Pas d'IPv6 sans multicast

Matthieu Herrb

Pas d’IPv6 sans Multicast

Matthieu Herrb
LAAS – CNRS – Université de Toulouse
7 avenue du Colonel Roche
BP 54200
31031 Toulouse Cedex 4

Résumé

Le protocole IPv6 n’utilise plus de broadcasts mais uniquement des paquets multicast pour diverses raisons, il est notamment question d’efficacité pour des réseaux locaux de grande taille, et aussi de limitation de la fuite d’informations. Qu’en est-il en pratique ?

Cet article décortique l’utilisation du multicast en IPv6 : comment il est utilisé pour différents composants : découverte des voisins, auto-configuration… Il donne une vue d’ensemble de ce qu’il se passe en termes de multicast lorsque qu’un nœud rejoint un réseau IPv6, qu’il ouvre une connexion TCP vers un autre et qu’un ou plusieurs commutateurs se trouvent sur le chemin physique.

Sur le réseau, les commutateurs ont un rôle important à jouer pour la diffusion des paquets multicast; c’est à ce niveau qu’intervient l’interception du protocole MLD pour assurer une diffusion efficace des trames multicast au sein du réseau.

La plupart des commutateurs implémentent l’interception des annonces IGMP et MLD, ce qui leur permet d’apprendre sur quels ports se trouvent les nœuds abonnés à un groupe multicast donné et ainsi (tout comme pour les adresses MAC) de ne relayer un paquet multicast qu’à destination des nœuds abonnés (contrairement à un paquet de type broadcast qui est envoyé sur tous les ports). Cela consomme des ressources (entrées dans les tables par port), qui peuvent manquer et perturber le bon fonctionnement du protocole par pertes de paquets.

Enfin, les options spécifiques d’outils Linux permettant d’analyser et de superviser ce trafic multicast seront présentées.

Mots-clefs

Réseau, IPv6, Multicast, Commutateur

1 Quelques rappels sur le protocole IPv6

Lors de la conception d’IPv6 il a été choisi de simplifier la spécification en ne conservant que trois protocoles principaux : ICMPv6, UDP et TCP. Et cette version élimine l’utilisation de paquets de type broadcast au profit de diffusion ciblée (multicast).

Par rapport au cas le plus général, cet article ne considère qu’un cas relativement simple (mais le plus courant en dehors des fournisseurs d’accès réseau) : celui de réseaux sur lesquels il n’y a qu’un seul routeur qui fournit la route par défaut à tous les nœuds.

1.1 ICMPv6

Le protocole ICMPv6 (*Internet Control Message Protocol* version 6, RFC 4443) est le protocole de configuration et de gestion des erreurs. La découverte des voisins (ARP en v4) est intégrée dans ICMPv6 ainsi que de nouvelles fonctionnalités comme l’auto-configuration des adresses réseau.

ICMPv6 est donc à la fois plus riche et plus complexe que son ancêtre ICMP (en v4) et repose largement sur l’utilisation de multicast [1].

1.2 Auto-configuration du réseau

IPv6 apporte la possibilité de configurer automatiquement une interface (allocation d'une adresse IP, découverte du routeur par défaut et des serveurs DNS) [2].
L'auto-configuration de l'adresse IP (SLAAC - *StateLess Address AutoConfiguration*, RFC 4862) permet à un nœud de choisir son adresse, soit en la dérivant de l'adresse MAC de l'interface (RFC 2373, Annexe A), soit à partir d'un générateur aléatoire (RFC 7217). Le protocole DAD (*Duplicate Address Detection*, Annexe A du RFC 4662) permet de s'assurer de l'unicité de l'adresse choisie avant de l'utiliser. DAD s'appuie sur une annonce de l'adresse choisie qui est diffusée en multicast vers le groupe de sollicitation de l'adresse en question. En cas de conflit, un autre nœud utilisant la même adresse fera partie de ce groupe multicast et répondra (toujours en multicast) que cette IP est déjà utilisée, afin d'initier un autre choix.

L'affectation des adresses IPv6 peut également être gérée par le protocole DHCPv6 (RFC 8415), qui utilise le protocole UDP (ports 546 et 547). Contrairement à DHCP en IPv4, les informations de routage ne sont jamais fournies par DHCPv6 mais toujours via le mécanisme de découverte des routeurs d'ICMPv6.

