Contents

- 1 Introduction
- 2 Lab topology
- 3 VM setup
  - 3.1 CSR1000V VM
    - 3.1.1 CSR1000V configuration
  - 3.2 Ubuntu Server VM with DHCPv6 server
    - 3.2.1 DHCP server configuration
  - 3.3 OpenWRT CPE VM
- 4 Lab verification
- 5 How to make the DHCPv6 option value ?

Introduction

This example describes the configuration of the MAP-T KVM-based lab with OpenWRT CPE(s), ISC DHCPD as a DHCPv6 server, and a CSR1000V as a MAP-T BR.

For simplicity (in terms of required dependencies), we describe the setup using pure KVM on 64-bit Ubuntu Linux 12.02 (no libvirt), and using built-in Linux bridging for connectivity. Also, to limit the number of devices, this setup does not have the endpoints behind the CPE - adding that is left as an exercise to the reader.

The similar environments can be created using OpenStack / VMWare environments.

We assume that the reader is familiar with installation of Ubuntu Server as well as CSR1000V as VMs. Also we assume the familiarity with basic Linux tasks, like editing the files, restarting services, etc.

Please leave the feedback on how this document can be improved, hit "Leave a Comment" link - or drop an email to ayourtch at cisco dot com, so that I can improve this document or publish a follow-up!

Lab topology

We will precreate the interfaces which we will assign to the VMs later, and connect them to enable the connection of the CPE to the CSR1000V to DHCPv6 server.

First, let's disable IPv6 on the interfaces that do not need it:

```
sudo sysctl -w net.ipv6.conf.all.disable_ipv6=1
sudo sysctl -w net.ipv6.conf.eth0.disable_ipv6=0
```

Let's create the interfaces:

```
sudo tunctl -u $USER -t map-br0e0
sudo tunctl -u $USER -t map-br0e1
sudo tunctl -u $USER -t map-br0e2
sudo tunctl -u $USER -t map-br0e3
```

```
sudo tunctl -u $USER -t map-dhcpd0e0
sudo tunctl -u $USER -t map-dhcpd0e1

sudo tunctl -u $USER -t cpe1e0
sudo tunctl -u $USER -t cpe1e1
```

Now, the newly created interfaces need to be connected in a few bridged segments.

Let's create the bridges:

```
sudo brctl addbr cpe1-in
sudo ifconfig cpe1-in up promisc
sudo brctl addbr cpe-access
sudo ifconfig cpe-access up promisc
sudo brctl addbr servers
sudo ifconfig servers up promisc
```

Now we can put the respective interfaces into these bridges:

```
sudo ifconfig map-br0e2 up promisc
sudo ifconfig map-br0e3 up promisc
sudo brctl addif cpe-access map-br0e2
sudo brctl addif servers map-br0e3
sudo ifconfig map-dhcpd0e1 up promisc
sudo brctl addif servers map-dhcpd0e1
sudo ifconfig cpe1e0 up promisc
sudo ifconfig cpe1e1 up promisc
sudo brctl addif cpe-access cpe1e1
sudo brctl addif cpe1-in cpe1e0
```

As a result, we should have the following bridge configuration:

```
$ brctl show
bridge name   bridge id               STP enabled     interfaces
cpe-access            8000.22596a974f7c       no              cpe1e1
                                                              map-br0e2
cpe1-in               8000.e239236f45ce       no              cpe1e0
servers               8000.2e6886ad3ab5       no              map-br0e3
                                                              map-dhcpd0e1
```

VM setup

We will need three VMs:

CSR1000V VM

Download the CSR1000V IOS XE 3.13a ISO image from this URL:
http://software.cisco.com/download/release.html?mdfid=284364978&softwareid=282046477&release=3.13.0aS&relind=A

Save the file as "csr1000v-universalk9.03.13.00a.S.154-3.S0a-ext.iso"

Create an empty sparse disk image of 16G for the hard disk image:

```
dd if=/dev/zero of=map-br0-hda.img bs=1 count=1 seek=16G
```

Create the network interfaces for the router:

```
sudo tunctl -u $USER -t map-br0e0
sudo tunctl -u $USER -t map-br0e1
sudo tunctl -u $USER -t map-br0e2
sudo tunctl -u $USER -t map-br0e3
```

