Understand how AireOS WLCs Handle DHCP Protocol

Contents

Introduction

This document describes the different DHCP operations on the Cisco AireOS wireless controller.

External DHCP Server

The Wireless LAN Controller (WLC) supports two modes of DHCP operations in case an external DHCP

server is used:

- DHCP proxy mode
- DHCP bridging mode

DHCP proxy mode serves as a DHCP helper function in order to achieve better security and control over DHCP transactions between the DHCP server and the wireless clients. DHCP bridging mode provides an option to make the controller role in a DHCP transaction entirely transparent to the wireless clients.

Comparison of DHCP Proxy and Bridging Modes

Handling Client DHCP	DHCP Proxy Mode	DHCP Bridging Mode
Modify giaddr	Yes	No
Modify siaddr	Yes	No
Modify packet content	Yes	No
Redundant offers not forwarded	Yes	No
Option 82 support	Yes	No
Broadcast to unicast	Yes	No
BOOTP support	No	Server
RFC non-compliant	Proxy and relay agents are not exactly the same concept. DHCP bridging mode is recommended for full RFC compliance.	No

DHCP Proxy Mode

The DHCP proxy is not ideal for all network environments. The controller modifies and relays all DHCP transactions in order to provide a helper function and address certain security issues.

The controller virtual IP address is normally used as the source IP address of all DHCP transactions to the client. As a result, the real DHCP server IP address is not exposed in the air. This virtual IP is displayed in debug output for DHCP transactions on the controller. However, the use of a virtual IP address can cause issues for certain types of clients.

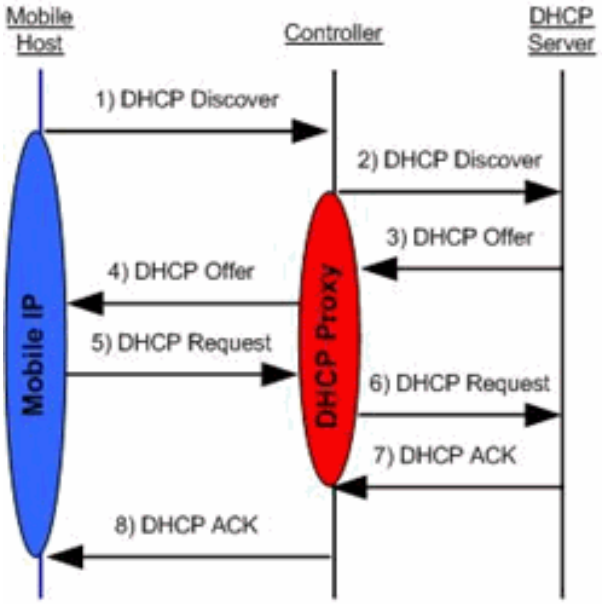
DHCP proxy mode operation maintains the same behavior for both, symmetric and asymmetric mobility protocols.

When multiple offers come from external DHCP servers, the DHCP proxy normally selects the first one that comes in and sets the IP address of the server in the client data structure. As a result, all subsequent transactions run through the same DHCP server until a transaction fails after retries. At this point, the proxy selects a different DHCP server for the client.

DHCP proxy is enabled by default. All controllers that communicate must have the same DHCP proxy setting.

Note: DHCP proxy must be enabled in order for DHCP option 82 to operate correctly.

Proxy Packet Flow



Handling of Packets for Local Clients

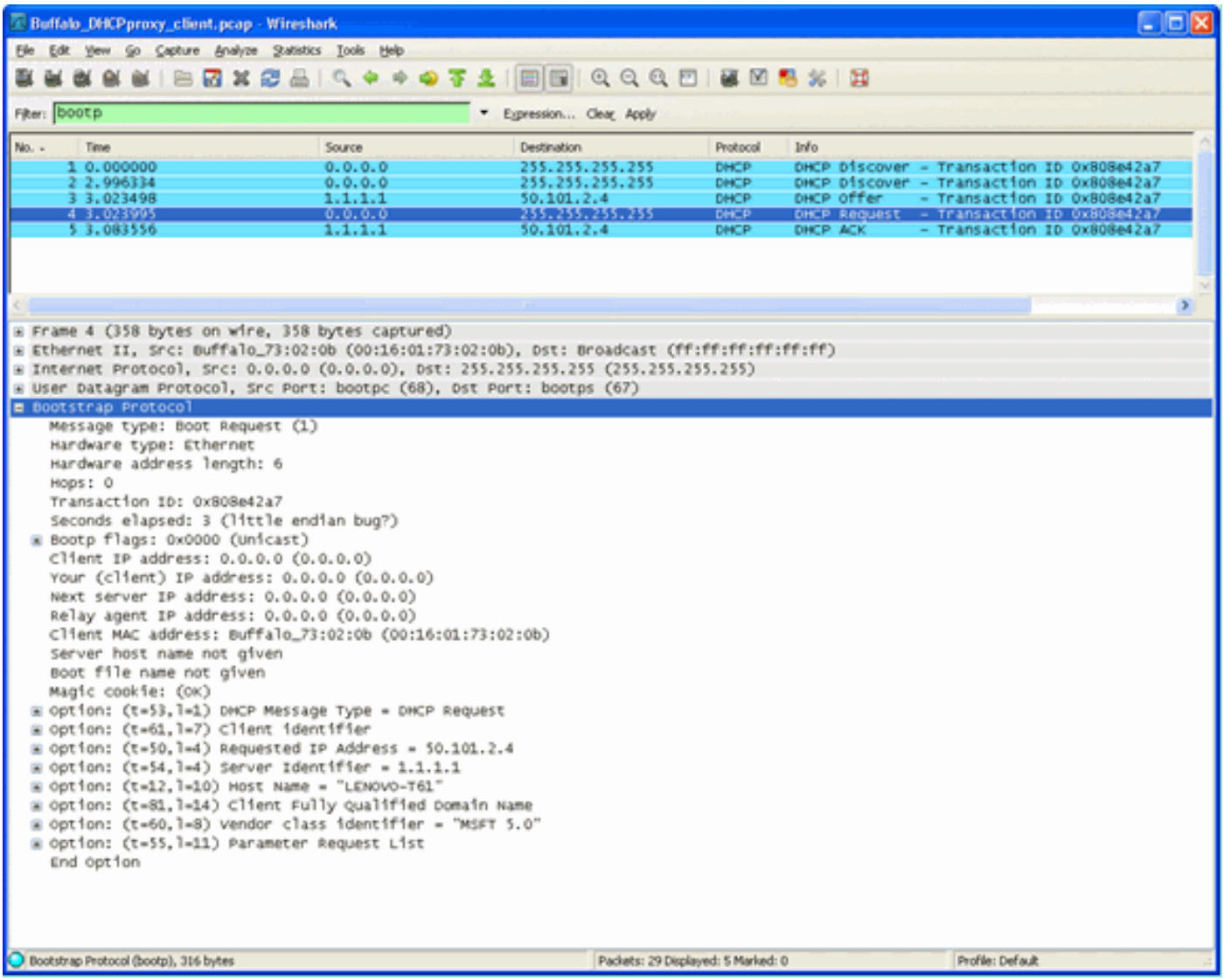
1) Client sends DHCP discover as all-subnets broadcast
2) Controller unicasts DHCP discover to DHCP servers configured on WLAN with WLAN IP address as source
3) DHCP server sends DHCP offer to controller (only first offer received by controller is processed. All others are dropped by proxy)
4) Controller unicasts DHCP offer to client with option 54 and source address set as controller's virtual IP (clients now believes controller is DHCP server)
5) Client sends DHCP request to virtual IP address
6) Controller unicasts DHCP request from WLAN IP address to DHCP server which returned the first offer to the client
7) DHCP server send ACK to controller
8) Controller unicasts ACK from the virtual IP to the client

Proxy Packet Capture

When the controller is in DHCP proxy mode, it not only directs DHCP packets to the DHCP server, it actually builds new DHCP packets to forward to the DHCP server. All DHCP options which are present in the client DHCP packets are copied into the controller DHCP packets. The next screenshot examples show this for a DHCP request packet.

