NAME

ndproxy — Neighbor Discovery Proxy

SYNOPSIS

ndproxy is a kernel module that implements IPv6 Neighbor Discovery proxying over Ethernet-like access networks, with many options to handle several use-cases.

ndproxy replies to a neighbor solicitation with a specific neighbor advertisement, in order to let the PE uplink router send further packets to a CPE downlink router, that may or may not be the same node that run ndproxy.

The hook-based `pfil`(9) framework is used to let ndproxy be invoked for every IPv6 incoming packet, in order to specifically handle and filter neighbor solicitations and reply with appropriate neighbor advertisements.

ND (Neighbor Discovery) packets are mainly targeted at solicited-node multicast addresses, but ndproxy has no information about the hosts to proxy, then it can not join the corresponding groups. Thus, the interface on which ndproxy listen to solicitations must be put into permanently promiscuous mode: add "promisc" to the ifconfig_<interface> variable in `rc.conf`(5).

For the same reason, MLD snooping must be disabled on the switches that share the PE/CPE interconnect (the layer-2 link the listening interface is attached to). Note that MLD snooping must not be disabled entirely on each switch, but only on the corresponding vlan.

The interface on which ndproxy listen to solicitations only need to be assigned a link-local address. No information about the delegated prefix and no global address are needed on this interface. It is sufficient to add "inet6 -ifdisabled -accept_rtadv auto_linklocal" to the ifconfig_<interface>_ipv6 variable in `rc.conf`(5).

DIFFERENCES WITH NDP

The target address to proxy must be given when using the `ndp`(8) command-line tool with the proxy option. On the contrary, ndproxy does not rely on a list of target addresses to proxy. Thus, RFC-4941 temporary addresses can be proxyfied. For security reasons, many operating systems use a temporary address when establishing outbound connections.

When using `ndp`(8) command-line tool with the proxy option, the proxyfied packets are redirected to the node that run ndp. With ndproxy, the host that run ndp can be used only to redirect packets to another IPv6 internal router, for instance a dedicated router with hardware support of IPv6 routing process.

PREFIX SUBNETTING

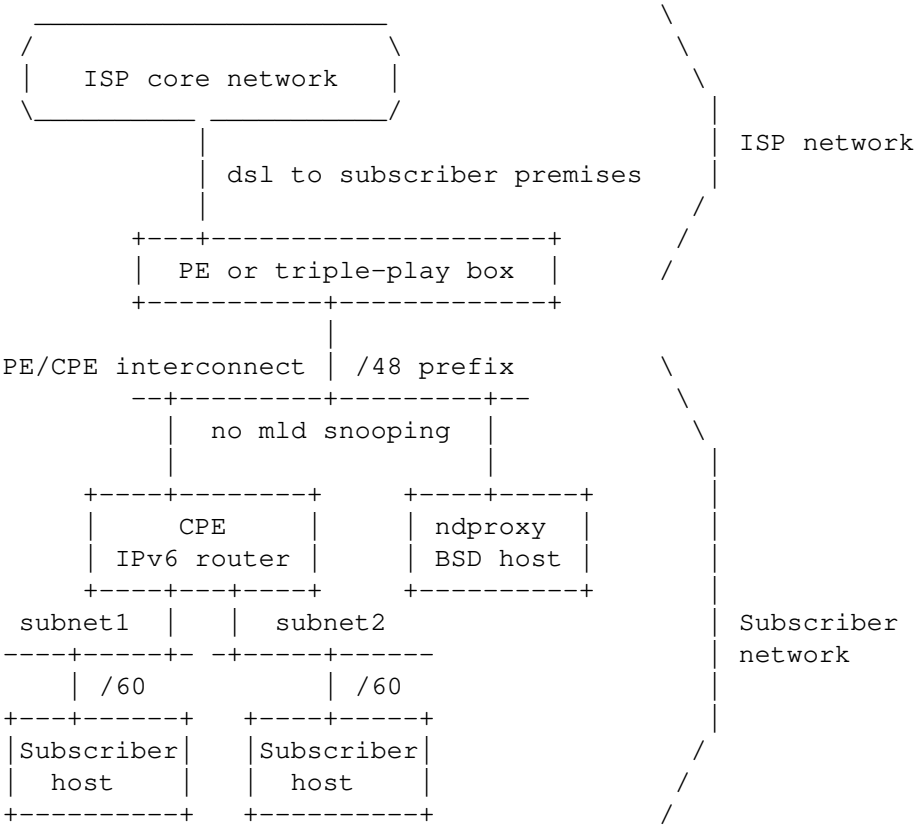
Connecting a flat IPv6 network to the Internet is easily done with the RFC-4861 ND protocol. But connecting a subnetted IPv6 prefix is more complicated, depending on the ISP network design choices. ndproxy can help subscribers to achieve this goal.

