"
  </fvTenant>

```

Example:

BFD on spine switch

```

<l3extLNodeP name="bSpine">
  <l3extRsNodeL3OutAtt tDn="topology/pod-1/node-103" rtrId="192.3.1.8">
    <l3extLoopBackIfP addr="10.10.3.1" />
    <l3extInfraNodeP fabricExtCtrlPeering="false" />
  </l3extRsNodeL3OutAtt>

  <l3extLIIfP name='portIf'>
    <l3extRsPathL3OutAtt tDn="topology/pod-1/paths-103/pathep-[eth5/10]" encap='vlan-4'
  ifInstT='sub-interface' addr="20.3.10.1/24"/>
    <ospfIfP>
      <ospfRsIfPol tnOspfIfPolName='ospf_intf_pol' />
    </ospfIfP>
    <bfdIfP name="test" type="sha1" key="hello" status="created,modified">
      <bfdRsIfPol tnBfdIfPolName='default' status="created,modified"/>
    </bfdIfP>
  </l3extLIIfP>

</l3extLNodeP>

```

Step 3 The following example shows the interface configuration for enabling BFD on BGP:

Example:

```

<fvTenant name="ExampleCorp">
  <l3extOut name="l3-out">
    <l3extLNodeP name="leaf1">
      <l3extRsNodeL3OutAtt tDn="topology/pod-1/node-101" rtrId="2.2.2.2"/>
      <l3extLIIfP name='portIpv4'>
        <l3extRsPathL3OutAtt tDn="topology/pod-1/paths-101/pathep-[eth1/11]" ifInstT='l3-port'
      addr="10.0.0.1/24" mtu="1500">
          <bgpPeerP addr="4.4.4.4/24" allowedSelfAsCnt="3" ctrl="bfd" descr="" name=""
peerCtrl="" ttl="1">
              <bgpRsPeerPfxPol tnBgpPeerPfxPolName="" />
              <bgpAsP asn="3" descr="" name="" />
            </bgpPeerP>
          </l3extRsPathL3OutAtt>
        </l3extLIIfP>

      </l3extLNodeP>
    </l3extOut>
  </fvTenant>

```

Step 4 The following example shows the interface configuration for enabling BFD on Static Routes:

Example:

BFD on leaf switch

```

<fvTenant name="ExampleCorp">
  <l3extOut name="l3-out">
    <l3extLNodeP name="leaf1">
      <l3extRsNodeL3OutAtt tDn="topology/pod-1/node-101" rtrId="2.2.2.2">
        <ipRouteP ip="192.168.3.4" rtCtrl="bfd">
          <ipNexthopP nhAddr="192.168.62.2"/>
        </ipRouteP>
      </l3extRsNodeL3OutAtt>
    </l3extLNodeP>
  </l3extOut>
</fvTenant>

```



```

        </ipRouteP>
    </l3extRsNodeL3OutAtt>
    <l3extLIfP name='portIpv4'>
        <l3extRsPathL3OutAtt tDn="topology/pod-1/paths-101/pathep-[eth1/3]" ifInstT='l3-port'
        addr="10.10.10.2/24" mtu="1500" status="created,modified" />
    </l3extLIfP>

    </l3extLNodeP>

</l3extOut>
</fvTenant>

```

Example:**BFD on spine switch**

```

<l3extLNodeP name="bSpine">

    <l3extRsNodeL3OutAtt tDn="topology/pod-1/node-103" rtrId="192.3.1.8">
        <ipRouteP ip="0.0.0.0" rtCtrl="bfd">
            <ipNextHopP nhAddr="192.168.62.2"/>
        </ipRouteP>
    </l3extRsNodeL3OutAtt>

    <l3extLIfP name='portIf'>
        <l3extRsPathL3OutAtt tDn="topology/pod-1/paths-103/pathep-[eth5/10]" encap='vlan-4'
        ifInstT='sub-interface' addr="20.3.10.1/24"/>

        <bfdIfP name="test" type="sha1" key="hello" status="created,modified">
            <bfdRsIfPol tnBfdIfPolName='default' status="created,modified"/>
        </bfdIfP>
    </l3extLIfP>

</l3extLNodeP>

```

Step 5 The following example shows the interface configuration for enabling BFD on IS-IS:

Example:

```

<fabricInst>
    <l3IfPol name="testL3IfPol" bfdIisis="enabled"/>
    <fabricLeafP name="LeNode" >
        <fabricRsLePortP tDn="uni/fabric/leportp-leaf_profile" />
    <fabricLeafS name="spsw" type="range">
    <fabricNodeBlk name="node101" to_"102" from_"101" />
</fabricLeafS>
    </fabricLeafP>

    <fabricSpineP name="SpNode" >
    <fabricRsSpPortP tDn="uni/fabric/sportp-spine_profile" />
    <fabricSpineS name="spsw" type="range">
        <fabricNodeBlk name="node103" to_"103" from_"103" />
    </fabricSpineS>
    </fabricSpineP>

    <fabricLePortP name="leaf_profile">
    <fabricLFPortS name="leafIf" type="range">
    <fabricPortBlk name="spBlk" fromCard="1" fromPort="49" toCard="1" toPort="49" />
        <fabricRsLePortPGrp tDn="uni/fabric/funcprof/leportgrp-LeTestPGrp" />
    </fabricLFPortS>
    </fabricLePortP>

    <fabricSpPortP name="spine_profile">
    <fabricSFPortS name="spineIf" type="range">

```

```
<fabricPortBlk name="spBlk" fromCard="5" fromPort="1" toCard="5" toPort="2" />
<fabricRsSpPortPGrp tDn="uni/fabric/funcprof/spportgrp-SpTestPGrp" />
</fabricSFPorts>
</fabricSpPortP>

<fabricFuncP>
  <fabricLePortPGrp name = "LeTestPGrp">
<fabricRsL3IfPol tnL3IfPolName="testL3IfPol"/>
  </fabricLePortPGrp>

  <fabricSpPortPGrp name = "SpTestPGrp">
<fabricRsL3IfPol tnL3IfPolName="testL3IfPol"/>
  </fabricSpPortPGrp>

</fabricFuncP>

</fabricInst>
```