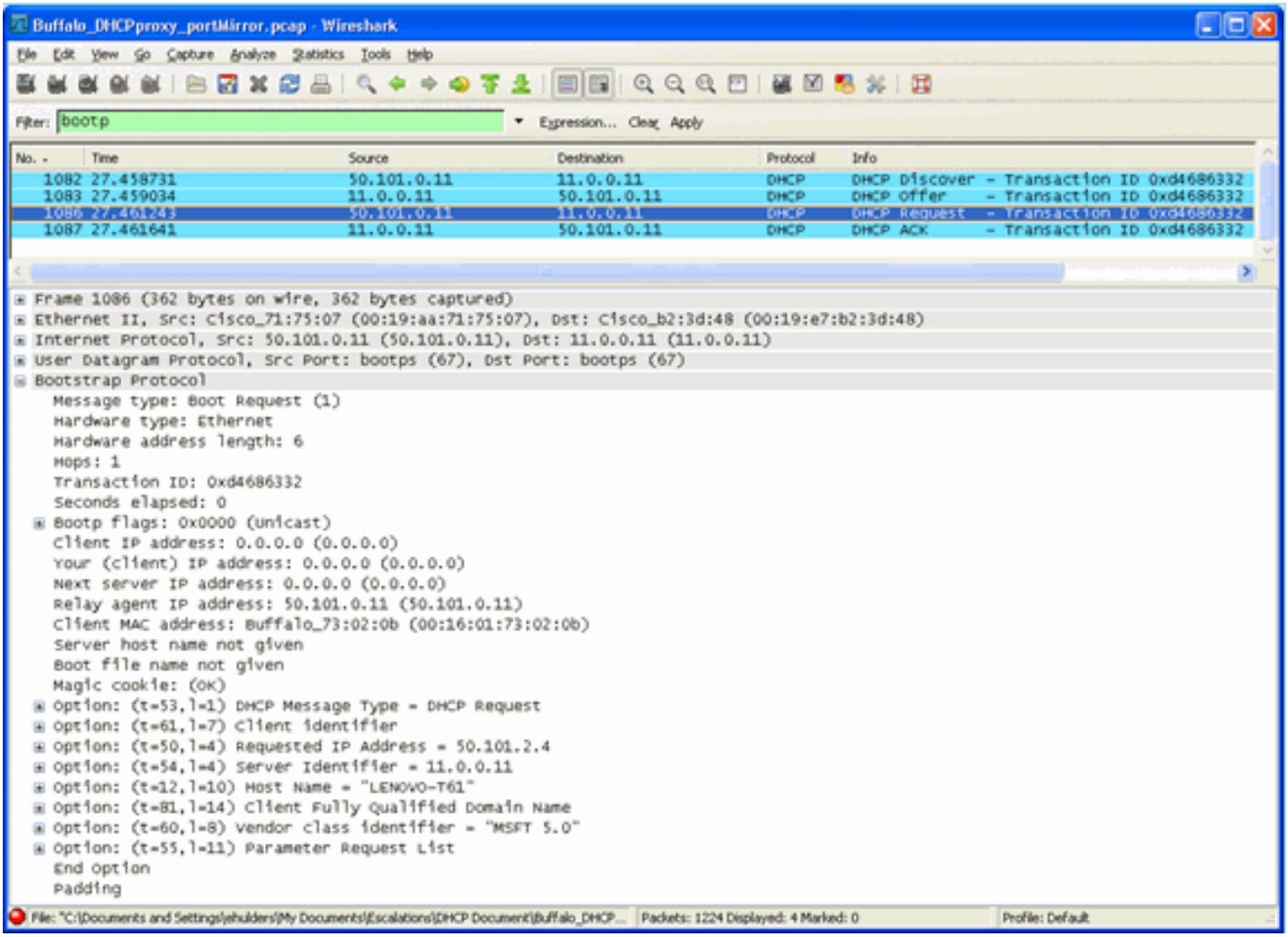
Client Perspective

This screenshot is of a packet capture taken from the perspective of the client. It shows a DHCP discover, DHCP offer, DHCP request, and a DHCP ACK. The DHCP request is highlighted and the boot p protocol detail is expanded, which shows the DHCP options.



Server Perspective

This screenshot is of a packet capture taken from the perspective of the server. Similar to the previous example, it shows a DHCP discover, DHCP offer, DHCP request, and a DHCP ACK. However, these are packets that the controller built as a function of the DHCP proxy. Again, the DHCP request is highlighted and the boot p protocol detail is expanded, which shows the DHCP options. Notice that they are the same as in the DHCP request packet of the client. Also, note that the WLC proxy relays the packet and highlights packet addresses.



Proxy Configuration Example

In order to use the controller as a DHCP proxy, the DHCP proxy feature must be enabled on the controller. By default, this feature is enabled. In order to enable the DHCP proxy, this CLI command can be used. The same is available in the GUI on the Controller page in the DHCP menu.

<#root>

```
(Cisco Controller) >
```

config dhcp proxy enable

```
(Cisco Controller) >
```

show dhcp proxy

```
DHCP Proxy Behavior: enabled
```

For the DHCP proxy to work, a primary DHCP server must be configured on each controller interface that requires DHCP services. A DHCP server can be configured on the management interface, the ap-manager interface, and on dynamic interfaces. These CLI commands can be used in order to configure a DHCP server for each interface.

<#root>

```
(Cisco Controller) >
config interface dhcp ap-manager primary <primary-server>
(Cisco Controller) >
config interface dhcp management primary <primary-server>
(Cisco Controller) >
config interface dhcp dynamic-interface <interface-name>


 primary <primary-server>
```

The DHCP bridging feature is a global setting, so it affects all DHCP transactions within the controller.

