IPv6 Deployment

In Local Area Networks

April 2011

François Kooman <francois.kooman@surfnet.nl>

Revision 19

Table of Contents

1. Introduction

This document describes implementing IPv6 in an existing LAN of an organization. Specifically in a network with end-user devices with IP connectivity like desktop, notebooks and mobile devices.

The goal of this document to describe how to set up a network infrastructure with both IPv4 and IPv6 (dual stack[1]) connectivity.

This document will primarily focus on practical issues:

1. Required modifications to the network infrastructure;
2. Status of IPv6 support in common operating systems for a variety of devices.

As a case study the LAN of SURFnet was taken. This network contains various different types of devices which makes it a good test case to evaluate their IPv6 (dual stack) support.

It is assumed that IPv6 connectivity is already available on the router (at the WAN side) connecting to the LAN. This IPv6 connectivity can be obtained through either native IPv6 as provided by SURFnet[2] or through some tunneling mechanism like 6to4 or 6in4 if no native IPv6 connectivity is available yet.

2. Configuration of Devices

2.1. IPv4 Configuration

In IPv4 networks there are two ways to configure an IP address on a device:

1. Static configuration;
2. Dynamic configuration using DHCP[3].

To configure DNS resolvers the same two options are available:

1. Static configuration;
2. Dynamic configuration using DHCP.

The first option is typically used for “fixed” elements in the network like routers and servers. The second option is primarily used for the (automatic) configuration of devices belonging to end-users.

2.2. IPv6 Configuration

In IPv6 networks there is an extra way to configure IP addresses:

1. Static configuration;
2. Dynamic configuration using stateless address auto configuration (SLAAC);
3. Dynamic configuration using DHCPv6.

For DNS resolvers there are three options as well:

1. Static configuration;
2. Dynamic configuration using RDNSS;

1 With *dual stack* we mean the design of a network such that IPv6 connectivity is completely independent on IPv4 connectivity and that both IPv6 and IPv4 are fully active and can be automatically configured without any manual intervention of the user.
2 Native IPv6 can be obtained (for no additional fee) by customers of SURFnet as part of the service “SURFinternet”. More information about this can be obtained at the institute's account manager.
3 In this document DHCP always means DHCPv4. DHCPv6 is always mentioned explicitly.

3. Dynamic configuration using DHCPv6.

These methods are interchangeable. It is for instance possible to use SLAAC for IP address configuration and DHCPv6 for providing the DNS resolvers. In the next section the different ways of configuring IPv6 are discussed.

3. IPv6 Configuration

3.1. Static Configuration

Static configuration is typically used for "fixed" devices in the network like routers and servers. This is no different from the situation in IPv4 networks. The prefix of a network is by default `/64` in a LAN[4]. It is possible to use smaller networks, although that will make it impossible to use SLAAC to configure devices.

3.2. Dynamic Configuration using SLAAC

Stateless address auto configuration (SLAAC) is documented in RFC 4862. To acquire a globally unique IP address the router plays an important role. The router announces a prefix on a network using "router advertisements" (RA). The end-user host will choose a unique IP address in this prefix.

For IPv4 networks a prefix could be for example `192.168.1.0/24` which contains 2^8 (= 256) addresses. In IPv6 networks a prefix is usually 64 bits for a (V)LAN, for example `2001:610:508:109::/64`. This prefix contains 2^{64} (= a lot) of addresses.

For choosing an address in the IPv6 prefix usually the MAC address of the (network) interface is used. This is helpful in determining an address because the 48 bit MAC address is (by definition) globally unique. This however can be a potential privacy problem as the chosen address will be always the same (and leaks at least the vendor identification of the network device). For this reason recent versions of Windows (Vista and later) use the IPv6 privacy extensions as documented in RFC 4941. This prevents the IPv6 address from being always the same and to not correlate with the MAC address of the interface. In Mac OS X and Linux[5] it is possible to activate the privacy extensions manually. Mobile devices do not currently expose user configurable options to enable the privacy extension.

