IPv6 Configuration on HP ProCurve Switches

Best Practice Document

Produced by CESNET led working group
on IPv6
(CBPD115)

Authors: Tomas Podermanski and Vladimir Zahorik
November 2010

Document No: GN3-NA3-T4-CBPD115
Version / date: November 2010
Original language: Czech
Original title: “Konfigurace IPv6 na přepínačích HP ProCurve”
Original version / date: 1.0 of 1 October 2010
Contact: tpoder@cis.vutbr.cz

CESNET bears responsibility for the content of this document. The work has been carried out by a CESNET led working group on IPv6 as part of a joint-venture project within the HE sector in the Czech Republic.

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 238875, relating to the project 'Multi-Gigabit European Research and Education Network and Associated Services (GN3)'.

Table of Contents

Executive Summary

New firmware for HP ProCurve switches was released on 15th November 2010. With this step, the manufacturer removed a significant shortcoming of the ProCurve switches – no full support for the IPv6 protocol. Partial IPv6 support was already introduced in earlier versions, but only for device management and filtering (ACL). Version K.15 brings IPv6 routing support in hardware with all features, including support of the OSPFv3 routing protocol. This firmware was released for the L3 switches series 54xx, 81xx – i.e., all switches with the "K" letter in their firmware name. The release number of the new version is 15 (K.15). The current document presents a detailed look at the implementation of IPv6 support. Giving examples, it will be shown that IPv6 configuration is not very complicated. Since for many people practical use of IPv6 is still unknown territory, some differences from IPv4 will be described in more detail below. Management and syntax of IPv6 commands copy the Cisco philosophy to a large degree. Yet, there are some small differences. The procedures below definitely do not represent all IPv6 possibilities in the K.15 firmware or IPv6 configuration possibilities, but are merely a manual to put IPv6 into production on these switches easily and quickly.

1 Setting Addresses on Interfaces

The first thing that must be done is to set an IPv6 address. The common IPv4 set-up was one address and a relevant subnet mask for each interface. The situation is slightly different for IPv6. First of all, each interface must be equipped with a ***Link-local address***. This address has only local significance and must be set automatically on each IPv6 interface immediately after the device is turned on. From the administrator's point of view, this process is fully automated. As far as configuration is concerned, it does therefore not require any special attention. The other generally used addresses are ***Global IPv6 addresses***. This type of address resembles more or less the addresses that we know from the IPv4 world. Most likely, the change of address length (to 128 bit) will not surprise anyone, but setting the ***prefix length*** for most cases to 64 bits is a new thing. In IPv4 terminology we used to refer to a subnet mask and the mask length. With IPv6, we are talking about prefix and prefix length.

As mentioned above, the Link-Local address is set up automatically. The global address configuration is done using an interface. Here, we have two options. Either the whole address, i.e,. both the network and the host part (host ID), can be set statically, or you can set the network part only and have the host part set by an EUI64 algorithm, based on the device's MAC address.

```
hp-test# configure
hp-test(config)# vlan 224
hp-test(vlan-224)# ipv6 address  2001:718:802:224::1/64
hp-test(vlan-224)# exit
hp-test(config)# vlan 225
hp-test(vlan-224)# ipv6 address  2001:718:802:225::0/64 eui-64
```

Just to be sure we can check the configuration

```
hp-test(vlan-225)# show ipv6

 Internet (IPv6) Service

  IPv6 Routing    : Enabled
  ND DAD          : Enabled
  DAD Attempts    : 3

  VLAN Name : DEFAULT_VLAN
  IPv6 Status     : Disabled
```

```
VLAN Name : VLAN224
IPv6 Status     : Enabled

Address    |                                          Address
Origin     | IPv6 Address/Prefix Length               Status
---------- + ---------------------------------------- -----------
manual     | 2001:718:802:224::1/64                   tentative
autoconfig | fe80::21d:b3ff:fe01:a700/64              tentative

VLAN Name : VLAN225
IPv6 Status     : Enabled

Address    |                                          Address
Origin     | IPv6 Address/Prefix Length               Status
---------- + ---------------------------------------- -----------
manual     | 2001:718:802:225:21d:b3ff:fe01:a700/64   tentative
autoconfig | fe80::21d:b3ff:fe01:a700/64              tentative
```

As shown in the listing, there are three different IPv6 addresses set on two IP interfaces (which are represented by VLAN). The first one is the address we set on the *2001:718:802:224* network. The first available address in the relevant network is used (with the number 1 in the host ID). The other address was created by the EUI64 algorithm. In this case, the network address is *2001:718:802:225* and the host ID is *21d:b3ff:fe01:a700*. The third address shown in the listing (fe80::21d:b3ff:fe01:a700) is a *Link-Local* address. Note that the *Link-Local* address has the same value on all interfaces. When working with this address, we must therefore add to this address after the % symbol the interface to which the relevant *Link-Local* address belongs (e.g., *fe80::21d:b3ff:fe01:a700%VLAN224*).

