Blackford Computing

Lab Test Plan

IPV6 Basic Test Plan - BCTP101.1

Feb 21 2012

Revision History

Revision Number	Revision Date	Summary of Changes	Author
101.0	2/18/2012	Initial	Bill Blackford
101.1	2/21/2012	Fixed site-local mask Cleaned up headings	Bill Blackford

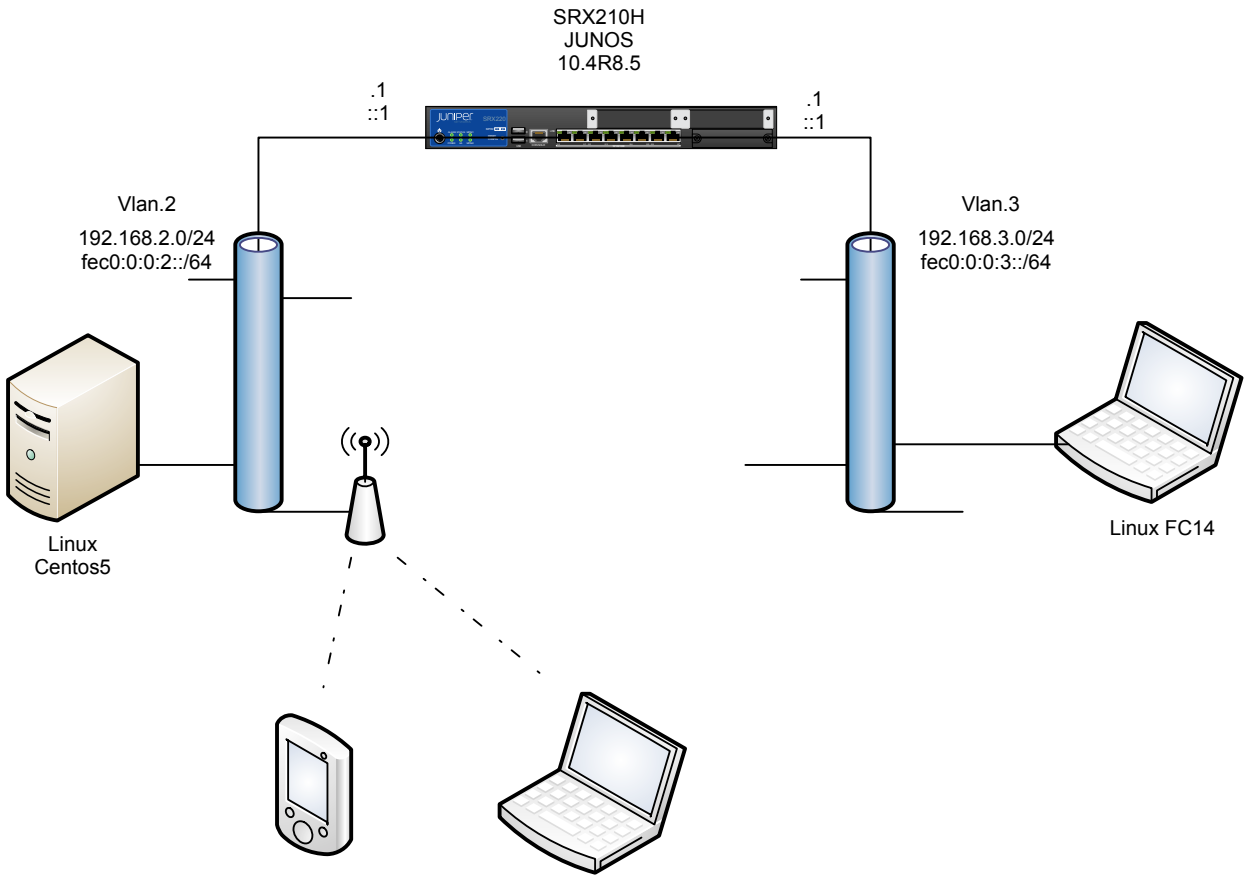
Table of Contents

1.0 Executive Summary

The goal here is to document a test plan of my home IPV6 implementation and to walk the reader through the process of set up and verification. I mostly focus on observed results and have left out some basics. This is not a tutorial, (although, I seem to be writing it like one) however, an assumption is made that the reader has some knowledge of networking, systems and in particular IPV4 and IPV6. I do offer a couple of helpful links for anyone who has too much time on their hands and would venture to explore these topics in more detail.

2.0 Test environment

2.1 Test Bed



2.2 Test systems

Router: Juniper SRX210
Hosts: Centos 5.2 server, various Linux, Mac and Windows clients. ipods/iphones, etc., running WIFI. (BTW, it appears that IPV6 is supported on my wife's iphone evidenced by seeing it in the neighbor cache later on, but it's not on HTC Android).
AP: Cisco 1100. Worth noting here that the AP is not IPV6 enabled. I was surprised that the standard framing type used on IPV6 is being passed via the BVI bridge. IOW, wireless clients are getting SLAAC addresses.

2.2.1 Primary Devices Under Test

```
bblackford@fw01.bchome> show version
Hostname: fw01.bchome
Model: srx210h
JUNOS Software Release [10.4R8.5]


shambhala[0]:~# cat /etc/redhat-release
CentOS release 5.2 (Final)
shambhala[0]:~# uname -r
2.6.18-92.el5

shambhala[0]:~# ethtool -i eth0
driver: tg3
version: 3.86
firmware-version: 5751-v3.44a
bus-info: 0000:02:00.0



[root@dell-6400 ~]# cat /etc/redhat-release
Fedora release 14 (Laughlin)

[root@dell-6400 ~]# uname -r
2.6.35.6-45.fc14.i686

[root@dell-6400 ~]# ethtool -i eth0
driver: b44
version: 2.0
firmware-version:
bus-info: 0000:03:00.0
```

3.0 Concepts Used

I thought I'd add this section simply as a background to some of the technologies and terminology used in the document. I'm including links for further review as well.

3.1 EUI-64

EUI-64 uses the MAC address of the local interface and inserts two bytes, FF: FE, in between the third and fourth byte forming a 64 bit address. It also modifies the first nibble to a "0", if the address is global or a "2" if the address is local. The following link contains a much better explanation than I can offer in the amount of time I feel like spending on this section.

http://packetlife.net/blog/2008/aug/4/eui-64-ipv6/

3.2 IPV6 Addressing Refresher

3.2.1 IPV6 Addressing

Each hex nibble equals four bits. So a 128 bit IPV6 address, contains 32 hex nibbles.
For example,
2001:1960:1111:2222:FEED:G00D:CAFE:BEEF

Each group, separated by colons, (four nibbles) represents 16 bits.
For example,
2001:1960, shows two groups of four nibbles each for a total of 32 bits.

2001:1960::/32 is an example of a typical assignment.
2001:1960:1111::/48
2001:1960:1111:2222::/64
etc.

Not all groups have to have each nibble populated. :1111:, :111, :11:, :1: are valid groups.
A complete tutorial on best practices for IPV6 addressing goes beyond the scope of this document. See: http://www.ipbcop.org/drafts/bcop-ipv6-subnetting/

3.2.2 Link-Local Addresses

fe80::/10 This is the address that gets auto-configured on each IPV6 host (similar to the 169.250/16 seen on Windows/Samba clients with IPV4). It is seen as a /64 mask that uses the EUI-64 format (pads FF:FE to the middle of the MAC address to form a 64 bit address). Unique to the link only. Cannot be routed beyond subnet boundary.

For example,

fe80::ca2a:14ff:fe2d:19a2/64

3.2.3 Site-local Unicast Addresses

Site-local. **fec0::/10** Analogous to RFC1918 address space in IPV4. Can be routed beyond subnet boundary, but is not routed globally. I will be using site-local addressing in my test plan.

3.2.4 Global Unicast Addresses

Currently IANA is assigning 001 in the first 3 bits, so 2 in hex.
2001::/x.
2001:1960::/32 is assigned to Frontier Communications for example.
2001:558::/31 is assigned to Comcast.

Here's a link that can explain the subject of IPV6 addressing much better than I:
http://ipv6.com/articles/general/IPv6-Addressing.htm

3.3 Addresses used in test environment

For the purpose of this test plan, I'm focusing on the link-local and site-local addresses.
Site-local used in this test plan:

fec0:0:0/48 broken into /64 subnets.

Vlan.2 fec0:0:0:2::/64
Vlan.3 fec0:0:0:3::/64
Vlan.4 fec0:0:0:4::/64 - family inet6 only.

3.4 Address Families

I will be using IPV4 and address family 'inet' somewhat interchangeably as well as IPV6 and address family 'inet6' throughout this document. These address families translate over to Linux and Mac as well so hopefully this won't introduce too much ambiguity.

IPV4 = inet
IPV6 = inet6

4.0 SLAAC

Here, we focus on the SLAAC (StateLess Address Auto Configuration).
Stated in the previous section, I will be using site-local addressing. Prefix fec0:0:0::/48 broken down into three subnets. It's important to note at this point that SLAAC addresses don't have to be site-local addresses. In most cases, they will be global unicast addresses. I'm just using site-local addresses for my test plan.
Since the environment is configured with two address family types, I've included the family inet addresses for each respective vlan for reference. This also helps to illustrate my wish to assign family inet6 addresses on a nibble boundary for simplicity. Vlan.4 however is address family inet6 only.

