

# Raw Formatting of DHCPv4/v6 Options in ESM

---

## In This Chapter

This section provides information about raw formatting of DHCPv4/v6 options in ESM.

Topics in this section include:

- [Applicability on page 2652](#)
- [Overview on page 2653](#)
- [Configuration on page 2662](#)
- [Conclusion on page 2685](#)

## Applicability

This example is applicable to DHCPv4/v6 subscriber-hosts using the Routed Central Office ESM model on 7750 SR-7/12/12e, 7750 SR-c4/12 and 7450 ESS 6/7/12 in Mixed-Mode with IOM3-XP or IMM.

A local DHCPv4/v6 server is used for address/prefix assignment, which implies a 7x50 DHCP *relay* scenario (as opposed to a 7x50 DHCP *proxy* scenario where the IP address/prefix is assigned via a RADIUS server or an LUDB).

The configuration was tested in a single homed environment using SR OS 12.0.R4.

## Overview

The 7x50 supports IP address assignment to its DHCP clients via two IP address assignment authorities:

- DHCP server — In this model the 7x50 behaves as a DHCP relay between the DHCP client and the 7x50 DHCP server.
- RADIUS/LUDB — In this model the IP address/prefix is assigned via a RADIUS server or an LUDB and the 7x50 internal or external behaves as a proxy<sup>1</sup> between the DHCP client and the non-DHCP aware RADIUS/LUDB.

Within these two fundamental address assignment models, there are several mechanisms available on the 7x50 by which DHCP parameters (DHCP options and various parameters within the options) can be passed to the DHCP client during the address assignment phase.

For example, in the RADIUS/LUDB address assignment model, the DHCP parameters can be supplied via RADIUS, LUDB and Python, while in the DHCP server model, the DHCP parameters can also be also supplied via the DHCP server itself (in addition to RADIUS, LUDB and Python).

Some of the more commonly used DHCP parameters have their own RADIUS and CLI constructs. For example, a default router has its own RADIUS attribute(s):

```
Alc-Default-Router (26-6527-18)
```

or even its own CLI keyword:

```
config>router>dhcp>server>pool>subnet>options# default-router
config>service>router>dhcp>server>pool>subnet>options# default-router
config>subscr-mgmt>ludb>ipoe>host>options# default-router
```

Other less common DHCP options can be defined and inserted by the 7x50 DHCP relay agent using the pre-formatted (IP address, domain, or string) or the non-formatted (hex) custom-option CLI command:

```
config>router>dhcp>server>pool>options# custom-option
config>router>dhcp>server>pool>subnet>options# custom-option
config>router>dhcp6>server>pool>options# custom-option
config>router>dhcp6>server>pool>prefix>options# custom-option
config>service>vprn>dhcp>server>pool>options# custom-option
config>service>vprn>dhcp>server>pool>subnet>options# custom-option
config>service>vprn>dhcp6>server>pool>options# custom-option
config>service>vprn>dhcp6>server>pool>prefix>options# custom-option
```

---

1. Note that the term *proxy* can also refer to the functionality where the DHCP server is used for address assignment. In this case, the 7x50 would hide the DHCP server from the client and pretend to be the DHCP server to the client, passing the DHCP parameters between the client and the server (lease times, etc).

```
config>subscriber-mgmt>ludb>ipoe>host>options# custom-option
```

The most flexible way of configuring DHCP parameters is by means of ‘raw’ (or hexadecimal) formatting. Any DHCP option can be hexadecimally (raw) formatted via the following RADIUS attributes:

```
Alc-ToClient-Dhcp-Options
Alc-ToClient-Dhcp6-Options
```

and/or via the custom-options CLI commands as outlined above. These options are then passed on to the DHCP client via the DHCP relay agent in the 7x50.

In addition to raw formatting via RADIUS or CLI, Python scripting can be used to intercept DHCP messages and modify their content.

The focus of this example is to demonstrate how the **raw** DHCP options are formatted via RADIUS. The messages can be optionally pre/post-processed by a Python script in the 7x50 before they are passed on to the DHCP client.

In this example, the following DHCP parameters are passed to the DHCP client using the **Alc-ToClient-Dhcp-Options** and the **Alc-ToClient-Dhcp6-Options** RADIUS attributes:

**Table 46: RADIUS Inserted Raw Options**

RADIUS	
DHCPv4 ToClient-Dhcp-Options	DHCPv6 ToClient-Dhcp6-Options
(default-)router [3] = 10.10.10.254	DNS server [23] = 2001:db8:1:1:1:1:1:1 2001:db8:1:1:1:1:1:2
DNS server [6] = 172.22.250.250 172.22.250/251	Domain search list [24] = ‘alcatel-lucent.com’ ‘test.com’
Domain name [15] = ‘alcatel.com’	Vendor specific-option [17] = ‘custom-test-option’
Custom option [230] = ‘custom test option’	
Renew time [58] = 5 min (300sec)	
Rebind time [59] = 6min 40sec (400sec)	

The DHCP parameters in the following DHCP messages are altered by a Python script:

**Table 47: Python Modified DHCP Fields**

Python	
DHCPv4 (DHCP Request)	DHCPv6 (LDRA DHCP Request)
Lease-time [51] = 8min 20sec (500sec)	IA-NA Preferred-Lifetime = 66min 40sec (4000sec)  IA-NA Valid-Lifetime = 66min 40sec (4000sec)  IA-NA Renew-Time (T1) = 33min 20sec (2000sec)  IA-NA Rebind-Time (T2) = 50min (3000sec)  IA-PD Preferred-Lifetime = 66min 40sec (4000sec)  IA-PD Valid-Lifetime = 66min 40sec (4000sec)  IA-PD Renew-Time (T1) = 33min 20sec (2000sec)  IA-PD Rebind-Time (T2) = 50min (3000sec)

The following DHCP parameters are configured via CLI in the 7x50 DHCPv4/v6 server:

**Table 48: CLI Inserted DHCP Options**

CLI DHCP Server Pool/prefix Options	
DHCPv4	DHCPv6
DNS server [6] = 172.22.250.253	DNS server [23] = 2001:db8:1:1:1:1:1:3
Custom option [231] = ‘dhcp injected custom option 231’	Custom option [232]= ‘v6 custom option 232’
	IA-NA Preferred-Lifetime = 20min (1200sec)
	IA-NA Valid-Lifetime = 20min (1200sec)
	IA-NA Renew-Time (T1) = 10 min (600 sec)
	IA-NA Rebind-Time (T2) = 15min (900 sec)
	IA-PD Preferred-Lifetime = 20min (1200sec)
	IA-PD Valid-Lifetime = 20min (1200sec)
	IA-PD Renew-Time (T1) = 10min (600 sec)