Launch the VM:

```
kvm  -net tap,vlan=0,ifname=map-br0e0,script=/bin/true \
     -net nic,vlan=0,macaddr=02:d4:98:5a:00:00 \
     -net tap,vlan=1,ifname=map-br0e1,script=/bin/true \
     -net nic,vlan=1,macaddr=02:d4:98:5a:00:01 \
     -net tap,vlan=2,ifname=map-br0e2,script=/bin/true \
     -net nic,vlan=2,macaddr=02:d4:98:5a:00:02 \
     -net tap,vlan=3,ifname=map-br0e3,script=/bin/true \
     -net nic,vlan=3,macaddr=02:d4:98:5a:00:03 \
     -vnc 127.0.0.1:34011 \
     -hda map-br0-hda.img \
     -cdrom csr1000v-universalk9.03.13.00a.S.154-3.S0a-ext.iso \
     -m 4G -daemonize
```

NB: the parameters specified here are intended for the lab testing only - for any other use, please consult the CSR1000V documentation on the resource usage!

Now, connect via VNC to the VM (display#34011), and follow the installation process. It should proceed automatically. If you need to connect from another host, the following command will allow to do it:

```
socat TCP-LISTEN:5900,reuseaddr TCP:localhost:39911 &
```

Note that socat uses TCP port number, which is for this case is 39911 = 34011+5900 (the base port for display 0).

CSR1000V configuration

The relevant part of the CSR1000V configuration is below:

```
!
hostname map-br
!
!
ipv6 unicast-routing
!
!
!
interface GigabitEthernet1
 description ipv6-only mgmt
 no ip address
 no shutdown
 ipv6 address autoconfig default
!
interface GigabitEthernet2
 description NAT44 outbound
 ip address dhcp
 no shutdown
!
interface GigabitEthernet3
 description CPE access
 no ip address
 no shutdown
 nat64 enable
```

```
 ipv6 address 2001:DB8:ACC3:55::1/64
 ipv6 nd managed-config-flag
 ipv6 nd other-config-flag
 ipv6 dhcp relay destination 2001:DB8:5555:5555:9B:1FF:FE82:1
!
interface GigabitEthernet4
 description DHCPv6 server and MAP-T outbound
 ip address 192.0.2.129 255.255.255.128
 no shutdown
 nat64 enable
 ipv6 address 2001:DB8:5555:5555::1/64
 ipv6 nd router-preference Low
!
!
!
nat64 map-t domain 1
 default-mapping-rule 2001:DB8:CAFE:CAFE::/64
 basic-mapping-rule
  ipv6-prefix 2001:DB8:FFFF:F000::/52
  ipv4-prefix 192.0.2.0/31
  port-parameters share-ratio 128 start-port 1024
!
```

Ubuntu Server VM with DHCPv6 server

Download the ISO image for the Ubuntu 14.04 server installation here:
http://www.ubuntu.com/download/server

Create the hard disk image for the VM:

```
dd if=/dev/zero of=map-dhcpd0-hda.img bs=1 count=1 seek=16G
```

Create the interfaces for the VM:

```
sudo tunctl -u $USER -t map-dhcpd0e0
sudo tunctl -u $USER -t map-dhcpd0e1
```

Bridge the map-dhcpd0e0 interface to the segment which has the internet access needed for Ubuntu installation. This can be IPv4-only, dual stack, or IPv6-only segment - the installation is possible using both pure IPv4 and pure IPv6.

