IPv6 Lab - DHCPv6-PD

Exploring DHCPv6 Prefix Delegation

In IPv4 there was no mechanism apart from email for an operator to let their customer know what IPv4 address block they could use. This isn't scalable, obviously, and needs weeks if not months notice for any changes to be made to the customer address space.

For IPv6, this problem is solved by the introduction of a prefix delegation function within DHCP. DHCP traditionally hands out a single IP address to a device requesting one - the new Prefix Delegation feature allows DHCP to hand out address blocks to requesting devices.

Preparation

Before we start this lab, remove any previous extra lab configuration on the Access and Customer routers and revert both routers to the configuration as it was at the end of the Static Lab. Check connectivity is working from the Customer router to the rest of the lab network as expected.

The following notes assume that changes need to be made from those required for the Static Lab.

Removing IPv6 on the link to the Customer

We need to remove any IPv6 configuration on the link between the Access Router and the Customer Router as it will be replaced by DHCPv6-PD. Here is an example for the Access Router:

```
interface FastEthernet0/1
 description P2P Link to CustX
 no ipv6 address 2001:DB8:X:20::0/127
 no ipv6 nd ra suppress all
!
no ipv6 route 2001:DB8:X:4000::/52 2001:DB8:X:20::1
!
```

and here is an example for the Customer Router:

```
interface FastEthernet0/1
 description P2P Link to ASX0
 no ipv6 address 2001:DB8:X:20::1/127
 no ipv6 nd ra suppress all
!
no ipv6 route 2001:DB8:X:4000::/52 Null0
no ipv6 route ::/0 2001:DB8:X:20::
!
```

Removing IPv6 configuration on Customer Router

And finally we need to remove the loopback interface from the Customer Router. Remember we created it as an anchor point for testing. Simply do:

```
no interface loopback 0
```

and it will be removed completely.

We leave `ipv6 unicast-routing` turned on though, as we still will need IPv6 on the router in the next part of the lab.

Setting up the Access Router

The lab instructors will have explained how DHCPv6-PD works during the presentations. But suffice to say, the router configuration on this CPE is the same across all CPE that would be deployed by the network operator – there is no per end user configuration which makes DHCPv6-PD a very simple to deploy technology, as all their CPE devices can have the same configuration prior to shipping to the end-user.

Right now the link between the ISP Access Router and the Customer router is a static set up. We have just removed the static IPv6 configurations in preparation for this lab, and we are now ready to deploy DHCPv6-PD.

Setting up the DHCPv6-PD pool

The first step is to set up the address pool to be used for DHCPv6-PD. We will be using the address block 2001:DB8:X:C000::/52 for DHCPv6-PD. All end users will be automatically assigned a /60 out of this /52. The following shows how to set up an IPv6 address pool in Cisco IOS:

```
ipv6 local pool dhcpv6-pool1 2001:DB8:X:C000::/52 60
!
ipv6 dhcp pool DHCPv6
 prefix-delegation pool dhcpv6-pool1 lifetime 1800 600
 dns-server 2001:DB8:FFFF::1
 domain-name lab.net
!
```

To explain these configuration commands:

- `ipv6 local pool dhcpv6-pool1` creates an IPv6 pool, using the 2001:DB8:X:C000::/52 address block, and handing out /60s on each request.
- `ipv6 dhcp pool DHCPv6` creates the DHCP pool configuration to be used on the interface connecting to customers.
- `prefix-delegation pool dhcpv6-pool1 lifetime 1800 600` says to use DHCPv6 Prefix Delegation, delegating prefixes from the pool called `dhcpv6-pool1` with a lifetime of `1800` seconds (30 minutes) and a preferred lifetime of `600` seconds (10 minutes).

- `dns-server` specifies a DNS resolver to use - this is just an example
- `domain-name` specifies the default domain-name for unqualified names - again this is just an example

Advertising the DHCPv6-PD pool to the rest of the network

Now that we have created the pool, we need to advertise it to the rest of the network. Doing this is very simple, and is why we set up iBGP earlier. Here is a configuration example:

```
ipv6 route 2001:DB8:X:C000::/52 null0
!
router bgp X0
 address-family ipv6
  network 2001:DB8:X:C000::/52
!
```

We point a static route to null for the whole address block being used by DHCPv6-PD (as the DHCPv6 configuration doesn't itself create any entries in the routing table).

