DHCP Relay in ACI

Overview, Configuration, Troubleshooting, and Caveats\Issues

Created by Tomas de Leon (ACI Solutions Delivery Team)

Version 6, updated November 17, 2016 (External Version)

Table of Contents

- **DHCP Relay Overview**
  - DHCP Relay in the MIT
  - Tenant DHCP Relay
  - DHCP Relay Modes

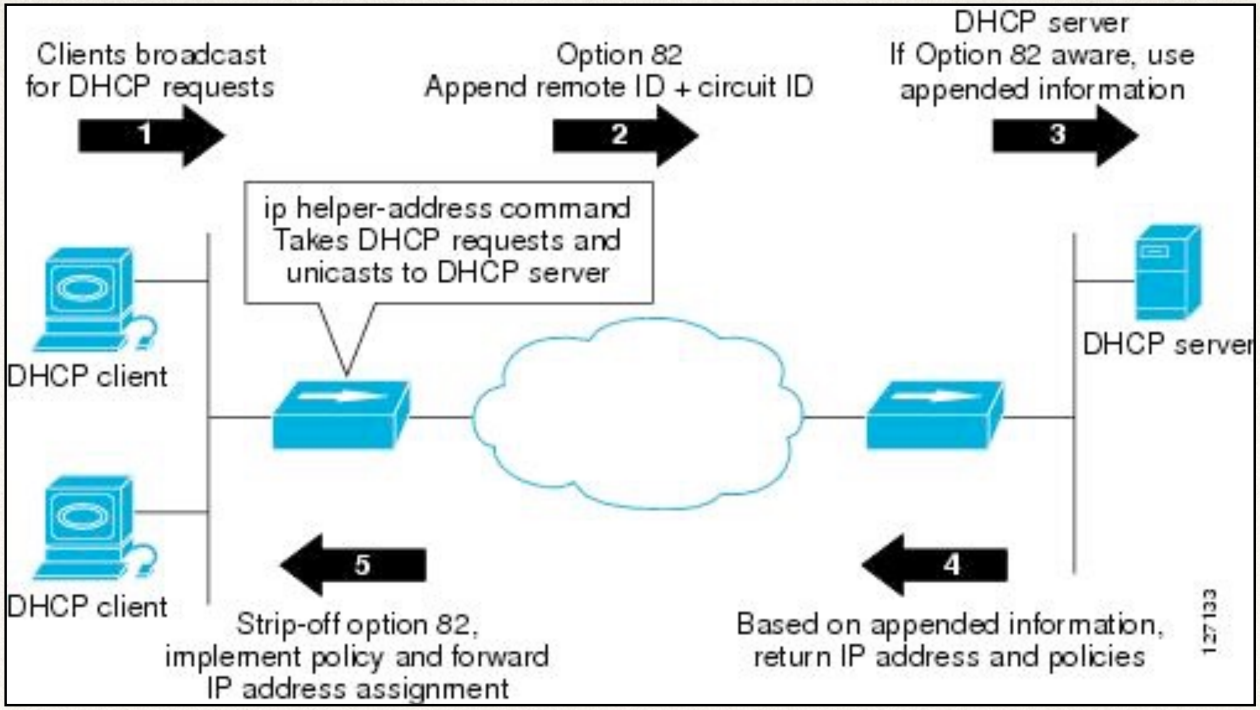
- **DHCP Relay Configuration**
  - Global DHCP Relay Configuration (Access)
  - Tenant DHCP Relay Configuration

Table of Contents (cont.)

- **DHCP Relay Troubleshooting**
    - Verify DHCP Relay Configuration
    - Debug Commands
    - Packet Traces

- **DHCP Relay Caveats - Issues**
    - DHCP Option 82
    - Bridge Domains - Subnets

- **References & Resources**


Clients broadcast for DHCP requests
1
Option 82
Append remote ID + circuit ID
2
DHCP server
If Option 82 aware, use appended information
3
ip helper-address command
Takes DHCP requests and unicasts to DHCP server
DHCP client
DHCP server
DHCP client
5
4
Strip-off option 82, implement policy and forward IP address assignment
Based on appended information, return IP address and policies
127133

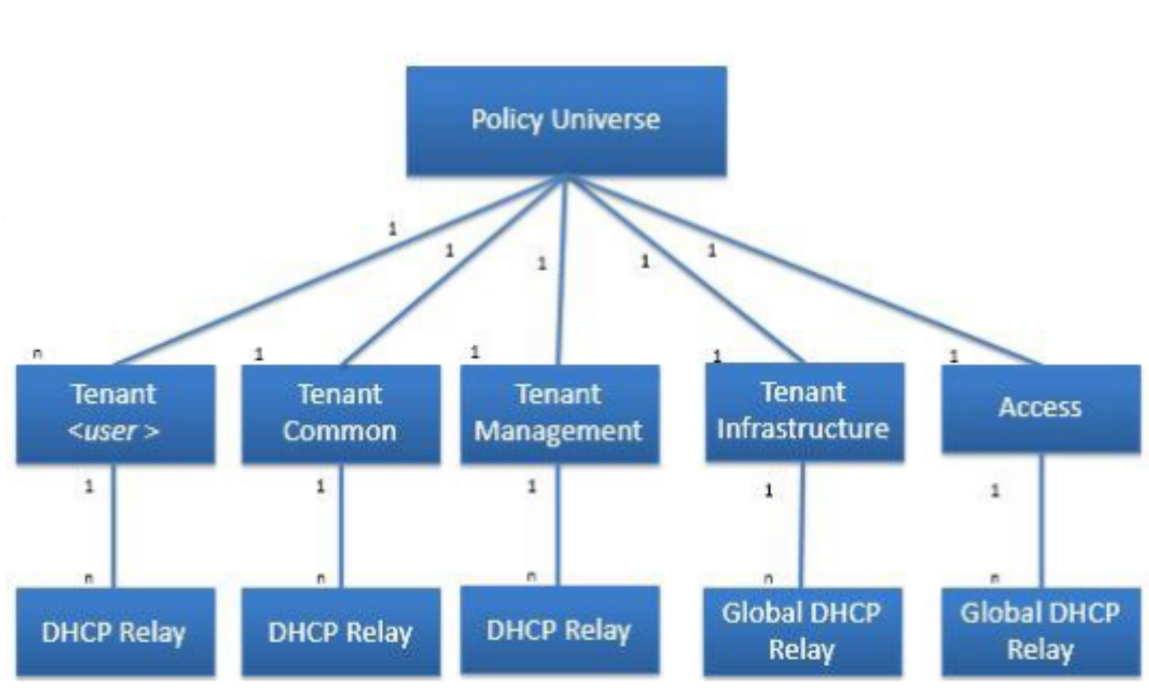
DHCP Relay Overview

While ACI fabric-wide flooding is disabled by default, flooding within a bridge domain is enabled by default. Because flooding within a bridge domain is enabled by default, clients can connect to DHCP servers within the same EPG. *However, when the DHCP server is in a different EPG, BD, or context (VRF) than the clients, DHCP Relay is required. Also, when Layer 2 flooding is disabled, DHCP Relay is required.*

DHCP Relay in the MIT

- The figure 7. shows the managed objects in the management information tree (MIT) for DHCP Relay policies.
  - User Tenant
  - Common Tenant
  - Management Tenant
  - Infrastructure Tenant
  - Fabric Access

Figure 7. DHCP Relay Locations in the MIT



DHCP Relay in the MIT (cont.)

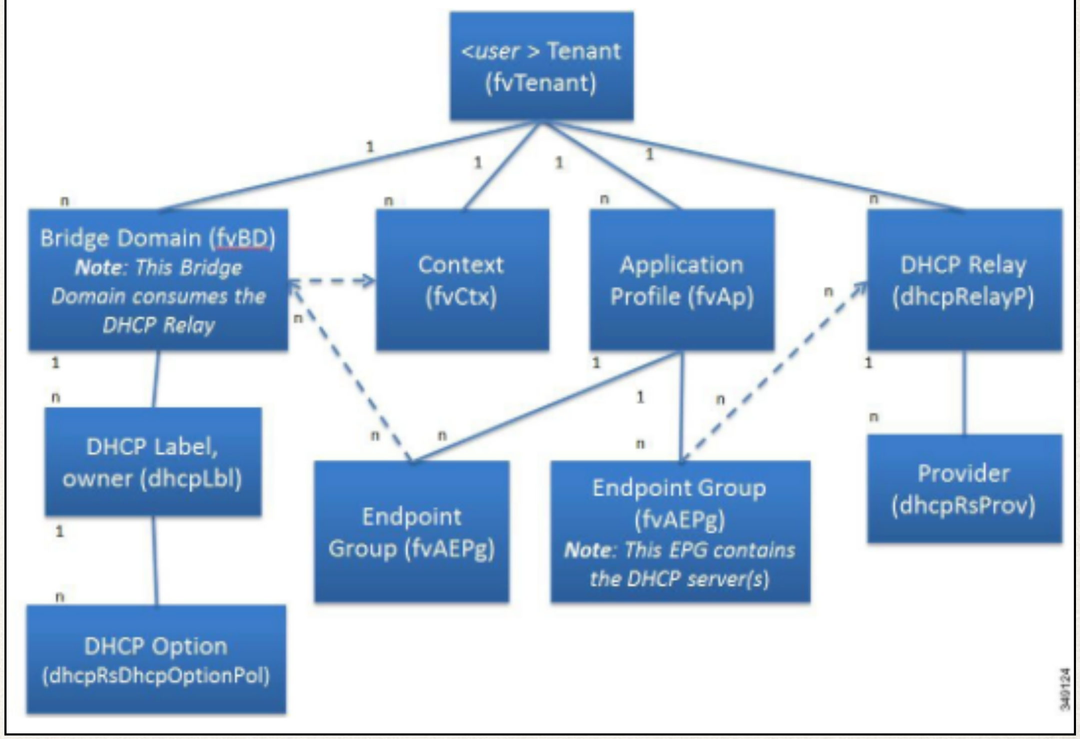
- What is the difference of the DHCP Relay Locations in the MIT?
    - Common Tenant DHCP Relay policies can be used by any tenant
    - Infrastructure Tenant DHCP Relay policies are exposed selectively by the ACI fabric service provider to other tenants
    - Fabric Access (infra:Infra) DHCP Relay Policies can be used by any tenant and they allow more granular configuration of the DHCP servers. In this case, it is possible to provision separate DHCP servers within the same bridge domain in the node profile.

Tenant DHCP Relay

- The figure 8. Tenant DHCP Relay shows the logical relationships of the DHCP Relay objects within a user tenant.

The DHCP Relay profile contains one or more providers. An EPG contains one or more DHCP servers, and the relation between the EPG and the DHCP Relay specifies the DHCP server ip address. The consumer bridge domain contains the DHCP label that associates the provider DHCP server with the bridge domain. Label matching enables the bridge domain to consume the DHCP Relay policy.

Figure 8. Tenant DHCP Relay



<user > Tenant (fvTenant)
Bridge Domain (fvBD)
Note: This Bridge Domain consumes the DHCP Relay
Context (fvCtx)
Application Profile (fvAp)
DHCP Relay (dhcpRelayP)
DHCP Label, owner (dhcpLbl)
Endpoint Group (fvAEPg)
Endpoint Group (fvAEPg)
Note: This EPG contains the DHCP server(s)
Provider (dhcpRsProv)
DHCP Option (dhcpRsDhcpOptionPol)
1
n
340124

Note: the bridge domain DHCP label MUST match the DHCP Relay name. The DHCP Label object also specifies the owner. The owner can be a tenant or the access infrastructure. If the owner is a tenant, the ACI fabric first looks within the tenant for a matching DHCP Relay policy. If there is no match within the user tenant, the ACI fabric then looks in the common tenant.

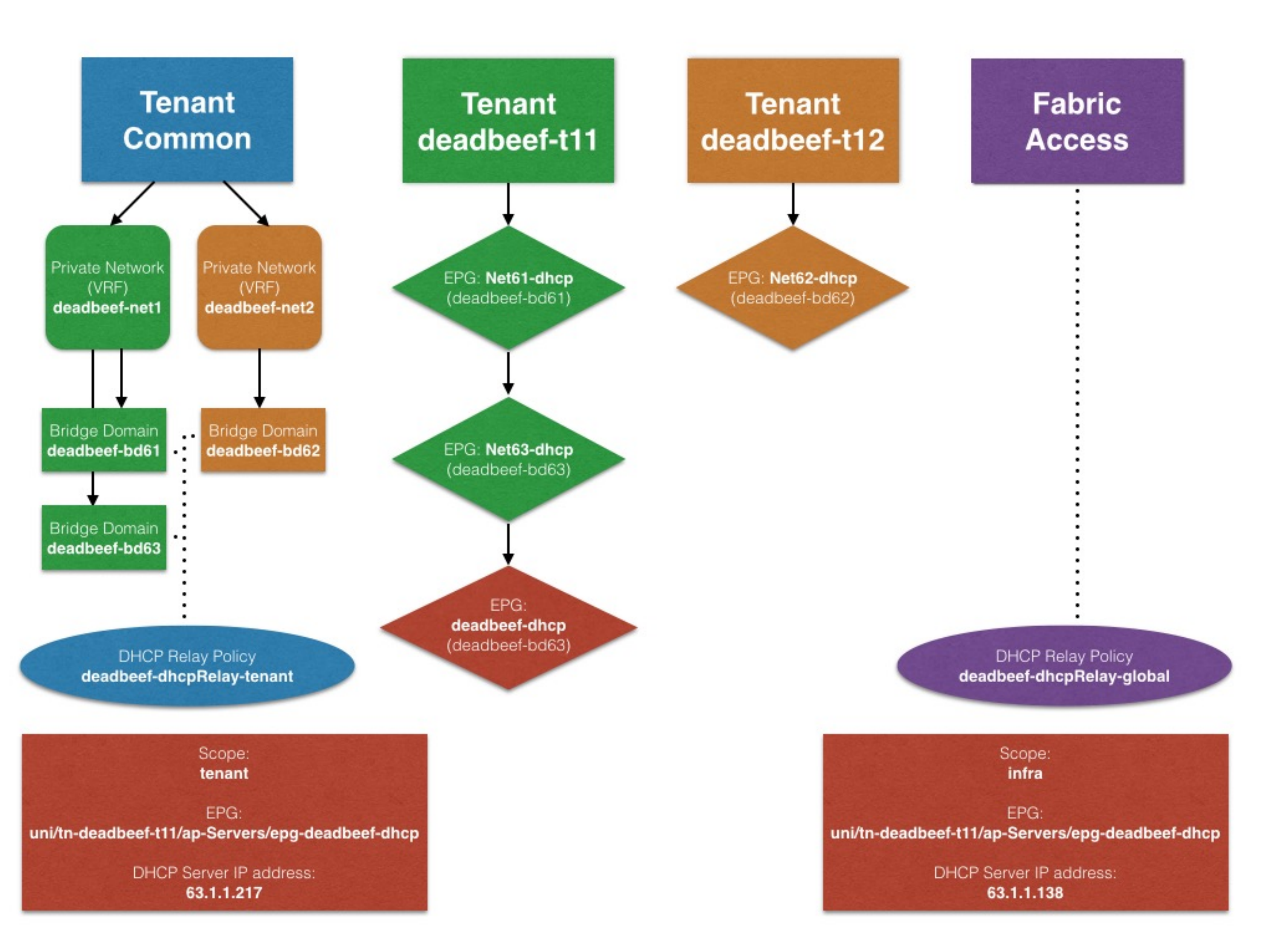
DHCP Relay Modes

- DHCP Relay operates in one of the following two modes:
  - Visible - the provider's ip address and subnet are leaked into the consumer's context. When the DHCP Relay is visible, it is exclusive to the consumer's context.
  - Not Visible - the provider's ip address and subnet are not leaked into the consumer's context.

    Note: *When the DHCP Relay operates in the not visible mode, the bridge domain of the provider must be on the same leaf switch as the consumer.*

DHCP Relay Configuration

For this topic, I will demonstrate configuring DHCP Relay as a Global Policy and as a Tenant Policy. The DHCP servers are located in a separate EPG. The DHCP clients will be in different BDs, EPGs, and Contexts (VRFs).


Tenant Common
Tenant deadbeef-t11
Tenant deadbeef-t12
Fabric Access
Private Network (VRF) deadbeef-net1
Private Network (VRF) deadbeef-net2
EPG: Net61-dhcp (deadbeef-bd61)
EPG: Net62-dhcp (deadbeef-bd62)
Bridge Domain deadbeef-bd61
Bridge Domain deadbeef-bd62
EPG: Net63-dhcp (deadbeef-bd63)
Bridge Domain deadbeef-bd63
EPG: deadbeef-dhcp (deadbeef-bd63)
DHCP Relay Policy deadbeef-dhcpRelay-tenant
DHCP Relay Policy deadbeef-dhcpRelay-global
Scope: tenant
EPG: uni/tn-deadbeef-t11/ap-Servers/epg-deadbeef-dhcp
DHCP Server IP address: 63.1.1.217
Scope: infra
EPG: uni/tn-deadbeef-t11/ap-Servers/epg-deadbeef-dhcp
DHCP Server IP address: 63.1.1.138

DHCP Relay Topology Example

The chart shown is the topology used for providing configuration examples in this presentation.

DHCP Relay Topology Overview

When configuring "shared" resources and services in the ACI fabric, it is best practice to create these managed objects in the Tenant Common. Shared resources and services in the Common Tenant can be used by any tenant. The goal of this lab topology is to provide examples of configurations which demonstrate DHCP Relay in a *multi-tenant* and *multi-context (VRF)* environment.

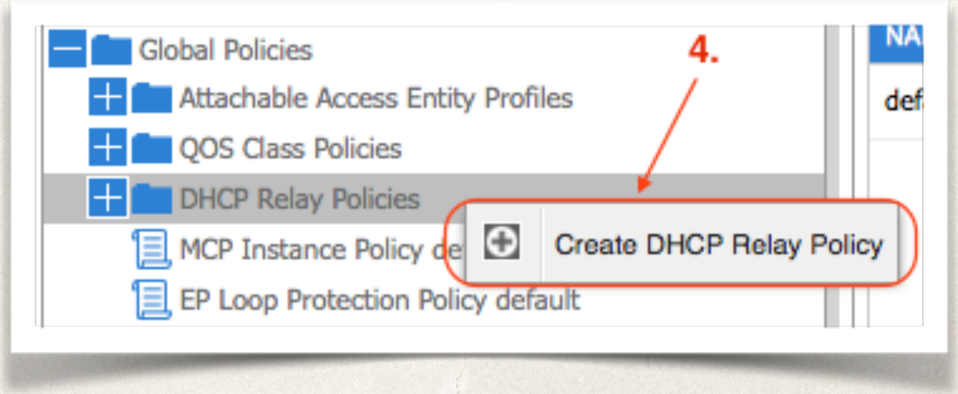
As shown in the previous slide; two private networks, three bridge domains, and the tenant DHCP Relay policy are configured in the Tenant Common. Two separate tenants (*deadbeef-t11 and deadbeef-t12*) are used for defining and segmenting endpoints into the appropriate End Point Groups (EPGs).

For this DHCP Relay configuration example, an assumption is made that the Tenants, BDs, Private Networks, Contracts, OOB mgmt addresses, and Route-leaking are already configured and verified. In order to show different DHCP deployment scenarios; *Microsoft Windows Server 2008*, *Microsoft Windows Server 2012*, and *CentOS 6.5* are used as DHCP Servers.

Global DHCP Relay Configuration (Access)

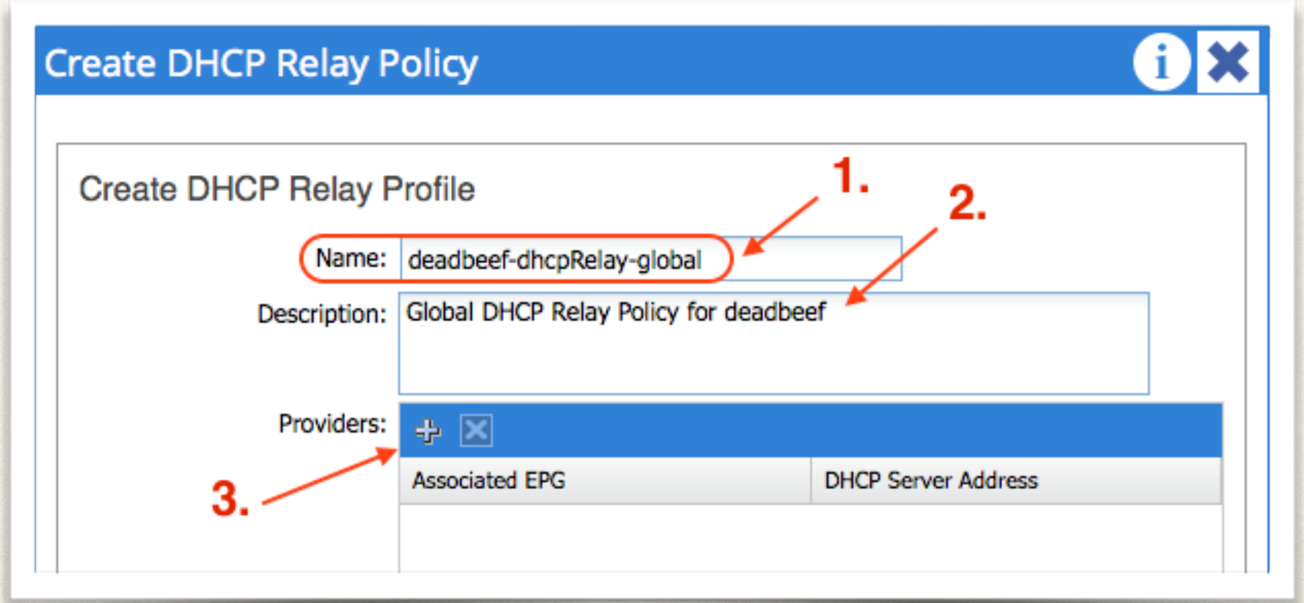
- Use case example of configuring a Global DHCP Relay Policy. The goal is for all DHCP clients in all Tenants to use the same DHCP Server. In this scenario, the DHCP provider is 63.1.1.138 (*Microsoft Windows 2012 Server*)

- Configuration Steps:
  1. Access the APIC Admin GUI.
  2. Select FABRIC -> ACCESS POLICIES.
  3. In the policies navigation panel on the left, select and expand the GLOBAL POLICIES -> DHCP RELAY POLICIES.
  4. Right Click and Select CREATE DHCP RELAY POLICY



Global DHCP Relay Configuration (Access) (cont.)