3.3. Dynamic Configuration using DHCPv6

In this case DHCPv6 is used as documented in RFC 3315. This means that, like in the IPv4 situation, the DHCPv6 server gives an address to an end-user host from a predetermined range. In the RA of the router the flag "*Managed address configuration*" should be set so the host knows that DHCPv6 is used for the address configuration (see RFC 2461).

It should be noted that it is not possible to announce the IP address of the router(s) using DHCPv6 as is possible in the case of DHCP, but it always has to be announced using RA.

3.4. IP Addressing Plan

To carefully design network addressing on a site, if for instance this site obtained a prefix of size /48, SURFnet wrote another document called "Preparing an IPv6 Addressing Plan"[6]. In this document methods are proposed of efficiently and meaningfully designing a addressing plan for a site.

---

4 The recommended minimal prefix is /64 is for a network (see RFC 4291). For a site (for instance an organization) a default of /48 is allocated (2^{16} /64 networks).

5 With Linux we actually mean GNU/Linux. This includes the Linux kernel and user space applications that are part of a Linux distribution like Red Hat Enterprise Linux or Debian.

6 This document can be obtained from the SURFnet website at http://www.surfnet.nl/Documents/handleiding_IPv6_nummerplan_EN.pdf.

4. IPv6 DNS Configuration

4.1. Static Configuration

Static configuration is typically used for “fixed” devices in the network like routers and servers. This is no different from the situation in IPv4 networks.

4.2. Dynamic Configuration using RDNSS

Recursive DNS Server (RDNSS) as documented in RFC 6106 “IPv6 Router Advertisement Options for DNS Configuration” is a method to announce addresses of DNS resolvers and search domains to end-user hosts. The addresses of the resolvers are added to the router advertisements.

4.3. Dynamic Configuration using DHCPv6

In this case DHCPv6 is used as documented in RFC 3315. Like with DHCP, with DHCPv6 the DNS resolver(s) and possibly other information can be announced. In the RA the flag “*Other stateful configuration*” should be set, so the host knows that it should use its DHCPv6 client to obtain this information (see RFC 2461).

5. RA, RDNSS or DHCPv6?

With these different ways to (automatically) configure IP addresses and DNS resolvers the question remains which one should be chosen. Different operating systems and (mobile) devices support a different (not necessarily overlapping) number of methods. So it will be “mix and match” to support all (or at least as much as possible) operating systems.

Operating System	**Version**	**SLAAC**	**RDNSS**	**DHCPv6**	**Privacy Extension**	**Manual**[7]	**Dual-stack**
Microsoft Windows[8]	7	Yes	No	Yes	Yes	**Yes**	**Yes**
Apple Mac OS X	10.6.5	Yes	No[9]	No	Yes	**Yes**	**No**
Apple iOS[10]	4.2.1	Yes	No	Yes	No	**No**	**Yes**
Google Android[11]	2.2	Yes	No	No	No	**No**	**No**
Linux (Ubuntu)[12]	10.04.1	Yes	Yes[13]	No	Yes	**Yes**	**No**
Linux (Fedora)	14	Yes	Yes	Yes	Yes	**Yes**	**Yes**[14]
Cisco IOS	15.x	Yes	No	Yes	No	**Yes**	**Yes**

7 This means whether or not it is possible to configure the operating system manually for dual stack connectivity.
8 When IPv4 is disabled manually in the network configuration a bug occurs with which the IPv6 address of the (automatically via DHCPv6) configured DNS resolver corrupts. See http://www.tunnelbroker.net/forums/index.php?topic=878.0 for more information.
9 Support for RDNSS will supposedly be available in Mac OS X 10.7, Lion. See http://seclists.org/nanog/2011/Feb/1923.
10 Tests with an iPhone 3GS and iOS 4.2.1 on an IPv6 only access point show that the iPhone is able to work without any IPv4 configuration. However, sometimes problems occurred during browsing the web in which sometimes a message is displayed that the server could not be found. A page refresh made it work again.
11 Issue with IPv6 support in Google Android: http://code.google.com/p/android/issues/detail?id=3389. This is (still) not resolved in Android 2.3 (Gingerbread).
12 Support of RDNSS and DHCPv6 is expected in Ubuntu 11.04 (available at the end of April 2011).
13 Requires the installation of the package “rdnssd”.
14 The default firewall blocks DHCPv6 responses (See: https://bugzilla.redhat.com/show_bug.cgi?id=656334 and http://www.redhat.com/archives/anaconda-devel-list/2010-November/msg00172.html). Furthermore, the IPv6 connection should be set to “Automatc” in NetworkManager to enable the automatic configuration of DNS resolver addresses using either RDNSS or DHCPv6.