2 IPv6 Management

Using IPv6 for switch management will probably remain rather marginal for some time. The main reason for this is the effort to focus on providing native IPv6 (or dual stack) connectivity for servers and client systems. IPv6 support for management was included in the K.14 firmware release, but customers probably never used this feature on a large scale.

If you decide to keep using IPv4 for management, you must not forget that each configured IPv6 address automatically becomes an address that can be used to manage the switch. You can limit access by defining the mgmt-vlan option. But you cannot always afford to use this method. When configuring the first IPv6 address on the switch you should always set up limitations for access to component management. Use the following command to limit management to selected networks:

```
hp-test(config)# ipv6 authorized-managers 2001:718:802:228::0
ffff:ffff:ffff:ffff:: access manager
```

The IPv6 management can be restricted completely using the follow command:

```
hp-test(config)# ipv6 authorized-managers 0::
ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
```

Note that the net address mask is entered in a somewhat unusual format. Unfortunately, at this moment there is no way to enter the net mask using, for instance, the prefix length.

If you wish to manage the switch via IPv6 only, you can do it. Most settings will not present major issues – i.e., access through the SNMP protocol to the switch MIB, sending remote logs via syslog, time servers etc. The only problem is the definition of RADIUS servers. The current version does not permit entering IPv6 addresses. Thus, if you use 802.1X user authentication or authenticate the management access to the switch through a RADIUS server, you will need to keep at least one IPv4 address on the component for management purposes.

3 Client Configuration

In the previous sections, we carried out the first step that is required to run IPv6 on the interface. Now, we will find out how to set up an IPv6 address on the endpoint systems. We will skip the possibility of configuring a static address (which can of course be done) and focus on tools that will make the job easier for us.

The IPv6 protocol introduces new mechanisms to configure addresses for endpoint systems. The ***router advertisement*** (RA) protocol, a part of *Neighbour Discovery* (RFC 4861) is one of these. Each router will send information about network addresses that are configured on the router interfaces at regular intervals or upon request (*Router Solicitation*). These data are used by an endpoint system or device to set its own IPv6 address. It is a completely different approach than we were accustomed to in the IPv4 environment, where assigning IPv4 by DHCP was the common way to configure an IPv4 address.

If the routing functionality is enabled on the switch, then *Router Advertisement* is generated automatically and it includes all networks configured on the interface. But in some cases you could suppress the spread of RA. This can be done globally for the whole switch

```
hp-test(config)# ipv6 nd suppress-ra
```

or within the configuration of a given interface

```
hp-test(vlan-224)# ipv6 nd ra suppress
```

In practice we will probably use these commands very rarely. But the *MANAGED* and *OTHER* commands are vastly more important to configure *RA*. The *MANAGED* flag says that the device's IPv6 address and other parameters may be discovered in the given network through DHCPv6. The *OTHER* flag tells the client that it can use DHCPv6 only to obtain other parameters such as DNS server addresses, DNS suffixes etc. Setting the *MANAGED* flag automatically has a higher priority. If the *MANAGED* flag is set, setting the *OTHER* flag is meaningless. By default, both flags are turned off. They can be turned on with the following commands.

```
hp-test(vlan-224)# ipv6 nd ra managed-config-flag
hp-test(vlan-224)# ipv6 nd ra other-config-flag
```

Most likely, in practice these options, especially the *MANAGED* flag, will be used very often.