- **In the Create DHCP Relay Policy Wizard, Create a DHCP Relay Policy:**
  1. Enter DHCP Relay Profile NAME.
  2. Enter DHCP Relay Profile DESCRIPTION.
  3. Click on "+" to add a DHCP Relay PROVIDER.



Create DHCP Relay Policy

Create DHCP Relay Profile

Name: deadbeef-dhcpRelay-global

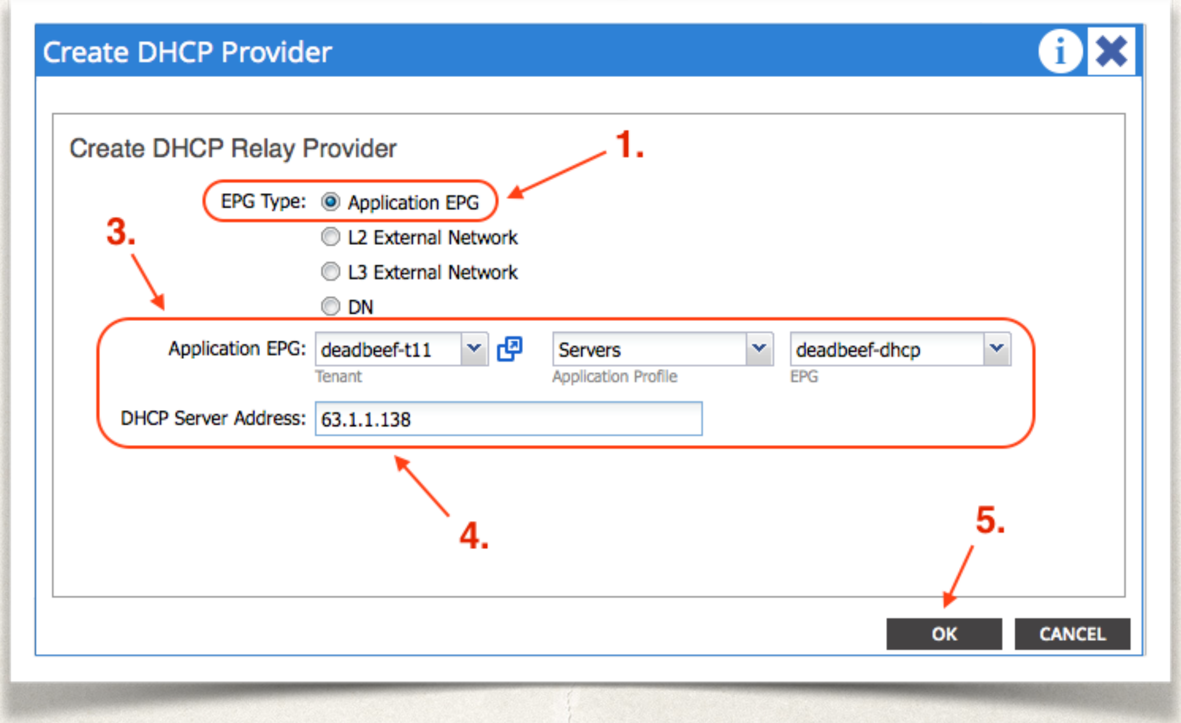
Description: Global DHCP Relay Policy for deadbeef

Providers:

Associated EPG	DHCP Server Address

Global DHCP Relay Configuration (Access) (cont.)

- **In the Create DHCP Provider Wizard, Create a DHCP Relay Provider:**
  1. Select the EPG Type for the provider.
  2. For this use case example, the EPG Type is APPLICATION EPG.
  3. Select APPLICATION EPG in which the DHCP provider is located.
  4. Enter the DHCP Server Address (*63.1.1.138*).
  5. Click OK when finished.



Create DHCP Provider

Create DHCP Relay Provider

EPG Type: ● Application EPG
○ L2 External Network
○ L3 External Network
○ DN

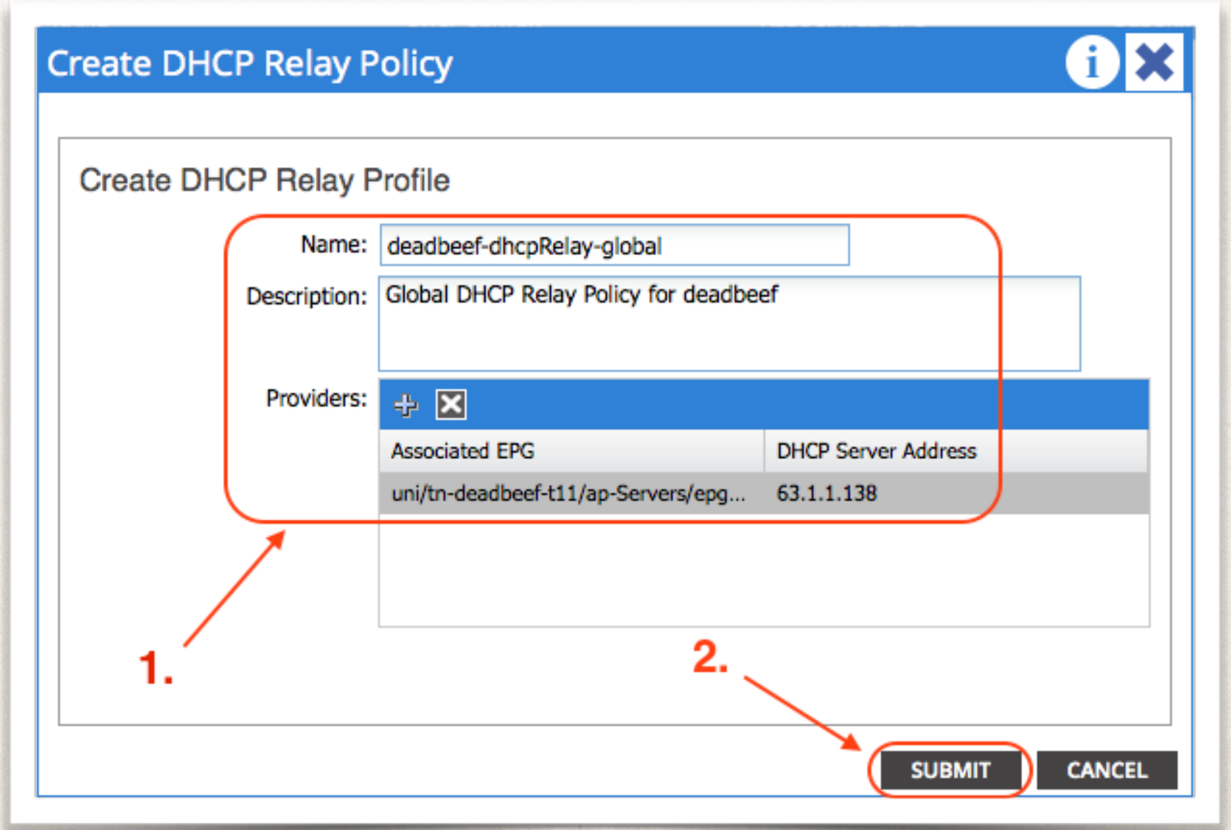
Application EPG: deadbeef-t11 (Tenant) | Servers (Application Profile) | deadbeef-dhcp (EPG)

DHCP Server Address: 63.1.1.138

Global DHCP Relay Configuration (Access) (cont.)

- **In the Create DHCP Relay Policy Wizard, verify configured parameters:**
  1. Verify NAME, DESCRIPTION, and PROVIDERS are correct.
  2. Click SUBMIT to complete creation of the DHCP Relay Policy.

Note:
Repeat previous steps to Create multiple DHCP Relay Policies if needed.



Create DHCP Relay Policy

Create DHCP Relay Profile

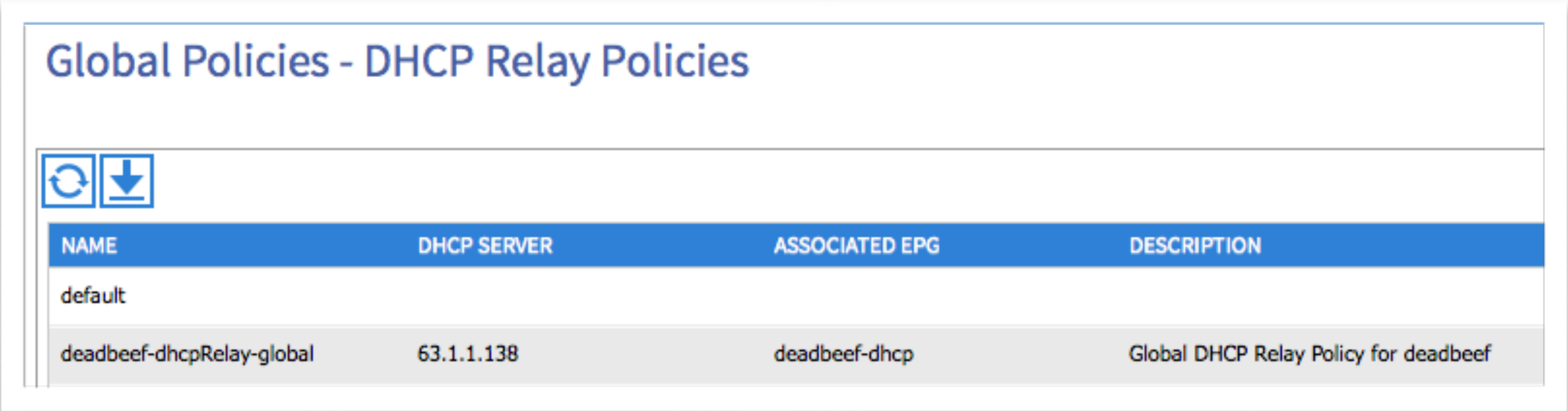
Name: deadbeef-dhcpRelay-global

Description: Global DHCP Relay Policy for deadbeef

Providers:

Associated EPG	DHCP Server Address
uni/tn-deadbeef-t11/ap-Servers/epg...	63.1.1.138

Global DHCP Relay Configuration (Access) (cont.)



Global Policies - DHCP Relay Policies

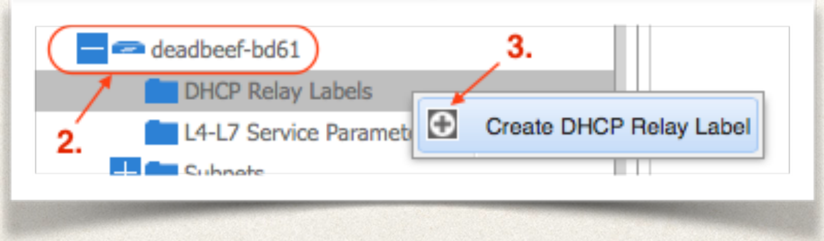
NAME	DHCP SERVER	ASSOCIATED EPG	DESCRIPTION
default			
deadbeef-dhcpRelay-global	63.1.1.138	deadbeef-dhcp	Global DHCP Relay Policy for deadbeef

- As mentioned earlier, the consumer bridge domain contains the DHCP label that associates the provider DHCP server with the bridge domain. Label matching enables the bridge domain to consume the DHCP Relay policy.
- After configuring the DHCP Relay policies, you will need to create a DHCP Relay Label for the consumer Bridge Domains.

Global DHCP Relay Configuration (Access) (cont.)

- **Create a DHCP Relay Label:**

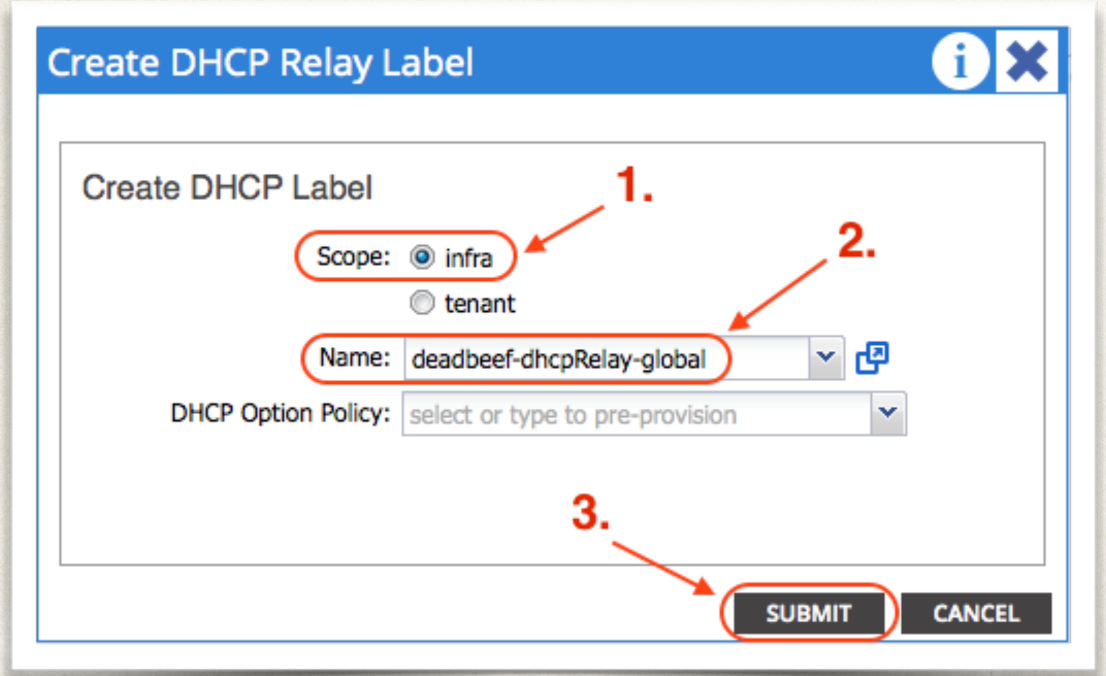
1. Navigate to the desired TENANT in which you want to apply the Global DHCP Relay Policy.
2. In the TENANT navigation panel, select NETWORKING -> BRIDGE DOMAINS -> Desired BD to add the DHCP Relay policy.
3. Right Click on the DHCP RELAY LABELS and select CREATE DHCP RELAY LABEL.
4. The CREATE DHCP RELAY LABEL WIZARD will be presented.



Global DHCP Relay Configuration (Access) (cont.)

- **Create a DHCP Relay Label Wizard:**
  1. Select SCOPE "**infra**" since this is a Global DHCP Relay Policy.
  2. Select the desired Global DHCP Relay Policy that you created earlier (*deadbeef-dhcpRelay-global*) in the drop down list.
  3. Click SUBMIT to complete the creation of the DHCP LABEL for the selected Bridge Domain.

Note: Repeat the steps for additional Bridge Domains that need to use a DHCP Relay Policy.



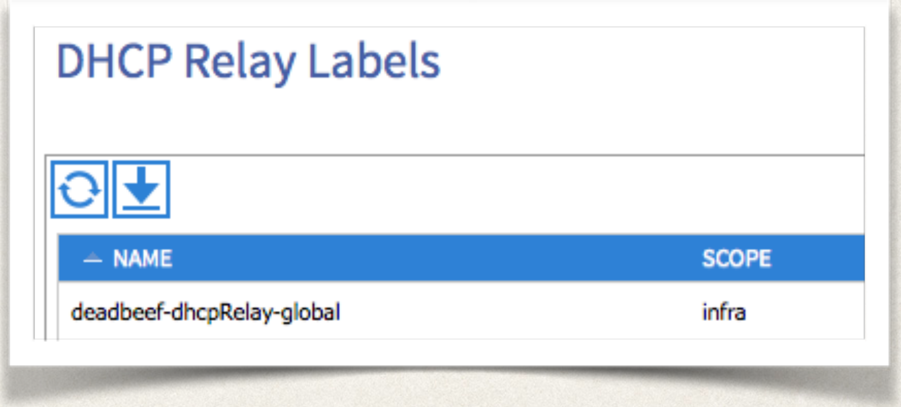
Create DHCP Relay Label

Create DHCP Label

Scope: ● infra ○ tenant

Name: deadbeef-dhcpRelay-global

DHCP Option Policy: select or type to pre-provision



DHCP Relay Labels

NAME	SCOPE
deadbeef-dhcpRelay-global	infra

Tenant DHCP Relay Configuration

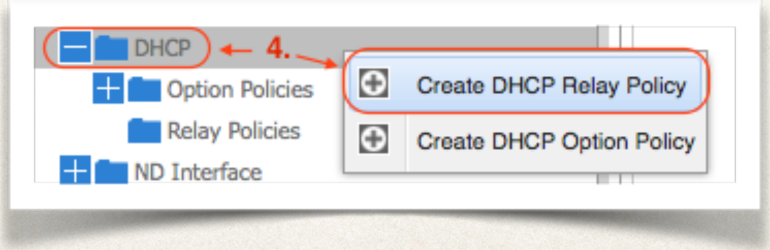
- Use case example of configuring a Tenant DHCP Relay Policy. The goal is for all DHCP clients in all Tenants to use the same DHCP Server. In this scenario, the DHCP provider is 63.1.1.217 (*Linux CentOS 6.5 DHCP Server*).

- Instead of configuring a "Global" DHCP Relay Policy, this use case scenario uses the Tenant Common which contains the Bridge Domains & Contexts (VRFs). The Client & Server EPGs are configured in separate Tenants but associate back to the Bridge Domains in Tenant Common.

Tenant DHCP Relay Configuration (cont.)

- **Configuring a Tenant DHCP Relay configuration policy:**

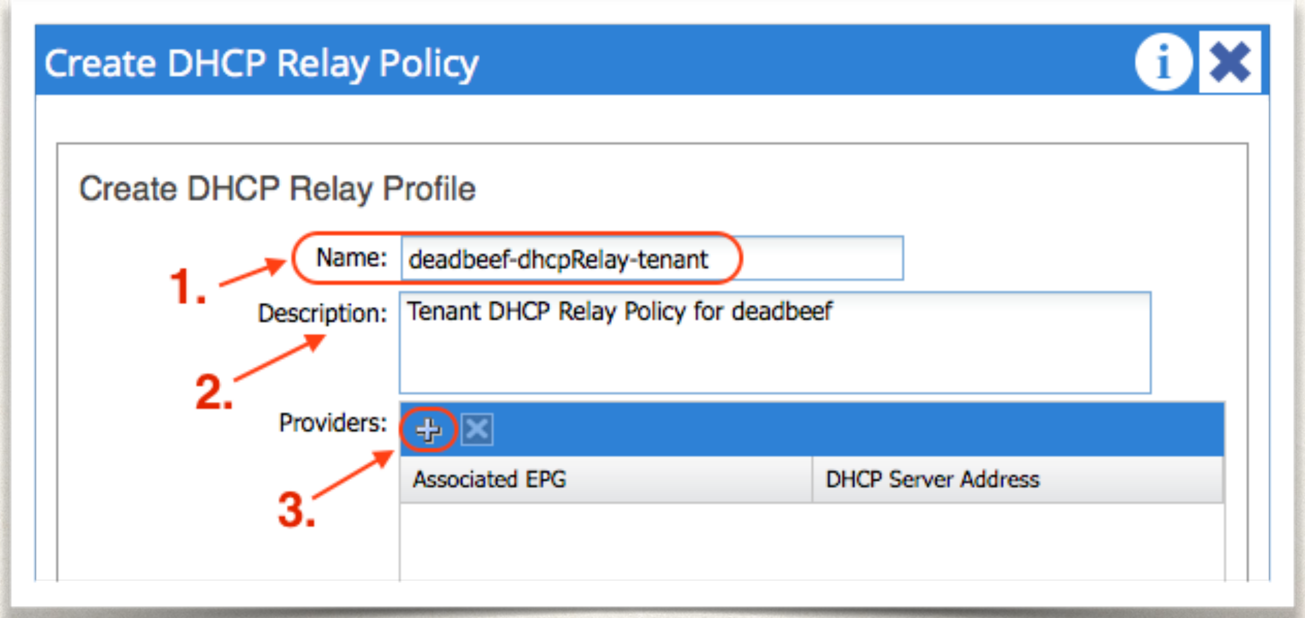
1. Access the APIC Admin GUI.
2. Select TENANTS -> COMMON.
3. In the navigation panel on the left, select and expand NETWORKING -> PROTOCOL POLICIES.
4. Select DHCP, Right Click and Select CREATE DHCP RELAY POLICY.
5. The CREATE DHCP RELAY POLICY WIZARD will be presented.