Vlan.2
    Inet 192.168.2.0/24
    Inet6 fec0:0:0:2::/64

Vlan.3
    Inet 192.168.3.0/24
    Inet6 fec0:0:0:3::/64

Vlan.4
    Inet6 fec0:0:0:4::/64

4.1 Router settings

The vlan.2 interface on the SRX has been statically assigned the site-local address of fec0:0:0:2::1/64. The 'show interfaces terse' output shows the site-local as well as the link-local address. (we're not paying any attention to the inet (IPV4) addressing but know that the systems are dual-stacked).

```
bblackford@fw01.bchome> show configuration interfaces vlan.2
description "_VLAN_, HOME LAN";
family inet {
    filter {
        input INSIDE;
    }
    address 192.168.2.1/24;
}
family inet6 {
    address fec0:0:0:2::1/64;
}
```

```
bblackford@fw01.bchome> show interfaces vlan.2 terse
Interface               Admin Link Proto    Local                 Remote
vlan.2                  up    up   inet     192.168.2.1/24
                                    inet6    fe80::5e5e:abff:fee0:7d88/64
                                             fec0:0:0:2::1/64
```

The ‘show configuration protocols’ shows the settings for SLAAC. This stanza creates the router advertisement used in auto configuration. Notice the prefix being used is in the same range as the configured address on the interfaces vlan.2.

```
bblackford@fw01.bchome> show configuration protocols
router-advertisement {
    interface vlan.2 {
        current-hop-limit 10;
        prefix fec0:0:0:2::/64;
    }
}
```

The SRX, by default, runs IPV6 in packet mode. I need to modify the security settings to allow for IPV6 to operate in flow mode.

```
bblackford@fw01.bchome> show configuration security forwarding-options
family {
    inet6 {
        mode flow-based;
    }
}
```

4.2 IPV6 verification

4.2.1 Router Advertisement

Showing that router advertisement (RA) is functioning. This shows the status of the RA and RS messages for vlan.2.

```
bblackford@fw01.bchome> show ipv6 router-advertisement
Interface: vlan.2
  Advertisements sent: 14, last sent 00:06:29 ago
  Solicits received: 1, last received 00:12:09 ago
  Advertisements received: 0
```

Note to self: I may enable traceoptions to gather more verbose output in the log. RA conversations, etc. Now comparing this output to the same output from a M/MX series, the SRX seems to be missing a lot here, so it may just be an SRX limitation.

4.2.2 Neighbor cache

The equivalent of 'show arp' with IPV4. Here we see the site-local address of the Centos server. Note that the interface is vlan.2 and the "2" fec0::2:218:8bff:fe10:e23a, in the site-local address indicating the /64 nibble set on vlan.2. We'll see a "3" and a "4" in later sections while testing vlan.3 and vlan.4.

```
bblackford@fw01.bchome> show ipv6 neighbors
IPv6 Address                  Linklayer Address  State       Exp Rtr Secure Interface
fe80::218:8bff:fe19:e23a      00:18:8b:19:e2:3a  stale       94  no  no     vlan.2
fe80::1439:3db1:ec2c:54ad     00:0e:a6:f4:5b:5b  stale       479 no  no     vlan.2
fec0::2:218:8bff:fe19:e23a    00:18:8b:19:e2:3a  stale       1134 no no     vlan.2
```

Note: the link-local address is seen in the neighbor cache as well.

```
bblackford@fw01.bchome> show ipv6 neighbors
IPv6 Address                  Linklayer Address  State       Exp Rtr Secure Interface
fe80::218:8bff:fe19:e23a      00:18:8b:19:e2:3a  stale       106 no  no     vlan.2
fe80::5a1f:aaff:fe51:4916     58:1f:aa:51:49:16  stale       191 no  no     vlan.2
fec0::2:218:8bff:fe19:e23a    00:18:8b:19:e2:3a  stale       361 no  no     vlan.2
```

4.2.3 Neighbor Discovery (ND)

The following illustrates Neighbor Discovery (ND). Similar to ARP. Each link-local address shows up in the ND cache. Replacing the 'fe80::' with 'fec0::2' (in this case due to both link-local addresses and the SLAAC addresses using the same EUI-64 format) in a ping will populate the ND cache.

Below, I'm attempting to populate the ND cache with the newly auto-configured addresses.

We'll take a look at the ND cache beforehand:

```
bblackford@fw01.bchome> show ipv6 neighbors
IPv6 Address                 Linklayer Address  State       Exp Rtr Secure Interface
fe80::218:8bff:fe19:e23a     00:18:8b:19:e2:3a  stale       828 no  no      vlan.2
fe80::1439:3db1:ec2c:54ad    00:0e:a6:f4:5b:5b  stale       469 no  no      vlan.2
fe80::5a1f:aaff:fe51:4916    58:1f:aa:51:49:16  stale       322 no  no      vlan.2
fe80::e2f8:47ff:fe3d:5352    e0:f8:47:3d:53:52  stale       1155 no no      vlan.2
fec0::2:218:8bff:fe19:e23a   00:18:8b:19:e2:3a  stale       521 no  no      vlan.2
```

Then, we'll start to ping some addresses. *Note, these are being ping'd for the first time on these addresses which explains the packet loss. This is cosmetic and has no bearing on actual performance. One device didn't ping. This is my wife's iPhone WIFI interface. I think she had just left the house when I ran this test. ☺.*

```
bblackford@fw01.bchome> ping fec0::2:e2f8:47ff:fe3d:5352
PING6(56=40+8+8 bytes) fec0:0:0:2::1 --> fec0::2:e2f8:47ff:fe3d:5352
16 bytes from fec0::2:e2f8:47ff:fe3d:5352, icmp_seq=1 hlim=10 time=5.540 ms
16 bytes from fec0::2:e2f8:47ff:fe3d:5352, icmp_seq=2 hlim=10 time=44.006 ms
^C
--- fec0::2:e2f8:47ff:fe3d:5352 ping6 statistics ---
4 packets transmitted, 2 packets received, 50% packet loss
round-trip min/avg/max/std-dev = 5.540/24.773/44.006/19.233 ms

bblackford@fw01.bchome> ping fec0::2:5a1f:aaff:fe51:4916
PING6(56=40+8+8 bytes) fec0:0:0:2::1 --> fec0::2:5a1f:aaff:fe51:4916
16 bytes from fec0::2:5a1f:aaff:fe51:4916, icmp_seq=1 hlim=10 time=91.925 ms
16 bytes from fec0::2:5a1f:aaff:fe51:4916, icmp_seq=2 hlim=10 time=17.928 ms
16 bytes from fec0::2:5a1f:aaff:fe51:4916, icmp_seq=3 hlim=10 time=33.780 ms
^C
--- fec0::2:5a1f:aaff:fe51:4916 ping6 statistics ---
4 packets transmitted, 3 packets received, 25% packet loss
round-trip min/avg/max/std-dev = 17.928/47.878/91.925/31.811 ms

bblackford@fw01.bchome> ping fec0::2:1439:3db1:ec2c:54ad
PING6(56=40+8+8 bytes) fec0:0:0:2::1 --> fec0::2:1439:3db1:ec2c:54ad
^C
--- fec0::2:1439:3db1:ec2c:54ad ping6 statistics ---
6 packets transmitted, 0 packets received, 100% packet loss
```

So now we take another look at the neighbor cache and see three new SLAAC addresses (fec0:) in the output.

```
bblackford@fw01.bchome> show ipv6 neighbors
IPv6 Address                 Linklayer Address  State       Exp Rtr Secure Interface
fe80::218:8bff:fe19:e23a     00:18:8b:19:e2:3a  stale       708 no  no     vlan.2
fe80::1439:3db1:ec2c:54ad    00:0e:a6:f4:5b:5b  stale       349 no  no     vlan.2
fe80::5a1f:aaff:fe51:4916    58:1f:aa:51:49:16  stale       202 no  no     vlan.2
fe80::e2f8:47ff:fe3d:5352    e0:f8:47:3d:53:52  stale       1035 no no     vlan.2
fec0::2:218:8bff:fe19:e23a   00:18:8b:19:e2:3a  stale       401 no  no     vlan.2
fec0::2:1439:3db1:ec2c:54ad  00:0e:a6:f4:5b:5b  reachable   22  no  no     vlan.2
fec0::2:5a1f:aaff:fe51:4916  58:1f:aa:51:49:16  stale       1198 no no     vlan.2
fec0::2:e2f8:47ff:fe3d:5352  e0:f8:47:3d:53:52  stale       1169 no no     vlan.2

bblackford@fw01.bchome>
```

Later, the iPhone was back home, so I scanned it:

```
shambhala[0]:~# nmap -6 fec0::2:5a1f:aaff:fe51:4916

Starting Nmap 5.00 ( http://nmap.org ) at 2012-02-22 13:54 PST
Interesting ports on fec0::2:5a1f:aaff:fe51:4916:
Not shown: 999 closed ports
PORT      STATE SERVICE
62078/tcp open  iphone-sync

Nmap done: 1 IP address (1 host up) scanned in 38.17 seconds
```

4.3 Verifying Connectivity

First I'll ping the Centos server. It's already in the neighbor cache and I don't need to ping it first in order to connect to it, I just think it looks cool.

```
bblackford@fw01.bchome> ping fec0::2:218:8bff:fe19:e23a
PING6(56=40+8+8 bytes) fec0:0:0:2::1 --> fec0::2:218:8bff:fe19:e23a
16 bytes from fec0::2:218:8bff:fe19:e23a, icmp_seq=0 hlim=10 time=15.784 ms
16 bytes from fec0::2:218:8bff:fe19:e23a, icmp_seq=1 hlim=10 time=3.742 ms
16 bytes from fec0::2:218:8bff:fe19:e23a, icmp_seq=2 hlim=10 time=4.308 ms
16 bytes from fec0::2:218:8bff:fe19:e23a, icmp_seq=3 hlim=10 time=9.281 ms
16 bytes from fec0::2:218:8bff:fe19:e23a, icmp_seq=4 hlim=10 time=3.676 ms
16 bytes from fec0::2:218:8bff:fe19:e23a, icmp_seq=5 hlim=10 time=8.896 ms
16 bytes from fec0::2:218:8bff:fe19:e23a, icmp_seq=6 hlim=10 time=3.434 ms
16 bytes from fec0::2:218:8bff:fe19:e23a, icmp_seq=7 hlim=10 time=3.392 ms
16 bytes from fec0::2:218:8bff:fe19:e23a, icmp_seq=8 hlim=10 time=3.510 ms
16 bytes from fec0::2:218:8bff:fe19:e23a, icmp_seq=9 hlim=10 time=3.306 ms
16 bytes from fec0::2:218:8bff:fe19:e23a, icmp_seq=10 hlim=10 time=3.162 ms
16 bytes from fec0::2:218:8bff:fe19:e23a, icmp_seq=11 hlim=10 time=3.097 ms
^C
```

SSH'ing from the Juniper router to the Centos server for the first time.