1.2.1 Découverte des routeurs

La découverte du routeur par défaut fait partie du sous-protocole de découverte des voisins *Neighbor Discovery Protocol* (NDP, RFC 4861).
La requête *Router Sollicitation* (RS) permet la découverte du routeur. Un routeur s'abonne au groupe multicast des routeurs du lien (`ff02::2`). Un nœud qui souhaite trouver le routeur par défaut du lien peut envoyer un paquet *Router Sollicitation* à cette adresse. Le routeur répondra avec un paquet *Router Advertisement* contenant entre autres son adresse IP.

Un routeur diffuse aussi régulièrement en multicast des annonces avec son adresse (*Router Advertisement*) ce qui permet à un nœud de collecter cette information de manière passive en s'abonnant au groupe de tous les nœuds `ff02::1`).

1.2.2 Découverte des informations DNS

Une extension des paquets RA permet de diffuser les adresses de serveurs DNS à utiliser (RFC 8106). Par ailleurs, le protocole *multicast DNS* (ou *mDNS*, RFC 6772) permet une résolution locale des noms en interrogeant les voisins. En IPv6, c'est le groupe multicast `ff02::fb` qui est utilisé pour diffuser les requêtes mDNS.

1.2.3 Découverte des voisins

En IPv6, c'est le sous-protocole NDP (*Neighbor Discovery Protocol*, RFC 4861) qui est chargé de la découverte des voisins (trouver l'adresse MAC qui correspond à une adresse IP destination pour pouvoir construire la trame ethernet et l'envoyer sur la bonne interface). Chaque interface s'abonne à un ou plusieurs groupes multicast dont le numéro est lié aux adresses de l'interface : le groupe multicast sollicité (*solicited node address*). Il peut y avoir plusieurs adresses par interface, (c'est couramment le cas en IPv6) donc plusieurs groupes de multicast sollicité. L'interface reçoit de ses voisins des demandes (*neighbor discovery*) auxquelles elle répond par un paquet *Neighbor Advertisement* (NA).

Pour une adresse IPv6 donnée, l'adresse du groupe de sollicitation est obtenue à partir du préfixe réservé `ff02::1:ff00:0/104` en complétant par les 24 derniers bits de l'adresse IPv6 associée. Par exemple le groupe de sollicitation associé à l'adresse `2001:db8:cafe:f00d::1234:5678` sera `ff02::1:ff34:5678`

1.2.4 Bilan

Du point de vue multicast, plusieurs groupes de multicast interviennent donc dans le processus de configuration du réseau :

`ff02::1`	tous les nœuds du réseau local
`ff02::2`	tous les routeurs du réseau local
`ff02::1:ff00:0/104`	Le groupe de sollicitation liée à une adresse du nœud

Un nœud IPv6 classique sera donc membre d'au moins 3 groupes multicast:

– le groupe de tous les nœuds du réseau,

– un groupe de sollicitation pour son adresse *link-local*, dérivée de l'adresse MAC,

– un groupe de sollicitation pour son adresse publique principale (qui sera très probablement différent du précédent si DHCPv6 ou la configuration d'adresse aléatoire du RFC 7217 est utilisée.

– Par ailleurs, il sera éventuellement membre d'autres groupes de sollicitation pour chaque adresse IPv6 temporaire affectée à l'interface.

La bonne réception des paquets destinés aux groupes de sollicitation est indispensable au fonctionnement du mécanisme de découverte des voisins, et donc à l'établissement de connexions IPv6.

1.3 Diffusion large ou restreinte ?

La disparition des paquets de type broadcast (diffusion large) au profit du multicast (diffusion ciblée), a pour but théorique de mieux s'adapter aux réseaux de grande taille.
En effet avec la diffusion large, tous les nœuds d'un domaine de diffusion (tous ceux qui sont dans le même sous-réseau) doivent traiter les paquets qui ont une adresse destination de type broadcast. Sur des réseaux de grande taille (beaucoup de nœuds dans le même domaine de diffusion) cela peut créer une surcharge de travail pour les piles réseau des nœuds et engendrer des problèmes de performance.

Cependant la diffusion est plus simple à implémenter dans les éléments actifs du réseau. En général un paquet à destination de l'adresse de broadcast du réseau est envoyé à une adresse MAC de broadcast (`ff:ff:ff:ff:ff:ff`), les commutateurs reconnaissent cette destination et diffusent le paquet sur tous leurs ports. De même, les interfaces réseau des nœuds implémentent un filtre relativement simple qui déclenche le traitement d'un paquet si l'adresse MAC destination est reconnue comme l'adresse du nœud ou l'adresse de broadcast.