Tenant DHCP Relay Configuration (cont.)

- **Create DHCP Relay Policy Wizard:**

  1. Enter DHCP Relay Policy NAME.
  2. Add a DESCRIPTION.
  3. Click "+" to add a DHCP Relay Provider.
  4. The CREATE DHCP RELAY PROVIDER WIZARD will be presented.



Create DHCP Relay Policy

Create DHCP Relay Profile

1. Name: deadbeef-dhcpRelay-tenant

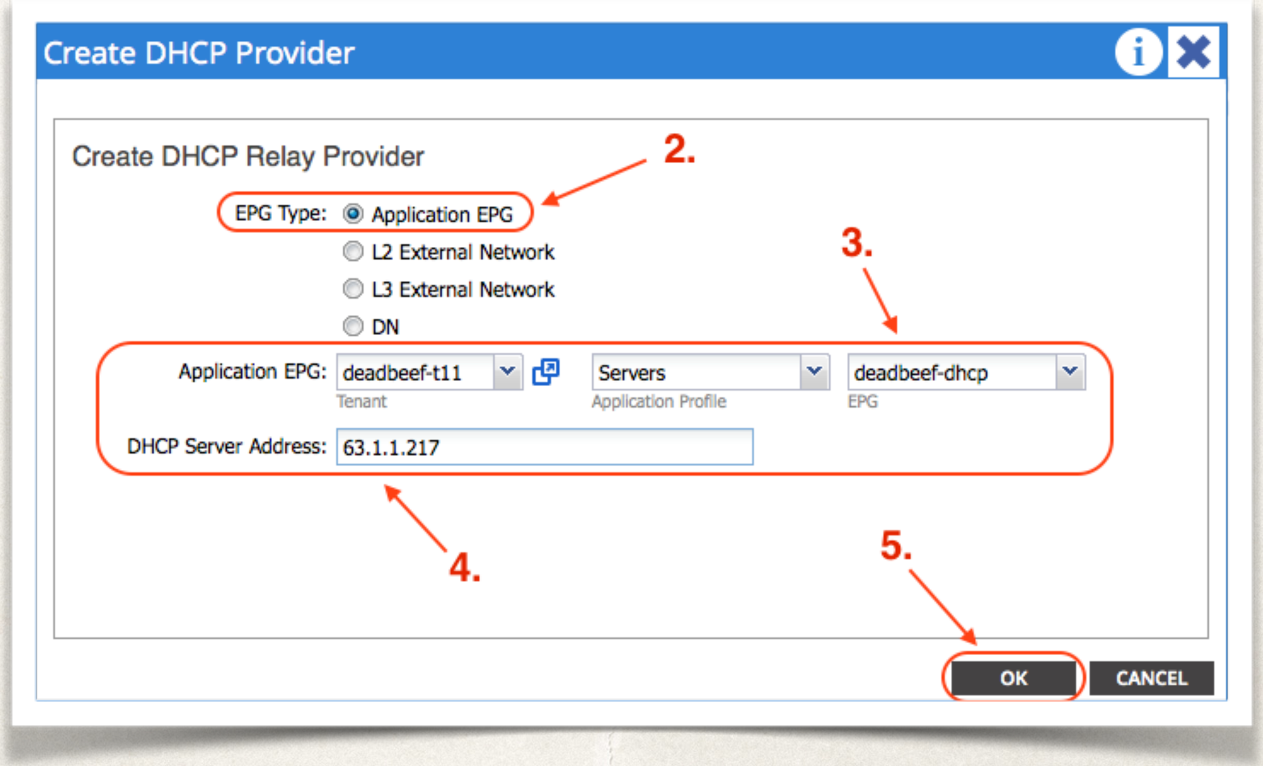
2. Description: Tenant DHCP Relay Policy for deadbeef

3. Providers:

Associated EPG	DHCP Server Address

Tenant DHCP Relay Configuration (cont.)

- **<u>Create DHCP Relay Provider Wizard:</u>**
  1. Select the EPG Type for the provider.
  2. For this use case example, the EPG Type is APPLICATION EPG.
  3. Select APPLICATION EPG in which the DHCP provider is located.
  4. Enter the DHCP Server Address (*63.1.1.217*).
  5. Click OK when finished.



Create DHCP Provider

Create DHCP Relay Provider

EPG Type: ● Application EPG
○ L2 External Network
○ L3 External Network
○ DN

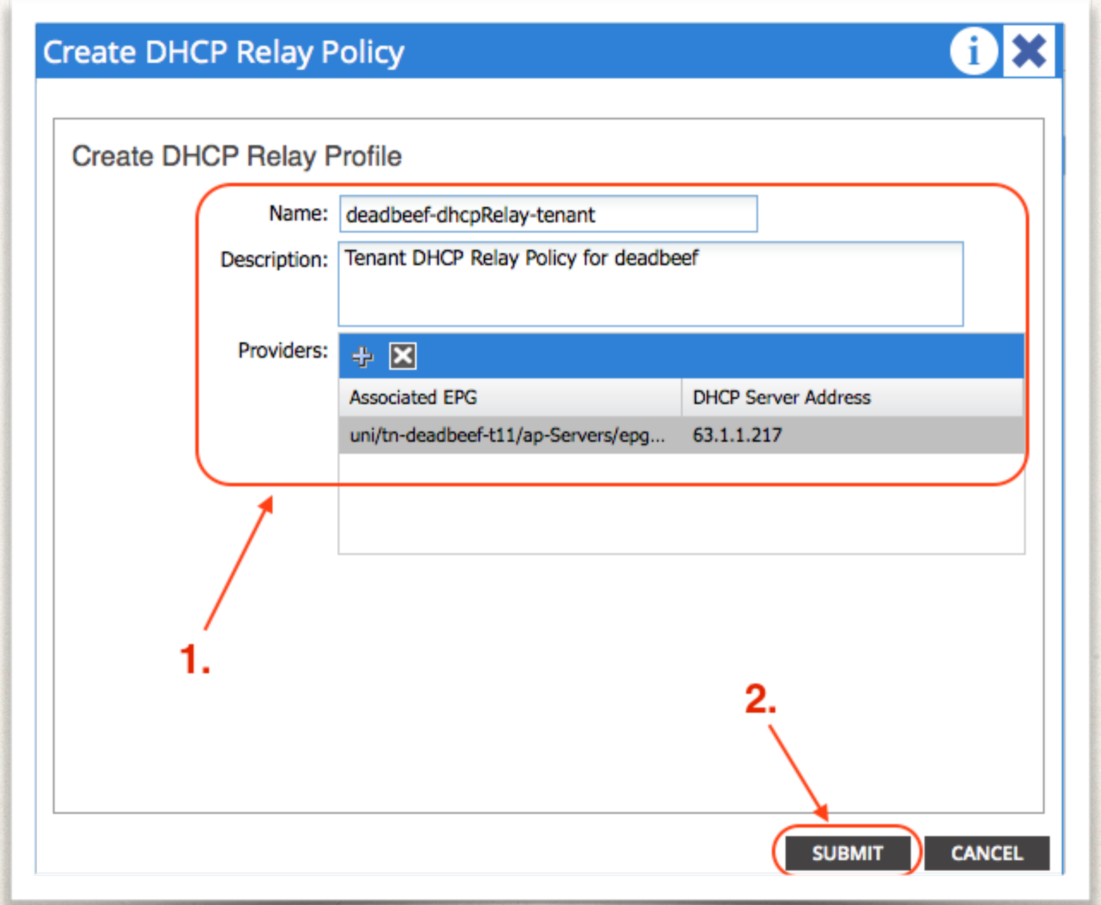
Application EPG: deadbeef-t11 (Tenant) | Servers (Application Profile) | deadbeef-dhcp (EPG)

DHCP Server Address: 63.1.1.217

Tenant DHCP Relay Configuration (cont.)

- **<u>In the Create DHCP Relay Policy Wizard, verify configured parameters:</u>**
    1. Verify NAME, DESCRIPTION, and PROVIDERS are correct.
    2. Click SUBMIT to complete creation of the DHCP Relay Policy.

Note:
Repeat previous steps to Create multiple DHCP Relay Policies if needed.



Create DHCP Relay Policy

Create DHCP Relay Profile

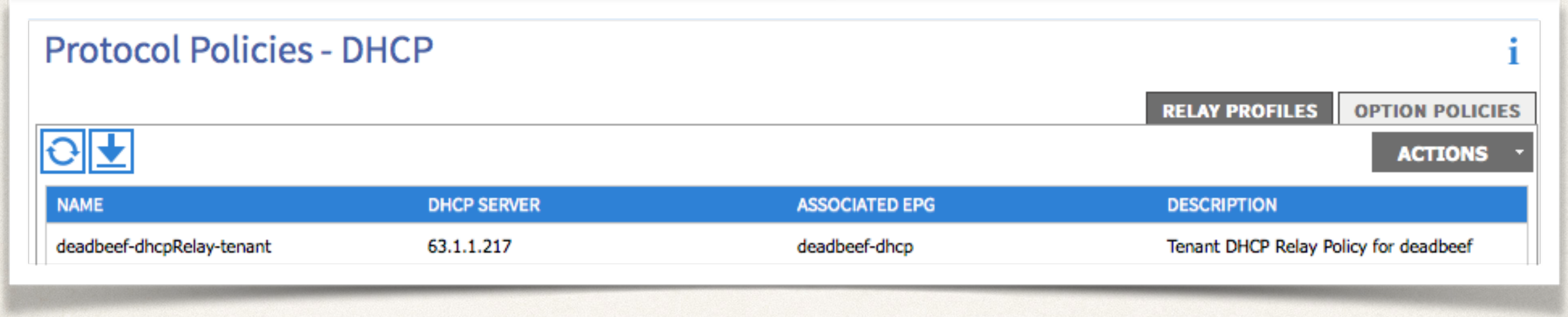
Name: deadbeef-dhcpRelay-tenant

Description: Tenant DHCP Relay Policy for deadbeef

Providers:

Associated EPG	DHCP Server Address
uni/tn-deadbeef-t11/ap-Servers/epg...	63.1.1.217

Tenant DHCP Relay Configuration (cont.)



Protocol Policies - DHCP

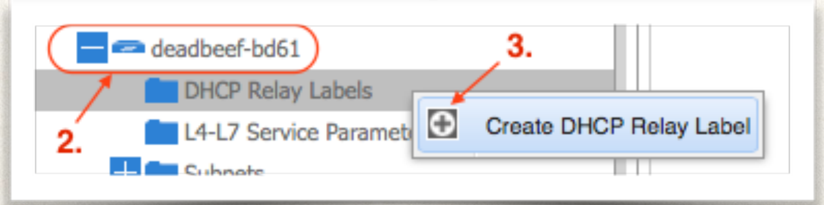
NAME	DHCP SERVER	ASSOCIATED EPG	DESCRIPTION
deadbeef-dhcpRelay-tenant	63.1.1.217	deadbeef-dhcp	Tenant DHCP Relay Policy for deadbeef

- As mentioned earlier, the consumer bridge domain contains the DHCP label that associates the provider DHCP server with the bridge domain. Label matching enables the bridge domain to consume the DHCP Relay policy.
- After configuring the DHCP Relay policies, you will need to create a DHCP Relay Label for the consumer Bridge Domains.

Tenant DHCP Relay Configuration (cont.)

- **Create a DHCP Relay Label:**

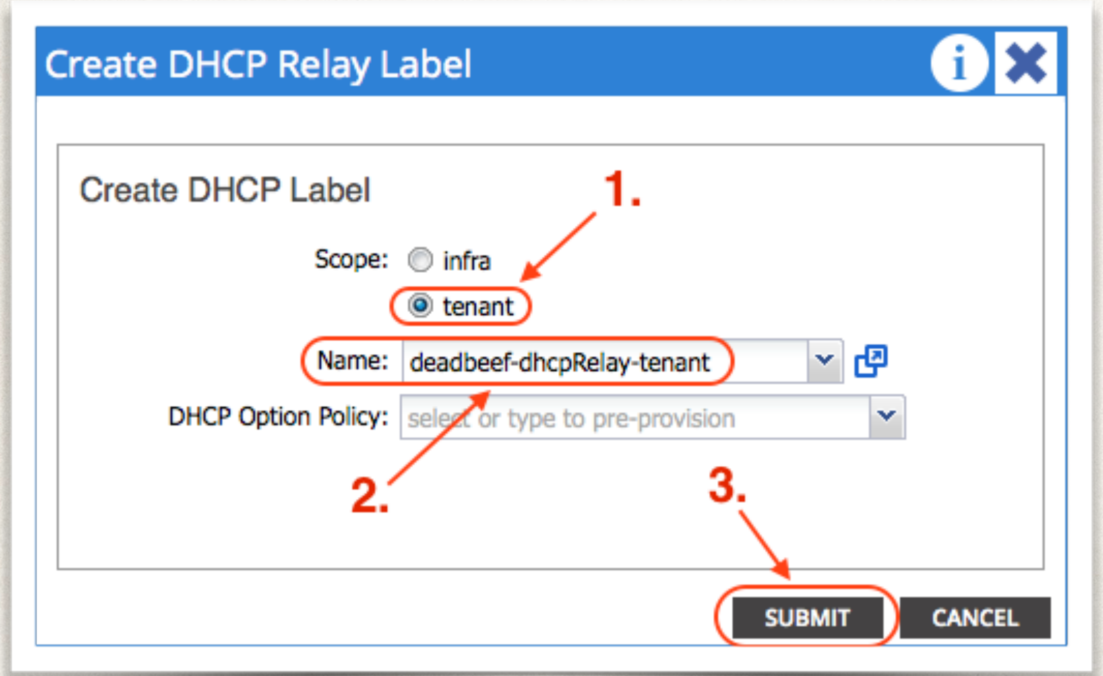
1. Navigate to the desired TENANT (*Common*) in which you want to apply the Tenant DHCP Relay Policy.
2. In the TENANT (*Common*) navigation panel, select NETWORKING -> BRIDGE DOMAINS -> Desired BD to add the DHCP Relay policy.
3. Right Click on the DHCP RELAY LABELS and select CREATE DHCP RELAY LABEL.
4. The CREATE DHCP RELAY LABEL WIZARD will be presented.



Tenant DHCP Relay Configuration (cont.)

- **<u>Create a DHCP Relay Label Wizard:</u>**
  1. Select SCOPE "**tenant**" since this is a Tenant DHCP Relay Policy.
  2. Select the desired Tenant DHCP Relay Policy that you created earlier (*deadbeef-dhcpRelay-tenant*) in the drop down list.
  3. Click SUBMIT to complete the creation of the DHCP LABEL for the selected Bridge Domain.

Note: Repeat the steps for additional Bridge Domains that need to use a DHCP Relay Policy.



Create DHCP Relay Label

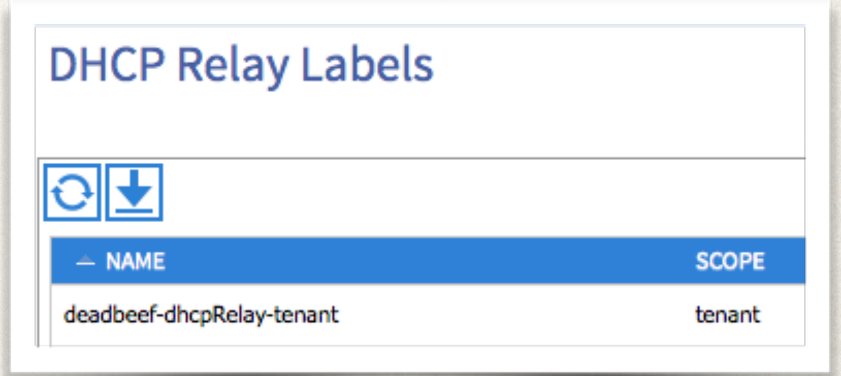
Create DHCP Label

1. Scope: ○ infra ● tenant

Name: deadbeef-dhcpRelay-tenant

2. DHCP Option Policy: select or type to pre-provision

3. SUBMIT



DHCP Relay Labels

NAME	SCOPE
deadbeef-dhcpRelay-tenant	tenant

DHCP Relay Troubleshooting

This section will provide an overview on generic troubleshooting DHCP Relay policies in the ACI Fabric. Once DHCP Relay policies are configured for Global Access and Tenants, verify that the configuration is pushed to the LEAF switches. Use the available CLI commands to verify configuration is enabled and applied. If needed, use of external tools and apps may be necessary.

Verify DHCP Relay Configuration

- **After completing the configuration of DHCP-Relay policies, verify configuration on Leaf Nodes. Note: You only have to check the Leaf Nodes that have endpoints which will be using the DHCP Relay services.**

  1. SSH to a Fabric APIC. Use the "*attach node-name*" command to connect to the desired Leaf Node.
  2. On each Leaf with DHCP-Relay configured, run "*show ip dhcp relay*". The output will verify that the "*DHCP relay service is enabled*". The output will also show the "*IP Helper Address*" information for the Leaf.

For Example:

```
fab2-leaf3# show ip dhcp relay
DHCP relay service is enabled
Insertion of option 82 is enabled
Insertion of cisco suboptions is disabled

Helper addresses are configured on the following interfaces:
 Interface          Relay Address      VRF Name
 -------------      -------------      ---------
 Vlan14             63.1.1.217         common:deadbeef-net1
 Vlan20             63.1.1.217         common:deadbeef-net1
 Vlan22             63.1.1.217         common:deadbeef-net1
```

Note: Repeat the "show ip dhcp relay" command on each Leaf node supporting DHCP Client endpoints.

Verify DHCP Relay Configuration (cont.)

- **Use the output from the "*show ip dhcp relay*" command to retrieve more detailed information on the DHCP Relay interfaces. Use the command "*show dhcp internal info relay address interface [leaf:interfaceVlan#]*".**

For Example:

```
fab2-leaf3# show dhcp internal info relay address interface vlan14
DHCP relay intf Vlan14 has 1 relay addresses:
DHCP relay addr: 63.1.1.217, vrf: common:deadbeef-net1, visible, gateway IP: 63.1.1.1

fab2-leaf3# show dhcp internal info relay address interface vlan20
DHCP relay intf Vlan20 has 1 relay addresses:
DHCP relay addr: 63.1.1.217, vrf: common:deadbeef-net1, visible, gateway IP: 63.1.1.1

fab2-leaf3# show dhcp internal info relay address interface vlan22
DHCP relay intf Vlan22 has 1 relay addresses:
DHCP relay addr: 63.1.1.217, vrf: common:deadbeef-net1, visible, gateway IP: 63.1.1.1
```

Note: Repeat the "show dhcp internal info relay address interface [leaf:interfaceVlan#]" command on each Leaf node supporting DHCP Client endpoints.

Verify DHCP Relay Configuration (cont.)

- **On each Leaf with DHCP Relay configured run "*show dhcp internal info relay discover*". This command will display any Custom DHCP option definitions configured for the DHCP Relay policies.**

For Example:

```
fab2-leaf3# show dhcp internal info relay discover
DHCP Relay Option Definition Information:
DHCP relay intf Vlan14 has 0 option defs
DHCP relay intf Vlan20 has 0 option defs
DHCP relay intf Vlan22 has 0 option defs

fab2-leaf4# show dhcp internal info relay discover
DHCP Relay Option Definition Information:
DHCP relay intf Vlan9 has 0 option defs
DHCP relay intf Vlan10 has 0 option defs
DHCP relay intf Vlan11 has 0 option defs
```

Note: Repeat the "show dhcp internal info relay discover" command on each Leaf node supporting DHCP Client endpoints.

Verify DHCP Relay Configuration (cont.)

- **Managed Object(MO) Queries is another way to verify configuration of DHCP Relay Policies. On each Leaf with DHCP Relay configured run “*moquery -c [object class]*” ie. (dhcpRelayP, dhcpProvDhcp, dhcpRtLblDefToRelayP).**

dhcpRelayP

```
fab2-leaf3# moquery -c dhcpRelayP

# dhcp.RelayP
name            : deadbeef-dhcpRelay-tenant
childAction     :
descr           : Tenant DHCP Relay Policy for deadbeef
dn              : uni/tn-common/relayp-deadbeef-dhcpRelay-tenant
lcOwn           : policy
modTs           : 2015-06-21T19:56:43.893-04:00
mode            : visible
monPolDn        : uni/tn-common/monepg-default
owner           : infra
ownerKey        :
ownerTag        :
rn              : relayp-deadbeef-dhcpRelay-tenant
status          :
uid             : 15374
```

Note: Repeat the “moquery -c dhcpRelayP” command on each Leaf node supporting DHCP Client endpoints.