Hey, it worked! Important note here. You might have clued on to this earlier, but the centos box is not configured for DHCP with IPV4 or IPV6. However, it grabbed a SLAAC address (the site-local in this case) using the EUI-64 format. If you look at the HWaddr and compare that to the inet6 addresses, you can see how the EUI-64 address is derived.

```
bblackford@fw01.bchome> ssh root@fec0::2:218:8bff:fe19:e23a
The authenticity of host 'fec0::2:218:8bff:fe19:e23a (fec0::2:218:8bff:fe19:e23a)'
can't be established.
RSA key fingerprint is 8e:7a:78:3b:b1:53:5b:67:43:b6:db:ac:da:bc:99:d5.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'fec0::2:218:8bff:fe19:e23a' (RSA) to the list of known
hosts.
root@fec0::2:218:8bff:fe19:e23a's password:
Last login: Wed Feb  8 04:52:21 2012
shambhala[0]:~#
shambhala[0]:~# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:18:8B:19:E2:3A
          inet addr:192.168.2.10  Bcast:192.168.2.255  Mask:255.255.255.0
          inet6 addr: fec0::2:218:8bff:fe19:e23a/64 Scope:Site
          inet6 addr: fe80::218:8bff:fe19:e23a/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:6982727 errors:0 dropped:0 overruns:0 frame:0
          TX packets:7627655 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4381861225 (4.0 GiB)  TX bytes:6691158384 (6.2 GiB)
          Interrupt:169
```

Link-local address compared to the MAC. Note the FF:FE:

```
shambhala[0]:~# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:18:8B:19:E2:3A
          inet addr:192.168.2.10  Bcast:192.168.2.255  Mask:255.255.255.0
          inet6 addr: fec0::2:218:8bff:fe19:e23a/64 Scope:Site
          inet6 addr: fe80::218:8bff:fe19:e23a/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:6982727 errors:0 dropped:0 overruns:0 frame:0
          TX packets:7627655 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4381861225 (4.0 GiB)  TX bytes:6691158384 (6.2 GiB)
          Interrupt:169
```

Also, notice that the inet6 loopback address is ::/128.

```
shambhala[1]:~# ifconfig lo
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:30701 errors:0 dropped:0 overruns:0 frame:0
          TX packets:30701 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:2670542 (2.5 MiB)  TX bytes:2670542 (2.5 MiB)
```

You can see that I'm logged in to the Centos host from the SRX's source interface address on vlan.2.

```
shambhala[0]:~# who am i
root     pts/0        2012-02-08 11:17 (fec0:0:0:2::1)
shambhala[0]:~#
```

```
shambhala[0]:~# w
 11:17:47 up 21 days,  1:40,  2 users,  load average: 0.02, 0.01, 0.00
USER     TTY      FROM              LOGIN@   IDLE   JCPU   PCPU WHAT
root     tty1     -                22Jan12  6:22m  2.54s  2.50s minicom
root     pts/0    fec0:0:0:2::1    11:17    1.00s  0.02s  0.00s w
```

Centos family inet6 route table. Since no other gateway has been defined, the system will use the next-hop link-local address. The output below illustrates that SLAAC doesn't appear to hand a default gateway on the local subnet. Link-local addresses are only unique to the local link, so until I configure a path out, I may be stuck in this subnet. I will test this in a later section. Note: addresses beginning with "FF" are multicast.

```
shambhala[0]:~# route -A inet6
Kernel IPv6 routing table
Destination                                 Next Hop
Flags Metric Ref    Use Iface
fe80::/64                                   *                                   U
256    0        0 eth0
fec0:0:0:2::/64                             *                                   UA
256    1388       0 eth0
*/0                                         fe80::5e5e:abff:fee0:7d88
UGDA  1024   142       0 eth0
localhost6.localdomain6/128                 *                                   U
0      29       1 lo
fe80::218:8bff:fe19:e23a/128                *                                   U
0      33       1 lo
fec0::2:218:8bff:fe19:e23a/128              *                                   U
0      343       1 lo
ff02::fb/128                                ff02::fb                            UC
0      15       0 eth0
ff00::/8                                    *                                   U
256    0        0 eth0
shambhala[0]:~#
```

4.4 SLAAC on Two Subnets

Now let's enable vlan.3 by adding address families to vlan.3 and bringing up a host. Also, we'll add another vlan interface to the router-advertisement and define a prefix. Note the "3" at the /64 boundary.

For this section, my host device under test in vlan.3 is a Dell Laptop running Linux:

On the router, I'll have to set up my test environment:

```
bblackford@fw01.bchome> show configuration interfaces vlan.3
description "_VLAN_, TEST LAN";
family inet {
    filter {
        input INSIDE;
    }
    address 192.168.3.1/24;
}
family inet6 {
    address fec0:0:0:3::1/64;
}
```

```
bblackford@fw01.bchome> show interfaces terse vlan
Interface               Admin Link Proto    Local                 Remote
vlan                    up    up
vlan.2                  up    up   inet     192.168.2.1/24
                                    inet6    fe80::5e5e:abff:fee0:7d88/64
                                             fec0:0:0:2::1/64
vlan.3                  up    up   inet     192.168.3.1/24
                                    inet6    fe80::5e5e:abff:fee0:7d88/64
                                             fec0:0:0:3::1/64
```

```
bblackford@fw01.bchome> show configuration protocols router-advertisement
interface vlan.2 {
    current-hop-limit 10;
    prefix fec0:0:0:2::/64;
}
interface vlan.3 {
    current-hop-limit 10;
    prefix fec0:0:0:3::/64;
}
```

Now, we'll verify the SLAAC behavior on the second subnet (vlan.3). We see the router-advertisement output now includes an additional interface.

```
bblackford@fw01.bchome> show ipv6 router-advertisement
Interface: vlan.2
  Advertisements sent: 1012, last sent 00:01:12 ago
  Solicits received: 265, last received 00:48:37 ago
  Advertisements received: 0
Interface: vlan.3
  Advertisements sent: 4, last sent 00:03:32 ago
  Solicits received: 0
  Advertisements received: 0
```

I see no Solicits received from the host.

```
bblackford@fw01.bchome> show ipv6 neighbors
IPv6 Address                  Linklayer Address  State       Exp Rtr Secure Interface
fe80::218:8bff:fe19:e23a      00:18:8b:19:e2:3a  stale       1033 no no      vlan.2
fe80::5a1f:aaff:fe51:4916     58:1f:aa:51:49:16  stale       292 no  no      vlan.2
fe80::66a7:69ff:feee:238d     64:a7:69:ee:23:8d  stale       391 no  no      vlan.2
fe80::a409:dd3f:6f19:6091     00:22:fb:b2:66:62  stale       646 no  no      vlan.2
fe80::e2f8:47ff:fe3d:5352     e0:f8:47:3d:53:52  stale       465 no  no      vlan.2
fec0::2:218:8bff:fe19:e23a    00:18:8b:19:e2:3a  stale       319 no  no      vlan.2
```

Hmm, not what I expected. I expected to see vlan.3 link-local address. I'll login to the Linux laptop via the IPV4 address and take a peak.

```
bblackford@fw01.bchome> ssh root@192.168.3.128
root@192.168.3.128's password:
Last login: Sun Feb 12 14:51:30 2012 from 192.168.3.1
[root@dell-6400 ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:15:C5:CD:46:D8
          inet addr:192.168.3.128  Bcast:192.168.3.255  Mask:255.255.255.0
          inet6 addr: fec0::3:215:c5ff:fecd:46d8/64 Scope:Site
          inet6 addr: fe80::215:c5ff:fecd:46d8/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:475958 errors:0 dropped:0 overruns:0 frame:0
          TX packets:274770 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:631944377 (602.6 MiB)  TX bytes:22374457 (21.3 MiB)
          Interrupt:17
```

Well, the addresses are there. I'll try to ping the IPV6 interface address of the router.

```
[root@dell-6400 ~]# ping6 fec0:0:0:3::1
PING fec0:0:0:3::1(fec0:0:0:3::1) 56 data bytes
64 bytes from fec0:0:0:3::1: icmp_seq=1 ttl=10 time=27.1 ms
64 bytes from fec0:0:0:3::1: icmp_seq=2 ttl=10 time=2.90 ms
64 bytes from fec0:0:0:3::1: icmp_seq=3 ttl=10 time=2.31 ms
^C
```

Back to the router.

```
[root@dell-6400 ~]# exit
logout
Connection to 192.168.3.128 closed.

bblackford@fw01.bchome> show ipv6 neighbors
IPv6 Address                  Linklayer Address  State       Exp Rtr Secure Interface
fe80::218:8bff:fe19:e23a      00:18:8b:19:e2:3a  stale       670 no  no      vlan.2
fe80::5a1f:aaff:fe51:4916     58:1f:aa:51:49:16  stale       1147 no no      vlan.2
fe80::66a7:69ff:feee:238d     64:a7:69:ee:23:8d  stale       28  no  no      vlan.2
fe80::a409:dd3f:6f19:6091     00:22:fb:b2:66:62  stale       283 no  no      vlan.2
fe80::e2f8:47ff:fe3d:5352     e0:f8:47:3d:53:52  stale       102 no  no      vlan.2
fec0::2:218:8bff:fe19:e23a    00:18:8b:19:e2:3a  stale       1174 no no      vlan.2
fec0::3:215:c5ff:fecd:46d8    00:15:c5:cd:46:d8  stale       1097 no no      vlan.3
```

