RFC 8106 : IPv6 Router Advertisement Options for DNS Configuration

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 18 mars 2017

Date de publication du RFC : Mars 2017

https://www.bortzmeyer.org/8106.html

Il existe deux méthodes pour configurer une machine IPv6 automatiquement, DHCP (RFC 8415[1]) et RA ("*Router Advertisement*", RFC 4862). Toutes les deux peuvent indiquer d'autres informations que l'adresse IP, comme par exemple les adresses des résolveurs DNS. Notre RFC normalise cette possibilité pour les RA. Il remplace le RFC 6106, avec peu de changements.

Si on gère un gros réseau, avec de nombreuses machines dont certaines, portables, vont et viennent, s'assurer que toutes ces machines ont les adresses IP des serveurs de noms à utiliser n'est pas trivial (section 1 du RFC). On ne peut évidemment pas utiliser le DNS, cela serait tenter de voler en tirant sur les lacets de ses chaussures. Et configurer à la main les adresses sur chaque machine (par exemple, sur Unix, en les écrivant dans le fichier `/etc/resolv.conf`) est bien trop difficile à maintenir. Se passer du DNS est hors de question. Pour les machines bi-protocoles (IPv4 et IPv6), une solution possible était d'utiliser un serveur de noms en v4. Mais pour une solution purement v6 ?

La solution la plus populaire était DHCP (RFC 8415 et RFC 3646). Son principal inconvénient est qu'elle est **à état** : le serveur DHCP doit se souvenir des **baux** qu'il a attribué. Sur un gros réseau local, le nombre de requêtes à traiter, chacune nécessitant une écriture dans une base de données, peut devenir très lourd.

Une autre solution est **sans état** et repose sur une nouveauté d'IPv6, les RA ("*Router Advertisements*", cette méthode est aussi appelée ND, pour "*Neighbor Discovery*", les RA en étant un cas particulier), décrits dans le RFC 4862. Ce sont des messages envoyés à intervalles réguliers par les routeurs et qui informent les machines non-routeuses des caractéristiques essentielles du réseau, comme le préfixe utilisé

1. Pour voir le RFC de numéro NNN, `https://www.ietf.org/rfc/rfcNNN.txt`, par exemple `https://www.ietf.org/rfc/rfc8415.txt`

(par exemple 2001:db8:beef:42::/64). Le routeur diffuse ses messages et n'a pas besoin d'écrire quoi que ce soit sur son disque, ni de faire des traitements compliqués lors d'une sollicitation, il répond toujours par le même message RA.

Ces RA peuvent diffuser diverses informations, par le biais d'un système d'options. Le principe de notre RFC est donc d'utiliser ces RA pour transporter l'information sur les serveurs de noms récursifs utilisables sur le réseau local, via des options notamment celle nommée RDNSS (le numéro 25 lui a été affecté par l'IANA).

La section 1.1 du RFC rappelle qu'il existe plusieurs choix, notre RFC 8106 n'étant qu'une possibilité parmi d'autres. Le RFC 4339 contient une discussion plus détaillée de ce problème du choix d'une méthode de configuration des serveurs de noms (notons qu'il existe d'autres méthodes comme l'"*anycast*" avec une adresse « bien connue »). La section 1.2 décrit ce qui se passe lorsque plusieurs méthodes (par exemple DHCP et RA) sont utilisées en même temps.

La méthode RA décrite dans notre RFC repose sur deux options, RDNSS, déjà citée, et DNSSL (section 4). La première permet de publier les adresses des serveurs de noms, la seconde une liste de domaine à utiliser pour compléter les noms courts (formés d'un seul composant). Les valeurs pour ces deux options doivent être configurées dans le routeur qui va lancer les RA. (Le routeur Turris Omnia <https://www.bortzmeyer.org/turris.html> le fait automatiquement <https://forum.turris.cz/t/how-to-configure-rdnss/2358>. Si on veut changer les paramètres, voici comment faire <https://forum.turris.cz/t/how-to-configure-ipv6-ra/3584>. En général, pour OpenWrt, il faut lire cette documentation <https://github.com/openwrt/odhcpd/blob/master/README>, l'ancien logiciel radvd n'étant plus utilisé.)

La première option, RDNSS, de numéro 25, est décrite en section 5.1. Elle indique une liste d'adresse IPv6 que le client RA mettra dans sa liste locale de serveurs de noms interrogeables.