Verify DHCP Relay Configuration (cont.)

dhcpProvDhcp

```
fab2-leaf3# moquery -c dhcpProvDhcp
```

```
# dhcp.ProvDhcp
epgDn         : uni/tn-deadbeef-t11/ap-Servers/epg-deadbeef-dhcp
addr          : 63.1.1.217
bdDefDn       : uni/bd-[uni/tn-common/BD-deadbeef-bd63]-isSvc-no
bdDefStQual   : none
childAction   :
ctxDefDn      : uni/ctx-[uni/tn-common/ctx-deadbeef-net1]
ctxDefStQual  : none
ctxSeg        : 2588672
descr         :
dn            : uni/tn-common/relayp-deadbeef-dhcpRelay-tenant/provdhcp-[uni/tn-deadbeef-t11/ap-Servers/epg-deadbeef-dhcp]
l3CtxEncap    : vxlan-2588672
lcOwn         : policy
modTs         : 2015-06-21T19:56:43.893-04:00
monPolDn      : uni/tn-common/monepg-default
name          : deadbeef-dhcp
ownerKey      :
ownerTag      :
pcTag         : 5477
rn            : provdhcp-[uni/tn-deadbeef-t11/ap-Servers/epg-deadbeef-dhcp]
scopeId       : 2588672
status        :
```

Note: Repeat the "moquery -c dhcpProvDhcp" command on each Leaf node supporting DHCP Client endpoints.

Verify DHCP Relay Configuration (cont.)

dhcpRtLblDefToRelayP

```
fab2-leaf3# moquery -c dhcpRtLblDefToRelayP
Total Objects shown: 3

# dhcp.RtLblDefToRelayP
tDn            : uni/bd-[uni/tn-common/BD-deadbeef-bd63]-isSvc-no/dhcplbldef-deadbeef-dhcpRelay-tenant
childAction    :
dn             : uni/tn-common/relayp-deadbeef-dhcpRelay-tenant/rtlblDefToRelayP-[uni/bd-[uni/tn-common/BD-
deadbeef-bd63]-isSvc-no/dhcplbldef-deadbeef-dhcpRelay-tenant]
lcOwn          : policy
modTs          : 2015-06-21T19:57:14.443-04:00
rn             : rtlblDefToRelayP-[uni/bd-[uni/tn-common/BD-deadbeef-bd63]-isSvc-no/dhcplbldef-deadbeef-
dhcpRelay-tenant]
status         :
tCl            : dhcpLblDef


# dhcp.RtLblDefToRelayP
tDn            : uni/bd-[uni/tn-common/BD-deadbeef-bd62]-isSvc-no/dhcplbldef-deadbeef-dhcpRelay-tenant
childAction    :
dn             : uni/tn-common/relayp-deadbeef-dhcpRelay-tenant/rtlblDefToRelayP-[uni/bd-[uni/tn-common/BD-
deadbeef-bd62]-isSvc-no/dhcplbldef-deadbeef-dhcpRelay-tenant]
lcOwn          : policy
modTs          : 2015-06-21T20:07:53.843-04:00
rn             : rtlblDefToRelayP-[uni/bd-[uni/tn-common/BD-deadbeef-bd62]-isSvc-no/dhcplbldef-deadbeef-
dhcpRelay-tenant]
status         :
tCl            : dhcpLblDef
```

Verify DHCP Relay Configuration (cont.)

dhcpRtLblDefToRelayP (cont.)

```
# dhcp.RtLblDefToRelayP
tDn          : uni/bd-[uni/tn-common/BD-deadbeef-bd61]-isSvc-no/dhcplbldef-deadbeef-
dhcpRelay-tenant

childAction  :
dn           : uni/tn-common/relayp-deadbeef-dhcpRelay-tenant/rtlblDefToRelayP-[uni/bd-
[uni/tn-common/BD-deadbeef-bd61]-isSvc-no/dhcplbldef-deadbeef-dhcpRelay-tenant]

lcOwn        : policy
modTs        : 2015-06-21T20:10:55.108-04:00
rn           : rtlblDefToRelayP-[uni/bd-[uni/tn-common/BD-deadbeef-bd61]-isSvc-no/
dhcplbldef-deadbeef-dhcpRelay-tenant]

status       :
tCl          : dhcpLblDef
```

Note: Repeat the "moquery -c dhcpRtLblDefToRelayP" command on each Leaf node supporting DHCP Client endpoints.

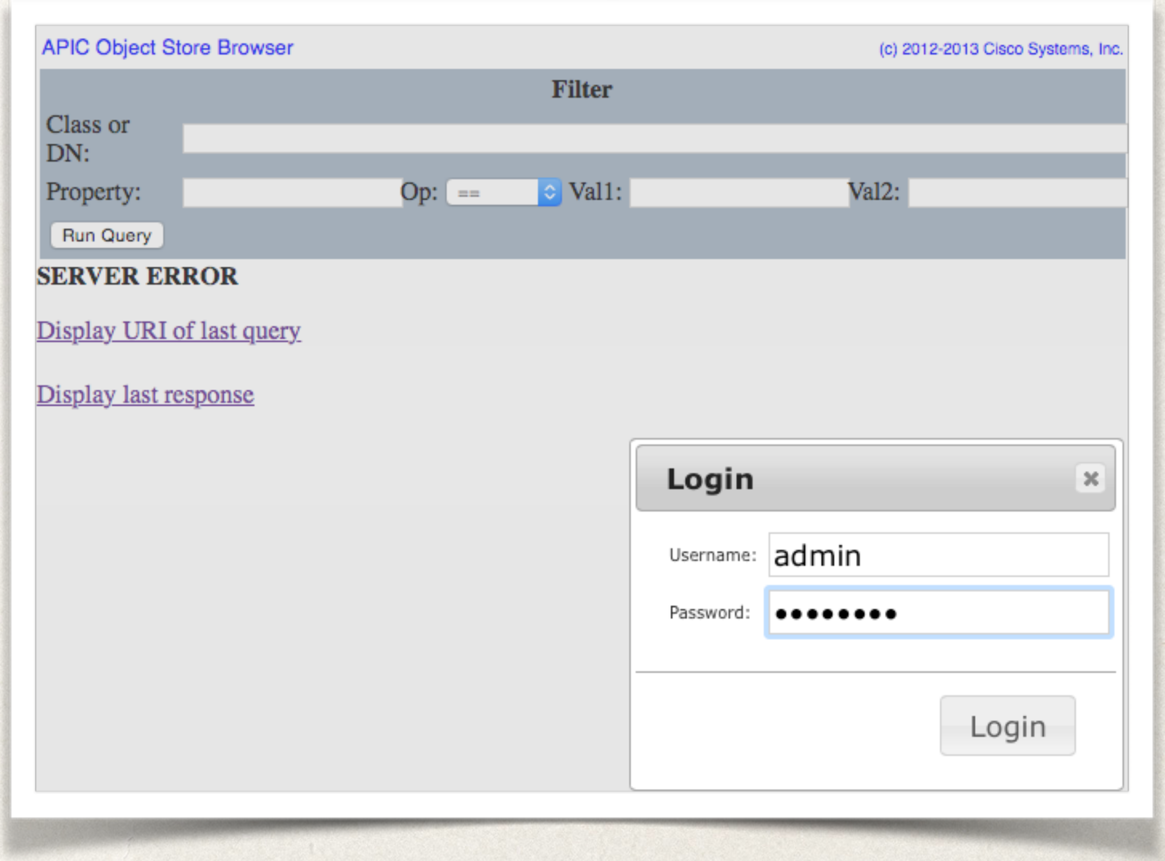
Verify DHCP Relay Configuration (cont.)

- Another tool to verify DHCP Relay configuration is VISORE. Enclosed are some samples of the VISORE information related to the DHCP Relay configuration.
(dhcpRelayP, dhcpRsProv, dhcpProvDhcp, dhcpRtLblDefToRelayP)

- To access VISORE, use a browser using the following address:

  https://<APIC_IP_address>/visore.html

  note: use your APIC Admin Credentials to login to VISORE



APIC Object Store Browser

(c) 2012-2013 Cisco Systems, Inc.

SERVER ERROR

Display URI of last query

Display last response

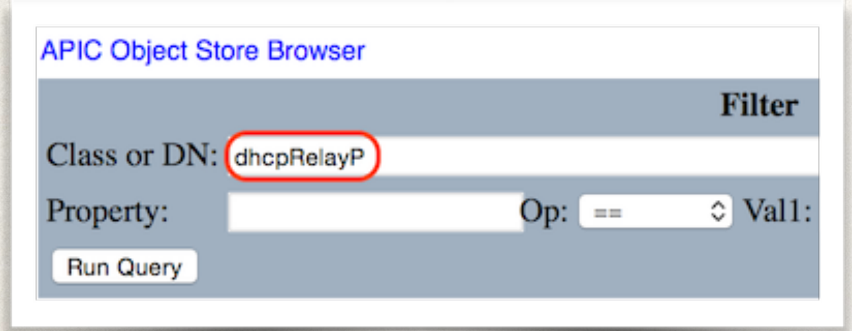
Login

Username: admin

Password: ●●●●●●●●

Verify DHCP Relay Configuration (cont.)

dhcpRelayP



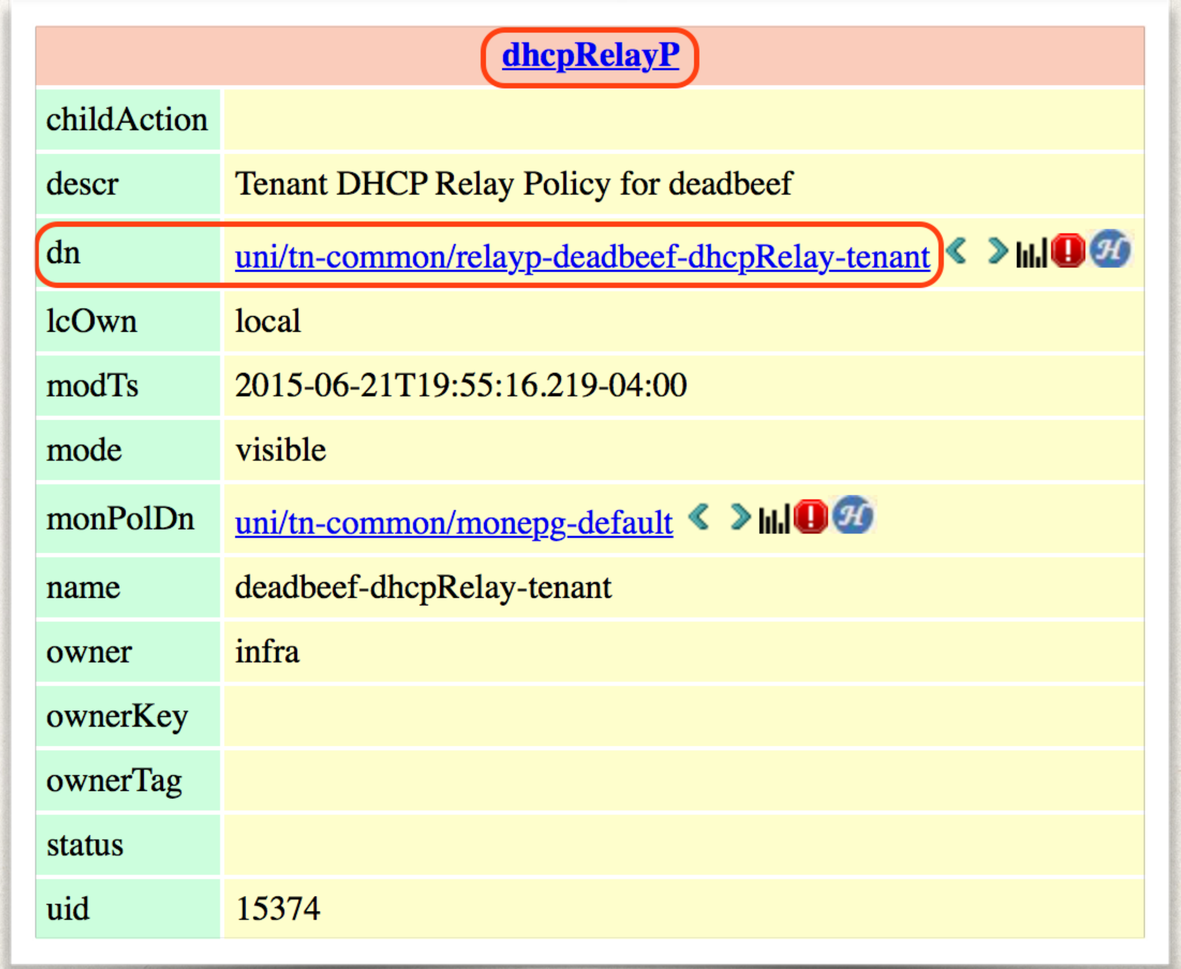
APIC Object Store Browser

Filter

Class or DN: dhcpRelayP

Property: Op: == Val1:

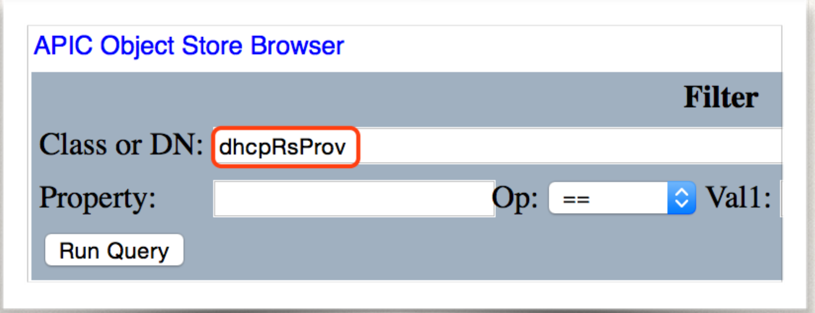
Run Query



dhcpRelayP	
childAction	
descr	Tenant DHCP Relay Policy for deadbeef
dn	uni/tn-common/relayp-deadbeef-dhcpRelay-tenant
lcOwn	local
modTs	2015-06-21T19:55:16.219-04:00
mode	visible
monPolDn	uni/tn-common/monepg-default
name	deadbeef-dhcpRelay-tenant
owner	infra
ownerKey	
ownerTag	
status	
uid	15374

Verify DHCP Relay Configuration (cont.)

dhcpRsProv



APIC Object Store Browser

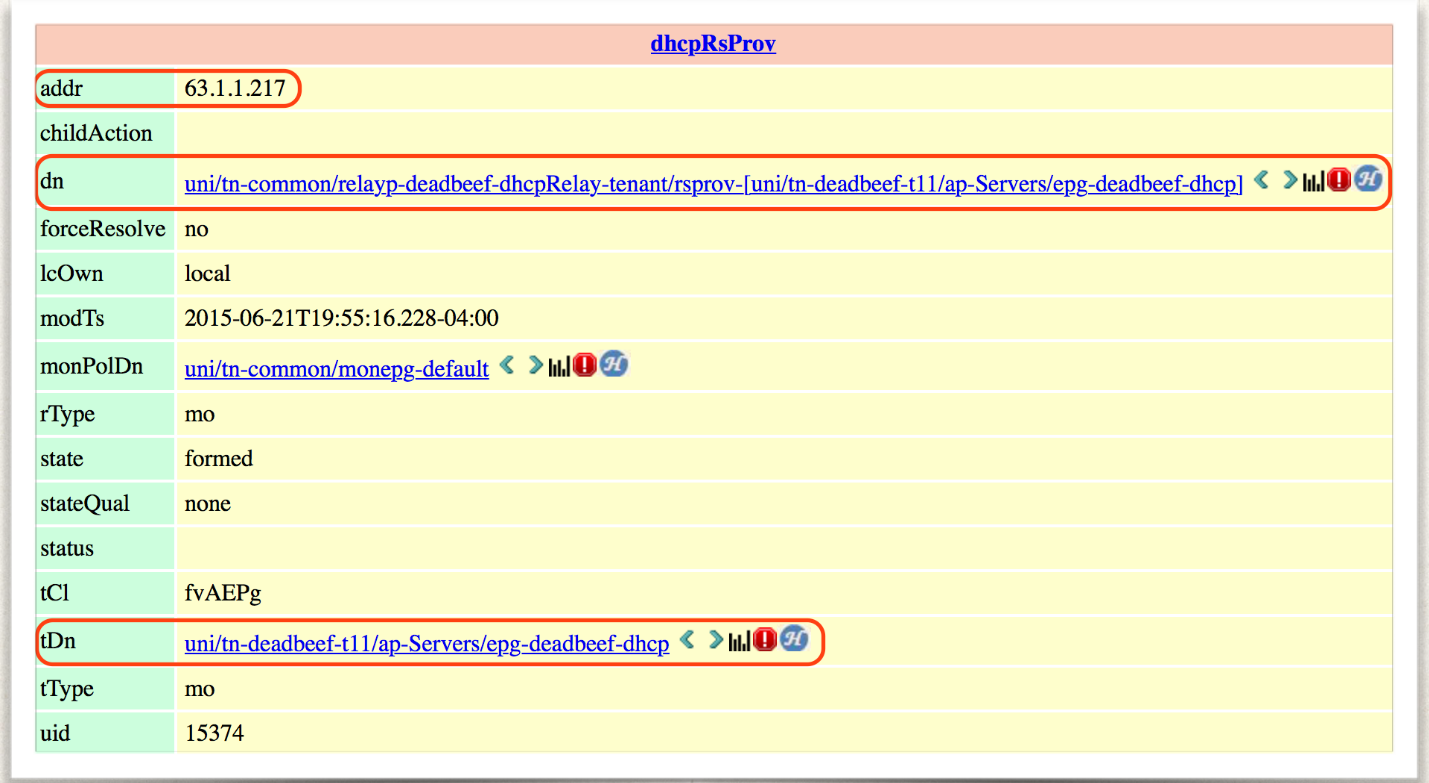
Filter

Class or DN: dhcpRsProv

Property:

Op: ==

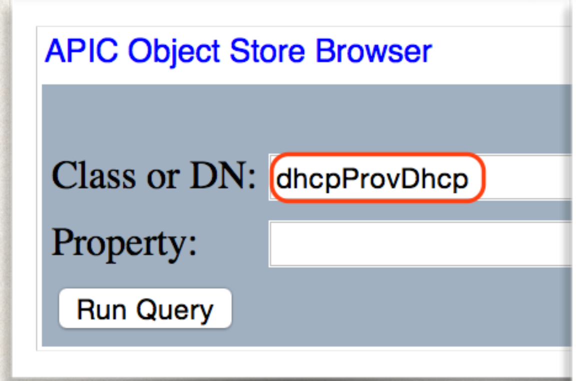
Val1:



dhcpRsProv	
addr	63.1.1.217
childAction	
dn	uni/tn-common/relayp-deadbeef-dhcpRelay-tenant/rsprov-[uni/tn-deadbeef-t11/ap-Servers/epg-deadbeef-dhcp]
forceResolve	no
lcOwn	local
modTs	2015-06-21T19:55:16.228-04:00
monPolDn	uni/tn-common/monepg-default
rType	mo
state	formed
stateQual	none
status	
tCl	fvAEPg
tDn	uni/tn-deadbeef-t11/ap-Servers/epg-deadbeef-dhcp
tType	mo
uid	15374