4 DHCPv6

It was already mentioned that in the IPv6 world DHCP support is not a necessary prerequisite to automatically configure a device, in contrast to what we are used to in the IPv4 world. The router advertisement (RA) mechanism mentioned above takes care of transferring data that are required to create the basic network connectivity. But in RA messages there is no way to provide other necessary data, such as DNS server addresses or search domain suffixes. These data can be received either via DHCP over IPv4 (with dual-stack support) or over DHCPv6. If no DHCPv6 server is connected directly to the given network you will have to use a remote DHCPv6 server and set up DHCPv6 relay on the switch. The set-up for DHCPv6 relay is very similar to the set-up of DHCPv4 relay. The configuration on the switch will be the same for stateful and for stateless configuration.

```
hp-test(vlan-224)# ipv6 helper-address unicast 2001:718:802:4::93e5:394
hp-test(vlan-224)# ipv6 helper-address unicast 2001:718:802:3::93e5:318
```

and to turn on the DHCPv6 relay support in the main configuration:

```
hp-test(config)# dhcpv6-relay
```

5 Neighbour Cache

Careful readers will have noticed that address assignment to endpoint systems is not managed centrally like we are used to with IPv4, where the DHCP server usually provides this service. In the case of IPv6, the endpoint system addresses are often randomly generated (RFC 4941), not influenced by an external authority. In many cases, in practical operation the relation between the communicating IPv6 address and the link-layer address (MAC address) will need to be known. With IPv4, this information was stored in an ARP table. With IPv6, the corresponding structure is called *neighbour cache*. The meaning and use are in principle the same as with an ARP table. You can list its contents with the following command:

```
hp-test# show ipv6 neighbours

 IPv6 ND Cache Entries

 IPv6 Address                              MAC Address   State Type    Port
 ----------------------------------------- ------------- ----- ------- ----
 ...
 2001:718:802:3:223:32ff:fe31:50d4         002332-3150d4 STALE dynamic 2
 2001:718:802:3:81f3:b2e7:f738:3bd8        000423-c915c4 STALE dynamic 2
 2001:718:802:3:915a:50d3:f16e:919a        000423-c915c4 STALE dynamic 2
 fe80::214:22ff:fe7b:8673%vlan223          001422-7b8673 STALE dynamic 23
 2001:718:802:80::1                        001ec1-daab81 STALE dynamic 4
 fe80::21e:c1ff:feda:ab81%vlan224          001ec1-daab81 STALE dynamic 4
 ...
```

As you see, *neighbour cache* contains records for all types of addresses, i.e. *Link-local* and global addresses. For the time being, browsing cached records is not very convenient: only VLAN ID is supported as a filtering option. Therefore, we must use some external tool for more advanced filtering or sorting. The neighbour cache records are also available through the MIB tree – as defined in RFC 4293 in the *ipNetToPhysicalTable*.

6 Unicast Routing

Having overcome all the hurdles of end network configuration, you can start the routing configuration. Routing support is activated with a single command:

```
hp-test(config)# ipv6 unicast-routing
```

It is obvious from the command that only unicast routing is activated this way. You would search in vain for a command to activate multicast routing. We must hope that support for multicast routing on the network layer including the related protocols (*PIM-SM, PIM-DM*) will be included in some future version.

The ***static routing*** configuration is also simple. In principle, record entry to the routing table is not different from the entry that is commonplace in the IPv4 world. The following command probably does not need further comments.

```
hp-test(config)# ipv6 route  2001:718:802:228::/64  2001:718:802:224::10
```

7 Routing Protocol - OSPFv3

The situation is slightly different for the configuration of the routing protocol. The components support the *OSPF* protocol, specifically its equivalent in the IPv6 world, i.e., *OSPFv3* (RFC 2740, 5340). The way in which the protocol works is largely similar to *OSPF*. The key change is the fact that communication between routers and the exchange of routing information are performed only over the *Link-Local* addresses. In practice, this means that global IPv6 addresses do not need to be configured on networks that interconnect OSPFv3 routers. The OSPFv3 interface configuration is simplified to allowing IPv6 on the given interface and assigning OSPFv3 area. The absence of a global IPv6 address on the interface causes some complications. Some diagnostic tools using the *ICMPv6* protocol, like *traceroute6* and *ping6,* cannot produce the proper information, because routers are not reachable by a global IPv6 address. This problem can be solved by setting up an IPv6 address on interconnecting networks. In that case, it is not important if the network is identical between routers. Only an arbitrary global address available from the rest of the network is necessary. Another option, very elegant in our opinion, is configuring a single global IPv6 address on the loopback interface of the L3 switch.