Troubleshoot

This is the output of the **debug dhcp packet enable** command. The debug shows a controller that receives a DHCP request from a client with MAC address 00:40:96:b4:8c:e1, transmits a DHCP request to the DHCP server, receives a reply from the DHCP server, and sends a DHCP offer to the client.

```
<#root>
(Cisco Controller) >
debug dhcp message enable


Thu Jun 25 21:48:55 2009: 00:40:96:b4:8c:e1 DHCP received op BOOTREQUEST (1)
 (len 312, port 29, encap 0xec03)

Thu Jun 25 21:48:55 2009: 00:40:96:b4:8c:e1 DHCP option len (including the magic cookie) 76
Thu Jun 25 21:48:55 2009: 00:40:96:b4:8c:e1 DHCP option: message type = DHCP REQUEST
Thu Jun 25 21:48:55 2009: 00:40:96:b4:8c:e1 DHCP option: 61 (len 7) - skipping
Thu Jun 25 21:48:55 2009: 00:40:96:b4:8c:e1 DHCP option: requested ip = 192.168.4.13
Thu Jun 25 21:48:55 2009: 00:40:96:b4:8c:e1 DHCP option: 12 (len 7) - skipping
Thu Jun 25 21:48:55 2009: 00:40:96:b4:8c:e1 DHCP option: 81 (len 11) - skipping
Thu Jun 25 21:48:55 2009: 00:40:96:b4:8c:e1 DHCP option: vendor class id = MSFT 5.0 (len 8)
Thu Jun 25 21:48:55 2009: 00:40:96:b4:8c:e1 DHCP option: 55 (len 11) - skipping
Thu Jun 25 21:48:55 2009: 00:40:96:b4:8c:e1 DHCP options end, len 76, actual 68
Thu Jun 25 21:48:55 2009: 00:40:96:b4:8c:e1 DHCP selecting relay 1 - control block settings:
                        dhcpServer: 0.0.0.0, dhcpNetmask: 0.0.0.0,
                        dhcpGateway: 0.0.0.0, dhcpRelay: 0.0.0.0  VLAN: 0

Thu Jun 25 21:48:55 2009: 00:40:96:b4:8c:e1 DHCP selected relay 1 - 192.168.3.1
 (local address 192.168.4.2, gateway 192.168.4.1, VLAN 101, port 29)
Thu Jun 25 21:48:55 2009: 00:40:96:b4:8c:e1 DHCP transmitting DHCP REQUEST (3)

Thu Jun 25 21:48:55 2009: 00:40:96:b4:8c:e1 DHCP   op: BOOTREQUEST, htype: Ethernet,
 hlen: 6, hops: 1
Thu Jun 25 21:48:55 2009: 00:40:96:b4:8c:e1 DHCP   xid: 0xfc3c9979 (4231829881), secs: 0,
 flags: 0
Thu Jun 25 21:48:55 2009: 00:40:96:b4:8c:e1 DHCP   chaddr: 00:40:96:b4:8c:e1
Thu Jun 25 21:48:55 2009: 00:40:96:b4:8c:e1 DHCP   ciaddr: 0.0.0.0,  yiaddr: 0.0.0.0
Thu Jun 25 21:48:55 2009: 00:40:96:b4:8c:e1 DHCP   siaddr: 0.0.0.0,  giaddr: 192.168.4.1
Thu Jun 25 21:48:55 2009: 00:40:96:b4:8c:e1 DHCP   requested ip: 192.168.4.13

Thu Jun 25 21:48:55 2009: 00:40:96:b4:8c:e1 DHCP Forwarding DHCP packet (332 octets)
```

```
          -- packet received on direct-connect port requires forwarding to external DHCP
 server. Next-hop is 192.168.4.1

Thu Jun 25 21:48:55 2009: 00:40:96:b4:8c:e1 DHCP sending REQUEST to 192.168.4.1
 (len 350, port 29, vlan 101)
Thu Jun 25 21:48:55 2009: 00:40:96:b4:8c:e1 DHCP selecting relay 2 - control block settings:
                        dhcpServer: 0.0.0.0, dhcpNetmask: 0.0.0.0,
                        dhcpGateway: 0.0.0.0, dhcpRelay: 192.168.4.1  VLAN: 101
Thu Jun 25 21:48:55 2009: 00:40:96:b4:8c:e1 DHCP selected relay 2 - NONE

Thu Jun 25 21:48:55 2009: 00:40:96:b4:8c:e1 DHCP received op BOOTREPLY (2) (len 316, port 29,
 encap 0xec00)

Thu Jun 25 21:48:55 2009: 00:40:96:b4:8c:e1 DHCP option len (including the magic cookie) 80
Thu Jun 25 21:48:55 2009: 00:40:96:b4:8c:e1 DHCP option: message type = DHCP ACK
Thu Jun 25 21:48:55 2009: 00:40:96:b4:8c:e1 DHCP option: 58 (len 4) - skipping
Thu Jun 25 21:48:55 2009: 00:40:96:b4:8c:e1 DHCP option: 59 (len 4) - skipping
Thu Jun 25 21:48:55 2009: 00:40:96:b4:8c:e1 DHCP option: lease time = 691200 seconds
Thu Jun 25 21:48:55 2009: 00:40:96:b4:8c:e1 DHCP option: server id = 192.168.3.1
Thu Jun 25 21:48:55 2009: 00:40:96:b4:8c:e1 DHCP option: netmask = 255.255.0.0
Thu Jun 25 21:48:55 2009: 00:40:96:b4:8c:e1 DHCP option: 15 (len 14) - skipping
Thu Jun 25 21:48:55 2009: 00:40:96:b4:8c:e1 DHCP option: gateway = 192.168.4.1
Thu Jun 25 21:48:55 2009: 00:40:96:b4:8c:e1 DHCP option: DNS server, cnt = 1, first = 192.168.3.1
Thu Jun 25 21:48:55 2009: 00:40:96:b4:8c:e1 DHCP option: WINS server, cnt = 1, first = 192.168.3.1
Thu Jun 25 21:48:55 2009: 00:40:96:b4:8c:e1 DHCP options end, len 80, actual 72
Thu Jun 25 21:48:55 2009: 00:40:96:b4:8c:e1 DHCP setting server from ACK (server 192.168.3.1,
 yiaddr 192.168.4.13)
Thu Jun 25 21:48:55 2009: 00:40:96:b4:8c:e1 Assigning Address 192.168.4.13 to mobile

Thu Jun 25 21:48:55 2009: 00:40:96:b4:8c:e1 DHCP sending REPLY to STA (len 424, port 29,
 vlan 20)
Thu Jun 25 21:48:55 2009: 00:40:96:b4:8c:e1 DHCP transmitting DHCP ACK (5)

Thu Jun 25 21:48:55 2009: 00:40:96:b4:8c:e1 DHCP   op: BOOTREPLY, htype: Ethernet, hlen: 6,
 hops: 0
Thu Jun 25 21:48:55 2009: 00:40:96:b4:8c:e1 DHCP   xid: 0xfc3c9979 (4231829881), secs: 0,
 flags: 0
Thu Jun 25 21:48:59 2009: 00:40:96:b4:8c:e1 DHCP   chaddr: 00:40:96:b4:8c:e1
Thu Jun 25 21:48:59 2009: 00:40:96:b4:8c:e1 DHCP   ciaddr: 0.0.0.0,  yiaddr: 192.168.4.13
Thu Jun 25 21:48:59 2009: 00:40:96:b4:8c:e1 DHCP   siaddr: 0.0.0.0,  giaddr: 0.0.0.0
Thu Jun 25 21:48:59 2009: 00:40:96:b4:8c:e1 DHCP   server id: 192.0.2.10  rcvd server id: 192.168.3.1
```

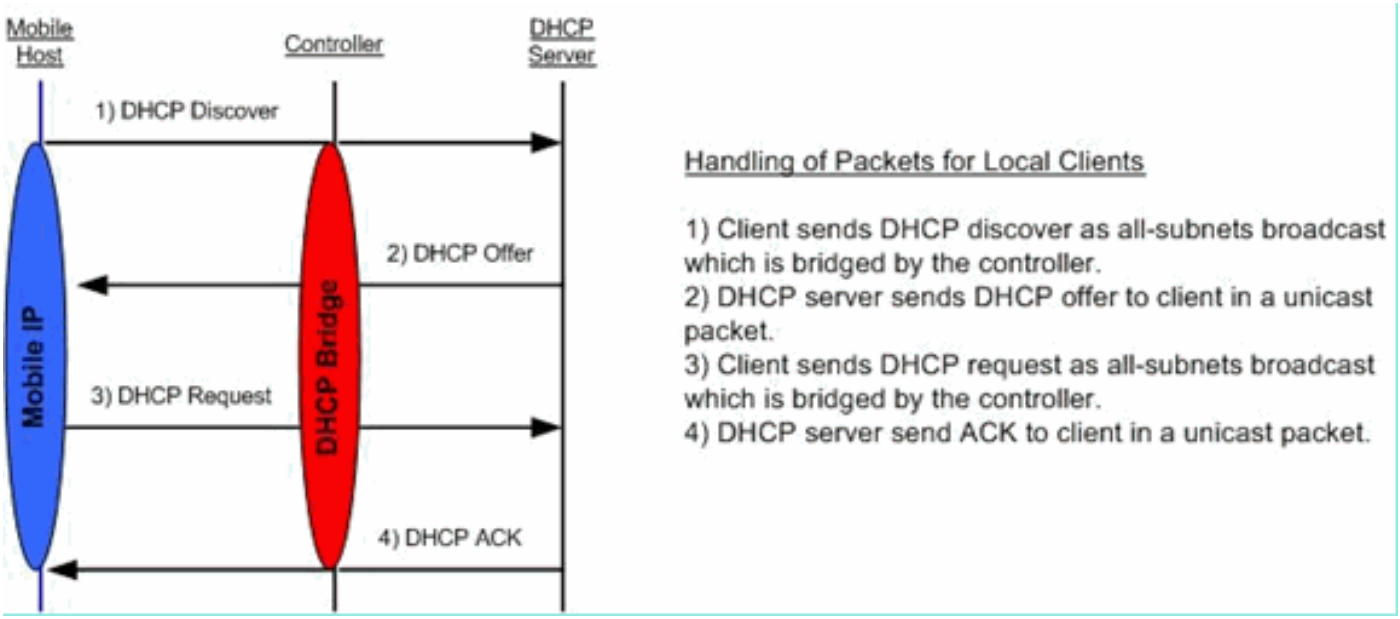
Caveats

- Interoperability issues can exist between a controller with DHCP proxy enabled and devices that act as both a firewall and DHCP server. This is most likely due to the firewall component of the device as firewalls generally do not respond to proxy requests. The workaround for this issue is to disable the DHCP proxy on the controller.
- When a client is in the DHCP REQ state on the controller, the controller drops DHCP inform packets. The client does not go into a RUN state on the controller (this is required for the client to pass traffic) until it receives a DHCP discover packet from the client. DHCP inform packets are forwarded by the controller when the DHCP proxy is disabled.
- All controllers that communicate with each other must have the same DHCP proxy setting.