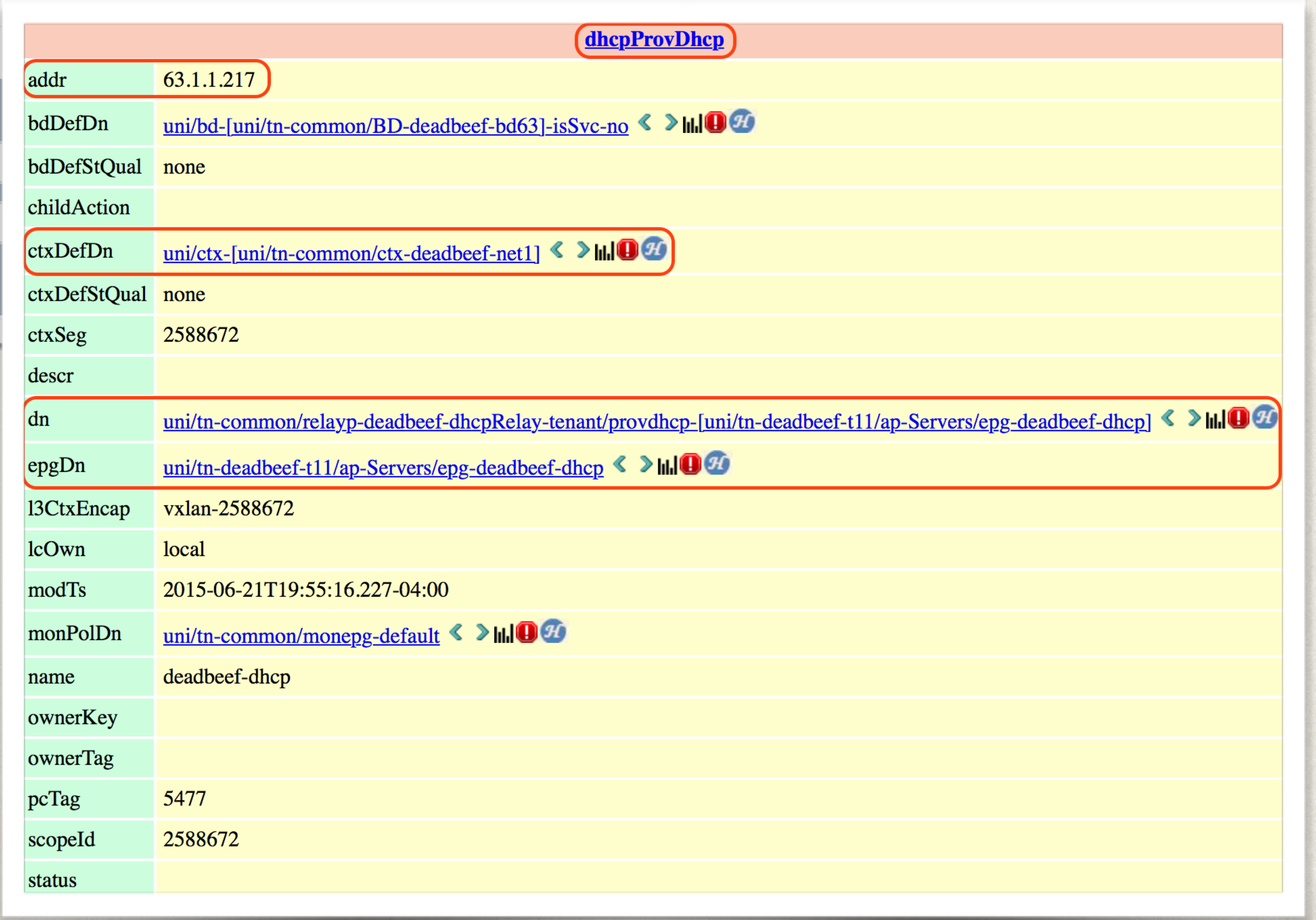
Verify DHCP Relay Configuration (cont.)

dhcpProvDhcp



APIC Object Store Browser

Class or DN:	dhcpProvDhcp
Property:	

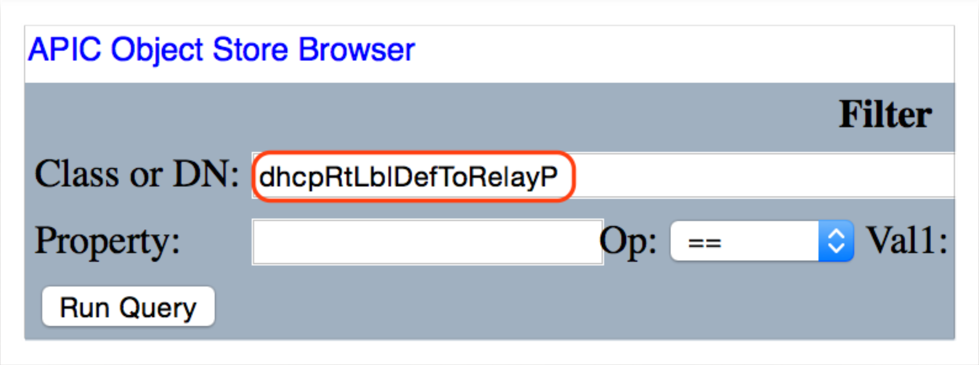


dhcpProvDhcp

addr	63.1.1.217
bdDefDn	uni/bd-[uni/tn-common/BD-deadbeef-bd63]-isSvc-no
bdDefStQual	none
childAction	
ctxDefDn	uni/ctx-[uni/tn-common/ctx-deadbeef-net1]
ctxDefStQual	none
ctxSeg	2588672
descr	
dn	uni/tn-common/relayp-deadbeef-dhcpRelay-tenant/provdhcp-[uni/tn-deadbeef-t11/ap-Servers/epg-deadbeef-dhcp]
epgDn	uni/tn-deadbeef-t11/ap-Servers/epg-deadbeef-dhcp
l3CtxEncap	vxlan-2588672
lcOwn	local
modTs	2015-06-21T19:55:16.227-04:00
monPolDn	uni/tn-common/monepg-default
name	deadbeef-dhcp
ownerKey	
ownerTag	
pcTag	5477
scopeId	2588672
status	

Verify DHCP Relay Configuration (cont.)

dhcpRtLblDefToRelayP



APIC Object Store Browser

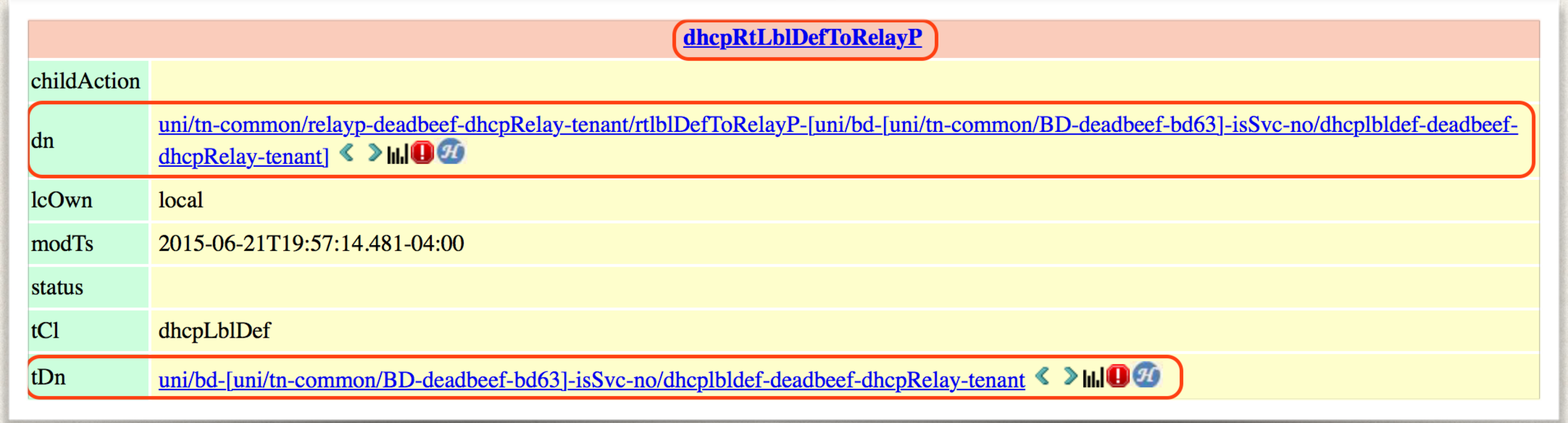
Filter

Class or DN: dhcpRtLblDefToRelayP

Property:

Op: ==

Val1:



dhcpRtLblDefToRelayP	
childAction	
dn	uni/tn-common/relayp-deadbeef-dhcpRelay-tenant/rtlblDefToRelayP-[uni/bd-[uni/tn-common/BD-deadbeef-bd63]-isSvc-no/dhcplbldef-deadbeef-dhcpRelay-tenant]
lcOwn	local
modTs	2015-06-21T19:57:14.481-04:00
status	
tCl	dhcpLblDef
tDn	uni/bd-[uni/tn-common/BD-deadbeef-bd63]-isSvc-no/dhcplbldef-deadbeef-dhcpRelay-tenant

Debug Commands

If the DHCP Relay configuration has been verified and you are still experiencing DHCP Relay issues, you can run some CLI commands from each Leaf experiencing issues.

- On each Leaf with DHCP-Relay configured use "**iping**" to test the connectivity to the DHCP SERVER.

```
iping [options] <target ip address>
options:
        -V vrf name (tenant:context)
        -c count
        -i wait
        -p pattern
        -s packet size -t timeout
        -S source ip address or source interface
```

For Example:

```
fab2-leaf3# iping -V common:deadbeef-net1 63.1.1.138
PING 63.1.1.138 (63.1.1.138) from 63.1.1.1: 56 data bytes
64 bytes from 63.1.1.138: icmp_seq=0 ttl=128 time=0.616 ms
64 bytes from 63.1.1.138: icmp_seq=1 ttl=128 time=0.504 ms
64 bytes from 63.1.1.138: icmp_seq=2 ttl=128 time=0.494 ms
64 bytes from 63.1.1.138: icmp_seq=3 ttl=128 time=0.605 ms
64 bytes from 63.1.1.138: icmp_seq=4 ttl=128 time=0.477 ms

--- 63.1.1.138 ping statistics ---
5 packets transmitted, 5 packets received, 0.00% packet loss
round-trip min/avg/max = 0.477/0.539/0.616 ms
```

Note: Repeat on each Leaf node supporting DHCP Client endpoints.

Debug Commands (cont.)

- On each Leaf with DHCP-Relay configured run “**show dhcp internal errors**”. This command will display any DHCP errors on the Leaf Node.

For Example:
(note: some output has been abbreviated for display purposes)

```
fab2-leaf3# show dhcp internal errors

150) 2015 Jul  1 09:04:01.508401 _snoop_handle_istack_packet: 1618 : After DHCP client processing DHCP response
packet. Drop net_l2_recv buffer.

154) 2015 Jul  1 09:03:50.503729 _parse_options_in_offer: 1851 : dhcp_parse_options_in_offer: TLV type 12 not
required

155) 2015 Jul  1 09:03:50.503661 _snoop_handle_istack_packet: 1618 : After DHCP client processing DHCP response
packet. Drop net_l2_recv buffer.

156) 2015 Jul  1 09:03:49.500015 _client_intf_ac_action_config_interface_select: 308 : Failed in the interface
selection to send DHCPREQUEST for interface Ethernet1/98.2

160) 2015 Jul  1 09:03:25.506882 _snoop_handle_istack_packet: 1741 : Snooping is not enabled globally or on vlan.
Drop net_l2_recv buffer.

161) 2015 Jul  1 09:03:25.490216 _client_intf_ac_action_config_interface_select: 308 : Failed in the interface
selection to send DHCPREQUEST for interface Ethernet1/97.1

163) 2015 Jul  1 09:03:13.485680 _client_create_clientintf: 4696 : dhcp_client_create_clientintf: Unable to create
new ClientIf while there is existing clientif with ifindex 335544320
```

Note: Repeat on each Leaf node supporting DHCP Client endpoints.

Debug Commands (cont.)

- On each Leaf with DHCP-Relay configured run "**show dhcp internal event-history msgs**". This command will display the DHCP event history on the Leaf Node.

For Example:
(note: some output has been abbreviated for display purposes)

```
fab2-leaf3# show dhcp internal event-history msgs

61) Event:E_MTS_RX, length:60, at 338159 usecs after Wed Jul  1 09:04:02 2015
    [NOT] Opc:MTS_OPC_CREATE_ImDhcptlvpolUInt32Policyelem(314348), Id:0X00004A78, Ret:SUCCESS
    Src:0x00000101/1248, Dst:0x00000101/0, Flags:None
    HA_SEQNO:0X00000000, RRtoken:0x00000000, Sync:UNKNOWN, Payloadsize:51
    Payload:
    0x0000:  fc 05 73 f6 ce 00 00 00 00 00 00 00 01 00 00 00

62) Event:E_MTS_RX, length:60, at 338082 usecs after Wed Jul  1 09:04:02 2015
    [NOT] Opc:MTS_OPC_MODIFY_ImDhcpClientIfPolicyelem(314365), Id:0X00004A72, Ret:SUCCESS
    Src:0x00000101/1248, Dst:0x00000101/0, Flags:None
    HA_SEQNO:0X00000000, RRtoken:0x00000000, Sync:UNKNOWN, Payloadsize:61
    Payload:
    0x0000:  00 06 73 f6 ce 00 00 00 00 00 00 00 01 00 00 00

63) Event:E_MTS_RX, length:60, at 329583 usecs after Wed Jul  1 09:04:02 2015
    [NOT] Opc:MTS_OPC_DELETE_ImDhcptlvpolUInt32Policyelem(314350), Id:0X00004A06, Ret:SUCCESS
    Src:0x00000101/1248, Dst:0x00000101/0, Flags:None
    HA_SEQNO:0X00000000, RRtoken:0x00000000, Sync:UNKNOWN, Payloadsize:51
    Payload:
    0x0000:  fc 05 73 f6 ce 00 00 00 00 00 00 00 01 00 00 00

64) Event:E_DEBUG, length:88, at 504952 usecs after Wed Jul  1 09:04:01 2015
    [108] dhcp_get_data_from_queue(903): dequeued timer msg: rid (0x1a061002), event_id (16)
```

Note: Repeat on each Leaf node supporting DHCP Client endpoints.

Debug Commands (cont.)

- On each Leaf with DHCP-Relay configured run “**show dhcp internal event-history traces**”. This command will display the DHCP event history on the Leaf Node.

For Example:
(note: some output has been abbreviated for display purposes)

```
fab2-leaf3# show dhcp internal event-history traces

583) 2015 Jul  1 15:05:31.551336 _obj_incr_clientrelayif_msg_stats: 1880 : parent client/relay if DN is:

584) 2015 Jul  1 15:05:31.551332 _objstore_open: 146 : dhcp_objstore_open

585) 2015 Jul  1 15:05:31.551327 _obj_incr_clientrelayif_msg_stats: 1858 : In saving client/relay if msg stat

587) 2015 Jul  1 15:05:31.551293 _relay_send_packet: 1615 : Sending packet on addr[63.1.1.138] port[67] iod[sin:0x0 tgt:0x0] ctx[vdc:1 vrf:5 top:0]

588) 2015 Jul  1 15:05:31.551264 _relay_send_packet: 1588 : DHCP relay add option82 cid. if_index added is Vlan30 and phys if index is Vlan30

589) 2015 Jul  1 15:05:31.551260 _relay_add_option82: 2577 :  Option82 Hex Dump = [T 52 L 14 V [T 1 L c V  1a 03 10 00 00 00 00 1f 00 00 00 00 ][T 2 L 4 V  a 00 c0 5b ] ]

590) 2015 Jul  1 15:05:31.551250 _relay_add_circuitid_rmtid: 2727 : Circuit Id and Remote Id suboptions are added

591) 2015 Jul  1 15:05:31.551248 _relay_add_circuitid_rmtid: 2708 : dhcp_relay_add_circuitid_rmtid: Add remote id suboption: tep ip is a00c05b.

592) 2015 Jul  1 15:05:31.551245 _relay_add_circuitid_rmtid: 2679 : Add circuit id suboption: if_index: Ethernet1/50 , svlan: 31, option def id: 0.

593) 2015 Jul  1 15:05:31.551229 _relay_add_option82: 2531 : Mac addr is 74:26:ac:eb:5e:cf

594) 2015 Jul  1 15:05:31.551226 _relay_add_option82: 2527 : Adding option82 suboptions

595) 2015 Jul  1 15:05:31.551224 _parse_dhcp_msg_type_option: 2578 : Val of dhcp msg type is 1

596) 2015 Jul  1 15:05:31.551222 _parse_dhcp_msg_type_option: 2574 : Got the DHCP msg type option.

597) 2015 Jul  1 15:05:31.551220 _relay_send_packet: 1576 : gi address is 61.1.1.1

598) 2015 Jul  1 15:05:31.551218 _relay_send_packet: 1568 : giaddr is 0

599) 2015 Jul  1 15:05:31.551217 _relay_send_packet: 1564 : Helper address is 63.1.1.138

600) 2015 Jul  1 15:05:31.551215 _relay_send_packet: 1555 : Client and Server are in the same VRF

603) 2015 Jul  1 15:05:31.551060 _relay_handle_packet_from_pkt_mgr: 423 : DHCPDISCOVER msg
```

Note: Repeat on each Leaf node supporting DHCP Client endpoints.

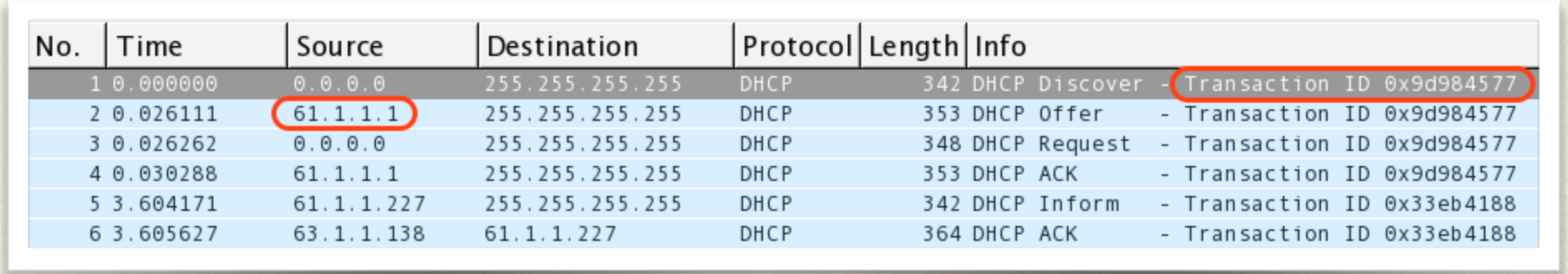
Packet Traces

- When dealing with a Client\Server application or service, It is best practice to gather packet traces from each device.

1. Use an analyzer tool and capture a packet trace from the CLIENT device.
2. From the same packet flow, use an analyzer tool and capture a packet trace from the SERVER device.
3. If available, capture packet traces from a known WORKING configuration. The packet trace should be a complete trace that displays expected behaviors. Compare the WORKING packet traces against the NON-WORKING traces to assist in problem determination.
4. If working capture packet are not available, compare NON-WORKING traces to RFCs or Software Design or Protocol Specifications to assist in problem determination.

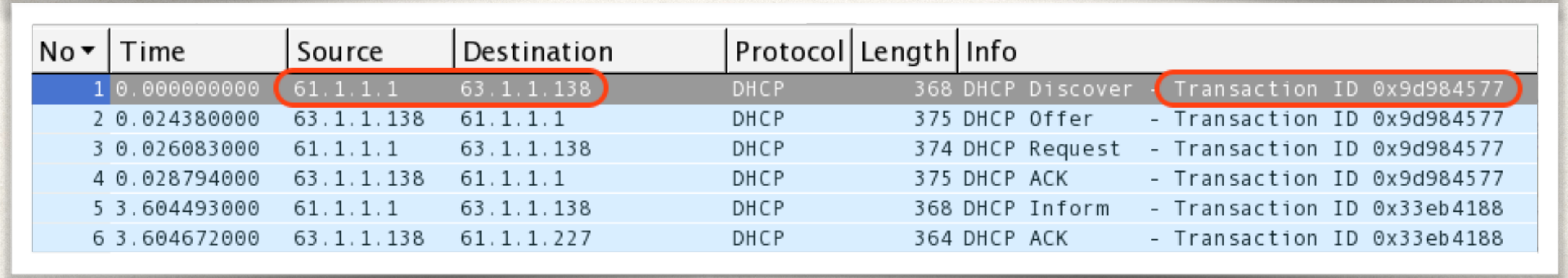
Note: The following DHCP-Relay example uses Wireshark to display a WORKING packet trace from the CLIENT\SERVER for DHCP in the ACI Fabric solution. 61.1.1.1 is the ACI BD default gateway (GI ADDR) and 63.1.1.138 is the DHCP Server.