Well, There's the SLAAC address. The IPV6 Neighbor Discovery does have odd behavior. Back to the host using the IPV6 address.

```
bblackford@fw01.bchome> ssh root@fec0::3:215:c5ff:fecd:46d8
The authenticity of host 'fec0::3:215:c5ff:fecd:46d8 (fec0::3:215:c5ff:fecd:46d8)'
can't be established.
RSA key fingerprint is a8:b8:01:52:81:0a:eb:57:f7:d8:3f:5a:f2:08:cf:65.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'fec0::3:215:c5ff:fecd:46d8' (RSA) to the list of known
hosts.
root@fec0::3:215:c5ff:fecd:46d8's password:
Last login: Sun Feb 12 14:53:52 2012 from 192.168.3.1
[root@dell-6400 ~]#

[root@dell-6400 ~]# route -A inet6
Kernel IPv6 routing table
Destination                                 Next Hop
Flags Metric Ref    Use Iface
fe80::/64                                   *                                   U
256    0        0 wlan0
fe80::/64                                   *                                   U
256    0        0 eth0
fec0:0:0:3::1/128                           fec0:0:0:3::1
UAC   0      3        1 eth0
fec0:0:0:3::/64                             *                                   UA
256    191       0 eth0
*/0                                         fe80::5e5e:abff:fee0:7d88
UGDA  1024   0        0 eth0
dell-6400/128                               *                                   U
0      10       1 lo
fe80::20e:a6ff:fef4:5df7/128                *                                   U
0      0        1 lo
fe80::215:c5ff:fecd:46d8/128                *                                   U
0      180       1 lo
fec0::3:215:c5ff:fecd:46d8/128              *                                   U
0      120       1 lo
ff00::/8                                    *                                   U
256    0        0 wlan0
ff00::/8                                    *                                   U
256    0        0 eth0
[root@dell-6400 ~]#
```

The output shows that the default gateway is via the link-local address. I'll test the gateway by pinging and tracing to the site-local address of the Centos server on vlan.2.

```
[root@dell-6400 ~]# ping6 fec0::2:218:8bff:fe19:e23a
PING fec0::2:218:8bff:fe19:e23a(fec0::2:218:8bff:fe19:e23a) 56 data bytes
64 bytes from fec0::2:218:8bff:fe19:e23a: icmp_seq=1 ttl=9 time=0.716 ms
64 bytes from fec0::2:218:8bff:fe19:e23a: icmp_seq=2 ttl=9 time=0.708 ms
^C
--- fec0::2:218:8bff:fe19:e23a ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1000ms
rtt min/avg/max/mdev = 0.708/0.712/0.716/0.004 ms

[root@dell-6400 ~]# traceroute6 fec0::2:218:8bff:fe19:e23a
traceroute to fec0::2:218:8bff:fe19:e23a (fec0::2:218:8bff:fe19:e23a), 30 hops max, 80
byte packets
 1  fec0:0:0:3::1 (fec0:0:0:3::1)  1.254 ms  1.192 ms  1.094 ms
 2  fec0::2:218:8bff:fe19:e23a (fec0::2:218:8bff:fe19:e23a)  6.860 ms  8.205 ms  8.426
ms
[root@dell-6400 ~]#
```

The output shows that the first hop is the vlan.3 interface address of the router even though the routing table suggested otherwise. *This confirms that the assumption made in section 4.3, the absence of a SLAAC handed gateway, is indeed expected behavior.*

Below is a reminder of the configured static interface address on the router vlan.3.

```
bblackford@fw01.bchome> show configuration interfaces vlan.3
description "_VLAN_, TEST LAN";
family inet {
    filter {
        input INSIDE;
    }
    address 192.168.3.1/24;
}
family inet6 {
    address fec0:0:0:3::1/64;
}
```

4.5 Conclusions and unsolicited opinions

SLAAC and EUI-64 have some limitations and some wonky behavior.

MAC's are unique. Certainly **not** recommended for systems that are dual-homed. SLAAC may have a built-in method to disambiguate the addresses in this case. But looking below, you can imagine a case where a host with a tagged interface on both vlans will get the same EUI-64 link-local address.

```
bblackford@fw01.bchome> show interfaces terse vlan
Interface               Admin Link Proto    Local                 Remote
vlan                    up    up
vlan.2                  up    up   inet     192.168.2.1/24
                                    inet6    fe80::5e5e:abff:fee0:7d88/64
                                             fec0:0:0:2::1/64
vlan.3                  up    up   inet     192.168.3.1/24
                                    inet6    fe80::5e5e:abff:fee0:7d88/64
                                             fec0:0:0:3::1/64
```

EUI-64 creates difficult to use/remember addresses.

SLAAC provides a gateway of the interface address, but the route table suggests the use of the link-local address.

Neighbor Discovery also exhibits odd behavior and may lead to some incorrect assumptions. (not seeing the RS and no ND cache even though the laptop did receive it's SLAAC address and was connected).

The Linux inet6 route table shows the link-local address of the router on that subnet as the default gateway, but tests showed that the interface address of the router on that subnet is used as the next-hop address. Again, somewhat mis-leading.

Note self: I may need to re-test SLAAC. Chris Grundemann's One Day guide http://forums.juniper.net/jnet/attachments/jnet/Day1Books/12/10/Exploring%20IPv6.pdf shows the use of the interface `eui-64` *keyword. I found that my connectivity verification and SLAAC behavior works fine without this, but I should use BCP.*

5.0 Static Addressing

5.1 Set Up and Verification

- Deactivate router-advertisement to remove SLAAC functionality
- Each test host will be statically assigned an IPV6 address for the respective VLAN.
- Verify connectivity, view ND cache, etc.

```
[edit protocols router-advertisement]
bblackford@fw01.bchome# deactivate interface vlan.2

[edit protocols router-advertisement]
bblackford@fw01.bchome# deactivate interface vlan.3

[edit protocols router-advertisement]
bblackford@fw01.bchome# show
inactive: interface vlan.2 {
    current-hop-limit 10;
    prefix fec0:0:0:2::/64;
}
inactive: interface vlan.3 {
    current-hop-limit 10;
    prefix fec0:0:0:3::/64;
}
```

Ok, now to statically assign inet6 address and gw on Centos server.

```
shambhala[0]:~# ifconfig eth0 inet6 add fec0:0:0:2::10/64
shambhala[0]:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:18:8B:19:E2:3A
          inet addr:192.168.2.10  Bcast:192.168.2.255  Mask:255.255.255.0
          inet6 addr: fec0:0:0:2::10/64 Scope:Site
          inet6 addr: fec0::2:218:8bff:fe19:e23a/64 Scope:Site
          inet6 addr: fe80::218:8bff:fe19:e23a/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:3019185 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3243546 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1786856715 (1.6 GiB)  TX bytes:2698085594 (2.5 GiB)
          Interrupt:169
```

I notice that the site-local address assigned through SLAAC is still bound to eth0. This may be expected behavior. I'm not certain if there is a mechanism to withdrawal SLAAC addresses as there is with DHCP release. *UPDATE: Later, my MacBook Pro which leaves the LAN, goes to work and comes back, did not get a SLAAC address when I came home with the router-advertisement still disabled. So at this point, my assumption is that the client won't withdrawal a SLAAC address (no lease time) if they stay connected, but will lose the address once they are.*

```
shambhala[0]:~# route -A inet6 add ::/0 gw fec0:0:0:2::1

shambhala[0]:~# route -A inet6
Kernel IPv6 routing table
Destination                                 Next Hop
Flags Metric Ref    Use Iface
fe80::/64                                   *                                   U
256    0        0 eth0
fec0:0:0:2::/64                             *                                   UA
256    1274       0 eth0
*/0                                         fec0:0:0:2::1                       UG
1      0        0 eth0
localhost6.localdomain6/128                 *                                   U
0      16       1 lo
fe80::218:8bff:fe19:e23a/128                *                                   U
0      1044       1 lo
fec0:0:0:2::10/128                          *                                   U
0      3        1 lo
fec0::2:218:8bff:fe19:e23a/128              *                                   U
0      368       1 lo
ff00::/8                                    *                                   U
256    0        0 eth0
shambhala[0]:~#
```

The above output shows the static configured gateway in the Linux inet6 route table vs. the link-local gateway shown in the SLAAC test case. For me, this is much more intuitive however, YMMV.

Ok, now to check the ND cache.

```
bblackford@fw01.bchome> show ipv6 neighbors
IPv6 Address                  Linklayer Address  State       Exp Rtr Secure Interface
fe80::215:c5ff:fecd:46d8      00:15:c5:cd:46:d8  stale       813 no  no      vlan.3
fe80::218:8bff:fe19:e23a      00:18:8b:19:e2:3a  stale       888 no  no      vlan.2
fe80::1439:3db1:ec2c:54ad     00:0e:a6:f4:5b:5b  stale       335 no  no      vlan.2
fe80::e2f8:47ff:fe3d:5352     e0:f8:47:3d:53:52  stale       674 no  no      vlan.2
fec0::2:218:8bff:fe19:e23a    00:18:8b:19:e2:3a  stale       196 no  no      vlan.2
fec0::3:215:c5ff:fecd:46d8    00:15:c5:cd:46:d8  stale       360 no  no      vlan.3
```

No sight of the new configured address.