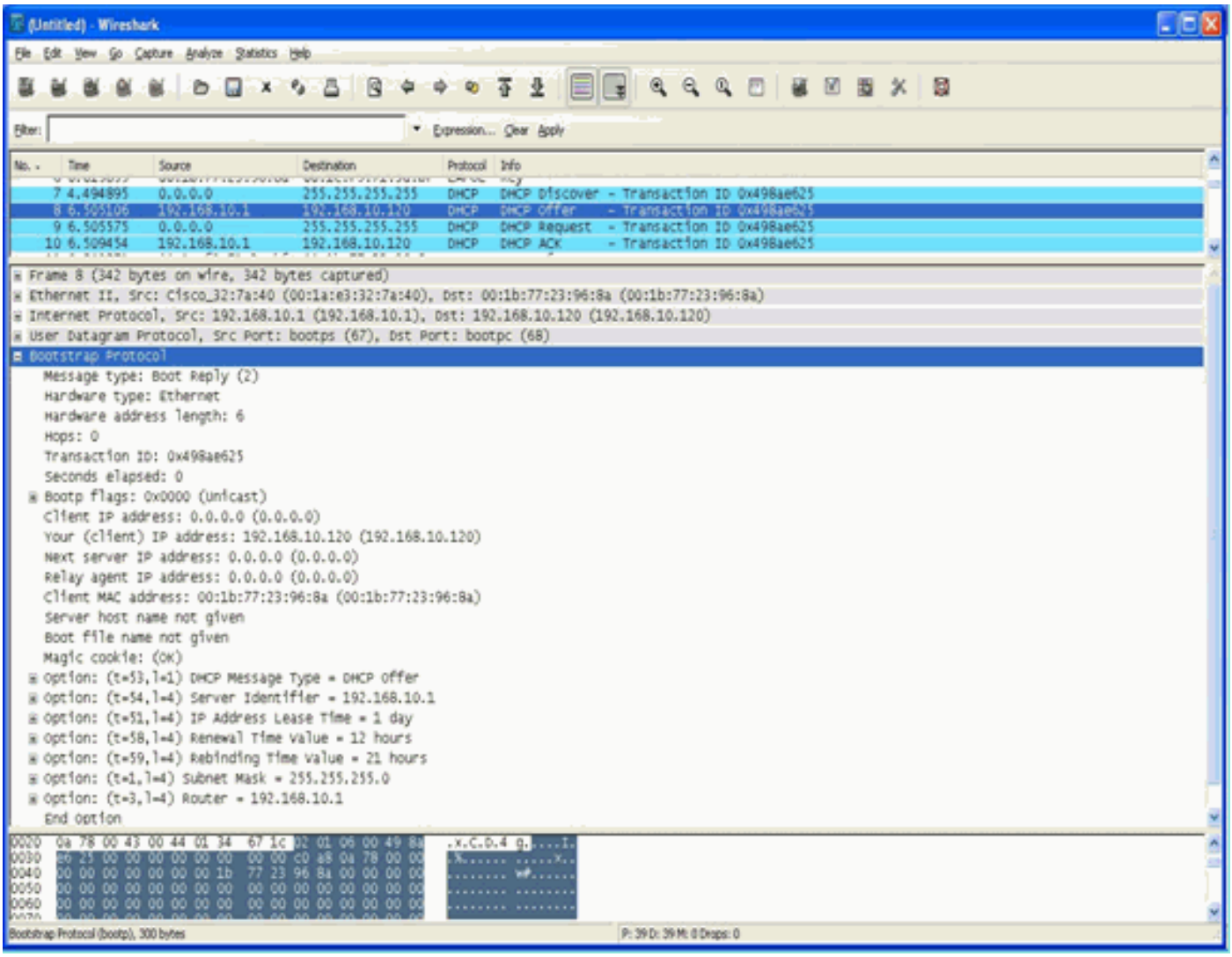
DHCP Bridging Mode

The DHCP bridging feature is designed to make the controller role in the DHCP transaction entirely transparent to the client. With the exception of 802.11 to Ethernet II conversion, packets from the client are bridged unmodified from the Light Weight Access Point Protocol (LWAPP) tunnel to the client VLAN (or Ethernet over IP (EoIP) tunnel in the L3 roaming case). Similarly, with the exception of Ethernet II to 802.11 conversions, packets to the client are bridged unmodified from the client VLAN (or EoIP tunnel in the L3 roaming case) to the LWAPP tunnel. Think of this as wiring a client into a switchport and then the client performs a traditional DHCP transaction.

DHCP Bridging Operations - Bridging Packet Flow

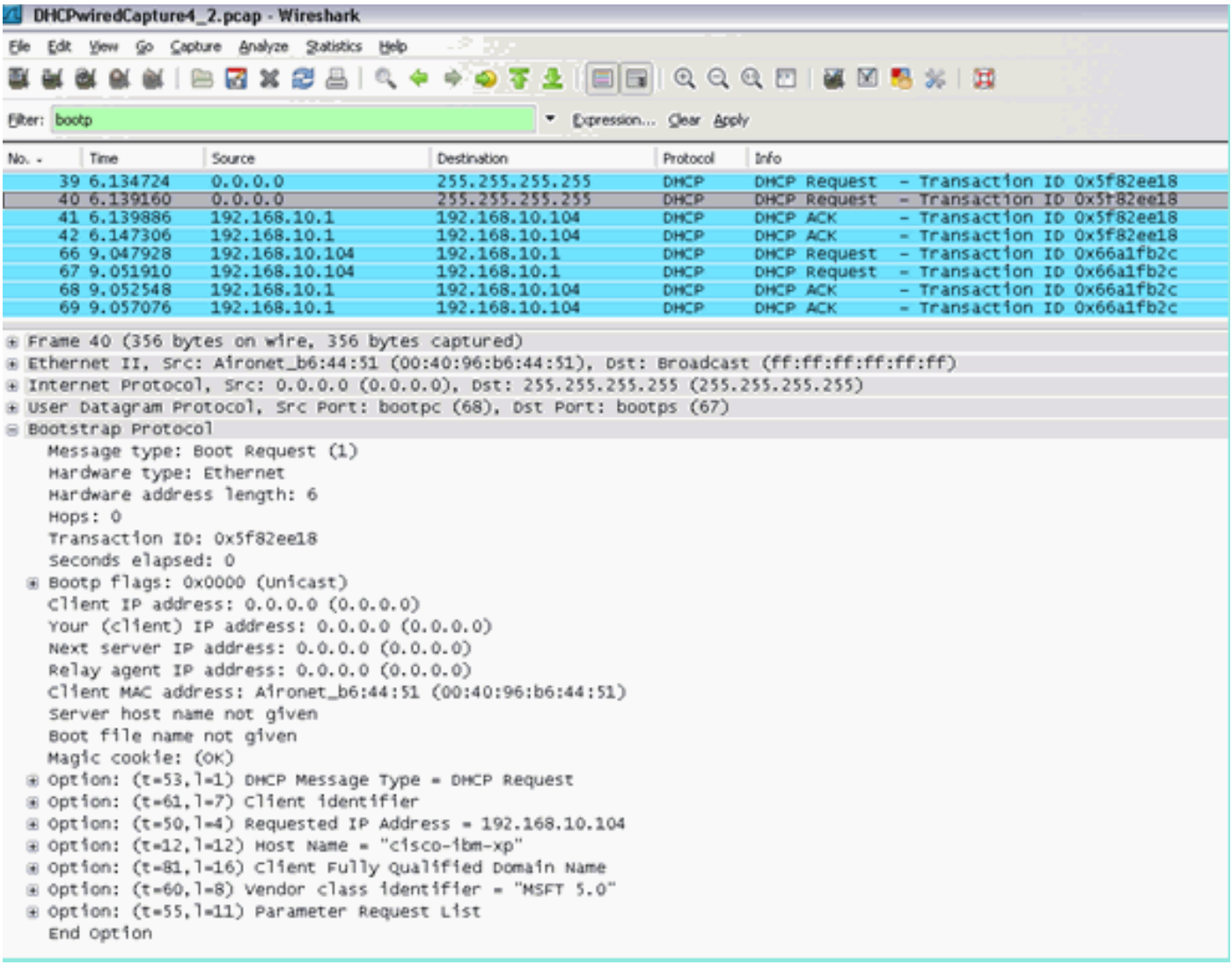


Bridging Packet Capture - Client Perspective



In the client-side packet capture screenshot, the main difference between the client capture in Proxy mode is the real IP of the DHCP server which is seen in the Offer and Ack packets instead of the controller virtual IP address.

Bridging Packet Capture - Server Perspective



In the wired packet capture screenshot you can see that packet 40 is the bridged DHCP Request broadcast from the test client 00:40:96:b6:44:51 to the wired network.