CLIENT capture



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0x9d984577
2	0.026111	61.1.1.1	255.255.255.255	DHCP	353	DHCP Offer - Transaction ID 0x9d984577
3	0.026262	0.0.0.0	255.255.255.255	DHCP	348	DHCP Request - Transaction ID 0x9d984577
4	0.030288	61.1.1.1	255.255.255.255	DHCP	353	DHCP ACK - Transaction ID 0x9d984577
5	3.604171	61.1.1.227	255.255.255.255	DHCP	342	DHCP Inform - Transaction ID 0x33eb4188
6	3.605627	63.1.1.138	61.1.1.227	DHCP	364	DHCP ACK - Transaction ID 0x33eb4188

SERVER capture

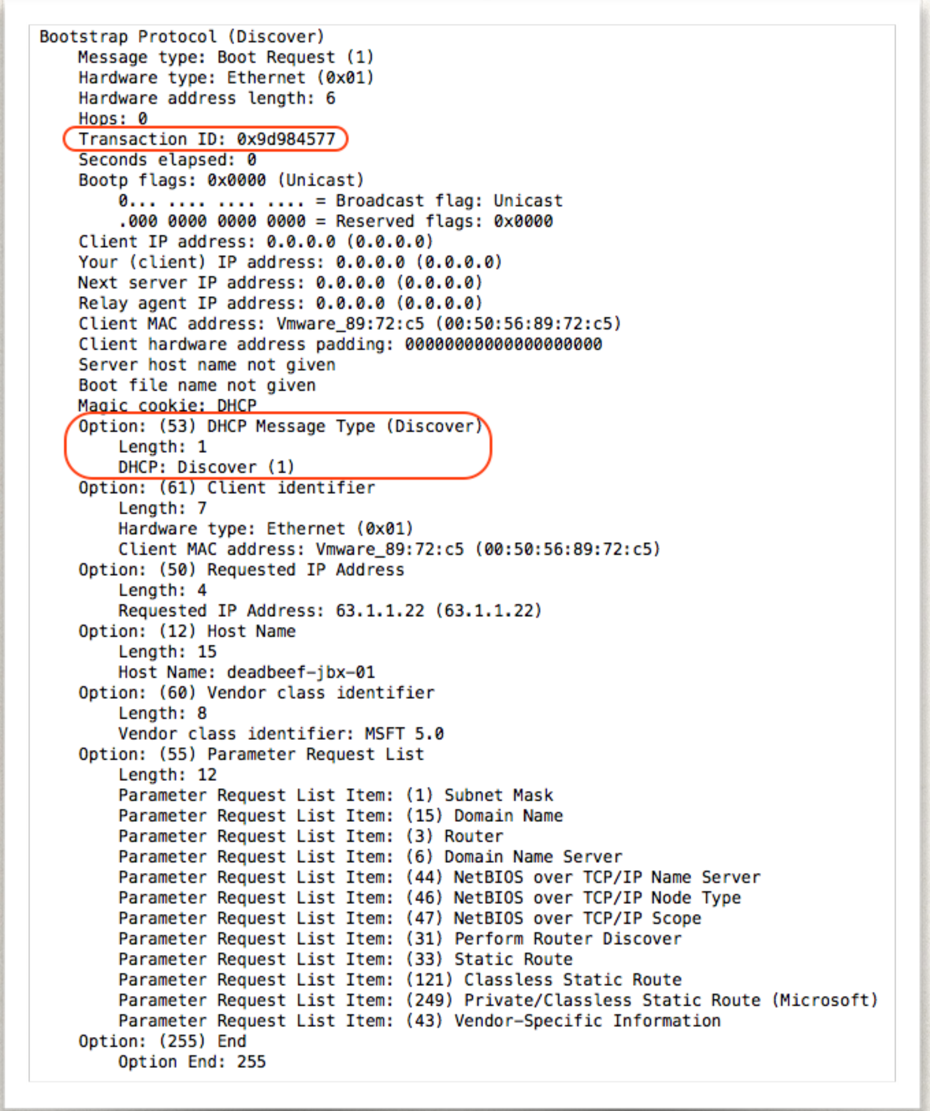


No ▾	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	61.1.1.1	63.1.1.138	DHCP	368	DHCP Discover - Transaction ID 0x9d984577
2	0.024380000	63.1.1.138	61.1.1.1	DHCP	375	DHCP Offer - Transaction ID 0x9d984577
3	0.026083000	61.1.1.1	63.1.1.138	DHCP	374	DHCP Request - Transaction ID 0x9d984577
4	0.028794000	63.1.1.138	61.1.1.1	DHCP	375	DHCP ACK - Transaction ID 0x9d984577
5	3.604493000	61.1.1.1	63.1.1.138	DHCP	368	DHCP Inform - Transaction ID 0x33eb4188
6	3.604672000	63.1.1.138	61.1.1.227	DHCP	364	DHCP ACK - Transaction ID 0x33eb4188

Packet Traces (cont.)

CLIENT - DHCP DISCOVER

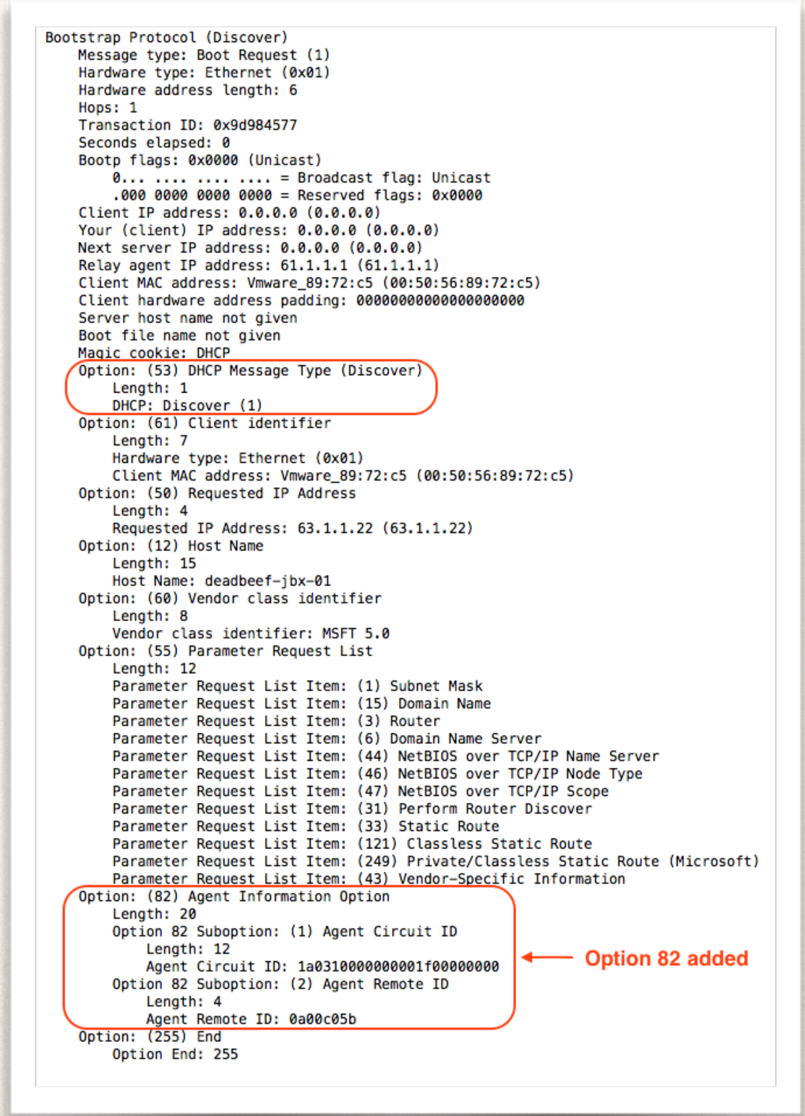
- Evaluate the Packet detail of what is transmitted from the client



```
Bootstrap Protocol (Discover)
    Message type: Boot Request (1)
    Hardware type: Ethernet (0x01)
    Hardware address length: 6
    Hops: 0
    Transaction ID: 0x9d984577
    Seconds elapsed: 0
    Bootp flags: 0x0000 (Unicast)
        0... .... .... .... = Broadcast flag: Unicast
        .000 0000 0000 0000 = Reserved flags: 0x0000
    Client IP address: 0.0.0.0 (0.0.0.0)
    Your (client) IP address: 0.0.0.0 (0.0.0.0)
    Next server IP address: 0.0.0.0 (0.0.0.0)
    Relay agent IP address: 0.0.0.0 (0.0.0.0)
    Client MAC address: Vmware_89:72:c5 (00:50:56:89:72:c5)
    Client hardware address padding: 00000000000000000000
    Server host name not given
    Boot file name not given
    Magic cookie: DHCP
    Option: (53) DHCP Message Type (Discover)
        Length: 1
        DHCP: Discover (1)
    Option: (61) Client identifier
        Length: 7
        Hardware type: Ethernet (0x01)
        Client MAC address: Vmware_89:72:c5 (00:50:56:89:72:c5)
    Option: (50) Requested IP Address
        Length: 4
        Requested IP Address: 63.1.1.22 (63.1.1.22)
    Option: (12) Host Name
        Length: 15
        Host Name: deadbeef-jbx-01
    Option: (60) Vendor class identifier
        Length: 8
        Vendor class identifier: MSFT 5.0
    Option: (55) Parameter Request List
        Length: 12
        Parameter Request List Item: (1) Subnet Mask
        Parameter Request List Item: (15) Domain Name
        Parameter Request List Item: (3) Router
        Parameter Request List Item: (6) Domain Name Server
        Parameter Request List Item: (44) NetBIOS over TCP/IP Name Server
        Parameter Request List Item: (46) NetBIOS over TCP/IP Node Type
        Parameter Request List Item: (47) NetBIOS over TCP/IP Scope
        Parameter Request List Item: (31) Perform Router Discover
        Parameter Request List Item: (33) Static Route
        Parameter Request List Item: (121) Classless Static Route
        Parameter Request List Item: (249) Private/Classless Static Route (Microsoft)
        Parameter Request List Item: (43) Vendor-Specific Information
    Option: (255) End
        Option End: 255
```

SERVER - DHCP DISCOVER

- Evaluate the Packet detail of what is received from the DHCP-Relay Proxy (ACI Leaf node)



```
Bootstrap Protocol (Discover)
    Message type: Boot Request (1)
    Hardware type: Ethernet (0x01)
    Hardware address length: 6
    Hops: 1
    Transaction ID: 0x9d984577
    Seconds elapsed: 0
    Bootp flags: 0x0000 (Unicast)
        0... .... .... .... = Broadcast flag: Unicast
        .000 0000 0000 0000 = Reserved flags: 0x0000
    Client IP address: 0.0.0.0 (0.0.0.0)
    Your (client) IP address: 0.0.0.0 (0.0.0.0)
    Next server IP address: 0.0.0.0 (0.0.0.0)
    Relay agent IP address: 61.1.1.1 (61.1.1.1)
    Client MAC address: Vmware_89:72:c5 (00:50:56:89:72:c5)
    Client hardware address padding: 00000000000000000000
    Server host name not given
    Boot file name not given
    Magic cookie: DHCP
    Option: (53) DHCP Message Type (Discover)
        Length: 1
        DHCP: Discover (1)
    Option: (61) Client identifier
        Length: 7
        Hardware type: Ethernet (0x01)
        Client MAC address: Vmware_89:72:c5 (00:50:56:89:72:c5)
    Option: (50) Requested IP Address
        Length: 4
        Requested IP Address: 63.1.1.22 (63.1.1.22)
    Option: (12) Host Name
        Length: 15
        Host Name: deadbeef-jbx-01
    Option: (60) Vendor class identifier
        Length: 8
        Vendor class identifier: MSFT 5.0
    Option: (55) Parameter Request List
        Length: 12
        Parameter Request List Item: (1) Subnet Mask
        Parameter Request List Item: (15) Domain Name
        Parameter Request List Item: (3) Router
        Parameter Request List Item: (6) Domain Name Server
        Parameter Request List Item: (44) NetBIOS over TCP/IP Name Server
        Parameter Request List Item: (46) NetBIOS over TCP/IP Node Type
        Parameter Request List Item: (47) NetBIOS over TCP/IP Scope
        Parameter Request List Item: (31) Perform Router Discover
        Parameter Request List Item: (33) Static Route
        Parameter Request List Item: (121) Classless Static Route
        Parameter Request List Item: (249) Private/Classless Static Route (Microsoft)
        Parameter Request List Item: (43) Vendor-Specific Information
    Option: (82) Agent Information Option
        Length: 20
        Option 82 Suboption: (1) Agent Circuit ID
            Length: 12
            Agent Circuit ID: 1a0310000000001f00000000
        Option 82 Suboption: (2) Agent Remote ID
            Length: 4
            Agent Remote ID: 0a00c05b
    Option: (255) End
        Option End: 255
```

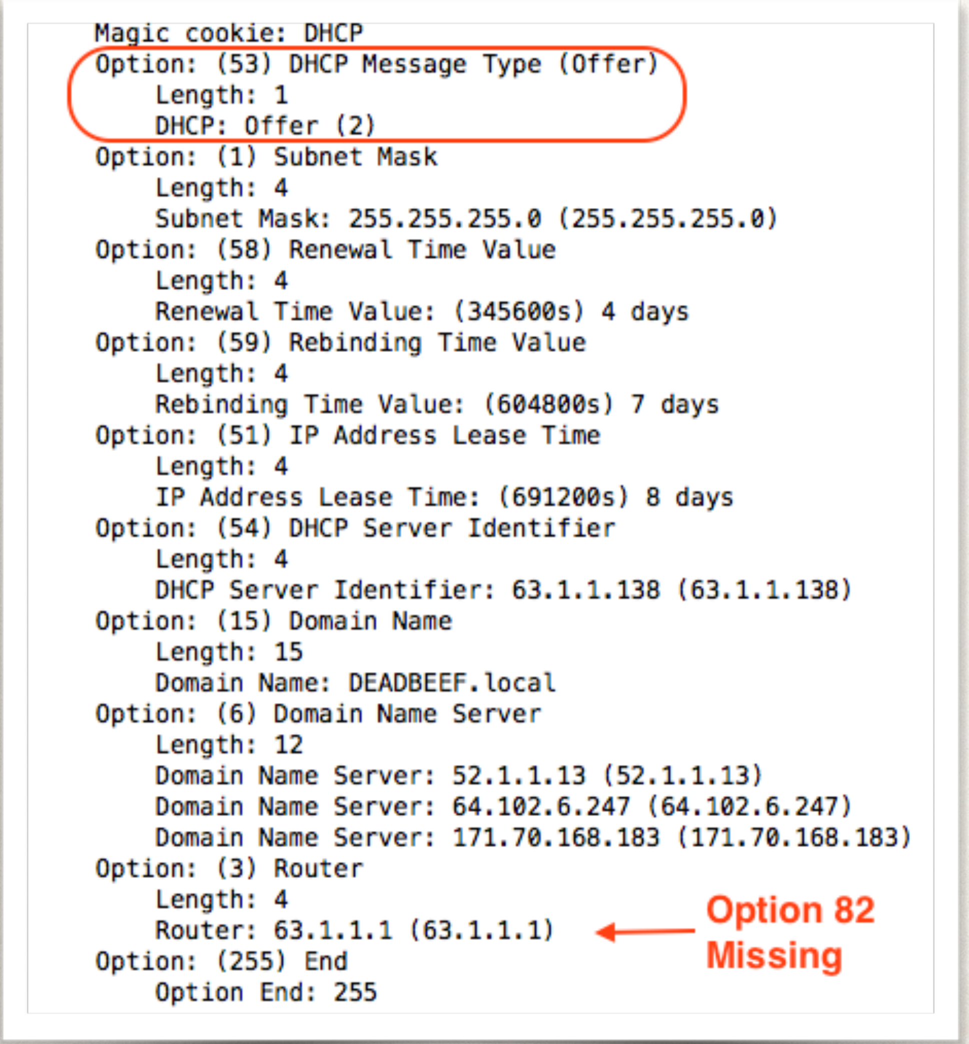
Packet Traces (cont.)

WINDOWS 2008 - DHCP OFFER

- Notice this DHCP OFFER DOES NOT contain OPTION 82. The DHCP-Relay Proxy (ACI Leaf Node) will drop this DHCP OFFER when received.

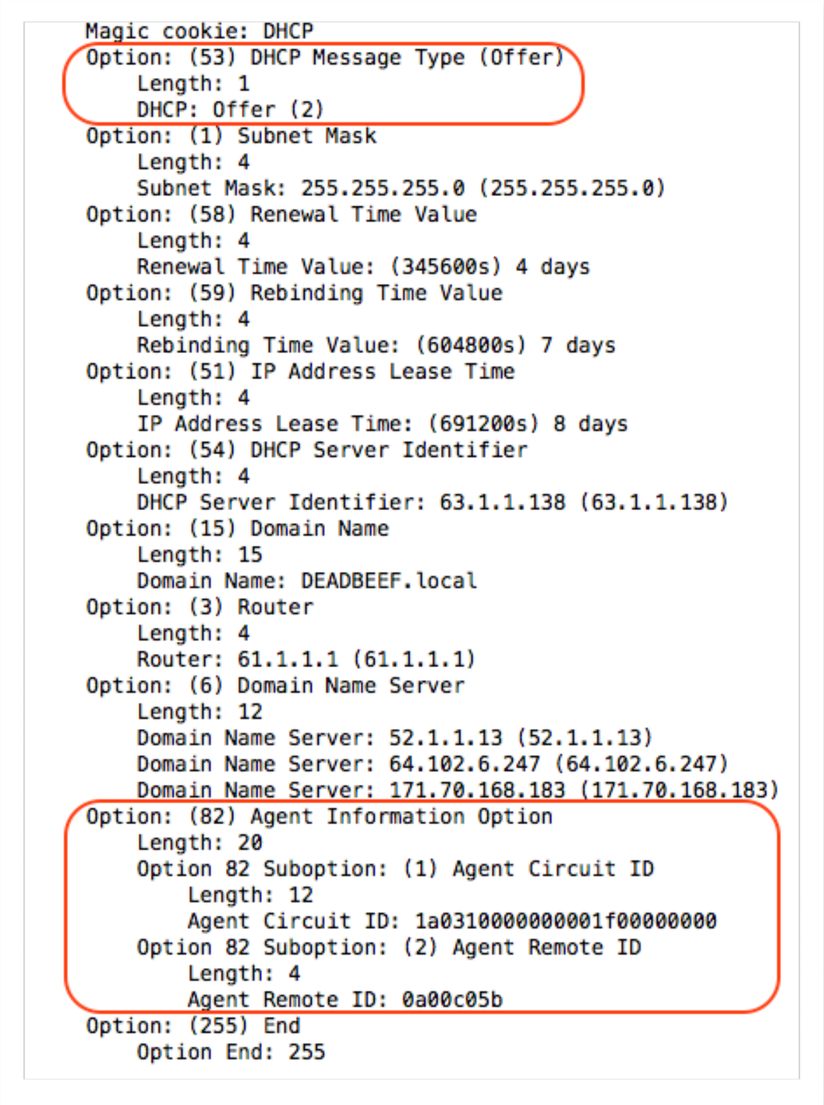
WINDOWS 2012 - DHCP OFFER

- Notice this DHCP OFFER contains OPTION 82 as requested in the DHCP DISCOVER from DHCP-Relay Proxy (ACI Leaf Node).



```
Magic cookie: DHCP
Option: (53) DHCP Message Type (Offer)
    Length: 1
    DHCP: Offer (2)
Option: (1) Subnet Mask
    Length: 4
    Subnet Mask: 255.255.255.0 (255.255.255.0)
Option: (58) Renewal Time Value
    Length: 4
    Renewal Time Value: (345600s) 4 days
Option: (59) Rebinding Time Value
    Length: 4
    Rebinding Time Value: (604800s) 7 days
Option: (51) IP Address Lease Time
    Length: 4
    IP Address Lease Time: (691200s) 8 days
Option: (54) DHCP Server Identifier
    Length: 4
    DHCP Server Identifier: 63.1.1.138 (63.1.1.138)
Option: (15) Domain Name
    Length: 15
    Domain Name: DEADBEEF.local
Option: (6) Domain Name Server
    Length: 12
    Domain Name Server: 52.1.1.13 (52.1.1.13)
    Domain Name Server: 64.102.6.247 (64.102.6.247)
    Domain Name Server: 171.70.168.183 (171.70.168.183)
Option: (3) Router
    Length: 4
    Router: 63.1.1.1 (63.1.1.1)
Option: (255) End
    Option End: 255
```

Option 82 Missing



```
Magic cookie: DHCP
Option: (53) DHCP Message Type (Offer)
    Length: 1
    DHCP: Offer (2)
Option: (1) Subnet Mask
    Length: 4
    Subnet Mask: 255.255.255.0 (255.255.255.0)
Option: (58) Renewal Time Value
    Length: 4
    Renewal Time Value: (345600s) 4 days
Option: (59) Rebinding Time Value
    Length: 4
    Rebinding Time Value: (604800s) 7 days
Option: (51) IP Address Lease Time
    Length: 4
    IP Address Lease Time: (691200s) 8 days
Option: (54) DHCP Server Identifier
    Length: 4
    DHCP Server Identifier: 63.1.1.138 (63.1.1.138)
Option: (15) Domain Name
    Length: 15
    Domain Name: DEADBEEF.local
Option: (3) Router
    Length: 4
    Router: 61.1.1.1 (61.1.1.1)
Option: (6) Domain Name Server
    Length: 12
    Domain Name Server: 52.1.1.13 (52.1.1.13)
    Domain Name Server: 64.102.6.247 (64.102.6.247)
    Domain Name Server: 171.70.168.183 (171.70.168.183)
Option: (82) Agent Information Option
    Length: 20
    Option 82 Suboption: (1) Agent Circuit ID
        Length: 12
        Agent Circuit ID: 1a0310000000001f00000000
    Option 82 Suboption: (2) Agent Remote ID
        Length: 4
        Agent Remote ID: 0a00c05b
Option: (255) End
    Option End: 255
```

DHCP Relay Caveats - Issues

This section will discuss some known caveats or issues with the DHCP Relay feature in the ACI Solution. A few notable Caveats or Issues are: DHCP Relay Proxy use of the DHCP Option 82 in the ACI Fabric and DHCP Relay support for multiple subnets under a single Bridge Domain (BD).

DHCP Option 82

- DHCP Servers must support Option 82 and Option 82 Sub-options when integrated with an ACI Fabric Solution.