Ping to populate the ND cache.

```
bblackford@fw01.bchome> ping fec0:0:0:2::10
PING6(56=40+8+8 bytes) fec0:0:0:2::1 --> fec0:0:0:2::10
16 bytes from fec0:0:0:2::10, icmp_seq=1 hlim=10 time=3.291 ms
16 bytes from fec0:0:0:2::10, icmp_seq=2 hlim=10 time=6.410 ms
16 bytes from fec0:0:0:2::10, icmp_seq=3 hlim=10 time=3.247 ms
^C
```

Ok, there it is.

```
bblackford@fw01.bchome> show ipv6 neighbors
IPv6 Address                 Linklayer Address  State       Exp Rtr Secure Interface
fe80::215:c5ff:fecd:46d8     00:15:c5:cd:46:d8  stale       793 no  no      vlan.3
fe80::218:8bff:fe19:e23a     00:18:8b:19:e2:3a  stale       868 no  no      vlan.2
fe80::1439:3db1:ec2c:54ad    00:0e:a6:f4:5b:5b  stale       315 no  no      vlan.2
fe80::e2f8:47ff:fe3d:5352    e0:f8:47:3d:53:52  stale       654 no  no      vlan.2
fec0:0:0:2::10               00:18:8b:19:e2:3a  reachable   19  no  no      vlan.2
fec0::2:218:8bff:fe19:e23a   00:18:8b:19:e2:3a  stale       176 no  no      vlan.2
fec0::3:215:c5ff:fecd:46d8   00:15:c5:cd:46:d8  stale       340 no  no      vlan.3

bblackford@fw01.bchome>
```

Now, onto vlan.3.

```
[root@dell-6400 ~]# ifconfig eth0 inet6 add fec0:0:0:3::128/64

[root@dell-6400 ~]# route -A inet6 add ::/0 gw fec0:0:0:3::1

[root@dell-6400 ~]# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:15:C5:CD:46:D8
          inet addr:192.168.3.128  Bcast:192.168.3.255  Mask:255.255.255.0
          inet6 addr: fec0::3:215:c5ff:fecd:46d8/64 Scope:Site
          inet6 addr: fe80::215:c5ff:fecd:46d8/64 Scope:Link
          inet6 addr: fec0:0:0:3::128/64 Scope:Site
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:565159 errors:0 dropped:0 overruns:0 frame:0
          TX packets:284534 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:638497099 (608.9 MiB)  TX bytes:24520812 (23.3 MiB)
          Interrupt:17
```

```
[root@dell-6400 ~]# route -nA inet6
Kernel IPv6 routing table
Destination                                 Next Hop
Flags Metric Ref    Use Iface
fe80::/64                                   ::                                      U
256    0        0 wlan0
fe80::/64                                   ::                                      U
256    0        0 eth0
fec0:0:0:3::1/128                           fec0:0:0:3::1
UAC   0      19       1 eth0
fec0:0:0:3::/64                             ::                                      UA
256    19       0 eth0
::/0                                        fec0:0:0:3::1                           UG
1      0        0 eth0
::1/128                                     ::                                      U
0      12       1 lo
fe80::20e:a6ff:fef4:5df7/128                ::                                      U
0      0        1 lo
fe80::215:c5ff:fecd:46d8/128                ::                                      U
0      147       1 lo
fec0:0:0:3::128/128                         ::                                      U
0      0        1 lo
fec0::3:215:c5ff:fecd:46d8/128              ::                                      U
0      890       1 lo
ff00::/8                                    ::                                      U
256    0        0 wlan0
ff00::/8                                    ::                                      U
256    0        0 eth0
[root@dell-6400 ~]#
```

Try some pings. First to the gateway, then to the Centos host on vlan.2

```
[root@dell-6400 ~]# ping6 fec0:0:0:3::1
PING fec0:0:0:3::1(fec0:0:0:3::1) 56 data bytes
64 bytes from fec0:0:0:3::1: icmp_seq=1 ttl=64 time=6.03 ms
64 bytes from fec0:0:0:3::1: icmp_seq=2 ttl=64 time=2.62 ms
64 bytes from fec0:0:0:3::1: icmp_seq=3 ttl=64 time=2.84 ms
^C

[root@dell-6400 ~]# ping6 fec0:0:0:2::10
PING fec0:0:0:2::10(fec0:0:0:2::10) 56 data bytes
64 bytes from fec0:0:0:2::10: icmp_seq=1 ttl=9 time=0.839 ms
64 bytes from fec0:0:0:2::10: icmp_seq=2 ttl=9 time=0.689 ms
64 bytes from fec0:0:0:2::10: icmp_seq=3 ttl=9 time=0.643 ms
^C
```

Now a traceroute,

```
[root@dell-6400 ~]# traceroute fec0:0:0:2::10
traceroute to fec0:0:0:2::10 (fec0:0:0:2::10), 30 hops max, 80 byte packets
 1  fec0:0:0:3::1 (fec0:0:0:3::1)  1.421 ms  1.395 ms  1.329 ms
 2  fec0:0:0:2::10 (fec0:0:0:2::10)  8.004 ms  8.011 ms  8.601 ms
[root@dell-6400 ~]#
```

Now, a re-check of the ND cache shows the new vlan.3 static address on the Dell Laptop.

```
bblackford@fw01.bchome> show ipv6 neighbors
IPv6 Address                  Linklayer Address  State       Exp Rtr Secure Interface
fe80::215:c5ff:fecd:46d8      00:15:c5:cd:46:d8  stale       619 no  no      vlan.3
fe80::218:8bff:fe19:e23a      00:18:8b:19:e2:3a  stale       861 no  no      vlan.2
fe80::1439:3db1:ec2c:54ad     00:0e:a6:f4:5b:5b  stale       1180 no no      vlan.2
fe80::6233:4bff:fe61:5bb9     60:33:4b:61:5b:b9  stale       763 no  no      vlan.2
fec0:0:0:2::10                00:18:8b:19:e2:3a  reachable   14  no  no      vlan.2
fec0::2:218:8bff:fe19:e23a    00:18:8b:19:e2:3a  stale       167 no  no      vlan.2
fec0:0:0:3::128               00:15:c5:cd:46:d8  stale       130 no  no      vlan.3
fec0::3:215:c5ff:fecd:46d8    00:15:c5:cd:46:d8  reachable   42  no  no      vlan.3
```

5.2 Mixed

Not really part of static, but I thought it would be good to show mixed SLAAC and static. So I activate both vlans in router-advertisement.

```
bblackford@fw01.bchome> show configuration protocols router-advertisement
traceoptions {
    file RA size 10m files 2;
    flag all;
}
interface vlan.2 {
    current-hop-limit 10;
    prefix fec0:0:0:2::/64;
}
interface vlan.3 {
    current-hop-limit 10;
    prefix fec0:0:0:3::/64;
}
```

Both static and SLAAC addresses are shown in the Linux hosts output below.

```
shambhala[1]:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:18:8B:19:E2:3A
          inet addr:192.168.2.10  Bcast:192.168.2.255  Mask:255.255.255.0
          inet6 addr: fec0:0:0:2::10/64 Scope:Site
          inet6 addr: fec0::2:218:8bff:fe19:e23a/64 Scope:Site
          inet6 addr: fe80::218:8bff:fe19:e23a/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:4839942 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5038523 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:3107696864 (2.8 GiB)  TX bytes:3885404054 (3.6 GiB)
          Interrupt:169
```

```
[root@dell-6400 ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:15:C5:CD:46:D8
          inet addr:192.168.3.128  Bcast:192.168.3.255  Mask:255.255.255.0
          inet6 addr: fec0::3:215:c5ff:fecd:46d8/64 Scope:Site
          inet6 addr: fe80::215:c5ff:fecd:46d8/64 Scope:Link
          inet6 addr: fec0:0:0:3::128/64 Scope:Site
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1588713 errors:1 dropped:0 overruns:0 frame:0
          TX packets:967862 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2271750512 (2.1 GiB)  TX bytes:75468666 (71.9 MiB)
          Interrupt:17
```

6.0 DHCPv6

At the time of this writing, DHCPv6 is still an evolving standard and too immature to do any worthwhile testing. Below I have some config snippets and I ran some basic tests, but so far I cannot get it to work. I will suspend this test case until a later date.

```
[edit]
bblackford@fw01.bchome# show system services
```
<output snipped>

```
dhcp-local-server {
    dhcpv6 {
        group V6-VLAN3 {
            interface vlan.3;
        }
    }
}
```
<output snipped>

```
[edit]
bblackford@fw01.bchome# show access
address-assignment {
    pool V6-VLAN3-POOL {
        family inet6 {
            prefix fec0:0:0:3::/64;
            range VLAN3-V6-RANGE {
                low fec0:0:0:3::192/128;
                high fec0:0:0:3::224/128;
            }
            dhcp-attributes {
                maximum-lease-time 300;
                dns-server {
                    fec0:0:0:3::1;
                }
            }
        }
    }
}
```

Now, I activate interface vlan.3 in protocols router-advertisement and make it a managed-configuration.

```
[edit]
bblackford@fw01.bchome# show protocols
router-advertisement {
    inactive: interface vlan.2 {
        current-hop-limit 10;
        prefix fec0:0:0:2::/64;
    }
    interface vlan.3 {
        managed-configuration;
        current-hop-limit 10;
        prefix fec0:0:0:3::/64;  <== no need to remove prefix.
    }
}
```

NOTE: at this point DHCPV6 is not working as expected.