Bridging Configuration Example

In order to enable the DHCP bridging functionality on the controller, you must disable the DHCP proxy feature on the controller. This can only be accomplished in the CLI with these commands:

<#root>

```
(Cisco Controller) >
config dhcp proxy disable
(Cisco Controller) >
show dhcp proxy
DHCP Proxy Behaviour: disabled
```

If the DHCP server does not exist on the same Layer 2 (L2) network as the client, then the broadcast must be forwarded to the DHCP server at the client gateway through the use of an IP helper. This is a sample of this configuration:

```
<#root>

Switch#

conf t

Switch(config)#

interface vlan <client vlan #>

Switch(config-if)#

ip helper-address <dhcp server IP>
```

The DHCP bridging feature is a global setting, so it affects all DHCP transactions within the controller. You must add IP helper statements in the wired infrastructure for all necessary VLANs on the controller.

Troubleshoot

The debugs listed here were enabled on the controller CLI and the DHCP portion of the output was extracted for this document.

```
<#root>

(Cisco Controller) >

debug client 00:40:96:b6:44:51

(Cisco Controller) >

debug dhcp message enable


00:40:96:b6:44:51 DHCP received op BOOTREQUEST (1) (len 308, port 1, encap 0xec03)
00:40:96:b6:44:51 DHCP option len (including the magic cookie) 72
00:40:96:b6:44:51 DHCP option: message type = DHCP DISCOVER
00:40:96:b6:44:51 DHCP option: 116 (len 1) - skipping
00:40:96:b6:44:51 DHCP option: 61 (len 7) - skipping
00:40:96:b6:44:51 DHCP option: 12 (len 12) - skipping
00:40:96:b6:44:51 DHCP option: vendor class id = MSFT 5.0 (len 8)
00:40:96:b6:44:51 DHCP option: 55 (len 11) - skipping
00:40:96:b6:44:51 DHCP options end, len 72, actual 64
00:40:96:b6:44:51 DHCP processing DHCP DISCOVER (1)
00:40:96:b6:44:51 DHCP   op: BOOTREQUEST, htype: Ethernet, hlen: 6, hops: 0
00:40:96:b6:44:51 DHCP   xid: 0x224dfab6 (575535798), secs: 0, flags: 0
00:40:96:b6:44:51 DHCP   chaddr: 00:40:96:b6:44:51
00:40:96:b6:44:51 DHCP   ciaddr: 0.0.0.0,  yiaddr: 0.0.0.0
00:40:96:b6:44:51 DHCP   siaddr: 0.0.0.0,  giaddr: 0.0.0.0

00:40:96:b6:44:51 DHCP successfully bridged packet to DS

00:40:96:b6:44:51 DHCP received op BOOTREPLY (2) (len 308, port 1, encap 0xec00)
00:40:96:b6:44:51 DHCP option len (including the magic cookie) 72
00:40:96:b6:44:51 DHCP option: message type = DHCP OFFER

00:40:96:b6:44:51 DHCP option: server id = 192.168.10.1

00:40:96:b6:44:51 DHCP option: lease time = 84263 seconds
00:40:96:b6:44:51 DHCP option: 58 (len 4) - skipping
00:40:96:b6:44:51 DHCP option: 59 (len 4) - skipping
00:40:96:b6:44:51 DHCP option: netmask = 255.255.255.0
```

```
00:40:96:b6:44:51 DHCP option: gateway = 192.168.10.1
00:40:96:b6:44:51 DHCP options end, len 72, actual 64
00:40:96:b6:44:51 DHCP processing DHCP OFFER (2)
00:40:96:b6:44:51 DHCP   op: BOOTREPLY, htype: Ethernet, hlen: 6, hops: 0
00:40:96:b6:44:51 DHCP   xid: 0x224dfab6 (575535798), secs: 0, flags: 0
00:40:96:b6:44:51 DHCP   chaddr: 00:40:96:b6:44:51
00:40:96:b6:44:51 DHCP   ciaddr: 0.0.0.0,  yiaddr: 192.168.10.104
00:40:96:b6:44:51 DHCP   siaddr: 0.0.0.0,  giaddr: 0.0.0.0
00:40:96:b6:44:51 DHCP   server id: 192.168.10.1  rcvd server id: 192.168.10.1
```

00:40:96:b6:44:51 DHCP successfully bridged packet to STA

```
00:40:96:b6:44:51 DHCP received op BOOTREQUEST (1) (len 328, port 1, encap 0xec03)
00:40:96:b6:44:51 DHCP option len (including the magic cookie) 92
00:40:96:b6:44:51 DHCP option: message type = DHCP REQUEST
00:40:96:b6:44:51 DHCP option: 61 (len 7) - skipping
00:40:96:b6:44:51 DHCP option: requested ip = 192.168.10.104
00:40:96:b6:44:51 DHCP option: server id = 192.168.10.1
00:40:96:b6:44:51 DHCP option: 12 (len 12) - skipping
00:40:96:b6:44:51 DHCP option: 81 (len 16) - skipping
00:40:96:b6:44:51 DHCP option: vendor class id = MSFT 5.0 (len 8)
00:40:96:b6:44:51 DHCP option: 55 (len 11) - skipping
00:40:96:b6:44:51 DHCP options end, len 92, actual 84
00:40:96:b6:44:51 DHCP processing DHCP REQUEST (3)
00:40:96:b6:44:51 DHCP   op: BOOTREQUEST, htype: Ethernet, hlen: 6, hops: 0
00:40:96:b6:44:51 DHCP   xid: 0x224dfab6 (575535798), secs: 0, flags: 0
00:40:96:b6:44:51 DHCP   chaddr: 00:40:96:b6:44:51
00:40:96:b6:44:51 DHCP   ciaddr: 0.0.0.0,  yiaddr: 0.0.0.0
00:40:96:b6:44:51 DHCP   siaddr: 0.0.0.0,  giaddr: 0.0.0.0
00:40:96:b6:44:51 DHCP   requested ip: 192.168.10.104
00:40:96:b6:44:51 DHCP   server id: 192.168.10.1  rcvd server id: 192.168.10.1
```

00:40:96:b6:44:51 DHCP successfully bridged packet to DS

```
00:40:96:b6:44:51 DHCP received op BOOTREPLY (2) (len 308, port 1, encap 0xec00)
00:40:96:b6:44:51 DHCP option len (including the magic
cookie) 72 00:40:96:b6:44:51 DHCP option: message type = DHCP ACK
00:40:96:b6:44:51 DHCP option: server id = 192.168.10.1
00:40:96:b6:44:51 DHCP option: lease time = 86400 seconds
00:40:96:b6:44:51 DHCP option: 58 (len 4) - skipping
00:40:96:b6:44:51 DHCP option: 59 (len 4) - skipping
00:40:96:b6:44:51 DHCP option: netmask = 255.255.255.0
00:40:96:b6:44:51 DHCP option: gateway = 192.168.10.1
00:40:96:b6:44:51 DHCP options end, len 72, actual 64
00:40:96:b6:44:51 DHCP processing DHCP ACK (5)
00:40:96:b6:44:51 DHCP   op: BOOTREPLY, htype: Ethernet, hlen: 6, hops: 0
00:40:96:b6:44:51 DHCP   xid: 0x224dfab6 (575535798), secs: 0, flags: 0
00:40:96:b6:44:51 DHCP   chaddr: 00:40:96:b6:44:51
00:40:96:b6:44:51 DHCP   ciaddr: 0.0.0.0,  yiaddr: 192.168.10.104
00:40:96:b6:44:51 DHCP   siaddr: 0.0.0.0,  giaddr: 0.0.0.0
00:40:96:b6:44:51 DHCP   server id: 192.168.10.1  rcvd server id: 192.168.10.1
```

00:40:96:b6:44:51 Assigning Address 192.168.10.104 to mobile
00:40:96:b6:44:51 DHCP successfully bridged packet to STA
00:40:96:b6:44:51 192.168.10.104 Added NPU entry of type 1

In this DHCP debug output, there are a few key indications that DHCP bridging is in use on the controller:

- DHCP successfully bridged packet to DS - This means that the original DHCP packet from the client was bridged, unaltered to the distribution system (DS). The DS is the wired infrastructure.
- DHCP successfully bridged packet to STA -This message indicates that the DHCP packet was

bridged, unaltered to the station (STA). The STA is the client machine that requests DHCP.

Also, you see the actual server IP address listed in the debugs, which is 192.168.10.1. If DHCP proxy is in use instead of DHCP bridging, you can see the controller virtual IP address listed for the server IP address.

Caveats

- By default, the DHCP proxy is enabled.
- All controllers that communicate with each other must have the same DHCP proxy setting.
- DHCP proxy must be enabled for DHCP option 82 to work.