Router advertisements are always required for every operating system. RDNSS is supported by only a few operating systems as is DHCPv6, but then a different set.

This table shows that DHCPv6 is really required for now[15]. RDNSS has limited use as only Fedora (and Ubuntu after installation of an optional software package) support it. However, RDNSS may become more important in the future as it will make DHCPv6 obsolete and will be needed for Mac OS X 10.7. Furthermore, it will be of great value after switching to secure neighbor discovery (see section 7.1).

It is noteworthy that it is currently impossible in Mac OS X, Google Android and Ubuntu to configure DNS resolvers automatically or configure IPv6 addresses using a DHCPv6 server. These systems are able to connect to IPv6 services, but they will always require the IPv4 DNS server that was obtained using DHCP (or IPv6 DNS resolver that was configured manually) for the resolving.

6. NAT64

Looking ahead towards a situation with an IPv6 only LAN it is almost certainly required to maintain connectivity to the IPv4 part of the Internet. This will be possible, for instance, using a NAT64/DNS64 gateway. This is a successor to the obsolete NAT-PT solution as documented in RFC 2766.

The idea behind NAT64 is that a DNS64 server creates a “virtual” AAAA record for host names if there currently is no AAAA record for that host. The virtual AAAA record points to a gateway machine in which the last 32 bits of the IP address encode the IPv4 address of the host. SURFnet has an experimental NAT64-gateway at IP address `2001:610:2001::610`[16]. By using this address as a DNS resolver (either in the DHCPv6, RNDSS configuration or manually specified) the NAT64 gateway will be used for targets that do not support IPv6.

Below an example of two target hosts is shown. In the case of `www.surfdiensten.nl` there is no AAAA record available and one is added by the DNS64 server. In the case of `www.surfnet.nl` there already is a AAAA record so it is not modified by the DNS64 server.

```
$ host www.surfdiensten.nl
www.surfdiensten.nl has address 194.171.53.6

$ host www.surfdiensten.nl 2001:610:2001::610
Using domain server:
Name: 2001:610:2001::610
Address: 2001:610:2001::610#53
Aliases:

www.surfdiensten.nl has address 194.171.53.6
www.surfdiensten.nl has IPv6 address 2001:610:2001:610::c2ab:3506

$ host www.surfnet.nl
www.surfnet.nl has address 194.171.26.203
www.surfnet.nl has IPv6 address 2001:610:1:80e1:194:171:26:203

$ host www.surfnet.nl 2001:610:2001::610
Using domain server:
Name: 2001:610:2001::610
Address: 2001:610:2001::610#53
Aliases:

www.surfnet.nl has address 194.171.26.203
www.surfnet.nl has IPv6 address 2001:610:1:80e1:194:171:26:203
```

15 As intermediate solution SLAAC could be used for IPv6 address configuration and DHCP for the (IPv4) DNS resolvers. Systems then will be able to communicate with IPv6 hosts, but it cannot be considered fully dual stack.

16 This experimental NAT64 gateway is only accessible from the SURFnet network.

The NAT64 gateway will take care of translating between IPv6 and IPv4. An implementation is available for Linux and OpenBSD and can be found in the Ecdysis project[17].

It should be noted that a NAT64 gateway has problems with some software, in particular software that passes IPv4 addresses as data inside the payload of the IP packets. Some examples of this are P2P software, SIP telephony, Skype and online games. An IETF document about IPv6 only experiences has some more information on this[18].