Here are some protocols or mechanisms the ISP need to support, when the delegated prefix must be subnetted and assigned to multiple links within the subscriber's network. For instance, the ISP could learn routes from the subscriber router using an IGP routing protocol, but the ISP and the subscriber must agree with a common routing protocol. The ISP could also feed the PE with a static route to the CPE router, but the ISP must be informed about the subscriber router address. Finally, the ISP could use the RFC-3633 IPv6 Prefix Options with DHCPv6 to delegate the prefix from its PE router to a requesting subscriber's router: in such a case, the ISP must support the DHCPv6 option.

ndproxy has been written for subscribers to ISP that do not support any of those mechanisms or protocols, thus not being able to natively subnet their IPv6 delegated prefix.

NETWORK DESIGN

Here is a generic network design using ndproxy to solve such situations:



Note that many other use-cases can be handled with ndproxy: the BSD host and the CPE router can be the same node, the delegated-prefix length can be /64, the PE router can have several interfaces on the ISP/Subscriber layer-2 boundary, there can be multiple PE routers, etc.

PREFIX LENGTH

Even if the IESG and the IAB first recommended the allocations of /48 prefixes in the general case, for the boundary between the public and the private topology (see RFC-3177), and that some Regional Internet Registries (APNIC, ARIN and RIPE) have subsequently revised the end site assignment policy to encourage the assignment of /56 blocks to end sites, and that RFC-6177 finally recommended giving home sites significantly more than a single /64, in order for home sites to be given multiple subnets, some ISP currently only delegate /64 prefixes.

In such a case, the subscriber should subnet a RFC-4193 Unique Local IPv6 Unicast Addresses prefix to the internal subnetworks, for internal-to-internal communications. The /64 global prefix should be routed to the only internal subnet in which RFC-4941 temporary addresses are used by hosts when establishing outbound connections. Static routes on the CPE router should be set to let hosts on other internal subnets be able to communicate with the Internet. Using temporary addresses for outbound connections to the Internet must be disabled on hosts on those other internal subnets.

IPv6 EXTENSION HEADERS

For security reasons, ndproxy explicitely rejects neighbor solicitation packets containing any extension header. Such a packet is mainly unattended:

Fragmentation:

According to RFC-6980, IPv6 fragmentation header is forbidden in all neighbor discovery messages.

Hop-by-hop header:

commonly used for jumbograms or for MLD. Should not involve neighbor solicitation packets.

Destination mobility headers:

commonly used for mobility, ndproxy does not support these headers.

Routing header:

commonly used for mobility or source routing, ndproxy does not support these headers.

AH & ESP headers:

securing the neighbor discovery process is not done with IPsec but with the SEcure Neighbor Discovery protocol (RFC-3971). ndproxy can not support RFC-3971, since proxifying ND packets is some kind of a spoofing process.

EXCEPTION ADDRESSES

Some neigbhor solicitations sent on the PE/CPE interconnect must not be proxyfied:

1. solicitations sent by other nodes than the PE;
2. solicitations sent by the PE to reach any on-link address (the address filled in the target address option) owned by nodes attached to the PE/CPE interconnect, for instance to reach the CPE, the ndproxy host or other hosts attached to this layer-2 interconnect.

The target addresses filled in those solicitations that ndproxy must ignore have to be declared via sysctl (net.inet6.ndproxyconf_exception_ipv6_addresses). This list must contain the link-local and global-scoped unicast and anycast addresses of the CPE, of the ndproxy host and of any other host than the PE attached to the PE/CPE interconnect.

Failing to maintain this list correctly could lead to badly redirect some packets to the CPE, but with a simple network design, this list can be let empty.

UPLINK ROUTER ADDRESSES

ndproxy only handles packets originating from one of the PE addresses. During its address resolution process, different source addresses can be choosen by the PE, depending on the packet that triggered the process or depending on other external constraints.