Launch the VM:

```
kvm  -net tap,vlan=0,ifname=map-dhcpd0e0,script=/bin/true \
     -net nic,vlan=0,macaddr=02:9b:01:82:00:00 \
     -net tap,vlan=1,ifname=map-dhcpd0e1,script=/bin/true \
     -net nic,vlan=1,macaddr=02:9b:01:82:00:01 \
     -vnc 127.0.0.1:24801 \
     -hda map-dhcpd0-hda.img \
     -cdrom ubuntu-14.04.1-server-amd64.iso \
     -m 1G -daemonize
```

If you do not have a bridge to connect to, you can use the host networking for the installation:

```
kvm  -net user,vlan=0 \
     -net nic,vlan=0,macaddr=02:9b:01:82:00:00 \
     -net tap,vlan=1,ifname=map-dhcpd0e1,script=/bin/true \
     -net nic,vlan=1,macaddr=02:9b:01:82:00:01 \
```

```
      -vnc 127.0.0.1:24801 \
      -hda map-dhcpd0-hda.img \
      -cdrom ubuntu-14.04.1-server-amd64.iso \
      -m 1G -daemonize
```

Then, connect via the VNC to the host (screen 24801) and follow through the install process Like in the previous example, you can connect to VNC from another host using socat as a temporary bridge:

```
socat TCP-LISTEN:5900,reuseaddr TCP:localhost:30701 &
```

30701 = 24801+5900

For the installation use all the defaults, select "eth0" as the installation interface, hostname "map-dhcpd", partitioning method: "Guided, use entire disk", no automatic updates, Package selection: "OpenSSH server only", install GRUB to MBR.

After the successful install, update the default software and install the ISC DHCP by issuing the commands:

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install isc-dhcp-server
```

DHCP server configuration

The eth1 on the Ubuntu Server guest will be the working interface for the setup, and it did not get configured in the process of the install.

We will need to edit the file /etc/network/interfaces and add the following to it:

```
auto eth1
iface eth1 inet static
  address 192.0.2.130/25
  post-up ip route add 192.0.2.0/24 via 192.0.2.129
iface eth1 inet6 auto
```

We will also need to create a file /etc/dhcp/dhcpd6.conf with the following contents:

```
subnet6 2001:db8:5555:5555::/64 {
 range6 2001:db8:5555:5555::100 2001:db8:5555:5555::1000;
}

# For ISC DHCPD dhcpd6.conf (ensure the lines are copied in full!)
option dhcp6.map-option code 95 = string;

subnet6 2001:DB8:ACC3:55::/64 {
  range6 2001:DB8:ACC3:55::1 2001:DB8:ACC3:55::1000;
  prefix6 2001:DB8:FFFF:F000:: 2001:DB8:FFFF:FFF0:: /60;
  option dhcp6.map-option 00:59:00:0f:00:08:1f:c0:00:02:00:34:20:01:0d:b8:ff:ff:f0:00:5b:00:09:4
}
```

After saving this file, we need to restart the networking service and the DHCPv6 server using the following commands:

```
sudo ifup eth1
sudo service isc-dhcp-server6 restart
```

If all goes well we will see the eth1 take on the new configuration and the DHCPv6 process starting to listen on port 547 which is the DHCPv6 server port:

```
user@map-dhcpd:~$ ifconfig eth1
eth1      Link encap:Ethernet  HWaddr 02:9b:01:82:00:01
          inet addr:192.0.2.130  Bcast:192.0.2.255  Mask:255.255.255.128
          inet6 addr: 2001:db8:5555:5555:9b:1ff:fe82:1/64 Scope:Global
          inet6 addr: 2001:db8:5555:5555:60c6:856f:8124:c482/64 Scope:Global
          inet6 addr: fe80::9b:1ff:fe82:1/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:70 errors:0 dropped:13 overruns:0 frame:0
          TX packets:80 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:7205 (7.2 KB)  TX bytes:13811 (13.8 KB)

user@map-dhcpd:~$ netstat -6 -na
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp6       0      0 :::22                   :::*                    LISTEN
udp6       0      0 :::34746                :::*
udp6       0      0 :::547                  :::*
udp6       0      0 :::57470                :::*
user@map-dhcpd:~$
```

OpenWRT CPE VM

You will create the lab VM using the hard disk image from the following URL: https://downloads.openwrt.org/barrier_breaker/14.07/x86/kvm_guest/openwrt-x86-kvm_guest-combined-ext4.img.gz - which is a regular nightly build of OpenWRT Barrier Breaker.

Uncompress the hard disk image:

```
gzip -d openwrt-x86-kvm_guest-combined-ext4.img.gz
```

Create two tap interfaces: cpe1e0, cpe1e1