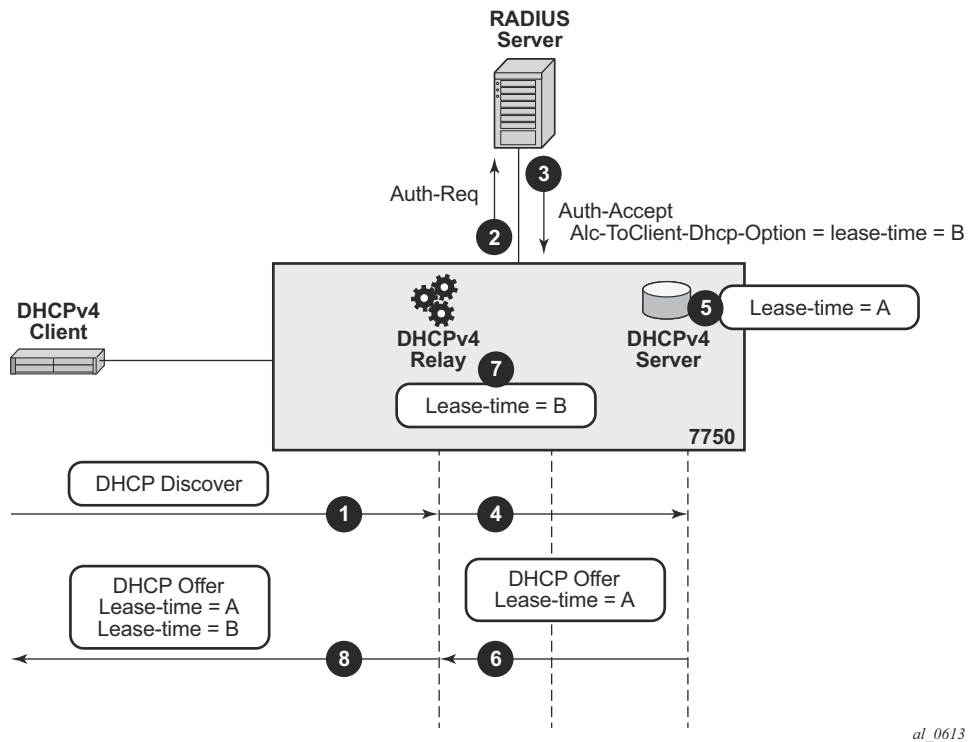
The RADIUS injected raw options are prepended by the DHCP relay agent in the 7x50 to any existing DHCP options already present in the DHCP message before being sent to the client. The existing options could be generated by the DHCP server (internal or external) or by the LUDB. No check is performed on the outgoing DHCP message towards the client in order to verify whether any of the RADIUS inserted options are already present in the DHCP message. This could potentially lead to duplication of DHCP options in the outgoing DHCP messages in case that the same option is inserted via the DHCP server and via RADIUS. To prove the point, this example supplies the same DHCP option (with different values) via multiple sources (RADIUS and CLI).

Configuration of DHCP lease related times requires closer examination. In DHCPv4, the DHCP lease-time option (51) is always supplied by the DHCPv4 server (this cannot be disabled). In case the lease-time is also supplied via RADIUS in an Alc-ToClient-Dhcp-Options VSA, the client would receive two lease-times for the same IP address. This can lead to unpredictable behavior not only on the client side but also on the 7x50 DHCPv4 server side since the DHCPv4 server (and the 7x50 DHCPv4 relay agent) creates the lease state only for the lease-time supplied by the DHCP server, and ignores the one supplied via RADIUS or LUDB. This scenario is shown in [Figure 402](#):

1. DHCP Discover arrives.
2. Radius authentication is triggered.

3. RADIUS returns lease-time value 'B' (Alc-ToClient-DHCP-Option) in Authentication-Accept message.
4. DHCP Discover is forwarded by the DHCP relay agent to the DHCP server.
5. DHCP server offers an IP lease with the configured lease-time of 'A'.
6. The DHCP offer is sent to the DHCP relay agent.
7. The DHCP relay agent appends the lease-time 'B' supplied by RADIUS to the DHCP message.
8. The DHCP relay forwards the message to the DHCP client with both lease-times 'A' and 'B'.

Note that the example in [Figure 402](#) does not represent a typical deployment case. This example is solely chosen to clarify the behavior in 7x50.



al\_0613

**Figure 402: DHCPv4 Lease-Time Inserted by RADIUS and DHCPv4 Server**

To ensure DHCPv4 lease time unambiguity, the lease-time should be supplied by a single source, in this case by the DHCPv4 server.

Since this eliminates RADIUS as a source of the DHCPv4 lease-time, an alternate method operating on the *raw level* is used to influence the automatic selection of the lease-time in the DHCPv4 server. This alternate method relies on the fact that the DHCPv4 server accepts hints received from the client as to what the desired lease-time should be. In other words, if the client

sends the option 51 (lease-time) with a specific value, the 7x50 DHCPv4 server will honor this hint<sup>2</sup>. To demonstrate this behavior, a Python script is invoked upon receipt of a DHCPv4 Request message during the IP address assignment process (DORA – Discover-Offer-Request-Ack). The Python script inserts a new option 51 with the desired value for the lease-time. The DHCPv4 server honors this hint from the client and it returns the requested lease-time back to the client. This scenario is shown in [Figure 403](#):

1. DHCP Discover arrives.
2. DHCP Discover is intercepted by the Python processing engine and the lease-time 'B' is inserted in DHCP Discover message. This is then used as a hint to the DHCP server.
3. DHCP Discover message is sent to the DHCP relay agent.
4. RADIUS authentication is triggered.
5. User is authenticated. This time lease-time is not returned via RADIUS.
6. DHCP Discover is forwarded to the DHCP server.
7. The DHCP server honors the hint from the DHCP Discover and offers lease-time 'B', even though the server is configured with lease-time 'A'.
8. The DHCP server replies with a DHCP Offer message.
9. DHCP Offer is forwarded by the DHCP relay agent to the client.

- 
2. As long as this value is within the configured range of values specified in the DHCP server.



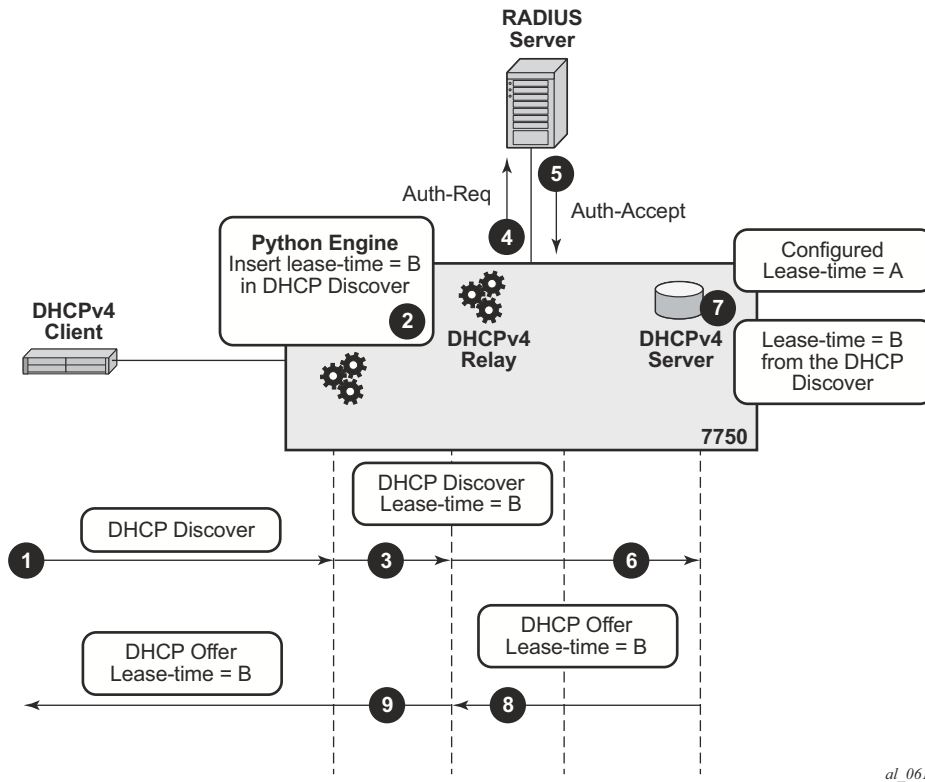


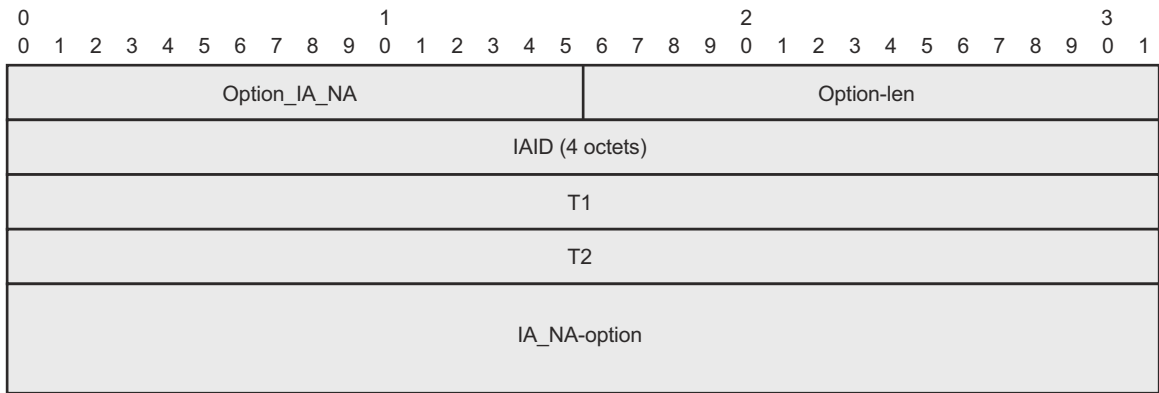
Figure 403: Python Injected Hint for Lease-Time

By default the local 7x50 DHCP server does **not** inject Renew (T1) and Rebind (T2) times so these two timers can still be supplied via RADIUS without duplication by the local 7x50 DHCP server.