Here are some cases when it can occur:

1. The PE may have multiple interfaces;
2. There may be multiple PE;
3. Many routers choose to use a link-local address when sending neighbor solicitations, but when an administrator of such a router, also having a global address assigned on the same link, tries to send packets (echo request, for instance) to an on-link destination global address, the source address of the echo request packet prompting the solicitation may be global-scoped according to the selection algorithm described in RFC-6724. Therefore, the source address of the Neighbor Solicitation packet should also be selected in the same global scope, according to RFC-4861;

4. When the uplink router does not yet know its own address, it must use the unspecified address, according to RFC-4861.

So, it can not be assumed that an uplink router will always use the same IPv6 address to send neighbor solicitations. Each assigned address that can be used as a source address by the PE on its downlink interface must then be declared to ndproxy via sysctl (net.inet6.ndproxyconf_uplink_ipv6_addresses).

ndproxy will only handle packets that come from one of these addresses.

A special care must be taken about the unsolicited address. It may be used by the PE, then it is part of the list of PE addresses and should therefore be added to the list of PE addresses. Since this address can also be used by other nodes during some initialization steps (for instance when hot-swapping an Ethernet board), another node could use this address to send neighbor solicitations that ndproxy should not handle, because they are not sent by the PE. In fact, this is not a problem because the target address option contained in a solicitation from this other node should be in the exception list. So, adding the unsolicited address in the PE addresses list should be safe.

Failing to maintain this list correctly could lead the PE not to be able to establish outbound connections to nodes on the PE/CPE interconnect, but if this list contains at least the PE link-local address, IPv6 connectivity should be correctly established between the Internet and the internal subscriber's subnets.

CONFIGURATION

An IPv6 address can be any valid textual representation according to RFC-4291 and RFC-5952 (this means that transitional textual representation is fully supported). Other representations will trigger an error event. IPv6 address lists must be formated as series of IPv6 adresses separated by semi-colons.

The sysctl utility or `rc.conf`(5) are used to set ndproxy configuration parameters.

If you have installed ndproxy as a port or as a package, set the following variables in `rc.conf`(5) and load the module at boot time by placing the following line in `rc.conf`(5):

```
ndproxy_enable="YES"
```

On the contrary, if you have NOT installed ndproxy as a port or as a package but as a standalone distribution, place the sysctl entries in `sysctl.conf`(5) and load the module at boot time by placing the following line in `loader.conf`(5):

```
ndproxy_load="YES"
```

net.inet6.ndproxyconf_uplink_interface sysctl entry or ndproxy_uplink_interface rc.conf variable:

Name of the interface talking to the broadcast multi-access network connecting the PE and CPE routers.

Example: "vlan2".

net.inet6.ndproxyconf_downlink_mac_address sysctl entry or ndproxy_downlink_mac_address rc.conf variable:

MAC address of the CPE router. Neighbor advertisements sent by ndproxy will be filled with this address in the target link-layer address option. The format of this parameter is the hexadecimal representation made of 6 groups of 2 hexadecimal numbers separated by colons.

Example: "00:0C:29:B6:43:D5".

net.inet6.ndproxyconf_exception_ipv6_addresses sysctl entry or ndproxy_exception_ipv6_addresses rc.conf variable:

Target addresses not to proxy. In a simple network design, this list can be let empty. See section "EXCEPTION ADDRESSES".

Example: "fe80::20d:edff:fe7b:68b7;fe80::222:15ff:fe3b:59a".

net.inet6.ndproxyconf_uplink_ipv6_addresses sysctl entry or ndproxy_uplink_ipv6_addresses rc.conf variable:

Addresses of the PE. This list should at least contain the PE link-local address. See section "UPLINK ROUTER ADDRESSES".

Example: "fe80::207:cbff:fe4b:2d20;2a01:e35:8aae:bc60::1;::".

net.inet6.ndproxycount sysctl entry:

Number of advertisements sent.

SEE ALSO

`inet6`(4), `rc.conf`(5), `loader.conf`(5), `sysctl.conf`(5), `sysctl`(8), `loader`(8), `pfil`(9)

AUTHOR

Alexandre Fenyo <alex@fenyo.net> - www.fenyo.net