And then we introduce the /52 into the network's iBGP in the usual fashion.

Activating the link to the Customer Router

With the pool created and announced by BGP to the rest of the network, we can now activate DHCPv6 towards the Customer Router. Here is an example of what would be configured on the Access Router:

```
interface FastEthernet0/1
 description P2P Link to CustX
 ipv6 enable
 ipv6 dhcp server DHCPv6 rapid-commit
!
```

What this does is enables IPv6 on the interface towards the Customer Router, and then assigns the prefix-delegation pool to hand out a prefix block to the router at the other end (when the router at the other end requests one). We do not need an IPv6 address on the point-to-point link - `ipv6 enable` command is sufficient to create a link-local address and turn on IPv6 processing on the link to the Customer Router.

Setting up the Customer Router

With the Access Router configuration now completed, we can turn our attention to the Customer router configuration.

We need to configure the point-to-point link to the Access Router to request an IPv6 address block by DHCPv6-PD. We will use a feature in Cisco IOS called `general-prefix`. This allows us to refer to learned addresses (by 6rd, DHCPv6PD etc) without configuring specific addresses on each prefix.

This following configuration example will do the needful:

```
interface FastEthernet0/1
 description P2P Link to ASX0
 ipv6 address autoconfig default
 ipv6 dhcp client pd PDLAB rapid-commit
!
```

Explaining these configuration lines:

- `ipv6 address autoconfig default` will set up the interface for auto configuration – this does two things, the first enabling IPv6 on the interface, and the second by setting IPv6 address depending on what is configured on the other end of the link. In our case it will result in just a link local address being used. The `default` keyword means that the router will install a default route on this interface to whatever the destination is at the other end of the link. This saves the requirement of configuring a static default route else where in the configuration.
- `ipv6 dhcp client pd PDLAB rapid-commit` says that the interface will operate in DHCPv6 client mode, using Prefix Delegation to learn the address block from the neighbouring router, and it will save the address block in `PDLAB`.
- `rapid-commit` simply speeds up the DHCPv6 process between client and server (2 messages are used rather than 4) and is generally recommended by most network operators.

Configuring Local Interfaces

Now we have the general prefix configured, we can use it to apply address to local interfaces. For example, FastEthernet0/0 on the edge/customer routers is a local LAN, and would get this configuration:

```
interface FastEthernet0/0
 description Local LAN
 ipv6 address PDLAB ::1:0:0:0:1/64
 no shutdown
!
```

For good measure, let’s recreate our Loopback interface too; this will restore our anchor point we can test against:

```
interface Loopback0
 description Customer Router Loopback
 ipv6 address PDLAB ::2:0:0:0:1/64
 no shutdown
!
```

And the resulting IPv6 addresses for the Customer router (using Group 6 as an example) would look like this:

```
Cust6# show ipv6 interface brief
FastEthernet0/0        [up/up]
    FE80::C81D:5FF:FEAF:8
    2001:DB8:6:C001::1
```

```
FastEthernet0/1        [up/up]
    FE80::C81D:5FF:FEAF:6
Loopback0              [up/up]
    FE80::C81D:5FF:FEAF:8
    2001:DB8:60:C002::1
```

Note that the DHCPv6 server on the Access Router has handed out the first subnet in the pool.

Refer to the earlier discussion about the “general-prefix”. We have `2001:DB8:6:C000::/60` as the IPv6 subnet created by 6rd. Taking this and appending `::1:0:0:0:0` results in the first subnet to be allocated to the network - the `FastEthernet0/0` interface there is automatically assigned `2001:DB8:6:C001::/64`, with the IPv6 address on the interface being `2001:DB8:6:C001::1/64`.

Testing

With the DHCPv6-PD now set up, trying some IPv4 and IPv6 traceroutes from the Customer router to the backbone Transit providers and out to the Internet. **Hint**: Use the Loopback interface as the source of your traceroutes and pings.

(Note that the lab doesn’t have IPv6 connectivity, but at least try IPv6 traceroutes into the Transit backbone and to other groups.)

What do you see?

Back to Agenda page

From:
https://www.bgp4all.com/pfs/ - **Philip Smith's Internet Development Site**

Permanent link:
https://www.bgp4all.com/pfs/training/sanog32/d-dhcpv6pd

Last update: **2018/08/17 08:00**