When it comes to lease-time related parameters, the behavior of the 7x50 DHCPv6 server is different from the behavior of the 7x50 DHCPv4 server.

DHCPv6 lease related timers are **not** DHCP options. Instead, they are parameters within the IPv6 addressing option. An IPv6 address or prefix is assigned to the client via the IA-NA or IA-PD option, which contains additional parameters (which are not considered options) such as the IP address/prefix and the lease related timers. [Figure 404](#) shows the IA-NA option that carries the T1/T2 parameters.

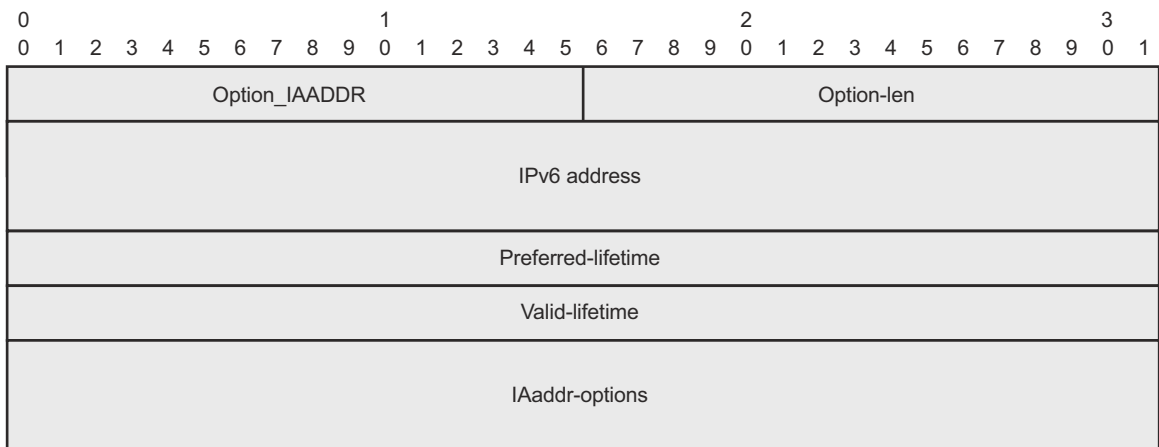
Overview



al\_0615

**Figure 404: Format of the IA-NA Option**

The format of the IA address option is shown in [Figure 405](#). This option carries preferred and valid lifetimes.



al\_0616

**Figure 405: Format of the IA Address Option**

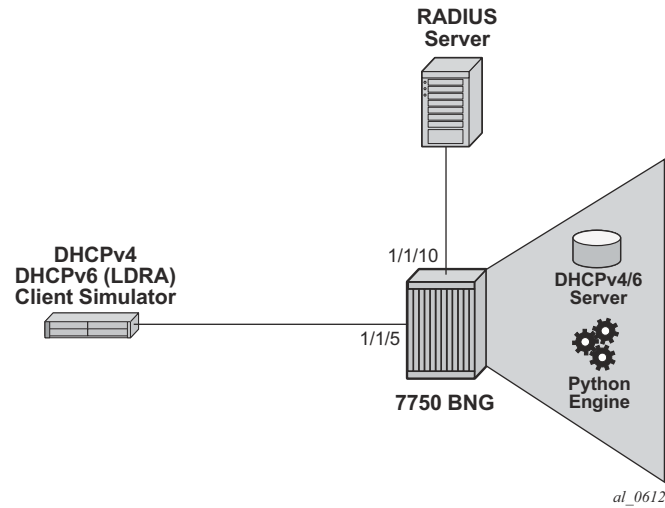
In this example, the IPv6 address/prefix is provided by the local 7x50 DHCPv6 server and as such, RADIUS cannot modify the parameters within the DHCPv6 options supplied by the DHCP server. Therefore, the desired IPv6 lease timers (preferred-life time, valid-lifetime, renew-time[T1], rebind-time[T2]) are part of the IPv6 pool configuration in the 7x50 DHCPv6 server.

Alternatively Python can be used to intercept the outgoing DHCPv6 message and then change the timers within the IA-NA and IA-PD options. Although this would configure the lease timers for the client, the action of modifying the outgoing DHCPv6 message occurs after the DHCPv6 server processing. This would result in different lease times in the client and the DHCPv6 server, without any intermediary between them (such as a DHCPv6 Proxy) to deal with the differences.

For consistency purposes with the DHCPv4 example, a Python script processes the incoming DHCPv6 message (DHCPv6 Request) altering the lease timers (preferred/valid/renew/rebind) as a hint to the DHCPv6 server to request those values. However, the 7x50 DHCPv6 server does not honor those hints and uses its own values (default or configured) instead.

## Configuration

The topology is shown in [Figure 406](#).



**Figure 406: Topology**

---

## Access Ethernet Port with QinQ Encapsulation

```
configure port 1/1/5
  ethernet
    mode access
    encap-type qinq
  exit
  no shutdown
```

## Capture SAP

A capture SAP is used to dynamically detect VLAN ID(s) in incoming DHCP (trigger) packets. This example uses RADIUS authentication along with Python scripting for DHCP message processing and therefore the authentication and Python policies must be configured under the capture SAP.

```
configure service vpls 10
  sap 1/1/5:1.* capture-sap create
    description "circuit-id authentication"
    trigger-packet dhcp dhcp6
    dhcp-python-policy "acg"
    dhcp6-python-policy "acg"
    authentication-policy "rad"
```

---

## MSAP-Policy Configuration

The MSAP-policy defines the anti-spoofing mode which is set to next-hop MAC (nh-mac) in this example. It also defines the default subscriber management parameters in case that they are not supplied via LUDB or RADIUS.

MSAP-policy configuration is mandatory when a capture-SAP is deployed. In this example, the MSAP-policy name is supplied via RADIUS:

```
Alc-MSAP-Policy = "msaps"

configure subscriber-mgmt msap-policy "msaps"
  sub-sla-mgmt
    sub-ident-policy "sub_ident_pol"
    multi-sub-sap limit 500
  exit
  ies-vprn-only-sap-parameters
    anti-spoof nh-mac
  exit
```

## Subscriber-Interface and Group-Interface Configuration

In this example the subscriber-interface is a 'numbered interface' in which the interface IPv4 address and the interface IPv6 prefixes are explicitly configured. The IPv4 address is used as the default-gateway by the IPE attached clients. The IPv4 subnet to which this address belongs and the configured IPv6 prefixes are used for routing aggregation and are treated as local subnets/prefixes in the 7x50 routing table.