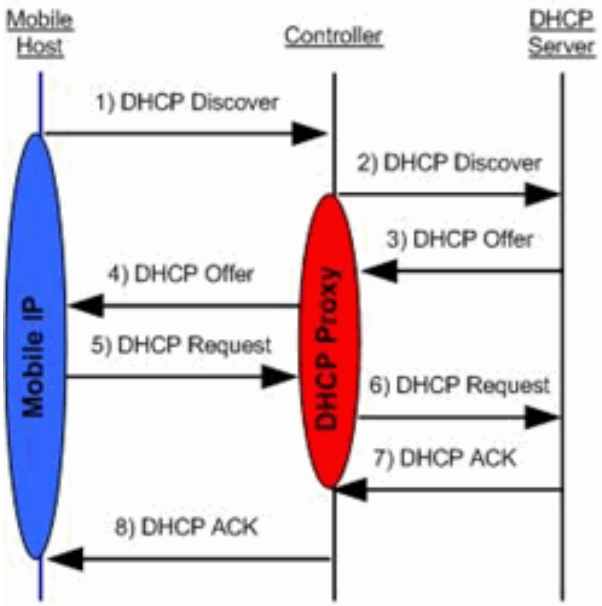
Internal DHCP Server

The internal DHCP server was introduced initially for branch offices where an external DHCP server is not available. It is designed in order to support a small wireless network with less than ten Access Points (APs) that are on the same subnet. The internal server provides IP addresses to wireless clients, direct-connect APs, appliance-mode APs on the management interface, and DHCP requests that are relayed from APs. It is not a full-blown general-purpose DHCP server. It only supports limited functionality and does not scale in a larger deployment.

Comparison of Internal DHCP and Bridging Modes

The two main DHCP modes on the controller are either DHCP proxy or DHCP bridging. With DHCP bridging the controller acts more like a DHCP back with autonomous APs. A DHCP packet comes into the AP via a client association to a Service Set Identifier (SSID) that is linked to a VLAN. Then, the DHCP packet goes out that VLAN. If an IP helper is defined on the Layer 3 (L3) gateway of that VLAN, the packet is forwarded to that DHCP server via directed unicast. The DHCP server then responds back directly to the L3 interface that forwarded that DHCP packet. With DHCP proxy, it is the same idea, but all of the forwarding is done directly at the controller instead of the L3 interface of the VLAN. For example, a DHCP request comes into the WLAN from the client, the WLAN then either uses the DHCP server defined on the interface of the VLAN *or* uses the DHCP override function of the WLAN in order to forward an unicast DHCP packet to the DHCP server with the DHCP packets GIADDR field filled out to be the IP address of the VLAN interface.

Internal DHCP Server - Packet flow



Handling of Packets for Local Clients

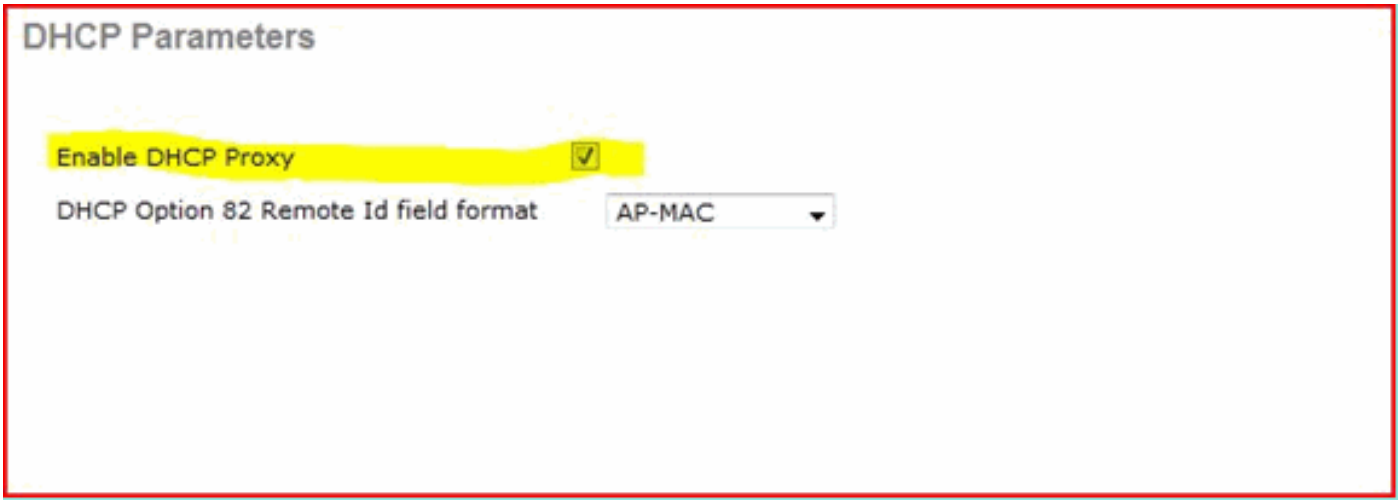
1) Client sends DHCP discover as all-subnets broadcast
2) Controller forwards the DHCP discover via the DHCP proxy service of the controller to the internal DHCP server (Note: the configured DHCP server IP address must be the management IP address of the controller).
3) Internal DHCP server sends DHCP offer back to the DHCP proxy agent on the controller.
4) Controller unicasts DHCP offer to client with option 54 and source address set as controller's management IP address.
5) Client sends DHCP request to the management IP address.
6) Controller unicasts DHCP request from WLAN IP address to DHCP proxy service which then forwards the request to the internal DHCP server.
7) Internal DHCP server sends ACK to the DHCP proxy service.
8) Controller unicasts ACK to the client.

Internal DHCP Server Configuration Example

You must enable the DHCP proxy on the controller in order to allow the internal DHCP server to function. This can be done via the GUI under this section:

Note: You are not able to set the DHCP proxy via the GUI in all versions.

```
Controller->Advanced->DHCP
```

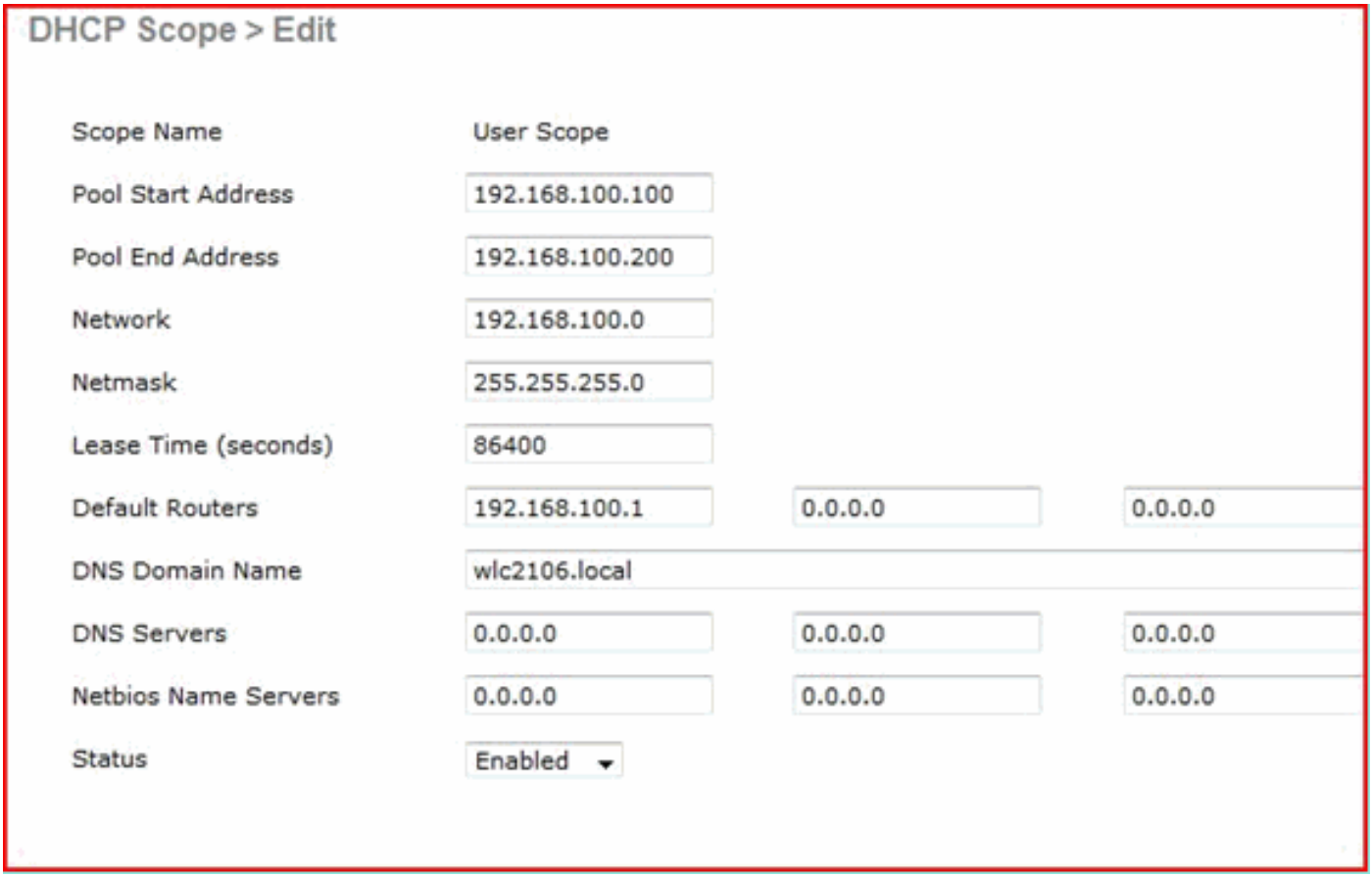


Or via the CLI:

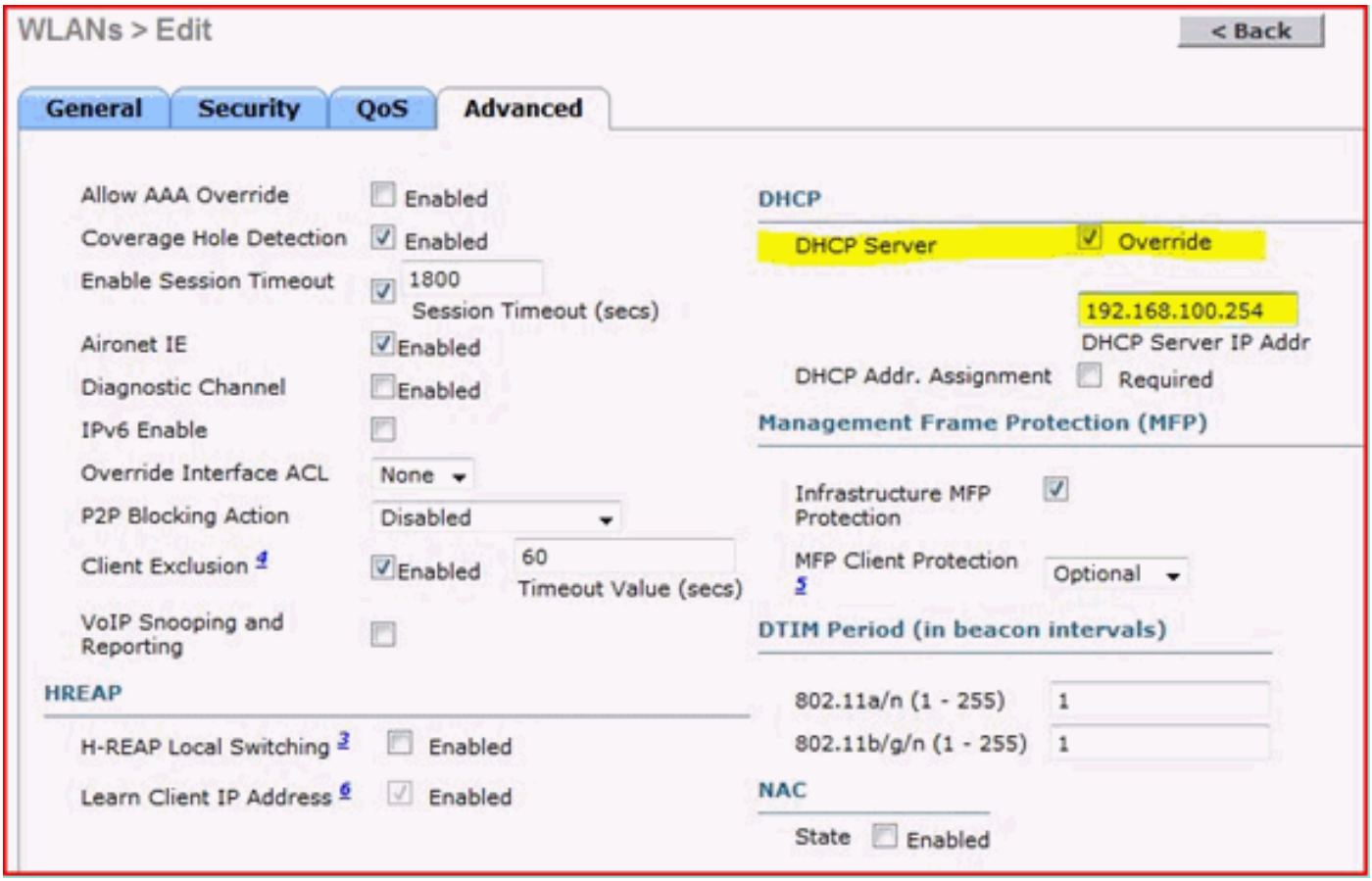
```
Config dhcp proxy enable
Save config
```

In order to enable the internal DHCP server, complete these steps:

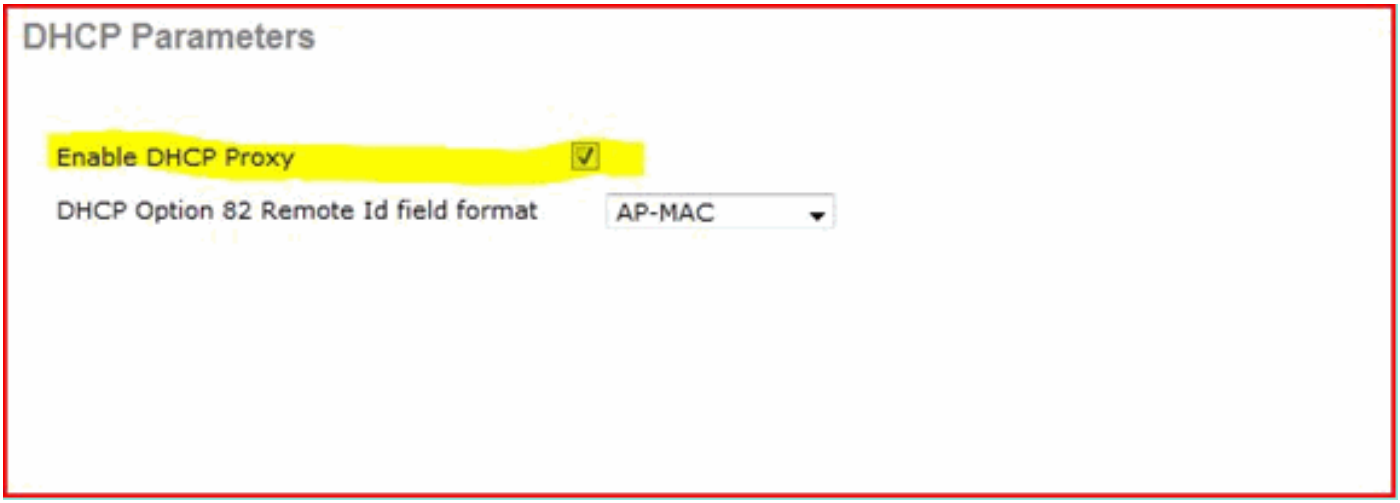
1. Define a scope that you use in order to pull IP addresses (Controller > Internal DHCP Server > DHCP Scope). Click New.



2. Point either your DHCP override to the management interface IP address of your controller.



3. Ensure that the DHCP proxy is enabled.



Troubleshoot

A debug of the internal DHCP server typically requires finding a client that has a problem obtaining an IP address. You must run these debugs.

```
debug client <MAC ADDRESS OF CLIENT>
```

The debug client is a macro that enables these debugs for you while it focuses the debug out only on the

client MAC address that you have entered.

```
debug dhcp packet enable
debug dot11 mobile enable
debug dot11 state enable
debug dot1x events enable
debug pem events enable
debug pem state enable
debug cckm client debug enable
```

The main one for DHCP issues is the debug dhcp packet enable command that is enabled automatically by the debug client command.