7. Securing IPv6 Networks

Just like with IPv4 networks, on IPv6 networks there can be both local attacks (by local users) or remote (perimeter) attacks. In this section only local networks are considered as the situation for securing the perimeter is similar to that of IPv4 networks and out of the scope in this document.

Common attacks on IPv4 LAN networks are ARP cache poisoning attacks and rogue DHCP servers. For IPv6 the situation is somewhat more complex.

7.1. Neighbor Discovery

In IPv6 networks ARP is replaced by neighbor discovery (ND). There are different kinds of ND packets, for example: router advertisements (RA) and duplicate address detection (DAD). While designing IPv6 the insecure LAN was not considered, no attacks were expected on the local “trusted” network. See for more information section 11 “Security Considerations“ of RFC 4861 and also RFC 3756 “IPv6 Neighbor Discovery (ND) Trust Models and Threats“.

The most important attack is probably a MITM attack[19] to redirect traffic. This can be done using rogue router advertisements as described in “IPv6 Router Advertisement Guard”[20] and “Rogue IPv6 Router Advertisement Problem Statement”[21]. The conclusion and solution according to this RFC:

> “While a number of the mitigations described above have their appeal, the simplest solutions probably lie in switch-based ACLs and RA-Guard style approaches. Where managed switches are not available, use of the Router Preference option and (more so in managed desktop environments) host firewalls may be appropriate.
>
> In the longer term wider experience of SeND will be beneficial, while the use of RA snooping will remain useful either to complement SeND (where a switch running RA Guard can potentially be a SeND proxy) or to assist in scenarios for which SeND is not deployed.”

SeND is documented in RFC 3971. It used public key cryptography to secure ND and a PKI for router discovery. Unfortunately there are no working implementation available that can be deployed right now, however, Cisco does have an implementation for their equipment.

The Cryptographically Generated Address (CGA) of SeND, documented in RFC 3972, works as follows: every station generates a public/private key and uses this to generate an IP address in a (published) network prefix using (secure) hashing. Thus it will be impossible to choose your own IP address, avoiding taking (over) an IP address for which the matching private key is not available. To verify RAs the advertisements would have to be signed by some trusted PKI root. Possibly this can be combined with RPKI router certification[22] (PKI for BGP) or DNSSEC.

Tools to experiment with for instance neighbor discovery and rogue router advertisements

17 See http://ecdysis.viagenie.ca/.
18 See http://tools.ietf.org/html/draft-arkko-ipv6-only-experience-00.
19 Man-In-The-Middle attack: in this attack traffic is (temporary) redirected through a machine controlled by an attacker to snoop, modify or block traffic without the user noticing this.
20 See http://datatracker.ietf.org/doc/draft-ietf-v6ops-ra-guard/.
21 See http://datatracker.ietf.org/doc/draft-ietf-v6ops-rogue-ra/.
22 See http://www.ripe.net/certification/.

can be found in the THC-IPV6-ATTACK-TOOLKIT[23].

To monitor ND the tools NDPMon[24] and SLAACer[25] are available. It will monitor (and optionally log) all (multicast) ND messages on the network and can be used to report on suspicious activity.

7.2. DHCPv6

In both IPv4 and IPv6 networks DHCP(v6) works using UDP. In the case of IPv4 via broadcast, in the case of IPv6 via multicast. For IPv4 this is documented in RFC 2131 and for IPv6 in RFC 3315.

A problem of DHCP(v6) is that it is possible to create a rogue DHCP(v6) server. Every system on the same (physical) LAN can do this. If the rogue DHCP server respond before the authoritative server responds this can result in a broken configuration or possibly MITM attacks. This can already happen by accident by enabling “Internet Connection Sharing” on some operating systems.

To solve this it is possible to block all DHCP(v6) responses coming from other hosts than the authoritative DHCP server in the LAN. This can be done on switches that support layer-3 filtering. See for example “IPv6 First Hop Security—Protecting Your IPv6 Access Network” written by Cisco[26].

For wireless networks, direct communication between hosts should be disabled, this can be done by enabling *access point isolation.*

To detect rogue DHCP(v6) servers if blocking them is not possible, can be done for DHCP wi the tool `dhcp_probe`[27]. It should not be too difficult to create a tool like this for DHCPv6, but so far implementations are not known.