In the APIC Getting Started Guide, under the section Configuring DHCP Relay Policy, the following text has been added:

When an ACI acts as a DHCP relay, it inserts the DHCP Option 82 (the DHCP Relay Agent Information Option) in DHCP requests that it proxies on behalf of clients. If a response (DHCP offer) comes back from a DHCP server without Option 82, it is silently dropped by the fabric. Therefore, when the ACI is acting as a DHCP relay, DHCP servers providing IP addresses to compute nodes attached to the ACI must support Option 82.

http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/getting-started/b_APIC_Getting_Started_Guide.html

Microsoft Windows Server 2003 & 2008 configured for DHCP Services do not support receiving DHCP DISCOVER requests with OPTION 82 enclosed. The DHCP Server parses the DHCP Request and extends a DHCP OFFER without Option 82 enclosed. As a result, the DHCP OFFER (without Option 82) received by the ACI Leaf Node is silently dropped by the fabric. The DHCP OFFER is never received at DHCP Client and the DHCP Request fails.

Note: Microsoft Windows Server 2012 configured for DHCP Services supports Option 82 in DHCP Requests for Single VRF environments. Linux Servers configured for DHCP Services supports Option 82 in DHCP Requests for Single & Multiple VRF environments.

- Refer to the troubleshooting section for discovering this issue:
  - show dhcp internal errors
  - show dhcp internal event-history traces
  - capture packet traces

- CSCuq78511 - Document mandatory requirement for DHCP server to support Option 82

DHCP Option 82 (cont.)

- Overview of the issues with OPTION 82 support in Single VRF (Intra-VRF) and Multiple VRF (Inter-VRF) environments

When the DHCP Relay Proxy adds OPTION 82 to DHCP Request, the gateway includes sub-options as part of the OPTION 82 body. The destination VRF will determine which sub-options to include. The VRF and sub-options are significant to determining which DHCP Scope will be used in assigning IP address to the requesting device.

Single VRF\Context (Intra-VRF)
Leaf relays DHCP Discover Packet with OPTION 82 with Sub-options:

- Agent Circuit ID
- Agent Remote ID

```
Option: (82) Agent Information Option
    Length: 24
    Option 82 Suboption: (1) Agent Circuit ID
        Length: 16
        Agent Circuit ID: 160000060000001b00000000000023fd
    Option 82 Suboption: (2) Agent Remote ID
        Length: 4
        Agent Remote ID: 0a00c85b
```

For intra-vrf DHCP requests, the scope decision can still be made on the **GIADDR** field. The **GIADDR** will be used for scope identification and ip address assignment.

- Microsoft Server 2012 supports Option 82 and sub-options: **Agent Circuit ID**, **Agent Remote ID**, and **VRF Name\VPN ID**. Microsoft Server 2012 will send a DHCP Offer with OPTION 82 and the Sub-options.
- Linux DHCP Server supports Option 82 and **all** of the sub-options. The Linux Server will send a DHCP Offer with OPTION 82 and the Sub-options.

DHCP Option 82 (cont.)

- What is the Option 82 Suboption “Agent Circuit ID” and “Agent Remote ID”? How do I translate the values?

When the DHCP Relay Proxy adds OPTION 82 to DHCP Request, the gateway includes sub-options as part of the OPTION 82 body. The destination VRF will determine which sub-options to include. The Default sub-options added by the ACI switches for DHCP Relay are

- Agent Circuit ID
- Agent Remote ID

```
Option: (82) Agent Information Option
    Option 82 Suboption: (1) Agent Circuit ID
        Agent Circuit ID: 160000060000001b00000000000023fd
    Option 82 Suboption: (2) Agent Remote ID
        Agent Remote ID: 0a00c85b
```

Agent Circuit ID: is the **Physical Interface, VLAN ID,** and **VLAN vnid** of where the Client resides on the DHCP Relay Proxy Gateway
Agent Remote ID: is the **TEP Address** of the DHCP Relay Proxy Gateway

You can decode these values to use for troubleshooting ACI DHCP Relay issues.

<u>Resources for decoding values:</u>

- A wireshark capture from the DHCP Server. Filter on “bootp” and capture the DHCP Discover or DHCP Request Packet.
- IP Address - HEX, Decimal, Binary Converter -> http://ncalculators.com/digital-computation/ip-address-hex-decimal-binary.htm
- Hexadecimal to Decimal Converter -> http://www.binaryhexconverter.com/hex-to-decimal-converter
- Access to ACI Leaf Nodes so that you can run some CLI commands

DHCP Option 82 (cont.)

- What is the Option 82 Suboption “Agent Circuit ID” and “Agent Remote ID”? How do I translate the values? (cont.)

Agent Circuit ID: *160000060000001b000000000000023fd*
Agent Remote ID: *0a00c85b*

Agent Circuit ID: is the **Physical Interface**, **VLAN ID**, and **VLAN vnid** of where the Client resides on the DHCP Relay Proxy Gateway Agent Circuit ID: 160000060000001b000000000000023fd

```
Agent Circuit ID: 160000060000 = Physical Interface (port-channel7 from the "show system internal epmc endpoint vlan 27”)
Agent Circuit ID: 001b00000000 = VLAN 27
Agent Circuit ID: 000023fd     = VLAN vnid  (9213 from the "show system internal epmc endpoint vlan 27")
Agent Circuit ID: port-channel7, VLAN 27, VLAN vnid 9213
```

Use the “show system internal epmc endpoint vlan ###” command output to decode the Option 82 sub-option values.

For example:

```
(vsh_lc)# show system internal epmc endpoint vlan ##

rtp-f2-p1-leaf4# vsh_lc

module-1# show system internal epmc endpoint vlan 27

Vlan 27

MAC : 0050.5689.286e ::: Num IPs : 1
IP# 0 : 192.2.25.101
Vlan id : 27 ::: Vlan vnid : 9213 ::: BD vnid : 16580488
Encap vlan : 802.1Q/51
VRF name : deadbeef-dhcp3:dhcp3-v1  ::: VRF vnid : 2981889
phy if : 0x16000006 ::: tunnel if : 0 ::: Interface : port-channel7
Ref count : 5 ::: sclass : 16388
```

DHCP Option 82 (cont.)

- What is the Option 82 Suboption “Agent Circuit ID” and “Agent Remote ID”? How do I translate the values? (cont.)

Agent Circuit ID: *160000060000001b00000000000023fd*
Agent Remote ID: *0a00c85b*

Agent Remote ID: is the TEP Address of the DHCP Relay Proxy Gateway

```
Agent Remote ID:     0a00c85b
Hex to IP Address:  10.0.200.91
```

```
# On APIC
# show switch | egrep -E  "10.0.200.91|ID|---"

rtp-f2-p1-apic1# show switch | egrep -E  "10.0.200.91|ID|---"

Abreviated Output

 ID     Pod    Address        Version             Serial Number  Name
 ----   ----   ------------   -----------------   -------------  ---------------
 214    1      10.0.200.91    n9000-12.2(0.64a)   SAL1816QWDQ    rtp-f2-p1-leaf4


# On LEAF
# acidiag fnvread | egrep -E  "10.0.200.91|ID|---"

rtp-f2-p1-leaf3# acidiag fnvread | egrep -E  "10.0.200.91|ID|---"
 ID    Pod ID  Name              Serial Number    IP Address        Role        State
-------------------------------------------------------------------------------------
214         1  rtp-f2-p1-leaf4   SAL1816QWDQ      10.0.200.91/32    leaf        active
```

DHCP Option 82 (cont.)

- When are the Option 82 Suboptions “VRF Name”, Server ID Override”, and “Link selection” used?

Multiple VRF\Context (Inter-VRF)

Leaf relays DHCP Discover Packet with OPTION 82 with Sub-options:

- Agent Circuit ID
- Agent Remote ID
- VRF Name\VPN ID
- Server ID Override
- Link selection

```
Option: (82) Agent Information Option
    Length: 55
    Option 82 Suboption: (1) Agent Circuit ID
        Length: 12
        Agent Circuit ID: 1a0310000000002c00000000
    Option 82 Suboption: (2) Agent Remote ID
        Length: 4
        Agent Remote ID: 0a00c05a
    Option 82 Suboption: (151) VRF name/VPN ID
        Length: 21
        VRF name:
    Option 82 Suboption: (11) Server ID Override
        Length: 4
        Server ID Override: 62.1.1.1 (62.1.1.1)
    Option 82 Suboption: (5) Link selection
        Length: 4
        Link selection: 62.1.1.0 (62.1.1.0)
```

DHCP Option 82 (cont.)

Multiple VRF\Context (Inter-VRF)

```
Option: (82) Agent Information Option

    Option 82 Suboption: (5) Link selection
        Length: 4
        Link selection: 62.1.1.0 (62.1.1.0)
```

For inter-vrf DHCP requests, the scope decision can still be made on the "**Option 82 Suboption: Link selection**". The "**Option 82 Suboption: Link selection**" will be used for scope identification and ip address assignment.

- **Microsoft Server 2012** supports Option 82 and **ONLY** sub-options: **Agent Circuit ID**, **Agent Remote ID**, and **VRF Name\VPN ID**. Microsoft Server 2012 does **NOT** support "**Option 82 Suboption: Link selection**" and will send a DHCP Offer with OPTION 82 and the Sub-options with an IP address from the **WRONG** subnet scope.
- **Linux DHCP Server** supports Option 82 and **all** of the sub-options. Linux Servers support "**Option 82 Suboption: Link selection**" and will send a DHCP Offer with OPTION 82 and the Sub-options with an IP address from the **CORRECT** subnet scope.

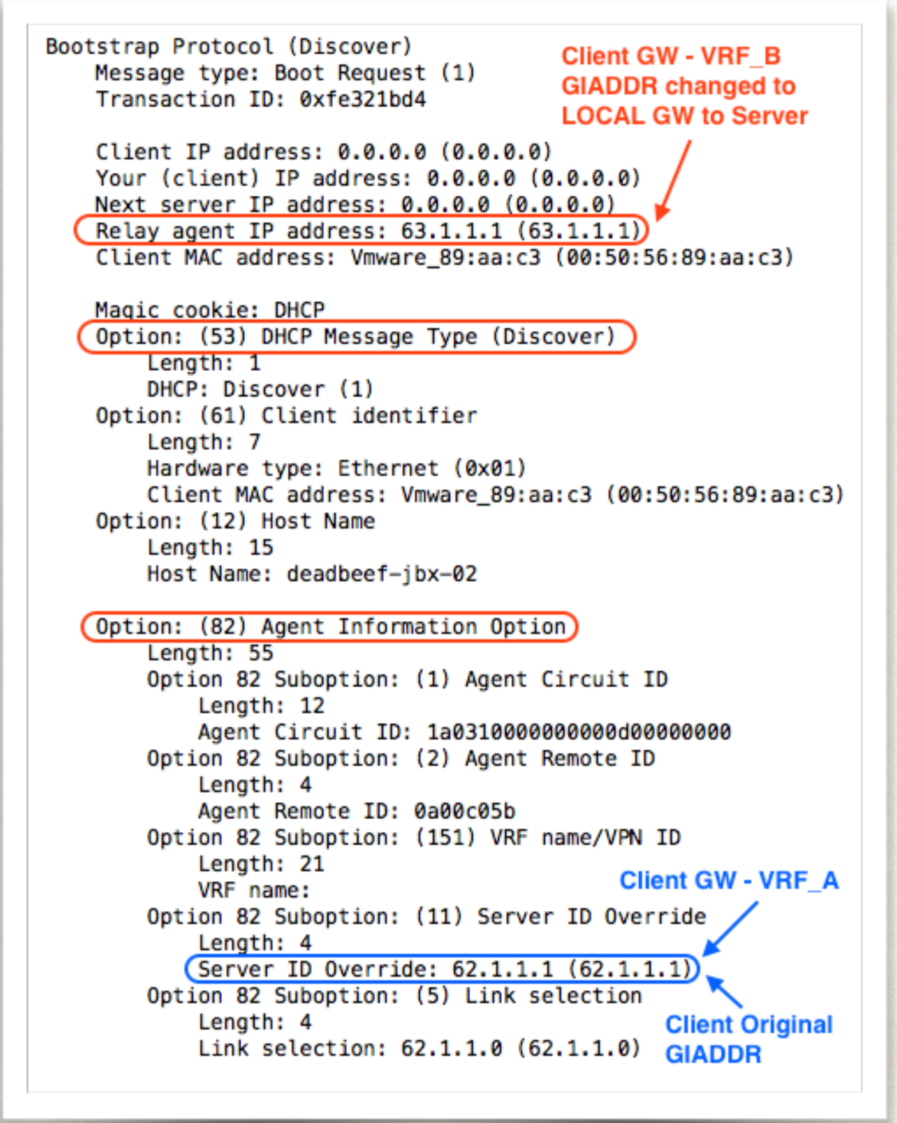
DHCP Option 82 - InterVRF (Failure)

CLIENT (VRF_A) - DHCP DISCOVER

- The DHCP Relay Proxy in VRF_B changes GIADDR to it's own SVI IP address per RFC specification.

WINDOWS 2012 SERVER (VRF_B) - DHCP OFFER

- Windows 2012 Server does not support "Link Selection" and uses GIADDR to select Client's Scope. Provides the Client an IP address from the wrong Scope.

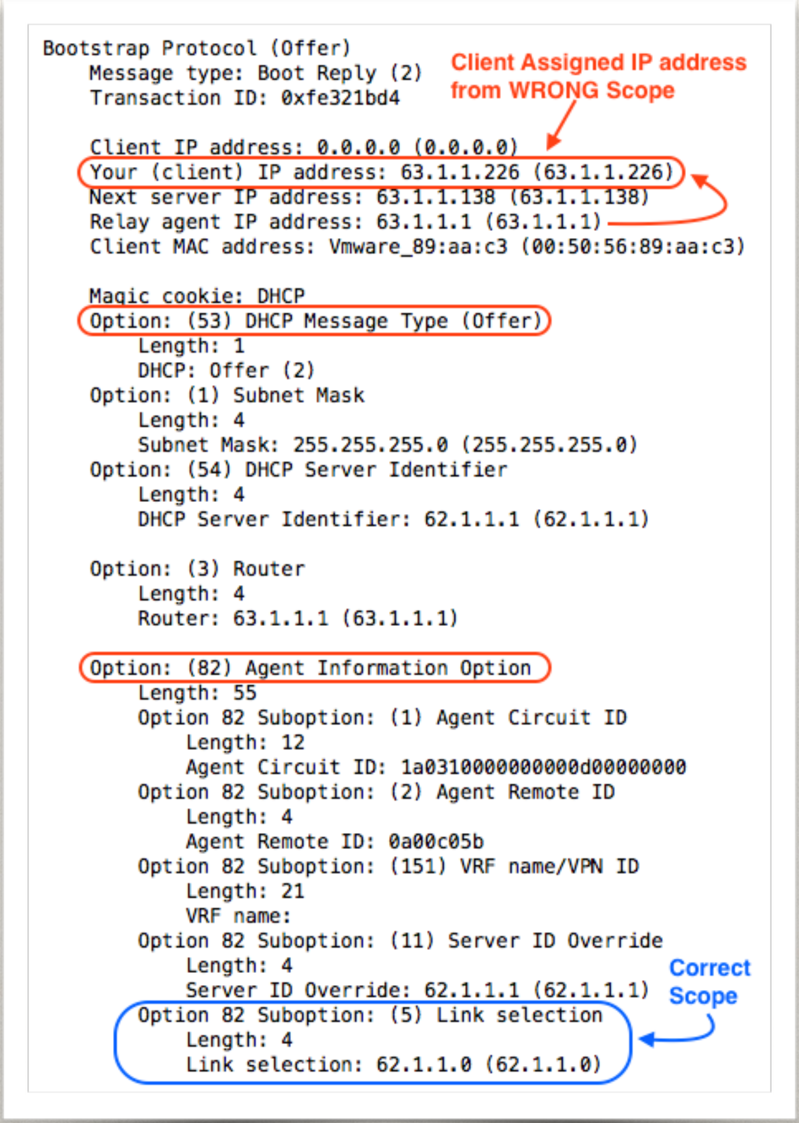


```
Bootstrap Protocol (Discover)
    Message type: Boot Request (1)
    Transaction ID: 0xfe321bd4

    Client IP address: 0.0.0.0 (0.0.0.0)
    Your (client) IP address: 0.0.0.0 (0.0.0.0)
    Next server IP address: 0.0.0.0 (0.0.0.0)
    Relay agent IP address: 63.1.1.1 (63.1.1.1)
    Client MAC address: Vmware_89:aa:c3 (00:50:56:89:aa:c3)

    Magic cookie: DHCP
    Option: (53) DHCP Message Type (Discover)
        Length: 1
        DHCP: Discover (1)
    Option: (61) Client identifier
        Length: 7
        Hardware type: Ethernet (0x01)
        Client MAC address: Vmware_89:aa:c3 (00:50:56:89:aa:c3)
    Option: (12) Host Name
        Length: 15
        Host Name: deadbeef-jbx-02

    Option: (82) Agent Information Option
        Length: 55
        Option 82 Suboption: (1) Agent Circuit ID
            Length: 12
            Agent Circuit ID: 1a0310000000000d00000000
        Option 82 Suboption: (2) Agent Remote ID
            Length: 4
            Agent Remote ID: 0a00c05b
        Option 82 Suboption: (151) VRF name/VPN ID
            Length: 21
            VRF name:
        Option 82 Suboption: (11) Server ID Override
            Length: 4
            Server ID Override: 62.1.1.1 (62.1.1.1)
        Option 82 Suboption: (5) Link selection
            Length: 4
            Link selection: 62.1.1.0 (62.1.1.0)
```

Client GW - VRF_B
GIADDR changed to
LOCAL GW to Server

Client GW - VRF_A

Client Original
GIADDR



```
Bootstrap Protocol (Offer)
    Message type: Boot Reply (2)
    Transaction ID: 0xfe321bd4

    Client IP address: 0.0.0.0 (0.0.0.0)
    Your (client) IP address: 63.1.1.226 (63.1.1.226)
    Next server IP address: 63.1.1.138 (63.1.1.138)
    Relay agent IP address: 63.1.1.1 (63.1.1.1)
    Client MAC address: Vmware_89:aa:c3 (00:50:56:89:aa:c3)

    Magic cookie: DHCP
    Option: (53) DHCP Message Type (Offer)
        Length: 1
        DHCP: Offer (2)
    Option: (1) Subnet Mask
        Length: 4
        Subnet Mask: 255.255.255.0 (255.255.255.0)
    Option: (54) DHCP Server Identifier
        Length: 4
        DHCP Server Identifier: 62.1.1.1 (62.1.1.1)

    Option: (3) Router
        Length: 4
        Router: 63.1.1.1 (63.1.1.1)

    Option: (82) Agent Information Option
        Length: 55
        Option 82 Suboption: (1) Agent Circuit ID
            Length: 12
            Agent Circuit ID: 1a0310000000000d00000000
        Option 82 Suboption: (2) Agent Remote ID
            Length: 4
            Agent Remote ID: 0a00c05b
        Option 82 Suboption: (151) VRF name/VPN ID
            Length: 21
            VRF name:
        Option 82 Suboption: (11) Server ID Override
            Length: 4
            Server ID Override: 62.1.1.1 (62.1.1.1)
        Option 82 Suboption: (5) Link selection
            Length: 4
            Link selection: 62.1.1.0 (62.1.1.0)
```

Client Assigned IP address
from WRONG Scope

Correct
Scope

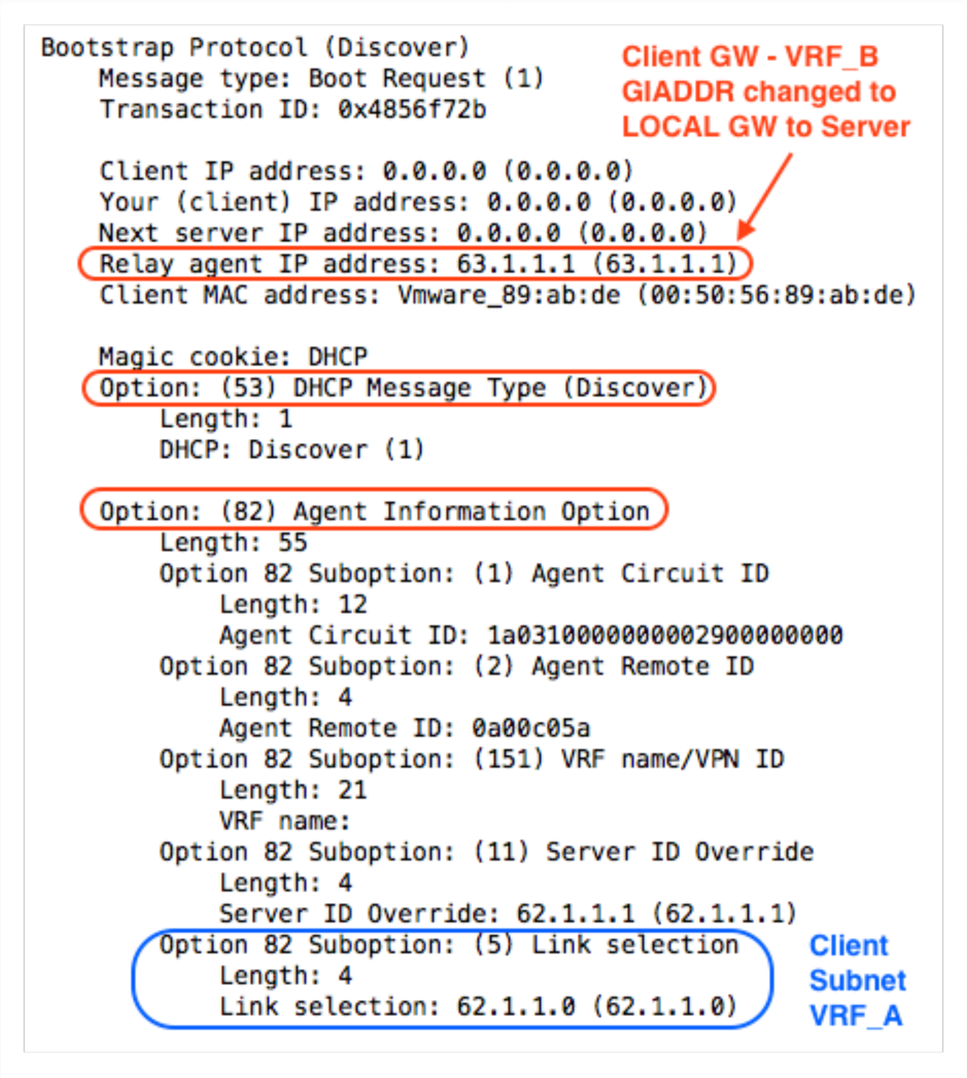
DHCP Option 82 - InterVRF (Success)

CLIENT (VRF_A) - DHCP DISCOVER

- The DHCP Relay Proxy in VRF_B changes GIADDR to it's own SVI IP address per RFC specification.