The managed (dynamic) SAPs are created under the group-interface which contains the reference to the authentication-policy name, the Python script, the v4/6 policy names and the DHCPv4/v6 relay related configuration settings (for example, a reference to DHCP servers). Both the authentication-policy name and the Python policy name referenced under the group-interface must match those configured under the capture-SAP.

```
configure service vprn 1
  subscriber-interface "int1-1" create
  address 10.10.10.254/24 # Numbered IPv4 subscriber interface.
  ipv6
    delegated-prefix-len 54
    subscriber-prefixes
      prefix 2001:db8:3::/48 pd # Numbered IPv6 subscriber interface.
      prefix 2001:db8:4::/48 wan-host # Numbered IPv6 subscriber interface.
    exit
  exit
  group-interface "gl-1" create
  ipv6
    router-advertisements
      no shutdown
    exit
    dhcp6
      python-policy "acg" # Python script for DHCPv6 messages.
      relay
        server 2001:db8::1001 # IPv6 address of the DHCPv6 server.
        client-applications dhcp
        no shutdown
      exit
    exit
  exit
  dhcp
    python-policy "acg" # Python script for DHCPv4 messages.
    option
      action keep # Keep option82 in the received DHCP packet.
      vendor-specific-option
        pool-name # Pool-name obtained via RADIUS (or LUDB) will be passed
                  # via DHCP relay to the local DHCP server. This name
                  # will be used for pool selection in DHCPv4 server.
    exit
  exit
  server 192.168.100.1 # IPv4 address of the DHCPv4 server.
  lease-populate 100 # Maximum number of DHCPv4 lease under each
                    # SAP of the group-interface.
  client-applications dhcp
  no shutdown
exit
```

```

        authentication-policy "rad" # RADIUS authentication policy.
    exit
exit

```

---

## Loopback (DHCP) Interface Configuration

The loopback interface is used for the DHCPv4/v6 server binding. It is configured with the IPv4/IPv6 addresses which are referenced from the DHCP relay configuration under the group-interface.

```

configure service vprn 1
    interface "loopback1-1" create
        address 192.168.100.1/32      # IPv4 address of the DHCPv4 server.
        ipv6
            address 2001:db8::1001/128 # IPv6 address of the DHCPv6 server.
            local-dhcp-server "v6"     # Binding of the DHCPv6 server
                                     # to this interface.
        exit
        local-dhcp-server "v4"       # Binding of the DHCPv4 server
                                     # to this interface.
    loopback
exit

```

---

## DHCPv4/6 Server Configuration

The local DHCP server configuration contains the pool selection method, pool information and DHCP options which are passed to the DHCP client at IP address/prefix assignment time.

```

configure service vprn 1
    dhcp
        local-dhcp-server "v4"
            use-pool-from-client # Pool-name received in the DHCP messages
                               # sent by the DHCP relay. The pool-name
                               # is used in pool selection.
        pool "non-shared-left"
            options
                dns-server 172.22.250.253 # DHCPv4 option passed on to the client.
                custom-option 231 string "dhcp server injected custom option 231"
            exit # DHCPv4 option passed on to the client.
            subnet 10.10.10.0/24 create
                address-range 10.10.10.5 10.10.10.100 # IPv4 address range available
                                                         # for address allocation.
            exit
        exit
    exit
    dhcp6
        local-dhcp-server "v6"
            use-pool-from-client

```

## DHCPv4/6 Server Configuration

```
pool "pd-left" create
  options
    dns-server 2001:db8:1:1:1:1:1:3
    custom-option 232 string "v6 custom option 232"
  exit
  prefix 2001:db8:4::/48 pd      # IPv6 prefix range available for delegated
                                # prefix allocation by this DHCPv6 server.
    preferred-lifetime min 20 # Preferred lifetime of the allocated
                                # delegated prefix.
    rebind-timer min 15       # Rebind (T2) time of the allocated
                                # delegated prefix.
    renew-timer min 10        # Renew (T1) time of the allocated
                                # delegated prefix.
    valid-lifetime min 20     # Valid lifetime of the allocated
                                # delegated prefix.
  exit
exit
pool "wan-left" create
  options
    dns-server 2001:db8:1:1:1:1:1:3
    custom-option 232 string "v6 custom option 232"
  exit
  prefix 2001:db8:3::/56 wan-host
    preferred-lifetime min 20 # Preferred lifetime of the
                                # allocated IPv6 address.
    rebind-timer min 15      # Rebind (T2) time of the
                                # allocated IPv6 address.
    renew-timer min 10       # Renew (T1) time of the
                                # allocated IPv6 address.
    valid-lifetime min 20    # Valid lifetime of the
                                # allocated IPv6 address.
  exit
exit
no shutdown
exit
exit
```



## RADIUS Authentication-Policy Configuration

The RADIUS authentication-policy is referenced under the capture-sap and under the group-interface configuration.

```
authentication-policy "rad" create
  password "ALU" hash2
  radius-authentication-server
  router "Base"
    server 1 address 192.168.114.1 secret "ALU" hash2
  exit
  user-name-format circuit-id
  include-radius-attribute
    circuit-id
    remote-id
    nas-port-id
    nas-identifier
  exit
exit
```

---

## Subscriber-Identification Policy

The subscriber-identification policy in this example defines a mapping method between the subscriber strings and the predefined subscriber profiles (*sub* and *sla*) locally configured on the 7x50. In our example the subscriber strings (*sub* and *sla*) are provided via RADIUS and are directly mapped to the preconfigured sub-profiles and sla-profiles with the matching names.

The subscriber-identification policy can be configured with default subscriber profiles in case the strings are not explicitly obtained via other means (RADIUS, LUSB, Python or statically provisioned). Subscriber-identification policy configuration is mandatory.

```
sub-ident-policy "sub_ident_pol" create
  sub-profile-map
    use-direct-map-as-default
  exit
  sla-profile-map
    use-direct-map-as-default
  exit
```

## Sla-Profile and Sub-Profile Configuration

The following is the configuration of the sub-profile and the sla-profile which are used to setup the subscriber-host. The sla and sub profiles are mandatory when creating subscriber-hosts in 7x50.

```
sla-profile "sla-profile-1" create
  ingress
    qos 2
    exit
  exit
  egress
    qos 2
    exit
  exit
exit

sub-profile "sub-profile-1" create
exit
```

---

## Python-Policy Configuration

The python-policy defined below is applied under the capture-sap and under the group-interface. It references the python-script command which defines the location of the script. A python-policy specifies the DHCP messages along with the direction to which the script processing applies.

The DHCPv4 script in this example is applied to incoming DHCPv4 Request messages. The python script inserts the lease-time option in the DHCPv4 Request message as a hint to the DHCPv4 server.

Similar logic is applied to the incoming Lightweight DHCPv6 Relay Agent

(LDRA) DHCPv6 messages where IA-NA and IA-PD related lease times are altered. Note that in the DHCPv6 case the local DHCPv6 server does not honor the hint and therefore the lease related times are explicitly configured in the DHCPv6 server.

```
python-script "acg" create
  action-on-fail passthrough #In case of python script failure, do not drop the message
                             but instead continue with message processing in 7750.
  primary-url "ftp://a.b.c.d/pub/configs/alu/SIMS/acg.py"
  no shutdown
exit
python-script "acg6" create
  action-on-fail passthrough
  primary-url "ftp://a.b.c.d/pub/configs/alu/SIMS/acg6.py"
  no shutdown
exit
python-policy "acg" create # Python policy that is applied under the capture-sap and
                           under the group-interface.
  dhcp request direction ingress script "acg"
```

```

dhcp6 relay-forward direction ingress script "acg6"
exit

```

---

## Python Script Configuration

In this example the Python script is located in an external location and downloaded to the 7x50 once the python-script CLI node is enabled (**no shutdown**).

The DHCPv4 Python script has exception code included (try/except statements). This makes script debugging easier in case one of the commands in the script fails.

For simplicity reasons, the exception code is removed from the DHCPv6 Python script. Note that in real deployments it is recommended for the exception code to be included in all Python scripts.