La seconde option, DNSSL, de numéro 31, est en section 5.2 (les deux options sont enregistrées dans le registre IANA <https://www.iana.org/assignments/icmpv6-parameters/icmpv6-parameters.xml#icmpv6-parameters-5>, cf. section 8). Elle publie une liste de domaines, typiquement ceux qui, sur une machine Unix, se retrouveront dans l'option search de /etc/resolv.conf.

Sur Linux, le démon rdnssd <http://rdnssd.linkfanel.net/> permet de recevoir ces RA et de modifier la configuration DNS. Pour FreeBSD, on peut consulter une discussion sur leur liste <http://lists.freebsd.org/pipermail/freebsd-net/2009-June/022248.html>. Les CPE de Free, les Freebox, émettent de telles options dans leurs RA (apparemment, la dernière fois que j'ai regardé, uniquement des RDNSS). Voici ce qu'affiche Wireshark :

```
...
Ethernet II, Src: FreeboxS_c3:83:23 (00:07:cb:c3:83:23),
              Dst: IPv6mcast_00:00:00:01 (33:33:00:00:00:01)
...
Internet Control Message Protocol v6
    Type: 134 (Router advertisement)
...
    ICMPv6 Option (Recursive DNS Server)
        Type: Recursive DNS Server (25)
        Length: 40
        Reserved
        Lifetime: 600
        Recursive DNS Servers: 2a01:e00::2 (2a01:e00::2)
        Recursive DNS Servers: 2a01:e00::1 (2a01:e00::1)
```

et les serveurs DNS annoncés répondent correctement. (Vous pouvez récupérer le paquet entier sur pcapr.net <http://www.pcapr.net/view/bortzmeyer+pcapr/2009/10/6/9/v6-fb.pcap.html>.)

Autre mise en œuvre de ces options, dans radvd (ainsi que pour les logiciels auxiliaires <http://lists.litech.org/pipermail/radvd-devel-l/2010-December/000528.html>). Wireshark, on l'a vu, sait décoder ces options.

La section 6 de notre RFC donne des conseils aux programmeurs qui voudraient mettre en œuvre ce document. Par exemple, sur un système d'exploitation où le client RA tourne dans le noyau (pour configurer les adresses IP) et où la configuration DNS est dans l'espace utilisateur, il faut prévoir un mécanisme de communication, par exemple un démon qui interroge le noyau régulièrement pour savoir s'il doit mettre à jour la configuration DNS.

RA pose divers problèmes de sécurité, tout comme DHCP, d'ailleurs. Le problème de ces techniques est qu'elles sont conçues pour faciliter la vue de l'utilisateur et de l'administrateur réseau et que « faciliter la vie » implique en général de ne pas avoir de fonctions de sécurité difficiles à configurer. La section 7 traite de ce problème, par exemple du risque de se retrouver avec l'adresse d'un serveur DNS méchant qui vous redirigerait Dieu sait où (les RA ne sont pas authentifiés). Ce risque n'a rien de spécifique aux options DNS, toute la technique RA est vulnérable (par exemple, avec un faux "*Neighbor Advertisement*"). Donc, notre RFC n'apporte pas de risque nouveau (cf. RFC 6104). Si on considère cette faiblesse de sécurité comme insupportable, la section 7.2 recommande d'utiliser le "*RA guard*" du RFC 6105, ou bien SEND (RFC 3971, mais il est nettement moins mis en avant que dans le précédent RFC).

Ce problème d'une auto-configuration simple des machines connectées à IPv6 est évidemment particulièrement important pour les objets connectés et c'est sans doute pour cela que le RFC contient la mention « "*This document was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) [10041244, Smart TV 2.0 Software Platform]*" ».

Les changements faits depuis le précédent RFC, le RFC 6106, figurent dans l'annexe A. On y trouve notamment :

— Une valeur par défaut plus élevée pour la durée de vie des informations envoyées (qui passe de deux fois `MaxRtrAdvInterval` à trois fois sa valeur, soit 1 800 secondes avec la valeur par défaut de cette variable), pour diminuer le nombre de cas où l'information expire parce que le réseau perdait trop de paquets,
— L'autorisation explicite des adresses locales au lien (celles en `fe80::/10`), comme adresses de résolveurs DNS,
— Suppression de la limite de trois résolveurs DNS, qui était dans l'ancien RFC.