```
sudo tunctl -u $USER -t cpe1e0
sudo tunctl -u $USER -t cpe1e1
```

Before launching the CPE, we need to install the packages necessary to run MAP-T: they are part of the standard OpenWRT packages, but are not installed by default.

To do that, let's start the OpenWRT

```
kvm  -net tap,vlan=0,ifname=cpe1e0,script=/bin/true \
     -net nic,model=virtio,vlan=0,macaddr=02:90:00:13:01:00 \
     -net user,vlan=1 \
     -net nic,model=virtio,vlan=1,macaddr=02:90:00:13:01:01 \
     -vnc 0.0.0.0:23042 \
     -hda openwrt-x86-kvm_guest-combined-ext4.img
```

Connect the VNC viewer to TCP port 28942 and issue the following commands:

```
# Add the configuration for the IPv4-only WAN interface
```

```
uci set network.wan=interface
uci set network.wan.ifname=eth1
uci set network.wan.proto=dhcp
uci commit
# Restart the network
/etc/init.d/network restart
# Install the MAP-T package and dependencies
opkg update; opkg install map-t
# Turn off the image
poweroff
```

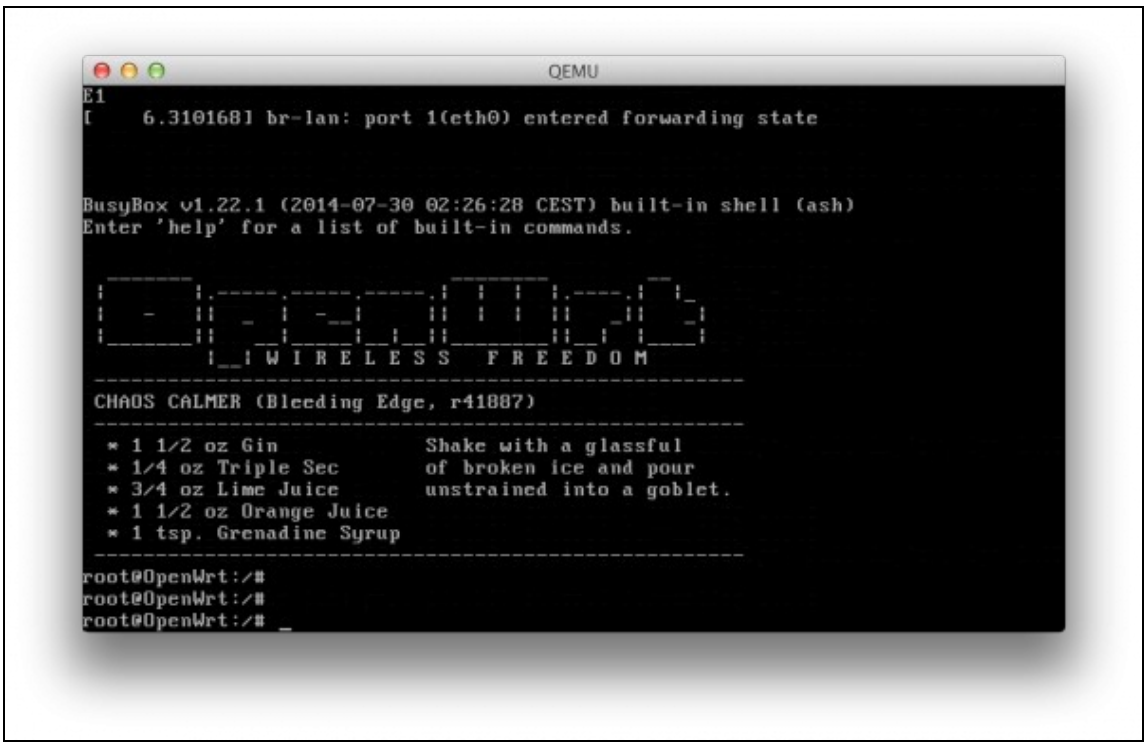
Now we can launch the VM:

```
kvm  -net tap,vlan=0,ifname=cpe1e0,script=/bin/true \
     -net nic,model=virtio,vlan=0,macaddr=02:90:00:13:01:00 \
     -net tap,vlan=1,ifname=cpe1e1,script=/bin/true \
     -net nic,model=vitrio,vlan=1,macaddr=02:90:00:13:01:01 \
     -vnc 127.0.0.1:23042 \
     -hda openwrt-x86-kvm_guest-combined-ext4.img  -daemonize
```

Again, if you need to connect from the remote host, here is the corresponding socat command to allow one connection:

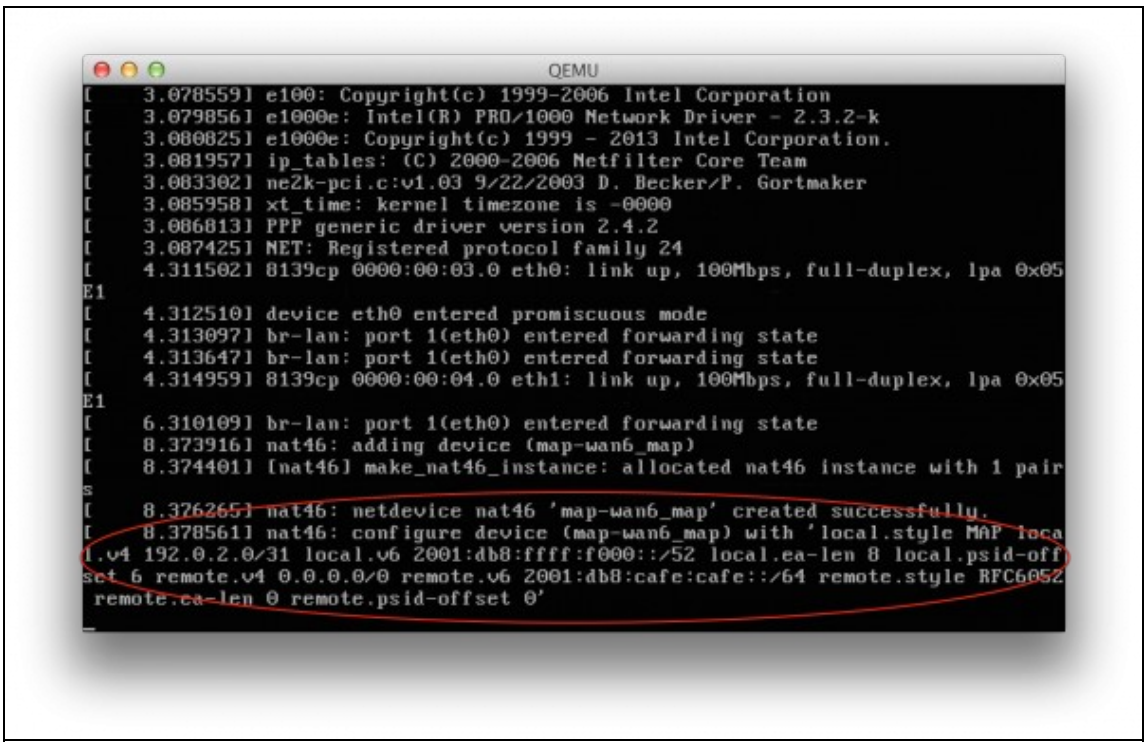
```
socat TCP-LISTEN:5900,reuseaddr TCP:localhost:28942 &
```

If in the VNC console you hit Enter key then you will be dropped into the root shell of the OpenWRT's busybox:



Lab verification

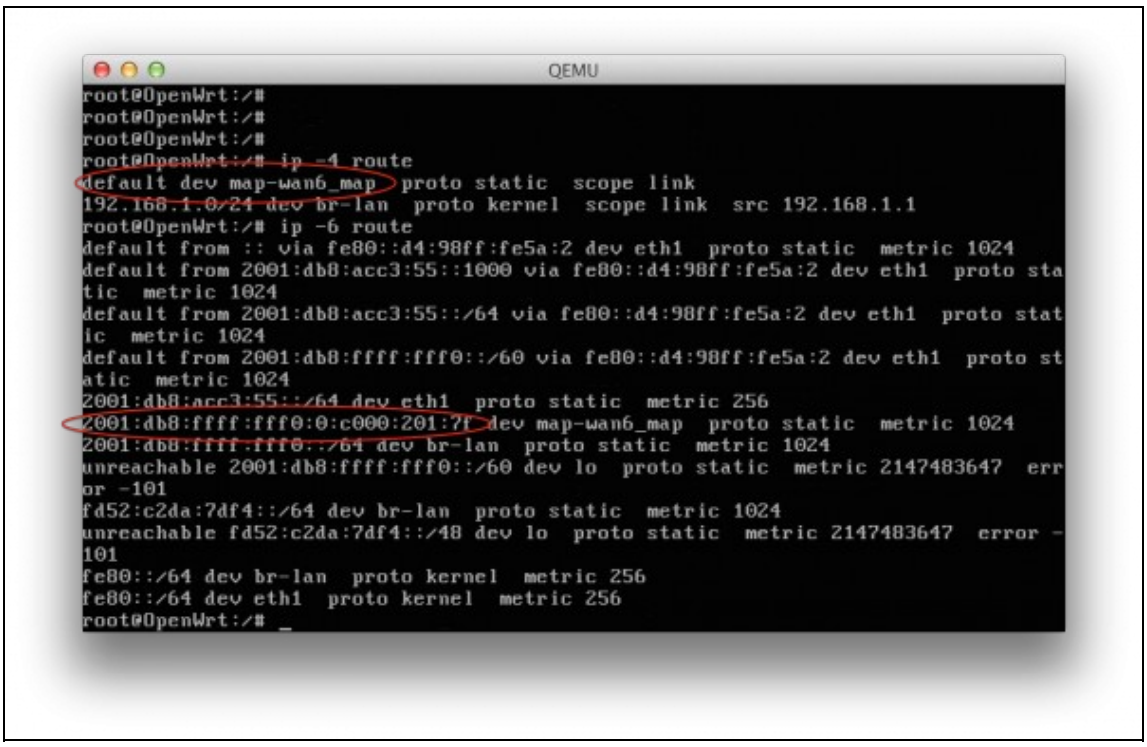