### DHCPv4 Python Script:

```

from alc import dhcpv4
try:
    myopt = dhcpv4.getOptionList()
    if myopt != []:
        print "option-list ", repr(myopt)
        print "\n"
except Exception:
    print "Can't retrieve DHCP options"
#lease 500s 8min 20sec
try:
    dhcpv4.set(51, ('\x00\x00\x01\xff',# Insert the lease-time (opt51) in the incoming
DHCPv4 request as a hint to the DHCPv4 server.
except Exception:
    print "Can't set time lease"

```

## RADIUS Access-Accept

### DHCPv6 Python Script:

```
from alc import dhcpv6
import struct
packet = dhcpv6.get_relaymsg()# Extract the original DHCPv6 packet within LDRA.

msgType = ord(packet.msg_type) # Get the message type.
ia_na = packet.get_iana()# Store the IA-NA option for further processing later on.
ia_pd = packet.get_iapd()# Store the IA-PD option for further processing later on.

if msgType == 3: # If the message in the LDRA packet is DHCPv6 Request, insert the lease
related times in address/prefic options.

    ia_na[0][1] = '\x00\x00\x07\xd0'# Set the renew time (T1) in IA-NA to 2000sec.
    ia_na[0][2] = '\x00\x00\x0b\xb8'# Set the rebind time (T2) in IA-NA to 3000sec.
    ia_na[0][3][5][0][1] = '\x00\x00\x0f\xa0' # Set the preferred time in IA-NA to
# 4000sec.
    ia_na[0][3][5][0][2] = '\x00\x00\x0f\xa0'# Set the valid time in IA-NA to 4000sec.
    packet.set_iana(ia_na) # Update the stored packet with the new values for IA-NA.

    ia_pd[0][1] = '\x00\x00\x07\xd0'# Set the renew time (T1) in IA-PD to 2000sec.
    ia_pd[0][2] = '\x00\x00\x0b\xb8'# Set the rebind time (T2) in IA-PD to 3000sec.
    ia_pd[0][3][26][0][0] = '\x00\x00\x0f\xa0' # Set the preferred time in IA-PD to
# 4000sec.
    ia_pd[0][3][26][0][1] = '\x00\x00\x0f\xa0'# Set the valid time in IA-PD to 4000sec.
    packet.set_iapd(ia_pd) # Update the stored packet with the new values for IA-PD.
    dhcpv6.set_relaymsg(packet) # Insert the packet in the LDRA message.
```

---

## RADIUS Access-Accept

Upon authentication, RADIUS returns the Access-Accept message with the following attributes:

```
Sending Access-Accept of id 66 to 192.168.114.2 port 64384
Alc-Subsc-Prof-Str = "sub-profile-1"
Alc-SLA-Prof-Str = "sla-profile-2"
Alc-MSAP-Interface = "g1-1"
Alc-MSAP-Policy = "msaps"
Alc-MSAP-Serv-Id = 1
Framed-Pool = "non-shared-left"
Framed-IPv6-Pool = "wan-left"
Alc-Delegated-IPv6-Pool = "pd-left"
Alc-ToClient-Dhcp-Options += 0x03040a0a0afe
Alc-ToClient-Dhcp-Options += 0x0608ac16fafaac16fafb
Alc-ToClient-Dhcp-Options += 0x0f0b616c636174656c2e636f6d
Alc-ToClient-Dhcp-Options += 0xe612637573746f6d2074657374206f7074696f6e
Alc-ToClient-Dhcp-Options += 0x3a040000012c
Alc-ToClient-Dhcp-Options += 0x3b0400000190
Alc-ToClient-Dhcp6-Options +=
```

## Raw Formatting of DHCPv4/v6 Options in ESM

```
0x0011001a0000197f00e60012637573746f6d2074657374206f7074696f6e
  Alc-ToClient-Dhcp6-Options +=
0x0017002020010db800010001000100010001000120010db800010001000100010002
  Alc-ToClient-Dhcp6-Options +=
0x0018001e0e616c636174656c2d6c7563656e7403636f6d00047465737403636f6d
```

It is possible to concatenate multiple DHCP options in a single RADIUS Alc-ToClient-DHCP6-Option but for clarity each option is in a separate attribute in this example.

The following table contains the explanation of the DHCP options inserted via RADIUS:

---

Alc-ToClient-Dhcp-Options += 0x03040a0a0afe  
(default) router (3) = 10.10.10.254

---

Alc-ToClient-Dhcp-Options += 0x0608ac16fafaac16fab  
dns (6) = 172.16.250.250 172.16.250.251

---

Alc-ToClient-Dhcp-Options += 0x0f0b616c636174656c2e636f6d  
domain-name (15) = alcatel.com

---

Alc-ToClient-Dhcp-Options += 0xe612637573746f6d2074657374206f7074696f6e  
custom -option (230) = "custom test option"

---

Alc-ToClient-Dhcp-Options += 0x3a040000012c  
renewal time T1 (58) = 300s (5min)

---

Alc-ToClient-Dhcp-Options += 0x3b0400000190  
rebind time T2 (59) = 400s (6min 40sec)

---

Alc-ToClient-Dhcp6-Options +=  
0x0011001a0000197f00e60012637573746f6d2074657374206f7074696f6e  
v6 vendor option (17) [opt-id(2) len(2) entp-id(4) vopt-code(2) vlen(2) vdata] = 17 26 6527 230  
18 "custom test option"

---

Alc-ToClient-Dhcp6-Options +=  
0x0017002020010db800010001000100010001000120010db800010001000100010002  
dns servers (23) [opt-id(2) len(2) servers-v6@] = 23 32  
2001:0db8:0001:0001:0001:0001:0001:0001 2001:0db8:0001:0001:0001:0001:0001:0002

---

Alc-ToClient-Dhcp6-Options +=  
0x0018001e0e616c636174656c2d6c7563656e7403636f6d00047465737403636f6d0  
domain list (24) = alcatel-lucent.com test.com [formatting as described in section 3.1 of RFC  
1035 (as referenced by RFC 4704 and RFC 3315)].

---

## Results and Verification

The results are verified via debug output and show commands on the 7x50, and also via pcap (Wireshark® packet capture) files on the DHCP client side.

Debug output on the 7x50 is enabled for DHCPv4/6 messages and for the Python script.

The DHCP debug output shows the options sent to the client in the DHCPv4/6 Ack/Reply messages.

The following commands enables debugging information to be sent to the current telnet/ssh session:

```
*A:BNGL# configure log
*A:BNGL>config>log# info
-----
log-id 50 # Capturing and displaying debug output is configured via log.
          from debug-trace # Capture debug output.
          to session # Output the debug to the current tcp/ssh session.
          exit
-----
```

The following commands enable DHCP related debugging:

```
*A:BNGL>config>log# show debug
debug
router "1"
  ip
    dhcp
      detail-level high
      mode egr-ingr-and-dropped
    exit
    dhcp6
      mode egr-ingr-and-dropped
      detail-level high
    exit
  exit
  local-dhcp-server "v4"
    detail-level high
    mode egr-ingr-and-dropped
  exit
  local-dhcp-server "v6"
    detail-level high
    mode egr-ingr-and-dropped
  exit
exit
```

## DHCPv4 Results

The following output displays the DHCPv4 Request message as it was received by the 7x50 DHCP server.

This message has been modified by the Python script on ingress and the lease-time option [51] has been inserted as a hint to the DHCPv4 server.

Option [82] is partially added by the **access-node** (relay-agent → circuit-id and remote-id) and partially by the internal 7x50 DHCP-relay (pool name).

```
32830 2014/07/24 03:02:46.44 WEST MINOR: DEBUG #2001 vprn1 DHCP server
"DHCP server: v4
Rx DHCP Request

ciaddr: 0.0.0.0          yiaddr: 0.0.0.0
siaddr: 0.0.0.0          giaddr: 10.10.10.254
chaddr: 00:00:65:01:03:01  xid: 0x159dd536

DHCP options:
[82] Relay agent information: len = 42
    [1] Circuit-id: ds-left
    [2] Remote-id: remote0
    [9] Vendor-Specific info: len = 22
        Enterprise [6527] : len = 17
        [13] dhcpPool: non-shared-left
[53] Message type: Request
[54] DHCP server addr: 192.168.100.1
[50] Requested IP addr: 10.10.10.34
[51] Lease time: 500
[255] End
```

The next output captures the DHCPv4 ACK message (within the 7x50) that is on its way to the client.