À noter que ce RFC n'intègre pas encore les résolveurs sécurisés du RFC 7858, car il se contente de réviser un RFC existant. Il n'y a donc pas de moyen de spécifier un résolveur sécurisé, pas de port 853.

Et pour finir, voici le RA émis par défaut par le routeur Turris <https://www.bortzmeyer.org/turris.html>, décodé par Wireshark :

```
Internet Protocol Version 6, Src: fe80::da58:d7ff:fe00:4c9e, Dst: ff02::1
    0110 .... = Version: 6
    .... 0000 0000 .... .... .... .... .... = Traffic class: 0x00 (DSCP: CS0, ECN: Not-ECT)
        .... 0000 00.. .... .... .... .... .... = Differentiated Services Codepoint: Default (0)
        .... .... ..00 .... .... .... .... .... = Explicit Congestion Notification: Not ECN-Capable Transport (0
    .... .... .... 0101 1110 1011 0100 0001 = Flow label: 0x5eb41
    Payload length: 152
    Next header: ICMPv6 (58)
    Hop limit: 255
    Source: fe80::da58:d7ff:fe00:4c9e
```

```
    [Source SA MAC: CzNicZSP_00:4c:9e (d8:58:d7:00:4c:9e)]
    Destination: ff02::1
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
Internet Control Message Protocol v6
    Type: Router Advertisement (134)
    Code: 0
    Checksum: 0x35ed [correct]
    [Checksum Status: Good]
    Cur hop limit: 64
    Flags: 0x80
        1... .... = Managed address configuration: Set
        .0.. .... = Other configuration: Not set
        ..0. .... = Home Agent: Not set
        ...0 0... = Prf (Default Router Preference): Medium (0)
        .... .0.. = Proxy: Not set
        .... ..0. = Reserved: 0
    Router lifetime (s): 1800
    Reachable time (ms): 0
    Retrans timer (ms): 0
    ICMPv6 Option (Source link-layer address : d8:58:d7:00:4c:9e)
        Type: Source link-layer address (1)
        Length: 1 (8 bytes)
        Link-layer address: CzNicZSP_00:4c:9e (d8:58:d7:00:4c:9e)
    ICMPv6 Option (MTU : 1480)
        Type: MTU (5)
        Length: 1 (8 bytes)
        Reserved
        MTU: 1480
    ICMPv6 Option (Prefix information : fde8:9fa9:1aba::/64)
        Type: Prefix information (3)
        Length: 4 (32 bytes)
        Prefix Length: 64
        Flag: 0xc0
            1... .... = On-link flag(L): Set
            .1.. .... = Autonomous address-configuration flag(A): Set
            ..0. .... = Router address flag(R): Not set
            ...0 0000 = Reserved: 0
        Valid Lifetime: 7200
        Preferred Lifetime: 1800
        Reserved
        Prefix: fde8:9fa9:1aba::
    ICMPv6 Option (Prefix information : 2a01:e35:8bd9:8bb0::/64)
        Type: Prefix information (3)
        Length: 4 (32 bytes)
        Prefix Length: 64
        Flag: 0xc0
            1... .... = On-link flag(L): Set
            .1.. .... = Autonomous address-configuration flag(A): Set
            ..0. .... = Router address flag(R): Not set
            ...0 0000 = Reserved: 0
        Valid Lifetime: 7200
        Preferred Lifetime: 1800
        Reserved
        Prefix: 2a01:e35:8bd9:8bb0::
    ICMPv6 Option (Route Information : Medium fde8:9fa9:1aba::/48)
        Type: Route Information (24)
        Length: 3 (24 bytes)
        Prefix Length: 48
        Flag: 0x00
            ...0 0... = Route Preference: Medium (0)
            000. .000 = Reserved: 0
        Route Lifetime: 7200
        Prefix: fde8:9fa9:1aba::
    ICMPv6 Option (Recursive DNS Server fde8:9fa9:1aba::1)
        Type: Recursive DNS Server (25)
        Length: 3 (24 bytes)
        Reserved
        Lifetime: 1800
```

```
    Recursive DNS Servers: fde8:9fa9:1aba::1
ICMPv6 Option (Advertisement Interval : 600000)
    Type: Advertisement Interval (7)
    Length: 1 (8 bytes)
    Reserved
    Advertisement Interval: 600000
```

On y voit l'option RDNSS (l'avant-dernière) mais pas de DNSSL.

Merci à Alexis La Goutte pour ses informations.