I'll enable traceoptions.

```
bblackford@fw01.bchome> show configuration protocols router-advertisement
traceoptions {
    file RA size 10m files 10;
    flag all;
}
interface vlan.2 {
    current-hop-limit 10;
    prefix fec0:0:0:2::/64;
}
interface vlan.3 {
    managed-configuration;
    current-hop-limit 10;
    inactive: prefix fec0:0:0:3::/64;  <== I remove it anyway
}
```

I'll now search the newly created log for the Linux Laptop's MAC (last four).

```
bblackford@fw01.bchome> show log RA | match 46d8
Feb 14 21:58:22.741902 ipv6_ra_receive_solicit: received solicit from
fe80::215:c5ff:fecd:46d8
Feb 14 21:58:22.742216 ipv6_ra_receive_solicit: task Router-Advertisement src
fe80::215:c5ff:fecd:46d8 dst ff02::2 hdr 0x1601000 count 16 intf 0x16435dc
Feb 14 21:58:26.751751 ipv6_ra_receive_solicit: received solicit from
fe80::215:c5ff:fecd:46d8
Feb 14 21:58:26.752043 ipv6_ra_receive_solicit: task Router-Advertisement src
fe80::215:c5ff:fecd:46d8 dst ff02::2 hdr 0x1601000 count 16 intf 0x16435dc
Feb 14 21:58:30.760444 ipv6_ra_receive_solicit: received solicit from
fe80::215:c5ff:fecd:46d8
Feb 14 21:58:30.760680 ipv6_ra_receive_solicit: task Router-Advertisement src
fe80::215:c5ff:fecd:46d8 dst ff02::2 hdr 0x1601000 count 16 intf 0x16435dc
```

The above output looks like the router is attempting to use SLAAC even though, this subnet should only be using managed-configuration. Not sure if this is expected behavior, or an SRX 10.4R8.5 bug.

I'm enabling traceoptions in system services dhcp-local-server as well

```
bblackford@fw01.bchome> ...tem services dhcp-local-server
traceoptions {
    file D size 5m files 2;
    flag all;
}
dhcpv6 {
    group V6-VLAN3 {
        interface vlan.3;
    }
}
```

On the Linux host on vlan.3:

```
[root@dell-6400 ~]# dhclient -6 eth0
^C
[root@dell-6400 ~]# exit
logout
Connection to 192.168.3.128 closed.
```