It can be observed that the DHCPv4 server inserted options are listed first:

- Opt[82] is echoed back by 7x50 DHCPv4 server
- Opt[53], [54], [51] and [1] are by default inserted by the local 7x50 DHCPv4 server and they cannot be disabled. The value for the lease-time [51] is set by the Python script.
- The next two options ([6] and [231]) are the options configured explicitly in the DHCPv4 server ([Table 48](#)).

The remaining options (with the exception of the *end* [255] option) are provided by RADIUS and they appear in the exact same order as they appear in the RADIUS Alc-ToClient-Dhcp-Options attributes ([Table 46](#)).

## DHCPv4 Results

There are two options [6] since they are inserted by both DHCP and RADIUS server.

Custom options [231] and [230] are decoded in [Table 46](#) and [Table 48](#).

```
32834 2014/07/24 03:02:46.44 WEST MINOR: DEBUG #2001 vprn1 PIP
"PIP: DHCP
instance 2 (1), interface index 11 (g1-1),
  transmitted DHCP Boot Reply to Interface g1-1 (1/1/5:1.3) Port 68

H/W Type: Ethernet(10Mb) H/W Address Length: 6
ciaddr: 0.0.0.0          yiaddr: 10.10.10.34
siaddr: 192.168.100.1   giaddr: 10.10.10.254
chaddr: 00:00:65:01:03:01  xid: 0x159dd536

DHCP options:
[82] Relay agent information: len = 18
    [1] Circuit-id: ds-left
    [2] Remote-id: remote0
[53] Message type: Ack
[54] DHCP server addr: 192.168.100.1
[51] Lease time: 500
    [1] Subnet mask: 255.255.255.0
[6] Domain name server: 172.22.250.253
[231] Unknown option: len = 38, value = 64 68 63 70 20 73 65 72 76 65 72
20 69 6e 6a 65 63 74 65 64 20 63 75 73 74 6f 6d 20 6f 70 74 69 6f 6e 20 32
33 31
[3] Router: 10.10.10.254
[6] Domain name server: length = 8
    172.22.250.250
    172.22.250.251
[15] Domain name: alcatel.com
[230] Unknown option: len = 18, value = 63 75 73 74 6f 6d 20 74 65 73 74
20 6f 70 74 69 6f 6e
[58] Renew timeout: 300
[59] Rebind timeout: 400
[255] End
```

The Wireshark® output shown on the next page is captured at the client side (N2X Ixia) and it effectively mirrors what is shown in the 7x50 debug output.



```

❑ Bootstrap Protocol
  Message type: Boot Reply (2)
  Hardware type: Ethernet (0x01)
  Hardware address length: 6
  Hops: 0
  Transaction ID: 0x159dd536
  Seconds elapsed: 0
  ⊕ Bootp flags: 0x8000 (Broadcast)
  Client IP address: 0.0.0.0 (0.0.0.0)
  Your (client) IP address: 10.10.10.34 (10.10.10.34)
  Next server IP address: 192.168.100.1 (192.168.100.1)
  Relay agent IP address: 10.10.10.254 (10.10.10.254)
  Client MAC address: NetworkG_01:03:01 (00:00:65:01:03:01)
  Client hardware address padding: 00000000000000000000
  Server host name not given
  Boot file name not given
  Magic cookie: DHCP
  ❑ Option: (53) DHCP Message Type
    Length: 1
    DHCP: ACK (5)
  ⊕ Option: (54) DHCP Server Identifier
  ❑ Option: (51) IP Address Lease Time
    Length: 4
    IP Address Lease Time: (500s) 8 minutes, 20 seconds
  ⊕ Option: (1) Subnet Mask
  ❑ Option: (6) Domain Name Server
    Length: 4
    Domain Name Server: 172.22.250.253 (172.22.250.253)
  ❑ Option: (231) Private
    Length: 38
    Value: 646863702073657276657220696e6a656374656420637573...
  ⊕ Option: (82) Agent Information Option
  ❑ Option: (3) Router
    Length: 4
    Router: 10.10.10.254 (10.10.10.254)
  ❑ Option: (6) Domain Name Server
    Length: 8
    Domain Name Server: 172.22.250.250 (172.22.250.250)
    Domain Name Server: 172.22.250.251 (172.22.250.251)
  ❑ Option: (15) Domain Name
    Length: 11
    Domain Name: alcatel.com
  ❑ Option: (230) Private
    Length: 18
    Value: 637573746f6d2074657374206f707469666e
  ❑ Option: (58) Renewal Time Value
    Length: 4
    Renewal Time Value: (300s) 5 minutes
  ❑ Option: (59) Rebinding Time value
    Length: 4
    Rebinding Time value: (400s) 6 minutes, 40 seconds
  ⊕ Option: (255) End

```

## DHCPv4 Results

The show command for the DHCP-relay lease state only displays the well known options inserted by the DHCPv4 server. The custom option inserted by the DHCPv4 server and any of the RADIUS supplied options are not kept as part of the DHCP-relay lease state.

```
*A:BNGL1# show service id 1 dhcp lease-state detail
=====
DHCP lease states for service 1
=====
Service ID           : 1
IP Address           : 10.10.10.34
Client HW Address    : 00:00:65:01:03:01
Subscriber-interface : int1-1
Group-interface      : g1-1
SAP                  : [1/1/5:1.3]
Up Time              : 0d 00:10:46
Remaining Lease Time : 0d 00:07:35
Remaining SessionTime : N/A
Persistence Key      : N/A

Sub-Ident            : "ds-left"
Sub-Profile-String   : "sub-profile-1"
SLA-Profile-String   : "sla-profile-2"
App-Profile-String   : ""
Lease ANCP-String    : ""
Lease Int Dest Id    : ""
Category-Map-Name    : ""

Lease Info origin    : DHCP

Ip-Netmask          : 255.255.255.0
Broadcast-Ip-Addr    : N/A
Default-Router       : N/A
Primary-Dns        : 172.22.250.253
Secondary-Dns        : N/A
Primary-Nbns         : N/A
Secondary-Nbns       : N/A

ServerLeaseStart     : 07/24/2014 03:02:46
ServerLastRenew      : 07/24/2014 03:12:46
ServerLeaseEnd       : 07/24/2014 03:21:06
Session-Timeout      : N/A
Lease-Time         : 0d 00:08:20
DHCP Server Addr   : 192.168.100.1

Relay Agent Information
  Circuit Id          : ds-left
  Remote Id           : remote0
  Radius User-Name    : "ds-left"
=====
Number of lease states : 1
=====
```

## DHCPv6 Results

The DHCPv6 server receives the DHCPv6 Request message with Python modified lease times (preferred, valid, renew and rebind) for IA-NA and IA-PD.

```
32877 2014/07/24 03:15:28.32 WEST MINOR: DEBUG #2001 vprnl DHCP server
"DHCP server: v6
Rx DHCPv6 RELAY_FORW
  Hop Count : 1
  Link Addr : 2001:db8:4::
  Peer Addr : fe80::200:65ff:fe01:301
  Option : RELAY_MSG (9), Length : 184
    Msg Type : RELAY_FORW (12)
    Hop Count : 0
    Link Addr : ::
    Peer Addr : fe80::200:65ff:fe01:301
    Option : INTERFACE_ID (18), Length : 7
      Interface Id : 64732d6c656674 (ds-left)
    Option : RELAY_MSG (9), Length : 135
      Msg Type : REQUEST (3)
      Trans Id : 0x060000
      Option : ELAPSED_TIME (8), Length : 2
        Time : 0 seconds
      Option : CLIENTID (1), Length : 10
        LL : HwTyp=0001,LL=000065010301
          00030001000065010301
      Option : SERVERID (2), Length : 10
        LL : HwTyp=0001,LL=d896ff000000
          00030001d896ff000000
      Option : ORO (6), Length : 4
        Requested Option : IA_NA (3)
        Requested Option : IA_PD (25)
      Option : IA_NA (3), Length : 40
        IAID : 0
        Time1: 2000 seconds
        Time2: 3000 seconds
      Option : IAADDR (5), Length : 24
        Address : 2001:db8:3:1::1
        Preferred Lifetime : 4000 seconds
        Valid Lifetime : 4000 seconds
      Option : IA_PD (25), Length : 41
        IAID : 0
        Time1: 2000 seconds
        Time2: 3000 seconds
      Option : IAPREFIX (26), Length : 25
        Prefix : 2001:db8:4:400::/54
        Preferred Lifetime : 4000 seconds
        Valid Lifetime : 4000 seconds
      Option : VENDOR_OPTS (17), Length : 37
        Enterprise : 0000197f
      Option : WAN_POOL (1), Length : 8
        wan-left
      Option : PFX_POOL (2), Length : 7
        pd-left
      Option : PFX_LEN (3), Length : 1
```

## DHCPv6 Results

The **hinted** DHCPv6 lease-times are not honored by the 7x50 DHCPv6 server and instead the 7x50 DHCPv6 server default values are inserted in the outgoing DHCPv6 Reply message towards the client as shown in the output below.