Some other parameters must also be set for OSPFv3. Most likely the need to configure an area will not surprise anyone. This is set in the same way as with OSPF – through a 32-bits identifier written in the form of four single byte numbers separated with dots. The value *0.0.0.0* is used to mark the *backbone area* just like in OSPF. With OSPFv3, you will certainly need to manually set the router ID parameter more often. It is a unique router identifier whose value is normally derived from the highest configured IPv4 address on the router. You did not have to deal with its configuration much in the IPv4 world, because the address was derived automatically. But if you want to have only IPv6 routing set up on a router, you need to set this parameter manually. The setting is done with a single command for the OSPF and the OSPFv3 routing process.

```
hp-test(config)# ip router-id 147.229.240.123
```

8 Playing with Multicast

Multicast support consists of two parts: link-layer support (multicast distribution optimisation) and support on the network layer (multicast routing). The first part includes mechanisms supporting effective distribution of multicast data. This mechanism was known as IGMP SNOOPING in the IPv4 world. The IGMP protocol is replaced with the *MLD* protocol (RFC2710 - Multicast Listener Discovery (MLD) for IPv6). The operation of this protocol is in principle identical to mechanisms known from *IGMPv2* and *IGMPv3* (RFC 2236, RFC 3376). The MLD protocol is automatically activated at the switch layer. When configuring, we will typically need to activate MLD on the IPv6 layer, i.e. VLAN:

```
hp-test(vlan-224)# ipv6 mld
```

Subsequently we can look at the connection status in individual groups with the following command:

```
hp-test(config)# show ipv6 mld vlan 224

  MLD Service Protocol Info

  VLAN ID : 310
  VLAN Name : list
  Querier Address : ::
  Querier Up Time : 0h:0m:0s
  Querier Expiry Time : 0h:0m:0s
  Ports with multicast routers :

  Active Group Addresses                  Type ExpiryTime Ports
  --------------------------------------- ---- ---------- --------------------
  ff02::c                                 FILT 0h:4m:20s  1
  ff02::1:3                               FILT 0h:4m:20s  1
  ff02::1:ff57:e0b2                       FILT 0h:4m:20s  1
  ff02::1:ffb5:2df1                       FILT 0h:4m:20s  1
  ff02::1:ffda:768d                       FILT 0h:4m:20s  1
```

Activating MLD snooping support is recommended as an automatic option for all VLANs.

The configuration mentioned above will provide an effective distribution of multicast operation within the local network. A logical subsequent step would be to activate the support of IPv6 multicast routing and an appropriate multicast routing protocol. But presently we would search in vain for such support. Multicast support on the network layer is planned for some future version.

9 Filtering – IPv6 Access Lists

If you start operating an IPv6 network, you will surely want to secure it in a suitable way. For this purpose, you can use an access-list-based packet filter on HP switches. Support for creating IPv6 access lists was included in the K.14 firmware release. The new version brings filtering support at the VLAN layer and routing support. The management is identical to creating access lists in the IPv4 environment.

First we must create a relevant access list in which we describe the filtering rules themselves:

```
hp-test(config)# ipv6 access-list "acl_1"
hp-test(config-ipv6-acl)# permit tcp any host 2001:718:802:4::93e5:394 eq 25
hp-test(config-ipv6-acl)# permit tcp host 2001:718:802:4::93e5:394 eq 25 any
hp-test(config-ipv6-acl)# deny tcp any any eq 25
hp-test(config-ipv6-acl)# permit ipv6 any any
```

The example describes a simple access list that blocks all SMTP traffic with the exception of the address 2001:718:802:4::93e5:394 which is the SMTP server. The access list created in this way must then be connected either to an interface (port):

```
hp-test(config-ipv6-acl)# interface a1
hp-test(eth-A1)# ipv6 access-group acl_1 in
```

or VLAN:

```
hp-test(vlan-223)# ipv6 access-group acl_1 in
hp-test(vlan-223)# ipv6 access-group acl_1 out
```

10 Conclusion

IPv6 support for components in the ProCurve series was released a bit later than with other manufacturers. You will need to wait a bit longer for support that provides all features, including multicast operation and various protection mechanisms. Despite small shortcomings, the implementation can be considered functional and it can be put into production on ordinary networks. The big advantage is that IPv6 support is released in the standard software release, which is available from on the ProCurve webpage, so that you do not have to pay anything extra to enable IPv6 features.