Note: The host never received a inet6 address.

```
bblackford@fw01.bchome> show log D
Feb 17 07:48:46 fw01.bchome clear-log[43378]: logfile cleared
Feb 17 07:51:19 DHCPv6: dhcpv6_io_reader: received 52 bytes, dst addr ff02::1:2
Feb 17 07:51:19 DHCPv6: Recv pkt in default/default, ifindex 70, if vlan.3
Feb 17 07:51:19 DHCPv6: dhcpv6 pkt parsing - Start
Feb 17 07:51:19 DHCPv6: --[     len == 52 ]--
Feb 17 07:51:19 DHCPv6: --[    from == fe80::215:c5ff:fecd:46d8/546 ]--
Feb 17 07:51:19 DHCPv6: --[ msgtype == DHCPV6_SOLICIT ]--
Feb 17 07:51:19 DHCPv6: --[     xid == 29b633 ]--
Feb 17 07:51:19 DHCPv6: --[ OPTION_CLIENTID
Feb 17 07:51:19 DHCPv6:       OPTION code   1, len  14, data 00 01 00 01 16 cd cb 70
00 15 c5 cd 46 d8 ]--
Feb 17 07:51:19 DHCPv6: --[ OPTION_OPT_REQ
Feb 17 07:51:19 DHCPv6:       OPTION code   6, len   4, data 00 17 00 18 ]--
Feb 17 07:51:19 DHCPv6: --[ Internally Unsupported Option
Feb 17 07:51:19 DHCPv6:       OPTION code   8, len   2, data 00 00 ]--
Feb 17 07:51:19 DHCPv6: --[ OPTION_IA_NA
Feb 17 07:51:19 DHCPv6:       OPTION code   3, len  12, iaid -976402728, T1 3600, T2
5400 ]--
Feb 17 07:51:19 DHCPv6:     Parsing suboptions of OPTION_IA_NA - Start
Feb 17 07:51:19 DHCPv6:     Parsing suboptions of OPTION_IA_NA - Done
Feb 17 07:51:19 DHCPv6: dhcpv6 pkt parsing - End
Feb 17 07:51:19 DHCPv6: dhcpv6_packet_handle: Could not find client table entry
Feb 17 07:51:19 DHCPv6 client entry ADDED, app_type 1 routing context default:default
being deletd
Feb 17 07:51:19 DHCPv6: new client table entry created for ifindex 70
Feb 17 07:51:19 Client 00-00-00-00-00-00 got event CLIENT_EVENT_SOLICIT_PDU in state
DHCPV6_LOCAL_SERVER_STATE_INIT
Feb 17 07:51:20 jdhcpd_sdb_create_session: AF_INET6, setting LR name  default
Feb 17 07:51:20 jdhcpd_sdb_create_session: AF_INET6, setting RI name  default
Feb 17 07:51:20 jdhcpd_sdb_create_session: AF_INET6, setting inner vlan_ID 3
Feb 17 07:51:20 jdhcpd_sdb_create_session: AF_INET6, setting underlying interface
vlan.3
Feb 17 07:51:20 jdhcpd_sdb_create_session: AF_INET6, setting logical if name  vlan.3
Feb 17 07:51:20 jdhcpd_sdb_create_session: AF_INET6, setting MAC addr
00:15:c5:cd:46:d8
Feb 17 07:51:20 DHCPv6: jdhcpd_sdb_create_session: setting IPv6 addr Key fec0:0:0:3::1
Feb 17 07:51:20 jdhcpd_sdb_create_session: session DB create got AF_INET6 entry id 9
Feb 17 07:51:20 Mirroring disabled (ha_graceful_switchover false) ::
Feb 17 07:51:20 Allocated memory to store in SDB, sizeof(jdhcpd_ha_client_header_t) 88
+ total_tlvs_len 98 (sizeof(client->session_id) is 9)
Feb 17 07:51:20 Saved NON-DEMUX interface name vlan.3 into mirroring ha_info
Feb 17 07:51:20 Added TLV for DHCPv6 CLIENT ID, length 14
Feb 17 07:51:20 Added TLV for DHCPv6 IA_PD or IA_NA ID, length 4
Feb 17 07:51:20 Added TLV for DHCPv6 IA_PD or IA_NA IA_TYPE, length 4
Feb 17 07:51:20 Added TLV for DHCPv6 LinkLocal Address, length 16
Feb 17 07:51:20 Size of ha_client (ptr+1 = 12137914)
Feb 17 07:51:20 Successfully saved private data for :: in SDB, client giaddr 0.0.0.0,
length 186
Feb 17 07:51:20 DHCPv6: SDB ACK will notify us for LOCAL-SERVER client session Id 9
Feb 17 07:51:20 Successful SDB Mirrored Ack: returned code SDB_SUCCESS for session id
9
Feb 17 07:51:20 Client 00-00-00-00-00-00 got event CLIENT_EVENT_CLIENT_SDB_MIRROR_ACK
in state DHCPV6_LOCAL_SERVER_STATE_WAIT_SDB_INIT_MIRROR_REQ
Feb 17 07:51:20 DHCPv6: SDB mirroring ACK received
Feb 17 07:51:20 DHCPv6: In DHCPV6_LOCAL_SERVER_STATE_WAIT_SDB_INIT_MIRROR_REQ: prof
[(null)] NDRA_PREFIX variable not present
Feb 17 07:51:20 asking authd for an address
Feb 17 07:51:20 Before adding to teardown list,
jdhcpd_inst.authd_restart_teardown_count 0
Feb 17 07:51:20 LIST IS EMPTY, jdhcpd_inst.authd_restart_teardown_count 0
```

```
Feb 17 07:51:20 Added client w/ session_id 9 in state
LOCAL_SERVER_STATE_WAIT_AUTH_REQ, to authd_restart_teardown list, count
authd_restart_teardown_count 1
Feb 17 07:51:20 Got reply for auth_request, reply auth_rqst_id = 65545- (session Id 9)
Feb 17 07:51:20 Request to remove client w. session_id 9 in state 1 from
authd_restart_teardown list, count authd_restart_teardown_count 1
Feb 17 07:51:20 Removed LAST client from authd_restart teardown list, zero out COUNT
authd_restart_teardown_count
Feb 17 07:51:20 Auth reply retval 1
Feb 17 07:51:20 DHCPv6: got SDB_USER_IPV6_ADDRESS dhcpv6_session_db_client_info_get
fec0:0:0:3::195
Feb 17 07:51:20 DHCPv6: dhcpv6_session_db_client_info_get: Got LS name: default
Feb 17 07:51:20 DHCPv6: dhcpv6_session_db_client_info_get: Got RI name default
Feb 17 07:51:20 DHCPv6: got ipv6 pool name V6-VLAN3, len 8
Feb 17 07:51:20 DHCPv6: failed to get V6 sdb auth response, return code -7
Feb 17 07:51:20 Client 00-00-00-00-00-00 got event CLIENT_EVENT_AUTH_REQ_ACK in state
DHCPV6_LOCAL_SERVER_STATE_WAIT_AUTH_REQ
Feb 17 07:51:20 DHCPv6: Auth request reply IPv6 fec0:0:0:3::195, pool name V6-VLAN3
Feb 17 07:51:20 DHCPv6: FOUND POOL 0xdb6000 w/ cfg container 0xb77800
Feb 17 07:51:20 DHCPv6: TRYING to send local server packet, pool 0xdb6000
Feb 17 07:51:20 DHCPv6: DHCPV6_PDU_SEND_TYPE_ADVERTISE: will fill in options for len 8
Feb 17 07:51:20 DHCPv6: STARTING dhcpv6_option_encode_req_option, pool 0xdb6000,
num_req_opt 2
Feb 17 07:51:20 DHCPv6: dhcpv6_option_encode_req_option, pool 0xdb6000, num_req_opt 2
Feb 17 07:51:20 DHCPv6: requested option from pool V6-VLAN3, opt_code 23,
opt_ptr_from_code 23, opt_ptr_from_len 16
Feb 17 07:51:20 SEND: incomming_safd 0xdafa00, demux 0x0, safd 0xdafa00
Feb 17 07:51:20 DHCPv6:
Dump of local server V6 PDU to be sent

Feb 17 07:51:20 DHCPv6: dhcpv6 pkt parsing - Start
Feb 17 07:51:20 DHCPv6: --[     len == 116 ]--
Feb 17 07:51:20 DHCPv6: --[    from == fe80::215:c5ff:fecd:46d8/546 ]--
Feb 17 07:51:20 DHCPv6: --[ msgtype == DHCPV6_ADVERTISE ]--
Feb 17 07:51:20 DHCPv6: --[     xid == 29b633 ]--
Feb 17 07:51:20 DHCPv6: --[ OPTION_CLIENTID
Feb 17 07:51:20 DHCPv6:       OPTION code   1, len  14, data 00 01 00 01 16 cd cb 70
00 15 c5 cd 46 d8 ]--
Feb 17 07:51:20 DHCPv6: --[ OPTION_SERVERID
Feb 17 07:51:20 DHCPv6:       OPTION code   2, len  26, data 00 02 00 00 05 83 41 44
30 33 31 31 41 41 30 30 34 31 00 00 00 00 00 00 00 00 ]--
Feb 17 07:51:20 DHCPv6: --[ OPTION_IA_NA
Feb 17 07:51:20 DHCPv6:       OPTION code   3, len  40, iaid -976402728, T1 150, T2
240 ]--
Feb 17 07:51:20 DHCPv6: --[ OPTION_DNS_SERVERS
Feb 17 07:51:20 DHCPv6:       OPTION code  23, len  16, data fe c0 00 00 00 00 00 03
00 00 00 00 00 00 00 01 ]--
Feb 17 07:51:20 DHCPv6:     Parsing suboptions of OPTION_IA_NA - Start
Feb 17 07:51:20 DHCPv6: --[ OPTION_IAADDR
Feb 17 07:51:20 DHCPv6:         OPTION code   5, len  24, pre-ltime 300, valid-ltime
300, addr fec0:0:0:3::195, data NULL ]--
Feb 17 07:51:20 DHCPv6:     Parsing suboptions of OPTION_IA_NA - Done
Feb 17 07:51:20 DHCPv6: dhcpv6 pkt parsing - End
Feb 17 07:51:20 DHCPv6: OUT from fe80::5e5e:abff:fee0:7d88 to
fe80::215:c5ff:fecd:46d8/546 out interface 70 routing instance default len 116

Feb 17 07:51:20 DHCPv6: Got AUTH reply- no profile-
DHCPV6_LOCAL_SERVER_STATE_CLIENT_SELECTING, updating SDB with client session state
SDB_SESSION_STATE_CONFIGURED
Feb 17 07:51:20 jdhcpd_session_db_client_session_state_set: setting session state 1
Feb 17 07:51:20 jdhcpd_session_db_client_session_state_set: setting action bits to 3,
config-bits:0 0 0 0
Feb 17 07:51:20 libstats get iflstats clear: passed session_id 9
```

```
Feb 17 07:51:20 Request to create baseline stats for 9 succeeded
Feb 17 07:51:20 DHCPv6: dhcpv6_io_reader: received 52 bytes, dst addr ff02::1:2
Feb 17 07:51:20 DHCPv6: Recv pkt in default/default, ifindex 70, if vlan.3
Feb 17 07:51:20 DHCPv6: dhcpv6 pkt parsing - Start
Feb 17 07:51:20 DHCPv6: --[     len == 52 ]--
Feb 17 07:51:20 DHCPv6: --[    from == fe80::215:c5ff:fecd:46d8/546 ]--
Feb 17 07:51:20 DHCPv6: --[ msgtype == DHCPV6_SOLICIT ]--
Feb 17 07:51:20 DHCPv6: --[     xid == 29b633 ]--
Feb 17 07:51:20 DHCPv6: --[ OPTION_CLIENTID
Feb 17 07:51:20 DHCPv6:       OPTION code   1, len  14, data 00 01 00 01 16 cd cb 70
00 15 c5 cd 46 d8 ]--
Feb 17 07:51:20 DHCPv6: --[ OPTION_OPT_REQ
Feb 17 07:51:20 DHCPv6:       OPTION code   6, len   4, data 00 17 00 18 ]--
Feb 17 07:51:20 DHCPv6: --[ Internally Unsupported Option
Feb 17 07:51:20 DHCPv6:       OPTION code   8, len   2, data 00 68 ]--
Feb 17 07:51:20 DHCPv6: --[ OPTION_IA_NA
Feb 17 07:51:20 DHCPv6:       OPTION code   3, len  12, iaid -976402728, T1 3600, T2
5400 ]--
Feb 17 07:51:20 DHCPv6:     Parsing suboptions of OPTION_IA_NA - Start
Feb 17 07:51:20 DHCPv6:     Parsing suboptions of OPTION_IA_NA - Done
Feb 17 07:51:20 DHCPv6: dhcpv6 pkt parsing - End
Feb 17 07:51:20 DHCPv6: dhcpv6_packet_handle: Found client table entry
Feb 17 07:51:20 DHCPv6: RECEIVE: incomming_safd 0xdafa00, demux 0x0, safd 0xdafa00
Feb 17 07:51:20 Client 00-00-00-00-00-00 got event CLIENT_EVENT_SOLICIT_PDU in state
DHCPV6_LOCAL_SERVER_STATE_CLIENT_SELECTING
Feb 17 07:51:20 DHCPv6: TRYING to send local server packet, pool 0xdb6000