The explicitly configured DHCPv6 options are inserted by the DHCPv6 server first (Table 48) followed by the RADIUS supplied options inserted by the DHCPv6 relay (Table 46).

There are two DNS options [23] since they are supplied via two sources (DHCPv6 server and RADIUS Alc-ToClient-DHCP-Option VSA).

```
32885 2014/07/24 03:15:28.32 WEST MINOR: DEBUG #2001 vprn1 TIP
"TIP: DHCP6_PKT
  Outgoing DHCP6 Msg : RELAY_REPLY (13)
  to itf g1-1
  Hop Count : 0
  Link Addr : ::
  Peer Addr : fe80::200:65ff:fe01:301
  Option : RELAY_MSG (9), Length : 265
    Msg Type : REPLY (7)
    Trans Id : 0x060000
    Option : SERVERID (2), Length : 10
      LL : HwTyp=0001,LL=d896ff000000
        00030001d896ff000000
    Option : CLIENTID (1), Length : 10
      LL : HwTyp=0001,LL=000065010301
        00030001000065010301
    Option : IA_NA (3), Length : 40
      IAID : 0
      Time1: 600 seconds
      Time2: 900 seconds
    Option : IAADDR (5), Length : 24
      Address : 2001:db8:3:1::1
      Preferred Lifetime : 1200 seconds
      Valid Lifetime : 1200 seconds
    Option : IA_PD (25), Length : 41
      IAID : 0
      Time1: 600 seconds
      Time2: 900 seconds
    Option : IAPREFIX (26), Length : 25
      Prefix : 2001:db8:4:400::/54
      Preferred Lifetime : 1200 seconds
      Valid Lifetime : 1200 seconds
    Option : DNS_NAME_SRVR (23), Length : 16
      Server : 2001:db8:1:1:1:1:1:3
    Option : UNKNOWN (232), Length : 20
      763620637573746f6d206f7074696f6e20323332
    Option : VENDOR_OPTS (17), Length : 26
      Enterprise : 0000197f
    Option : UNKNOWN (230), Length : 18
      637573746f6d2074657374206f7074696f6e
    Option : DNS_NAME_SRVR (23), Length : 32
      Server : 2001:db8:1:1:1:1:1:1
      Server : 2001:db8:1:1:1:1:1:2
    Option : DOM_SRCH_LIST (24), Length : 30
      SearchList : .alcatel-lucent.com..test.com.
    Option : INTERFACE_ID (18), Length : 7
      Interface Id : 64732d6c656674 (ds-left)
```

The Wireshark® capture of the DHCPv6 Reply message on the client side mirrors the debug information captured by the 7x50:

```

DHCPv6
  Message type: Relay-reply (13)
  Hopcount: 0
  Link address: :: (::)
  Peer address: fe80::200:65ff:fe01:301 (fe80::200:65ff:fe01:301)
  Relay Message
    Option: Relay Message (9)
    Length: 265
    Value: 07060000002000a00030001d896ff000000001000a0003...
DHCPv6
  Message type: Reply (7)
  Transaction ID: 0x060000
  Server Identifier
    Option: Server Identifier (2)
    Length: 10
    Value: 00030001d896ff000000
    DUID: 00030001d896ff000000
    DUID Type: link-layer address (3)
    Hardware type: Ethernet (1)
    Link-layer address: d8:96:ff:00:00:00
  Client Identifier
    Option: Client Identifier (1)
    Length: 10
    Value: 00030001000065010301
    DUID: 00030001000065010301
    DUID Type: link-layer address (3)
    Hardware type: Ethernet (1)
    Link-layer address: 00:00:65:01:03:01
  Identity Association for Non-temporary Address
    Option: Identity Association for Non-temporary Address (3)
    Length: 40
    Value: 00000000000000258000003840005001820010db800030001...
    IAID: 00000000
    T1: 600
    T2: 900
  IA Address
    Option: IA Address (5)
    Length: 24
    Value: 20010db800030001000000000000001000004b0000004b0
    IPv6 address: 2001:db8:3:1::1 (2001:db8:3:1::1)
    Preferred lifetime: 1200
    Valid lifetime: 1200

```

- ▣ Identity Association for Prefix Delegation
  - Option: Identity Association for Prefix Delegation (25)
  - Length: 41
  - Value: 000000000000025800000384001a0019000004b0000004b0...
  - IAID: 00000000
  - T1: 600
  - T2: 900
- ▣ IA Prefix
  - Option: IA Prefix (26)
  - Length: 25
  - Value: 000004b0000004b03620010db8000404000000000000000...
  - Preferred lifetime: 1200
  - Valid lifetime: 1200
  - Prefix length: 54
  - Prefix address: 2001:db8:4:400:: (2001:db8:4:400::)
- ▣ DNS recursive name server
  - Option: DNS recursive name server (23)
  - Length: 16
  - Value: 20010db8000100010001000100010003
  - DNS server address: 2001:db8:1:1:1:1:1:3 (2001:db8:1:1:1:1:1:3)
- ▣ DHCP option 232
  - Option: Unknown (232)
  - Length: 20
  - Value: 763620637573746f6d206f7074696f6e20323332
- ▣ Vendor-specific Information
  - Option: Vendor-specific Information (17)
  - Length: 26
  - Value: 0000197f00e60012637573746f6d2074657374206f707469...
  - Enterprise ID: Panthera Networks, Inc. (6527)
- ⊕ option
  - ▣ DNS recursive name server
    - Option: DNS recursive name server (23)
    - Length: 32
    - Value: 20010db800010001000100010001000120010db800010001...
    - DNS server address: 2001:db8:1:1:1:1:1:1 (2001:db8:1:1:1:1:1:1)
    - DNS server address: 2001:db8:1:1:1:1:1:2 (2001:db8:1:1:1:1:1:2)
  - ▣ Domain Search List
    - Option: Domain Search List (24)
    - Length: 30
    - Value: 0e616c636174656c2d6c7563656e7403636f6d0004746573...
    - DNS Domain Search List
    - Domain: alcatel-lucent.com
    - Domain: test.com
- ▣ Interface-Id
  - Option: Interface-Id (18)
  - Length: 7
  - Value: 64732d6c656674
  - Interface-ID: ds-left