Type "reboot" at the OpenWRT root console and hit Enter. If you did everything correctly, then upon the reboot you will see some of the MAP-T parameters in the console messages:



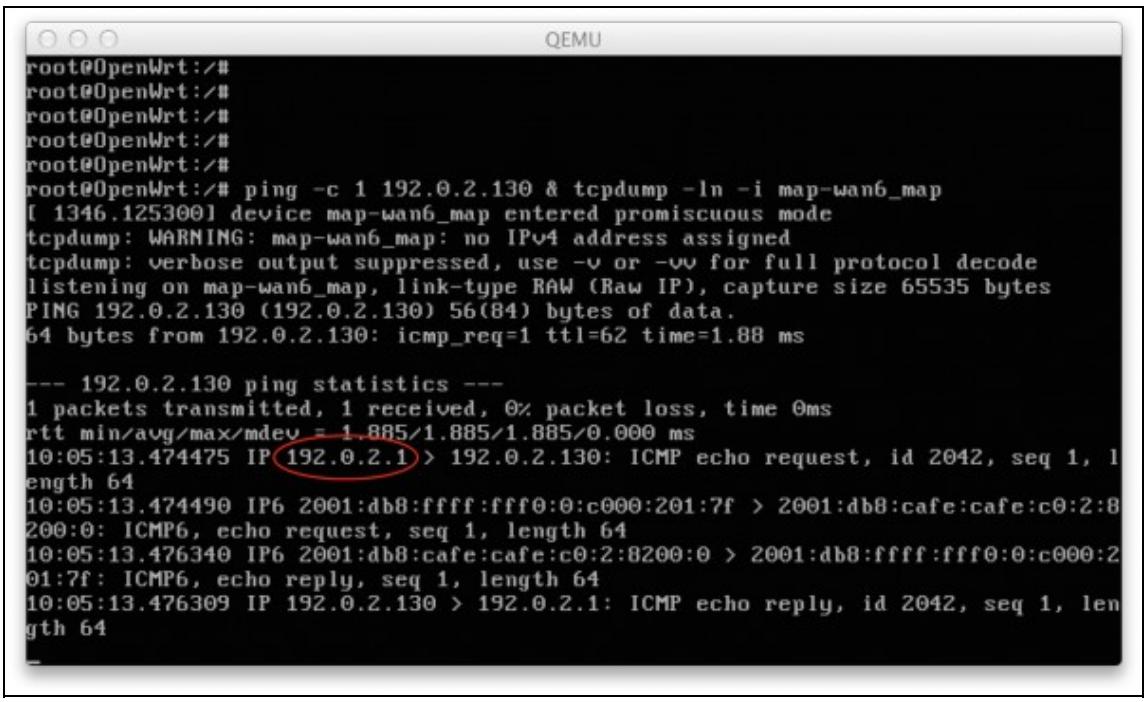
We can confirm that everything is working as expected, first of all by verifying the output of "ifstatus mapt":

if we use "ifstatus mapt | less" we can scroll through all the parameters of the interface, configured by the handler.

We can verify that the IPv4 routing and IPv6 routing are configured correctly to forward the inner packets before translation and outer packets on their return path to the mapt pseudointerface:



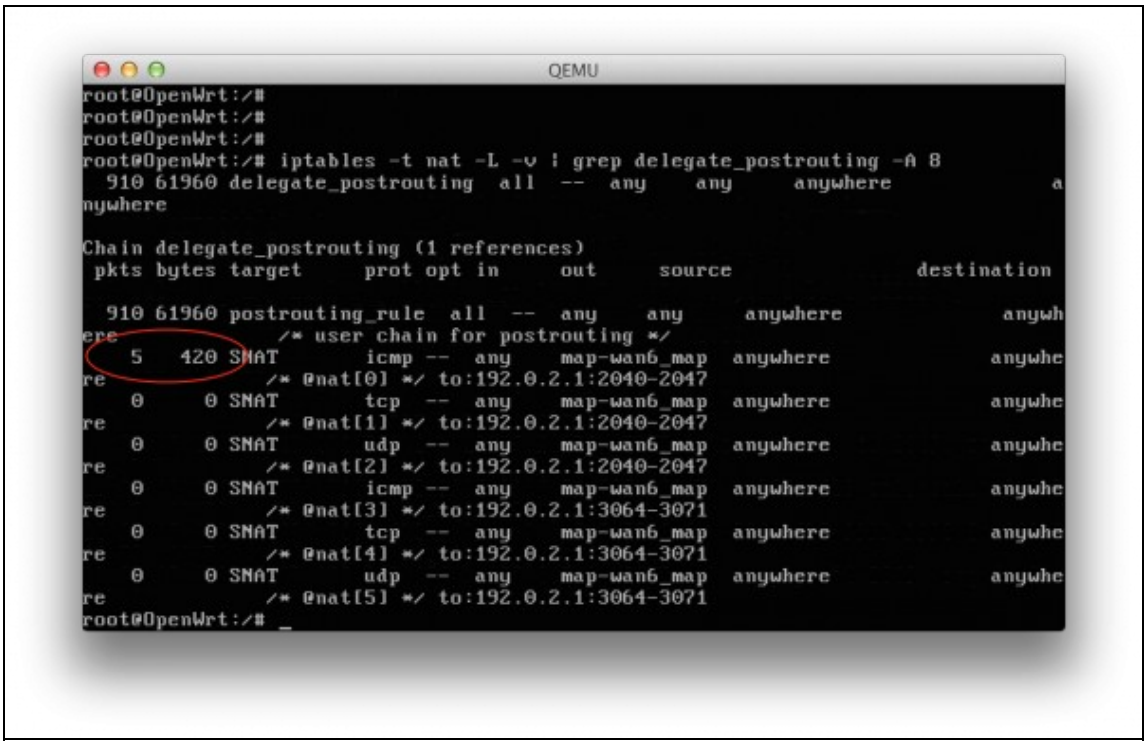
We can also launch a tcpdump on the map-mapt pseudointerface to verify the translation IPv4<->IPv6 is taking place correctly:



Note, that the source address of the IPv4 packets is 192.0.2.1, that is the MAP-T domain IPv4 address.

However, the original ping packets are sourced using the interface lan-br and have the RFC1918 source address, because this is the only IPv4 address available on the device:

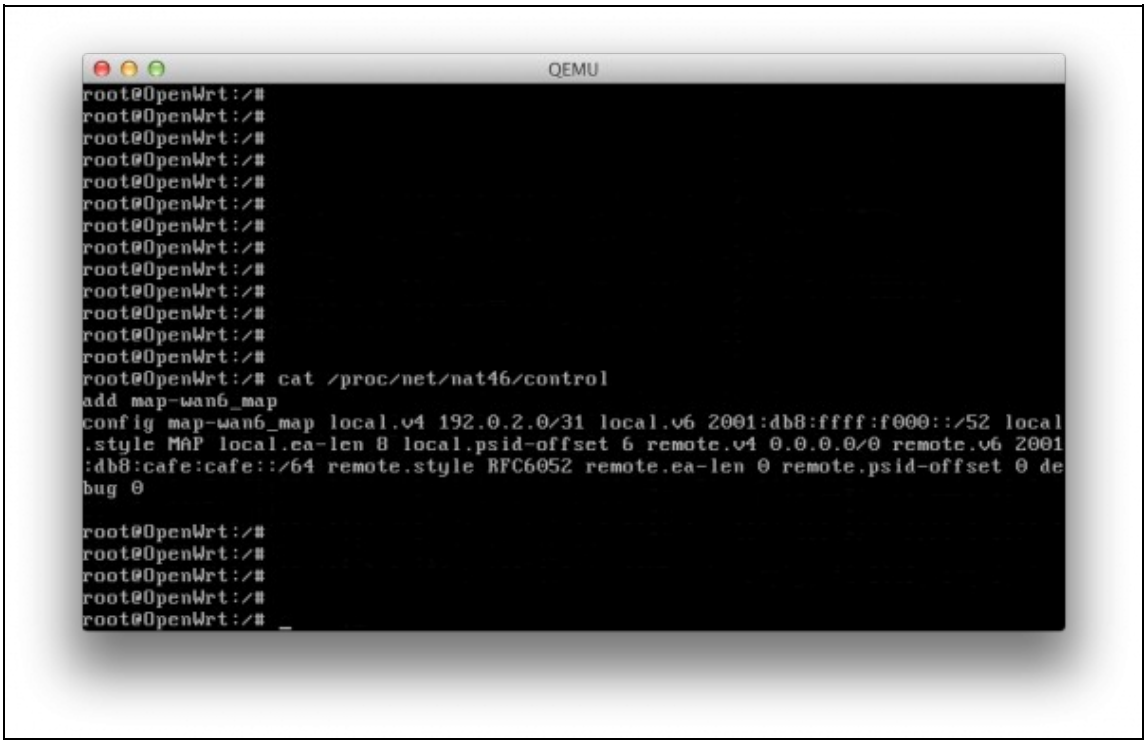
Because the default route points to map-mapt interface, the packets are forwarded out there, and they get caught by the iptables SNAT rules, taking care of the translation. You can see that using standard iptables counters:



after the translation is performed on the map-mapt interface, the IPv6 packets are forwarded out the eth1 interface, hit the BR, which translates them back into IPv4 and sends to the server. The server replies come back, get translated into IPv6 on the BR and send to the CPE - the CPE forwards them to map-mapt pseudointerface, which translates them into the IPv4 packets, which then get translated into the one with

private addresses using the iptables, and forwarded back to the originating stack.

You can verify the active parameters on the map-mapt interface by reading from /proc/net/nat46/control:



How to make the DHCPv6 option value ?

First, let's model the setup in the MAP calculator tool:
http://6lab.cisco.com/map/MAPnew.php?eyJydWxlcyI6W1siVGVzdCIsIjIwMDE6ZGI4OmZmZmY6ZjAwMC81MiIsIjE

In order to generate the DHCPv6 option, just click the "Generate DHCPv6 option value" button, and copy-paste the relevant portion of the text into your DHCPv6 server configuration.

You can also doublecheck the configurations by clicking the appropriate "Generate ..." buttons.