<#root>

00:1b:77:2b:cf:75 dhcpd: received DISCOVER

00:1b:77:2b:cf:75 dhcpd: Sending DHCP packet (giaddr:192.168.100.254)to 127.0.0.1:67 from 127.0.0.1:1067

00:1b:77:2b:cf:75 sendto (548 bytes) returned 548
00:1b:77:2b:cf:75 DHCP option len (including the magic cookie) 312

00:1b:77:2b:cf:75 DHCP option: message type = DHCP OFFER

00:1b:77:2b:cf:75 DHCP option: server id = 192.168.100.254
00:1b:77:2b:cf:75 DHCP option: lease time = 86400 seconds
00:1b:77:2b:cf:75 DHCP option: gateway = 192.168.100.1
00:1b:77:2b:cf:75 DHCP option: 15 (len 13) - skipping
00:1b:77:2b:cf:75 DHCP option: netmask = 255.255.255.0
00:1b:77:2b:cf:75 DHCP options end, len 312, actual 64
00:1b:77:2b:cf:75 DHCP option len (including the magic cookie) 81

00:1b:77:2b:cf:75 DHCP option: message type = DHCP REQUEST

00:1b:77:2b:cf:75 DHCP option: 61 (len 7) - skipping
00:1b:77:2b:cf:75 DHCP option: requested ip = 192.168.100.100
00:1b:77:2b:cf:75 DHCP option: server id = 192.0.2.10
00:1b:77:2b:cf:75 DHCP option: 12 (len 14) - skipping
00:1b:77:2b:cf:75 DHCP option: vendor class id = MSFT 5.0 (len 8)
00:1b:77:2b:cf:75 DHCP option: 55 (len 11) - skipping
00:1b:77:2b:cf:75 DHCP option: 43 (len 3) - skipping
00:1b:77:2b:cf:75 DHCP options end, len 81, actual 73
00:1b:77:2b:cf:75 DHCP Forwarding packet locally (340 octets) from 192.168.100.254 to 192.168.100.254
dhcpd: Received 340 byte dhcp packet from 0xfe64a8c0 192.168.100.254:68

00:1b:77:2b:cf:75 dhcpd: packet 192.168.100.254 -> 192.168.100.254 using scope "User Scope"

00:1b:77:2b:cf:75 dhcpd: received REQUEST

00:1b:77:2b:cf:75 Checking node 192.168.100.100 Allocated 1246985143, Expires 1247071543 (now: 1246985143)
00:1b:77:2b:cf:75 dhcpd: server_id = c0a864fe
00:1b:77:2b:cf:75 dhcpd: server_id = c0a864fe adding option 0x35 adding option 0x36 adding option 0x33 adding option 0x03 adding option 0x0f adding option 0x01

00:1b:77:2b:cf:75 dhcpd: Sending DHCP packet (giaddr:192.168.100.254)to 127.0.0.1:67

```
  from 127.0.0.1:1067

00:1b:77:2b:cf:75 sendto (548 bytes) returned 548
00:1b:77:2b:cf:75 DHCP option len (including the magic cookie) 312

00:1b:77:2b:cf:75 DHCP option: message type = DHCP ACK

00:1b:77:2b:cf:75 DHCP option: server id = 192.168.100.254
00:1b:77:2b:cf:75 DHCP option: lease time = 86400 seconds
00:1b:77:2b:cf:75 DHCP option: gateway = 192.168.100.1
00:1b:77:2b:cf:75 DHCP option: 15 (len 13) - skipping
00:1b:77:2b:cf:75 DHCP option: netmask = 255.255.255.0
00:1b:77:2b:cf:75 DHCP options end, len 312, actual 64
```

Clear the DHCP Leases on the WLC Internal DHCP Server

You can issue this command in order to clear the DHCP leases on the Internal DHCP server of the WLC:

<#root>

```
config dhcp clear-lease <all/IP Address>
```

Here is an example:

<#root>

```
config dhcp clear-lease all
```

Caveats

- DHCP proxy must be enabled for the Internal DHCP server to function
- Use of DHCP to port 1067 when you use the Internal DHCP server, which is affected by the CPU ACL
- The Internal DHCP server listens on the controller loopback interface via 127.0.0.1 UDP port 67

End User Interface

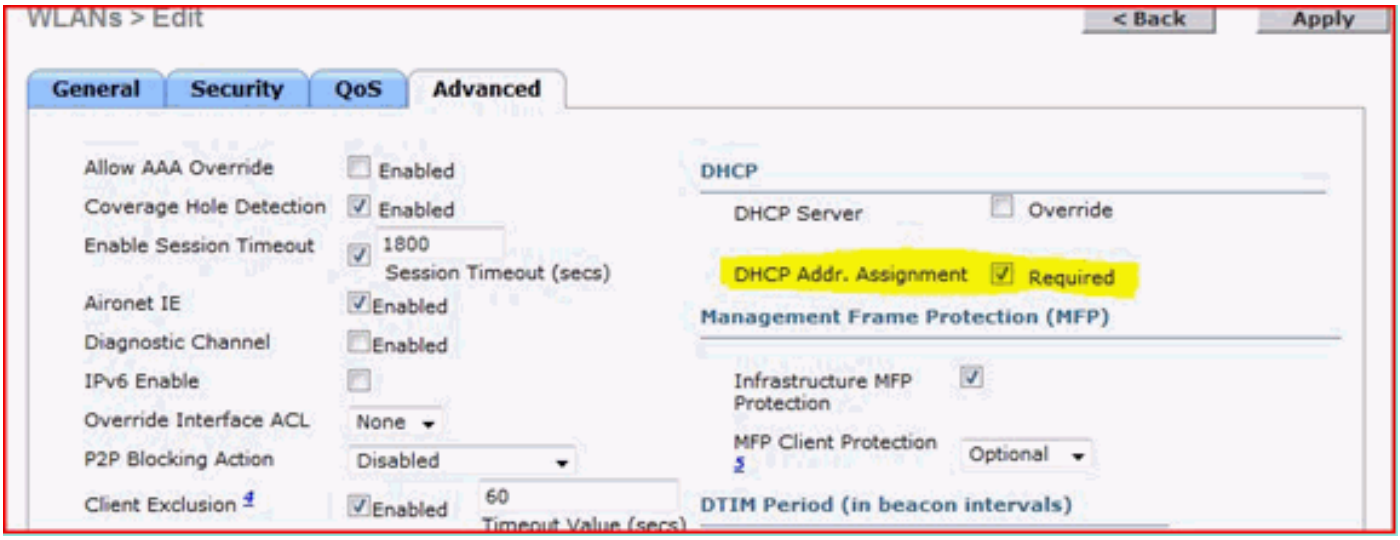
- The **config dhcp proxy disable** command implies the use of the DHCP bridging function. This is a global command (not a per-WLAN command).
- DHCP proxy remains enabled by default.
- When the DHCP proxy is disabled, the Internal DHCP server cannot be used by local WLANs. The bridging operation is not consistent with the operations required to redirect a packet to the internal server. Bridging really does mean bridging, with the exception of 802.11 to Ethernet II conversion. DHCP packets are passed unmodified from the LWAPP tunnel to the client VLAN (and vice-versa).
- When the proxy is enabled, a DHCP server must be configured on the WLAN’s interface (or in the

WLAN itself) in order for the WLAN to be enabled. No server needs to be configured when the proxy is disabled as these servers are not used.

- When a user attempts to enable the DHCP proxy, you internally verify that all WLANs (or associated interfaces) have a DHCP server configured. If not, the enable operation fails.

DHCP Required

The WLAN advanced configuration has an option that requires users to pass DHCP before they go into the RUN state (a state where the client can pass traffic through the controller). This option requires the client to do a full or half DHCP request. The main thing the controller looks for from the client is a DHCP request and an ACK that comes back from the DHCP server. As long as the client performs these steps, the client passes the DHCP required step and moves to the RUN state.



L2 and L3 Roaming

L2 Roam - If the client has a valid DHCP lease and performs an L2 roam between two different controllers on the same L2 network, the client must not need to reDHCP and the client entry must be completely moved to the new controller from the original controller. Then, if the client needs to DHCP again, the DHCP bridging or proxy process on the current controller transparently bridges the packet again.

L3 Roam - In an L3 roam scenario, the client moves between two different controllers in different L3 networks. In this situation, the client is anchored to the original controller and listed in the client table on the new foreign controller. During the anchor scenario, the client DHCP is handled by the anchor controller as the client data is tunneled within an EoIP tunnel between the foreign and anchor controllers.

Related Information

- DHCP OPTION 43 for Lightweight Cisco Aironet Access Points Configuration Example
- Technical Support & Documentation - Cisco Systems