The following command captures the information kept in the 7x50 DHCPv6 relay lease state:

```
*A:BNGL1# show service id 1 dhcp6 lease-state detail
=====
DHCP lease states for service 1
=====
Service ID          : 1
IP Address          : 2001:db8:3:1::1/128
Client HW Address   : 00:00:65:01:03:01
Subscriber-interface : int1-1
Group-interface     : g1-1
SAP                 : [1/1/5:1.3]
Up Time            : 0d 00:02:41
Remaining Lease Time : 0d 00:17:18
Remaining SessionTime: N/A
Persistence Key     : N/A

Sub-Ident          : "ds-left"
Sub-Profile-String : "sub-profile-1"
SLA-Profile-String : "sla-profile-2"
App-Profile-String : ""
Lease ANCP-String  : ""
Lease Int Dest Id  : ""
Category-Map-Name  : ""
Dhcp6 ClientId (DUID): 00030001000065010301
Dhcp6 IAID         : 0
Dhcp6 IAID Type    : non-temporary
Dhcp6 Client Ip    : fe80::200:65ff:fe01:301
Primary-Dns        : N/A
Secondary-Dns      : N/A
Pool Name          : "wan-left"
Dhcp6 Server Addr  : 2001:db8::1001
Dhcp6 ServerId (DUID): 00030001d896ff000000
Dhcp6 InterfaceId  : ds-left
Dhcp6 RemoteId     : N/A

Lease Info origin  : DHCP

ServerLeaseStart   : 07/24/2014 03:15:28
ServerLastRenew    : 07/24/2014 03:15:28
ServerLeaseEnd     : 07/24/2014 03:35:27
Session-Timeout    : N/A
Radius User-Name   : "ds-left"
-----
Service ID          : 1
IP Address          : 2001:db8:4:400::/54
Client HW Address   : 00:00:65:01:03:01
Subscriber-interface : int1-1
Group-interface     : g1-1
SAP                 : [1/1/5:1.3]
Up Time            : 0d 00:02:41
Remaining Lease Time : 0d 00:17:18
Remaining SessionTime: N/A
Persistence Key     : N/A

Sub-Ident          : "ds-left"
Sub-Profile-String : "sub-profile-1"
SLA-Profile-String : "sla-profile-2"
```

## DHCPv6 Results

```
App-Profile-String : ""
Lease ANCP-String  : ""
Lease Int Dest Id  : ""
Category-Map-Name  : ""
Dhcp6 ClientId (DUID): 00030001000065010301
Dhcp6 IAID         : 0
Dhcp6 IAID Type    : prefix
Dhcp6 Client Ip    : fe80::200:65ff:fe01:301
Primary-Dns        : N/A
Secondary-Dns      : N/A
Pool Name          : "pd-left"
Dhcp6 Server Addr  : 2001:db8::1001
Dhcp6 ServerId (DUID): 00030001d896ff000000
Dhcp6 InterfaceId  : ds-left
Dhcp6 RemoteId     : N/A
```

```
Lease Info origin : DHCP
```

```
ServerLeaseStart   : 07/24/2014 03:15:28
ServerLastRenew    : 07/24/2014 03:15:28
ServerLeaseEnd     : 07/24/2014 03:35:27
Session-Timeout    : N/A
Radius User-Name    : "ds-left"
```

```
-----
Number of lease states : 2
=====
```



## Python Debug Output

---

### DHCPv4

For debugging purpose a line is added to the Python script printing all DHCP option numbers present in the incoming DHCP packets.

It can also be observed that all Python induced modifications to the original DHCP message are also displayed in the debugging output (inserting option [51] in this case).

Python script:

```
from alc import dhcpv4
myopt = dhcpv4.getOptionList()
print "option-list =", repr(myopt)
#lease 500s 8min 20sec
dhcpv4.set(51, ('\x00\x00\x01\xf4',))
```

Debug Output:

```
32826 2014/07/24 03:02:46.44 WEST MINOR: DEBUG #2001 Base Python Output
"Python Output: acg
option-list (53, 54, 50, 82, 255)
"
32827 2014/07/24 03:02:46.44 WEST MINOR: DEBUG #2001 Base Python Result
"Python Result: acg
DHCPv4 Option 51, SET
      '\x00\x00\x01\xf4'
"
```

---

### DHCPv6

Also the DHCPv6 Python script has some lines added to demonstrate Python debugging capabilities. The new lines print assigned values to the debugging output.

DHCPv6 script

```
from alc import dhcpv6
import struct
packet = dhcpv6.get_relaymsg()
msgTop = ord(dhcpv6.msg_type)
msgBot = ord(packet.msg_type)
ia_na = packet.get_iana()
ia_pd = packet.get_iapd()
print 'ia-na = ', ia_na
print '\n'
```

## Python Debug Output

```
print 'ia-pd = ', ia_pd
print '\n'
print 'msg type Top = ', msgTop
print 'msg type Bot = ', msgBot

msgType = struct.unpack('B',packet.msg_type)[0]
print "relay packet: ", msgType

# in relay request insert DHCPv6 lease times
if msgBot == 3:

    ia_na[0][1] = '\x00\x00\x07\xd0'
    ia_na[0][2] = '\x00\x00\x0b\xb8'
    ia_na[0][3][5][0][1] = '\x00\x00\x0f\xa0'
    ia_na[0][3][5][0][2] = '\x00\x00\x0f\xa0'
    packet.set_iana(ia_na)
    ia_pd[0][1] = '\x00\x00\x07\xd0'
    ia_pd[0][2] = '\x00\x00\x0b\xb8'
    ia_pd[0][3][26][0][0] = '\x00\x00\x0f\xa0'
    ia_pd[0][3][26][0][1] = '\x00\x00\x0f\xa0'
    packet.set_iapd(ia_pd)
    dhcpv6.set_relaymsg(packet)
```

## Python debugging output

```
32873 2014/07/24 03:15:28.32 WEST MINOR: DEBUG #2001 Base Python Output
"Python Output: acg6
```

```
ia-na = [['\x00\x00\x00\x00', '\x00\x00\x02X', '\x00\x00\x03\x84', {5: [['\x01
\r\xb8\x00\x03\x00\x01\x00\x00\x00\x00\x00\x00\x00\x01', '\x00\x00\x04\xb0', '\x
00\x00\x04\xb0', {}]]}]
```

```
ia-pd = [['\x00\x00\x00\x00', '\x00\x00\x02X', '\x00\x00\x03\x84', {26: [['\x00
\x00\x04\xb0', '\x00\x00\x04\xb0', '6', '\x01\r\xb8\x00\x04\x04\x00\x00\x00\x00
\x00\x00\x00\x00\x00', {}]]}]
```

```
msg type Top = 12
msg type Bot = 3
relay packet: 3
"
```

```
32874 2014/07/24 03:15:28.32 WEST MINOR: DEBUG #2001 Base Python Result
"Python Result: acg6
```

```
DHCPv6 Option 9, SET
'\x03\x06\x00\x00\x08\x00\x02\x00\x00\x01\x00\n\x00\x03\x00\x01\x00\x00
e\x01\x03\x01\x00\x02\x00\n\x00\x03\x00\x01\xd8\x96\xff\x00\x00\x00\x06\x00\
x04\x00\x03\x00\x19\x00\x03\x00(\x00\x00\x00\x00\x00\x00\x07\xd0\x00\x00\x0b\xb8
\x00\x05\x00\x18 \x01\r\xb8\x00\x03\x00\x01\x00\x00\x00\x00\x00\x01\x00\
x00\x0f\xa0\x00\x00\x0f\xa0\x00\x19\x00)\x00\x00\x00\x00\x00\x07\xd0\x00\x00
\x0b\xb8\x00\x1a\x00\x19\x00\x00\x0f\xa0\x00\x00\x0f\xa06 \x01\r\xb8\x00\x04\x04
\x00\x00\x00\x00\x00\x00\x00\x00'
"
```

## Conclusion

The most common DHCP options that need to be passed by the 7x50 to the clients can be directly configured in CLI with a DHCP option specific command (such as DNS or a router option in IPv4). The DHCP option specific commands hide the complexity of the option encoding from the operator.

Less common options can be configured via a custom-option command in CLI. This scenario requires the operator to be familiar with the encoding of the option.

Similarly, RADIUS provides the means to pass the DHCP options destined to the client in the form of option specific RADIUS attributes (lease-time, etc). For less common options, two Alcatel-Lucent specific RADIUS attributes are provided: **Alc-ToClient-Dhcp-Options** and **Alc-ToClient-Dhcp6-Options**. These two attributes allow the operator to encode client destined DHCP options using hexadecimal notation. Although this process requires manual encoding it provides a very flexible way of providing options to the client.

The custom options supplied via LUDB or RADIUS are appended by the 7x50 DHCP-relay agent to any existing options that may have been already inserted by the DHCP server in the DHCP packet.

Python processing can additionally assist in DHCP message processing where the options or the parameters within the existing options can be added, removed or modified.

Conclusion