LINUX SERVER (VRF_B) - DHCP OFFER

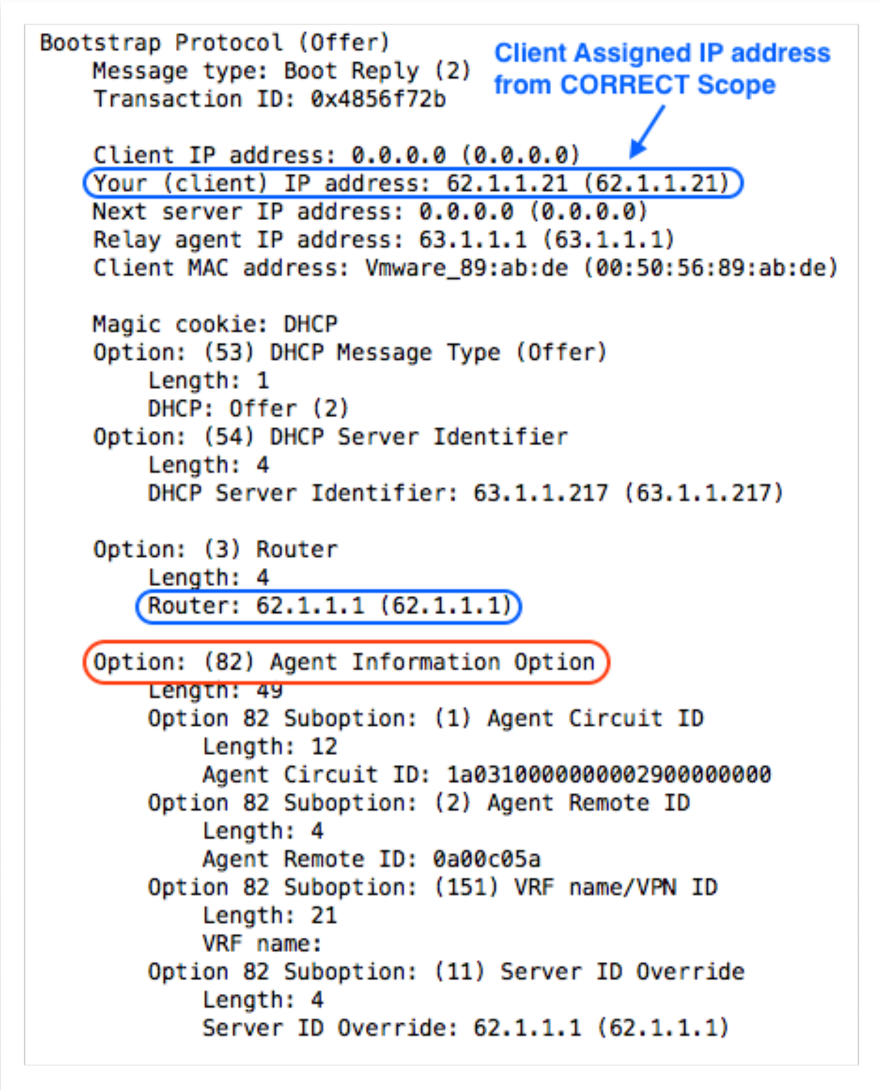
- Linux Server supports "Link Selection" and uses "Link Selection" to select Client's Scope. Provides the Client an IP address from the correct Scope.



```
Bootstrap Protocol (Discover)
    Message type: Boot Request (1)
    Transaction ID: 0x4856f72b

    Client IP address: 0.0.0.0 (0.0.0.0)
    Your (client) IP address: 0.0.0.0 (0.0.0.0)
    Next server IP address: 0.0.0.0 (0.0.0.0)
    Relay agent IP address: 63.1.1.1 (63.1.1.1)
    Client MAC address: Vmware_89:ab:de (00:50:56:89:ab:de)

    Magic cookie: DHCP
    Option: (53) DHCP Message Type (Discover)
        Length: 1
        DHCP: Discover (1)

    Option: (82) Agent Information Option
        Length: 55
        Option 82 Suboption: (1) Agent Circuit ID
            Length: 12
            Agent Circuit ID: 1a0310000000002900000000
        Option 82 Suboption: (2) Agent Remote ID
            Length: 4
            Agent Remote ID: 0a00c05a
        Option 82 Suboption: (151) VRF name/VPN ID
            Length: 21
            VRF name:
        Option 82 Suboption: (11) Server ID Override
            Length: 4
            Server ID Override: 62.1.1.1 (62.1.1.1)
        Option 82 Suboption: (5) Link selection
            Length: 4
            Link selection: 62.1.1.0 (62.1.1.0)
```

Client GW - VRF_B GIADDR changed to LOCAL GW to Server

Client Subnet VRF_A



```
Bootstrap Protocol (Offer)
    Message type: Boot Reply (2)
    Transaction ID: 0x4856f72b

    Client IP address: 0.0.0.0 (0.0.0.0)
    Your (client) IP address: 62.1.1.21 (62.1.1.21)
    Next server IP address: 0.0.0.0 (0.0.0.0)
    Relay agent IP address: 63.1.1.1 (63.1.1.1)
    Client MAC address: Vmware_89:ab:de (00:50:56:89:ab:de)

    Magic cookie: DHCP
    Option: (53) DHCP Message Type (Offer)
        Length: 1
        DHCP: Offer (2)
    Option: (54) DHCP Server Identifier
        Length: 4
        DHCP Server Identifier: 63.1.1.217 (63.1.1.217)

    Option: (3) Router
        Length: 4
        Router: 62.1.1.1 (62.1.1.1)

    Option: (82) Agent Information Option
        Length: 49
        Option 82 Suboption: (1) Agent Circuit ID
            Length: 12
            Agent Circuit ID: 1a0310000000002900000000
        Option 82 Suboption: (2) Agent Remote ID
            Length: 4
            Agent Remote ID: 0a00c05a
        Option 82 Suboption: (151) VRF name/VPN ID
            Length: 21
            VRF name:
        Option 82 Suboption: (11) Server ID Override
            Length: 4
            Server ID Override: 62.1.1.1 (62.1.1.1)
```

Client Assigned IP address from CORRECT Scope

Microsoft - DHCP Option 82 support (update)

As noted earlier, Microsoft Server 2012 supports Option 82 and ONLY sub-options: *Agent Circuit ID*, *Agent Remote ID*, and *VRF Name\VPN ID*. Microsoft Server 2012 does **NOT** support "Option 82 Suboption: Link selection" and will send a DHCP Offer with OPTION 82 and the Sub-options with an IP address from the WRONG subnet scope. *Cisco has a lot of ACI customers that deploy Microsoft as their DHCP Services Solution in their datacenter(s). Switching to Linux for DHCP services in most cases is not an option. So Cisco & Microsoft are working together to address the "Option 82" challenges.*

Currently their is work in progress to add enhancements to the DHCP Services in a version of the Windows 2016 Server releases. Microsoft internal reference numbers are:

Reference#s 7436729\7464838:
Cisco ACI: DHCP server does not honor Sub-Option 5 (Link Selection) as specified in RFC 3527

Reference#s 7435323\7464885:
Cisco ACI: DHCP server fails to include option-82 when issuing a NACK (negative acknowledgement) message to the client

I am testing the fixes and so far the testing has been positive and I look forward to the future software releases with the enhancements. There is "no" release date as of yet for these fixes. So, stayed tuned for updates. The purpose of this update is to make you all aware of the upcoming changes the will help our ACI customers address the "Option 82" challenges.

Another Enhancement that has been requested but not yet committed to be addressed:

Reference# 7471789:
Cisco ACI: Add support for VRF Name\VPN ID (RFC6607) to DHCP Server

Update as of 03-june-2016

Bridge Domains - Subnets

- DHCP Relay configuration for Bridge Domains with multiple subnets

When you configure a Bridge Domain with multiple subnets, the first subnet added becomes the “PRIMARY” IP address on the SVI interface. Subsequent subnets are configured as “SECONDARY” IP addresses.   Why is this an issue or caveat?

  - DHCP Relay policy can only be configured for the “PRIMARY” IP address on the SVI interface.
  - Under certain conditions, “PRIMARY” IP address on the SVI interface may change to one of the configured “SECONDARY” IP addresses. This would break your DHCP-Relay policy for this bridge domain. Possible scenarios would be configuring multiple addresses during a single transaction or importing a configuration with a bridge domain with multiple subnets.
  - use “**show ip interface vrf all**” to verify IP address assignments for the configured SVI Interfaces.

- CSCuq20803 - DHCP: Way to specify primary subnet for BD

References & Resources

References and Resources

Reference Links

- (Video) Cisco APIC - Configuring a DHCP Server Policy
  http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/getting-started/video/cisco_apic_configuring_dhcp_server_policy_using_gui.html
- Cisco Application Centric Infrastructure Fundamentals: Networking and Management Connectivity - DHCP Relay
  http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/aci-fundamentals/b_ACI-Fundamentals/b_ACI_Fundamentals_Beta_chapter_01111.html#concept_1D4F7C5492704AE0ACD6B8034A53C63A
- DHCP Relay Policy Examples
  http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/aci-fundamentals/b_ACI-Fundamentals/b_ACI_Fundamentals_Beta_appendix_01110.html
- rfc3046 - DHCP Relay Agent Information Option
  https://tools.ietf.org/rfc/rfc3046.txt
- rfc3256 - The DOCSIS (Data-Over-Cable Service Interface Specifications) Device Class DHCP (Dynamic Host Configuration Protocol) Relay Agent Information Sub-option
  https://tools.ietf.org/rfc/rfc3256.txt
- rfc3527 - Link Selection sub-option for the Relay Agent Information Option for DHCPv4
  https://tools.ietf.org/rfc/rfc3527.txt
- rfc3942- Reclassifying Dynamic Host Configuration Protocol version 4 (DHCPv4) Options
  https://tools.ietf.org/rfc/rfc3942.txt
- rfc3993 - Subscriber-ID Suboption for the Dynamic Host Configuration Protocol (DHCP) Relay Agent Option
  https://tools.ietf.org/rfc/rfc3993.txt

References and Resources (cont.)

Reference Links (cont.)

- rfc4243 - Vendor-Specific Information Suboption for the Dynamic Host Configuration Protocol (DHCP) Relay Agent Option. *https://tools.ietf.org/rfc/rfc4243.txt*
- rfc5107 - DHCP Server Identifier Override Suboption *https://tools.ietf.org/rfc/rfc5107.txt*
- rfc6607 - Virtual Subnet Selection Options for DHCPv4 and DHCPv6. *https://tools.ietf.org/rfc/rfc6607.txt*

Switch Node CLI Commands

- Show dhcp internal errors
- Show dhcp internal event-history msgs
- Show dhcp internal event-history traces
- Show dhcp internal info relay address interface [leaf:vlan#]
- Show dhcp internal info relay discover
- Show ip dhcp relay
- Show ip interface vrf [tenant:context]
- Show ip route vrf [tenant:context]

VISORE Class or DN

- (dhcpProvDhcp, dhcpRelayP, dhcpRsProv, dhcpRtLblDefToRelayP)

References and Resources (cont.)

Sample Linux Server “dhcpd.conf” file to support suboption “link-selection”

```
ddns-update-style interim;
ignore client-updates;
authoritative;
stash-agent-options true;
option agent.link-selection ip-address;
option routers 191.11.42.1;

class "deadbeef-19111x" {
    match if(binary-to-ascii(10, 8, ".", option agent.link-selection) = "191.1.1.0");
}

shared-network deadbeef-19111x {
    subnet 191.1.1.0 netmask 255.255.255.0 {
        option routers      191.1.1.1;
        option subnet-mask  255.255.255.0;

        pool {
            allow members of "deadbeef-19111x";
            range 191.1.1.201 191.1.1.209;
        }
    }
}

class "deadbeef-19112x" {
    match if(binary-to-ascii(10, 8, ".", option agent.link-selection) = "191.1.2.0");
}

shared-network deadbeef-19112x {
    subnet 191.1.2.0 netmask 255.255.255.0 {
        option routers      191.1.2.1;
        option subnet-mask  255.255.255.0;

        pool {
            allow members of "deadbeef-19112x";
            range 191.1.2.201 191.1.2.209;
        }
    }
}
```

References and Resources (cont.)

Sample Linux Server “dhcpd.conf” file to support suboption “link-selection” (cont.)

```
class "deadbeef-19113x" {
    match if(binary-to-ascii(10, 8, ".", option agent.link-selection) = "191.1.3.0");
}

shared-network deadbeef-19113x {
    subnet 191.1.3.0 netmask 255.255.255.0 {
        option routers      191.1.3.1;
        option subnet-mask  255.255.255.0;

        pool {
            allow members of "deadbeef-19113x";
            range 191.1.3.201 191.1.3.209;
        }
    }
}

class "deadbeef-19114x" {
    match if(binary-to-ascii(10, 8, ".", option agent.link-selection) = "191.1.4.0");
}

shared-network deadbeef-19114x {
    subnet 191.1.4.0 netmask 255.255.255.0 {
        option routers      191.1.4.1;
        option subnet-mask  255.255.255.0;

        pool {
            allow members of "deadbeef-19114x";
            range 191.1.4.201 191.1.4.209;
        }
    }
}
```

References and Resources (cont.)

Sample Linux Server “dhcpd.conf” file to support suboption “link-selection” (cont.)

```
class "deadbeef-19115x" {
    match if(binary-to-ascii(10, 8, ".", option agent.link-selection) = "191.1.5.0");
}

shared-network deadbeef-19115x {
    subnet 191.1.5.0 netmask 255.255.255.0 {
        option routers      191.1.5.1;
        option subnet-mask  255.255.255.0;

        pool {
            allow members of "deadbeef-19115x";
            range 191.1.5.201 191.1.5.209;
        }
    }
}

class "deadbeef-19116x" {
    match if(binary-to-ascii(10, 8, ".", option agent.link-selection) = "191.1.6.0");
}

shared-network deadbeef-19116x {
    subnet 191.1.6.0 netmask 255.255.255.0 {
        option routers      191.1.6.1;
        option subnet-mask  255.255.255.0;

        pool {
            allow members of "deadbeef-19116x";
            range 191.1.6.201 191.1.6.209;
        }
    }
}
```

References and Resources (cont.)

Sample Linux Server "dhcpd.conf" file to support suboption "link-selection" (cont.)

```
class "deadbeef-19117x" {
    match if(binary-to-ascii(10, 8, ".", option agent.link-selection) = "191.1.7.0");
}

shared-network deadbeef-19117x {
    subnet 191.1.7.0 netmask 255.255.255.0 {
        option routers      191.1.7.1;
        option subnet-mask  255.255.255.0;

        pool {
            allow members of "deadbeef-19117x";
            range 191.1.7.201 191.1.7.209;
        }
    }
}

class "deadbeef-19118x" {
    match if(binary-to-ascii(10, 8, ".", option agent.link-selection) = "191.1.8.0");
}

shared-network deadbeef-19118x {
    subnet 191.1.8.0 netmask 255.255.255.0 {
        option routers      191.1.8.1;
        option subnet-mask  255.255.255.0;

        pool {
            allow members of "deadbeef-19118x";
            range 191.1.8.201 191.1.8.209;
        }
    }
}
```

References and Resources (cont.)

Sample Linux Server “dhcpd.conf” file to support suboption “link-selection” (cont.)

```
class "deadbeef-19119x" {
    match if(binary-to-ascii(10, 8, ".", option agent.link-selection) = "191.1.9.0");
}

shared-network deadbeef-19119x {
    subnet 191.1.9.0 netmask 255.255.255.0 {
        option routers      191.1.9.1;
        option subnet-mask  255.255.255.0;

        pool {
            allow members of "deadbeef-19119x";
            range 191.1.9.201 191.1.9.209;
        }
    }
}

class "deadbeef-191110x" {
    match if(binary-to-ascii(10, 8, ".", option agent.link-selection) = "191.1.10.0");
}

shared-network deadbeef-191110x {
    subnet 191.1.10.0 netmask 255.255.255.0 {
        option routers      191.1.10.1;
        option subnet-mask  255.255.255.0;

        pool {
            allow members of "deadbeef-191110x";
            range 191.1.10.201 191.1.10.209;
        }
    }
}
```

References and Resources (cont.)

Sample Linux Server “dhcpd.conf” file to support suboption “link-selection” (cont.)

```
class "deadbeef-191111x" {
    match if(binary-to-ascii(10, 8, ".", option agent.link-selection) = "191.1.11.0");
}

shared-network deadbeef-191111x {
    subnet 191.1.11.0 netmask 255.255.255.0 {
        option routers      191.1.11.1;
        option subnet-mask  255.255.255.0;

        pool {
            allow members of "deadbeef-191111x";
            range 191.1.11.201 191.1.11.209;
        }
    }
}


class "deadbeef-191112x" {
    match if(binary-to-ascii(10, 8, ".", option agent.link-selection) = "191.1.12.0");
}

shared-network deadbeef-191112x {
    subnet 191.1.12.0 netmask 255.255.255.0 {
        option routers      191.1.12.1;
        option subnet-mask  255.255.255.0;

        pool {
            allow members of "deadbeef-191112x";
            range 191.1.12.201 191.1.12.209;
        }
    }
}
```

References and Resources (cont.)

Sample Linux Server “dhcpd.conf” file to support suboption “link-selection” (cont.)

```
class "deadbeef-191113x" {
    match if(binary-to-ascii(10, 8, ".", option agent.link-selection) = "191.1.13.0");
}

shared-network deadbeef-191113x {
    subnet 191.1.13.0 netmask 255.255.255.0 {
        option routers      191.1.13.1;
        option subnet-mask  255.255.255.0;

        pool {
            allow members of "deadbeef-191113x";
            range 191.1.13.201 191.1.13.209;
        }
    }
}

class "deadbeef-191114x" {
    match if(binary-to-ascii(10, 8, ".", option agent.link-selection) = "191.1.14.0");
}

shared-network deadbeef-191114x {
    subnet 191.1.14.0 netmask 255.255.255.0 {
        option routers      191.1.14.1;
        option subnet-mask  255.255.255.0;

        pool {
            allow members of "deadbeef-191114x";
            range 191.1.14.201 191.1.14.209;
        }
    }
}
```

References and Resources (cont.)

Sample Linux Server “dhcpd.conf” file to support suboption “link-selection” (cont.)

```
class "deadbeef-191184x" {
    match if(binary-to-ascii(10, 8, ".", option agent.link-selection) = "191.1.84.0");
}

shared-network deadbeef-191184x {
    subnet 191.1.84.0 netmask 255.255.255.0 {
        option routers      191.1.84.1;
        option subnet-mask  255.255.255.0;

        pool {
            allow members of "deadbeef-191184x";
            range 191.1.84.200 191.1.84.209;
        }
    }
}


subnet 191.11.42.0 netmask 255.255.255.0 {
    option routers      191.11.42.1;
    option subnet-mask  255.255.255.0;

    pool {
            range 191.11.42.200 191.11.42.209;
    }
}

subnet 191.1.24.0 netmask 255.255.255.0 {
    option routers      191.1.24.1;
    option subnet-mask  255.255.255.0;

    pool {
            range 191.1.24.201 191.1.24.209;
    }
}
```