Feb 17 07:51:20 DHCPv6: DHCPV6_PDU_SEND_TYPE_ADVERTISE: will fill in options for len 8
Feb 17 07:51:20 DHCPv6: STARTING dhcpv6_option_encode_req_option, pool 0xdb6000,
num_req_opt 2
Feb 17 07:51:20 DHCPv6: dhcpv6_option_encode_req_option, pool 0xdb6000, num_req_opt 2
Feb 17 07:51:20 DHCPv6: requested option from pool V6-VLAN3, opt_code 23,
opt_ptr_from_code 23, opt_ptr_from_len 16
Feb 17 07:51:20 SEND: incomming_safd 0xdafa00, demux 0x0, safd 0xdafa00
Feb 17 07:51:20 DHCPv6:
Dump of local server V6 PDU to be sent

Feb 17 07:51:20 DHCPv6: dhcpv6 pkt parsing - Start
Feb 17 07:51:20 DHCPv6: --[     len == 116 ]--
Feb 17 07:51:20 DHCPv6: --[    from == fe80::215:c5ff:fecd:46d8/546 ]--
Feb 17 07:51:20 DHCPv6: --[ msgtype == DHCPV6_ADVERTISE ]--
Feb 17 07:51:20 DHCPv6: --[     xid == 29b633 ]--
Feb 17 07:51:20 DHCPv6: --[ OPTION_CLIENTID
Feb 17 07:51:20 DHCPv6:       OPTION code   1, len  14, data 00 01 00 01 16 cd cb 70
00 15 c5 cd 46 d8 ]--
Feb 17 07:51:20 DHCPv6: --[ OPTION_SERVERID
Feb 17 07:51:20 DHCPv6:       OPTION code   2, len  26, data 00 02 00 00 05 83 41 44
30 33 31 31 41 41 30 30 34 31 00 00 00 00 00 00 00 00 ]--
Feb 17 07:51:20 DHCPv6: --[ OPTION_IA_NA
Feb 17 07:51:20 DHCPv6:       OPTION code   3, len  40, iaid -976402728, T1 150, T2
240 ]--
Feb 17 07:51:20 DHCPv6: --[ OPTION_DNS_SERVERS
Feb 17 07:51:20 DHCPv6:       OPTION code  23, len  16, data fe c0 00 00 00 00 00 03
00 00 00 00 00 00 00 01 ]--
Feb 17 07:51:20 DHCPv6:     Parsing suboptions of OPTION_IA_NA - Start
Feb 17 07:51:20 DHCPv6: --[ OPTION_IAADDR
Feb 17 07:51:20 DHCPv6:         OPTION code   5, len  24, pre-ltime 300, valid-ltime
300, addr fec0:0:0:3::195, data NULL ]--
Feb 17 07:51:20 DHCPv6:     Parsing suboptions of OPTION_IA_NA - Done
Feb 17 07:51:20 DHCPv6: dhcpv6 pkt parsing - End
Feb 17 07:51:20 DHCPv6: OUT from fe80::5e5e:abff:fee0:7d88 to
fe80::215:c5ff:fecd:46d8/546 out interface 70 routing instance default len 116
```

```
Feb 17 07:51:23 DHCPv6: dhcpv6_io_reader: received 52 bytes, dst addr ff02::1:2
Feb 17 07:51:23 DHCPv6: Recv pkt in default/default, ifindex 70, if vlan.3
Feb 17 07:51:23 DHCPv6: dhcpv6 pkt parsing - Start
Feb 17 07:51:23 DHCPv6: --[     len == 52 ]--
Feb 17 07:51:23 DHCPv6: --[    from == fe80::215:c5ff:fecd:46d8/546 ]--
Feb 17 07:51:23 DHCPv6: --[ msgtype == DHCPV6_SOLICIT ]--
Feb 17 07:51:23 DHCPv6: --[     xid == 29b633 ]--
Feb 17 07:51:23 DHCPv6: --[ OPTION_CLIENTID
Feb 17 07:51:23 DHCPv6:       OPTION code   1, len  14, data 00 01 00 01 16 cd cb 70
00 15 c5 cd 46 d8 ]--
Feb 17 07:51:23 DHCPv6: --[ OPTION_OPT_REQ
Feb 17 07:51:23 DHCPv6:       OPTION code   6, len   4, data 00 17 00 18 ]--
Feb 17 07:51:23 DHCPv6: --[ Internally Unsupported Option
Feb 17 07:51:23 DHCPv6:       OPTION code   8, len   2, data 01 3f ]--
Feb 17 07:51:23 DHCPv6: --[ OPTION_IA_NA
Feb 17 07:51:23 DHCPv6:       OPTION code   3, len  12, iaid -976402728, T1 3600, T2
5400 ]--
Feb 17 07:51:23 DHCPv6:     Parsing suboptions of OPTION_IA_NA - Start
Feb 17 07:51:23 DHCPv6:     Parsing suboptions of OPTION_IA_NA - Done
Feb 17 07:51:23 DHCPv6: dhcpv6 pkt parsing - End
Feb 17 07:51:23 DHCPv6: dhcpv6_packet_handle: Found client table entry
Feb 17 07:51:23 DHCPv6: RECEIVE: incomming_safd 0xdafa00, demux 0x0, safd 0xdafa00
Feb 17 07:51:23 Client 00-00-00-00-00-00 got event CLIENT_EVENT_SOLICIT_PDU in state
DHCPV6_LOCAL_SERVER_STATE_CLIENT_SELECTING
Feb 17 07:51:23 DHCPv6: TRYING to send local server packet, pool 0xdb6000
Feb 17 07:51:23 DHCPv6: DHCPV6_PDU_SEND_TYPE_ADVERTISE: will fill in options for len 8
Feb 17 07:51:23 DHCPv6: STARTING dhcpv6_option_encode_req_option, pool 0xdb6000,
num_req_opt 2
Feb 17 07:51:23 DHCPv6: dhcpv6_option_encode_req_option, pool 0xdb6000, num_req_opt 2
Feb 17 07:51:23 DHCPv6: requested option from pool V6-VLAN3, opt_code 23,
opt_ptr_from_code 23, opt_ptr_from_len 16
Feb 17 07:51:23 SEND: incomming_safd 0xdafa00, demux 0x0, safd 0xdafa00
Feb 17 07:51:23 DHCPv6:
Dump of local server V6 PDU to be sent

Feb 17 07:51:23 DHCPv6: dhcpv6 pkt parsing - Start
Feb 17 07:51:23 DHCPv6: --[     len == 116 ]--
Feb 17 07:51:23 DHCPv6: --[    from == fe80::215:c5ff:fecd:46d8/546 ]--
Feb 17 07:51:23 DHCPv6: --[ msgtype == DHCPV6_ADVERTISE ]--
Feb 17 07:51:23 DHCPv6: --[     xid == 29b633 ]--
Feb 17 07:51:23 DHCPv6: --[ OPTION_CLIENTID
Feb 17 07:51:23 DHCPv6:       OPTION code   1, len  14, data 00 01 00 01 16 cd cb 70
00 15 c5 cd 46 d8 ]--
Feb 17 07:51:23 DHCPv6: --[ OPTION_SERVERID
Feb 17 07:51:23 DHCPv6:       OPTION code   2, len  26, data 00 02 00 00 05 83 41 44
30 33 31 31 41 41 30 30 34 31 00 00 00 00 00 00 00 00 ]--
Feb 17 07:51:23 DHCPv6: --[ OPTION_IA_NA
Feb 17 07:51:23 DHCPv6:       OPTION code   3, len  40, iaid -976402728, T1 150, T2
240 ]--
Feb 17 07:51:23 DHCPv6: --[ OPTION_DNS_SERVERS
Feb 17 07:51:23 DHCPv6:       OPTION code  23, len  16, data fe c0 00 00 00 00 00 03
00 00 00 00 00 00 00 01 ]--
Feb 17 07:51:23 DHCPv6:     Parsing suboptions of OPTION_IA_NA - Start
Feb 17 07:51:23 DHCPv6: --[ OPTION_IAADDR
Feb 17 07:51:23 DHCPv6:         OPTION code   5, len  24, pre-ltime 300, valid-ltime
300, addr fec0:0:0:3::195, data NULL ]--
Feb 17 07:51:23 DHCPv6:     Parsing suboptions of OPTION_IA_NA - Done
Feb 17 07:51:23 DHCPv6: dhcpv6 pkt parsing - End
Feb 17 07:51:23 DHCPv6: OUT from fe80::5e5e:abff:fee0:7d88 to
fe80::215:c5ff:fecd:46d8/546 out interface 70 routing instance default len 116

Feb 17 07:51:27 DHCPv6: dhcpv6_io_reader: received 52 bytes, dst addr ff02::1:2
Feb 17 07:51:27 DHCPv6: Recv pkt in default/default, ifindex 70, if vlan.3
```

```
Feb 17 07:51:27 DHCPv6: dhcpv6 pkt parsing - Start
Feb 17 07:51:27 DHCPv6: --[     len == 52 ]--
Feb 17 07:51:27 DHCPv6: --[    from == fe80::215:c5ff:fecd:46d8/546 ]--
Feb 17 07:51:27 DHCPv6: --[ msgtype == DHCPV6_SOLICIT ]--
Feb 17 07:51:27 DHCPv6: --[     xid == 29b633 ]--
Feb 17 07:51:27 DHCPv6: --[ OPTION_CLIENTID
Feb 17 07:51:27 DHCPv6:       OPTION code   1, len  14, data 00 01 00 01 16 cd cb 70
00 15 c5 cd 46 d8 ]--
Feb 17 07:51:27 DHCPv6: --[ OPTION_OPT_REQ
Feb 17 07:51:27 DHCPv6:       OPTION code   6, len   4, data 00 17 00 18 ]--
Feb 17 07:51:27 DHCPv6: --[ Internally Unsupported Option
Feb 17 07:51:27 DHCPv6:       OPTION code   8, len   2, data 03 01 ]--
Feb 17 07:51:27 DHCPv6: --[ OPTION_IA_NA
Feb 17 07:51:27 DHCPv6:       OPTION code   3, len  12, iaid -976402728, T1 3600, T2
5400 ]--
Feb 17 07:51:27 DHCPv6:     Parsing suboptions of OPTION_IA_NA - Start
Feb 17 07:51:27 DHCPv6:     Parsing suboptions of OPTION_IA_NA - Done
Feb 17 07:51:27 DHCPv6: dhcpv6 pkt parsing - End
Feb 17 07:51:27 DHCPv6: dhcpv6_packet_handle: Found client table entry
Feb 17 07:51:27 DHCPv6: RECEIVE: incomming_safd 0xdafa00, demux 0x0, safd 0xdafa00
Feb 17 07:51:27 Client 00-00-00-00-00-00 got event CLIENT_EVENT_SOLICIT_PDU in state
DHCPV6_LOCAL_SERVER_STATE_CLIENT_SELECTING
Feb 17 07:51:27 DHCPv6: TRYING to send local server packet, pool 0xdb6000
Feb 17 07:51:27 DHCPv6: DHCPV6_PDU_SEND_TYPE_ADVERTISE: will fill in options for len 8
Feb 17 07:51:27 DHCPv6: STARTING dhcpv6_option_encode_req_option, pool 0xdb6000,
num_req_opt 2
Feb 17 07:51:27 DHCPv6: dhcpv6_option_encode_req_option, pool 0xdb6000, num_req_opt 2
Feb 17 07:51:27 DHCPv6: requested option from pool V6-VLAN3, opt_code 23,
opt_ptr_from_code 23, opt_ptr_from_len 16
Feb 17 07:51:27 SEND: incomming_safd 0xdafa00, demux 0x0, safd 0xdafa00
Feb 17 07:51:27 DHCPv6:
Dump of local server V6 PDU to be sent

Feb 17 07:51:27 DHCPv6: dhcpv6 pkt parsing - Start
Feb 17 07:51:27 DHCPv6: --[     len == 116 ]--
Feb 17 07:51:27 DHCPv6: --[    from == fe80::215:c5ff:fecd:46d8/546 ]--
Feb 17 07:51:27 DHCPv6: --[ msgtype == DHCPV6_ADVERTISE ]--
Feb 17 07:51:27 DHCPv6: --[     xid == 29b633 ]--
Feb 17 07:51:27 DHCPv6: --[ OPTION_CLIENTID
Feb 17 07:51:27 DHCPv6:       OPTION code   1, len  14, data 00 01 00 01 16 cd cb 70
00 15 c5 cd 46 d8 ]--
Feb 17 07:51:27 DHCPv6: --[ OPTION_SERVERID
Feb 17 07:51:27 DHCPv6:       OPTION code   2, len  26, data 00 02 00 00 05 83 41 44
30 33 31 31 41 41 30 30 34 31 00 00 00 00 00 00 00 00 ]--
Feb 17 07:51:27 DHCPv6: --[ OPTION_IA_NA
Feb 17 07:51:27 DHCPv6:       OPTION code   3, len  40, iaid -976402728, T1 150, T2
240 ]--
Feb 17 07:51:27 DHCPv6: --[ OPTION_DNS_SERVERS
Feb 17 07:51:27 DHCPv6:       OPTION code  23, len  16, data fe c0 00 00 00 00 00 03
00 00 00 00 00 00 00 01 ]--
Feb 17 07:51:27 DHCPv6:     Parsing suboptions of OPTION_IA_NA - Start
Feb 17 07:51:27 DHCPv6: --[ OPTION_IAADDR
Feb 17 07:51:27 DHCPv6:         OPTION code   5, len  24, pre-ltime 300, valid-ltime
300, addr fec0:0:0:3::195, data NULL ]--
Feb 17 07:51:27 DHCPv6:     Parsing suboptions of OPTION_IA_NA - Done
Feb 17 07:51:27 DHCPv6: dhcpv6 pkt parsing - End
Feb 17 07:51:27 DHCPv6: OUT from fe80::5e5e:abff:fee0:7d88 to
fe80::215:c5ff:fecd:46d8/546 out interface 70 routing instance default len 116


bblackford@fw01.bchome>
```

7.0 Extending Beyond the Subnet Boundary

Suspending this case as well. Layer3 functionality was tested in above sections.