7.3. Servers and Hosts

To secure end-user hosts and servers on the network IPv6 should be considered as well. Firewalls should work for both IPv4 and IPv6. In case this is forgotten it might be possible the server and end-user hosts are secured against attacks via IPv4, but open for attacks using IPv6. On Linux-systems `ip6tables` should be used as well as `iptables`. On Windows-systems the default firewall will also block IPv6 traffic the same way it blocks IPv4 traffic. However some third party Windows firewall solutions either block all IPv6 traffic or just let it pass through unfiltered. The first can result in broken connections, the second can result in (more) insecure systems as with IPv6 one usually gets a public IP address.

8. Identification of hosts

It is useful for network administrators to be able to trace problems with hosts in the network. In case a system is infected with a virus or configured in a wrong way and “attacks” other systems on the Internet. Sometimes it may even be necessary to trace an attack to a certain user as the MAC address of a network device can be manipulated and cannot be considered a unique mapping to a host or user.

8.1. IPv4

An approach used in IPv4 networks is creating a white list of network devices that are allowed to connect based on the MAC address of said device. In this case the user has to register their devices at the help desk so they can be linked to the owner so it is possible to find out who is

23 See http://freeworld.thc.org/thc-ipv6/ and the presentation slides during Chaos Computer Club 27C3 conference at http://www.youtube.com/watch?v=c7hq2q4jQYw (2010-12-27).
24 See http://NDPMon.sourceforge.net/.
25 See http://www.digriz.org.uk/slaacer.
26 See http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6553/whitepaper_c11-602135.html.
27 See http://www.net.princeton.edu/software/dhcp_probe/.

responsible for the device if problems occur. This typically only works if the user uses a fixed port on the wired LAN. If for instance students take all kinds of (mobile) devices to the university it will become very cumbersome to register them all.

In case 802.1X is used on the (wireless) network, linking a user to a device (MAC address) becomes much easier. Access points capable of 802.1X will usually send the IP address (and MAC address of the device) to the access point controller (or RADIUS server).

8.2. IPv6

Access point controllers we investigated, that support 802.1X, do not register the IPv6 address in the RADIUS log like they do for IPv4. This is because they do not perform DHCPv6 relaying like in the IPv4 situation. Of course, it will remain to be seen whether or not all networks deploying IPv6 will actually also deploy DHCPv6 as it is not strictly necessary, so it is unknown how this will be solved in the future.

However, if dual stack networks are created and a host gets both an IPv4 and IPv6 address it will be possible to use the link created for IPv4 (between MAC address and IPv4 address) to use this knowledge to determine the actual user of an IPv6 address by analyzing the neighbor discovery messages over the network(s). Tools that can do this are for example NDPMon and SLAACer.

9. Conclusion

To deploy “dual-stack” IPv6 networks in an organization and support as many as possible platforms it is necessary to deploy both stateless address auto configuration (SLAAC) and DHCPv6. DHCPv6 is then used only for passing the DNS resolvers to the hosts. Below a list of operating systems and whether or not they are “dual stack” capable. If the column “Dual stack” says “Yes” this means that the Internet connection remains working even when the IPv4 connectivity is completely dropped. If it says “No” that means an IPv4 connection is still required to be able to communicate with IPv6 services. In this table manual configuration was not considered as that does not scale in organizations with lots of hosts on the network.

Operating System	**Version**	**Dual stack**
Microsoft Windows	7	**Yes**
Apple Mac OS X	10.6.5	No
Apple iOS	4.2.1	**Yes**
Google Android	2.2	No
Linux (Ubuntu)	10.04.1	No
Linux (Fedora)	14	**Yes**
Cisco IOS	15.x	**Yes**

Right now, the SURFnet office network is a real dual stack network. If at some point it is decided to complete disable IPv4 on the LAN, the dual stack operating systems as shown in the table able will keep working. The list will be more positive in the near future, except in the case of Android where no (short term) plan is known for improving IPv6 support.