Le principe du multicast (diffusion ciblée) est de créer des groupes identifiés par des adresses IP de destination spécifiques (dans le préfixe `ff::/8` en v6). Des adresses MAC dérivées du numéro de groupe sont utilisées comme destination d'un paquet multicast. Et chaque nœud du réseau choisi de quels groupes de diffusion il souhaite recevoir les paquets en positionnant un filtre sur les adresses MAC correspondant aux groupes qu'il souhaite recevoir, le filtre devenant plus complexe en fonction du nombre d'adresses à reconnaître.

Cette approche est donc plus complexe à implémenter dans les éléments de commutation du réseau. Deux approches sont possibles : dans une approche simpliste les paquets avec une adresse de destination de type multicast sont traités comme des broadcast et diffusés à tous les ports. Tous les paquets multicast sont donc diffusés sur tous les VLANS et les ports physiques et on n'a pas de réduction du trafic.

Le multicast IP a été conçu à l'origine avec la possibilité de router un tel trafic. Il est bien entendu hors de question d'appliquer la stratégie précédente aux paquets multicast qui vont sortir du réseau local. C'est pourquoi il existe un protocole spécifique d'abonnement aux groupes de multicast : IGMP en v4, MLD en v6. Il permet à un routeur de connaître les groupes multicast auxquels sont abonnés les nœuds de l'une de ses interfaces et de demander à ses voisins de ne lui envoyer que les paquets à destination de ces groupes.

Le routage du trafic multicast (qui n'est quasiment pas utilisé à l'échelle de l'internet en raison de trop nombreuses difficultés pratiques) ne sera pas étudié ici. Cependant l'existence de ce protocole permet aux commutateurs d'optimiser le trafic multicast en interceptant (*snooping*) les paquets IGMP ou MLD échangés entre les nœuds et les routeurs pour découvrir à quels groupes de multicast un nœud est abonné et par la suite de ne diffuser les paquets multicast que sur les ports sur lesquels se trouve un nœud abonné. Pour les commutateurs d'extrémité cela représente une diminution non négligeable du trafic envoyé à un port avec un seul nœud (surtout si celui-ci n'a pas d'adresse IPv4 et n'utilise donc plus ARP).

Mais l'interception du trafic (MLD snooping ou IGMP snooping) a un coût pour le logiciel du commutateur : en parallèle de la table des adresses MAC il doit gérer une table des destinations multicast par port. Et cette fonction n'est généralement pas active par défaut sur les commutateurs. De plus cette fonctionnalité rend les commutateurs vulnérables à des attaques de type déni de service par saturation de ces tables. Un exemple remonte à 2014, lorsqu'un défaut des pilotes de certaines cartes réseau Intel provoquait (sans même qu'IPv6 soit actif dans la configuration réseau) des centaines d'abonnements à des groupes de sollicitation, saturant ainsi les commutateurs. Le micro-logiciel certains commutateurs cessaient alors complètement de fonctionner. [3]

1.4 MLD

Le but du protocole MLD est de permettre à chaque routeur de connaître les adresses multicast sur lesquelles écoutent des nœuds sur chaque segment de réseau connecté. Le protocole de routage multicast utilise cette information pour permettre le transport des flux de paquets multicast entre routeurs.
Le protocole est relativement complexe (RFC 3810). Ce qui suit est une description simplifiée dans le cas d'un seul routeur et d'une seule interface par nœud.

1.4.1 Fonctionnement de MLD

MLD est un protocole asymétrique : il sépare les nœuds qui écoutent sur des adresses multicast et les routeurs.
L'adresse multicast de tous les nœuds (`ff02::1`) est traitée à part, puisque tous les nœuds écoutent cette adresse en permanence, il n'y a pas de trafic MLD pour cette adresse.

Lorsqu'un nœud rejoint un groupe multicast (écoute sur un port et une adresse données) ou quitte le groupe (cesse d'écouter sur le port de l'adresse en question) il envoie un paquet *MLD report* informant le routeur du changement de situation.

De plus pour se protéger d'éventuelles pertes de paquets, ce rapport va être retransmis périodiquement (tant qu'il n'y a pas de modification de l'état du nœud).

Enfin le routeur envoie périodiquement des requêtes (*MLD query*) demandant aux nœuds de mettre à jour leur état. Quand un nœud reçoit une telle requête d'un routeur, il répond après un délai d'attente qui permet de fusionner plusieurs événements (abonnements ou désabonnements) qui se produiraient dans l'intervalle.

Un routeur écoute sur l'adresse multicast de tous les routeurs qui gèrent MLD mais doit aussi recevoir toutes les trames ethernet multicast afin de traiter le trafic à router.

La figure 1 illustre la diffusion d'un paquet multicast dans un réseau sans interception de MLD. Tous les commutateurs doivent diffuser le paquet sur toutes leurs interfaces et tous les nœuds le reçoivent.

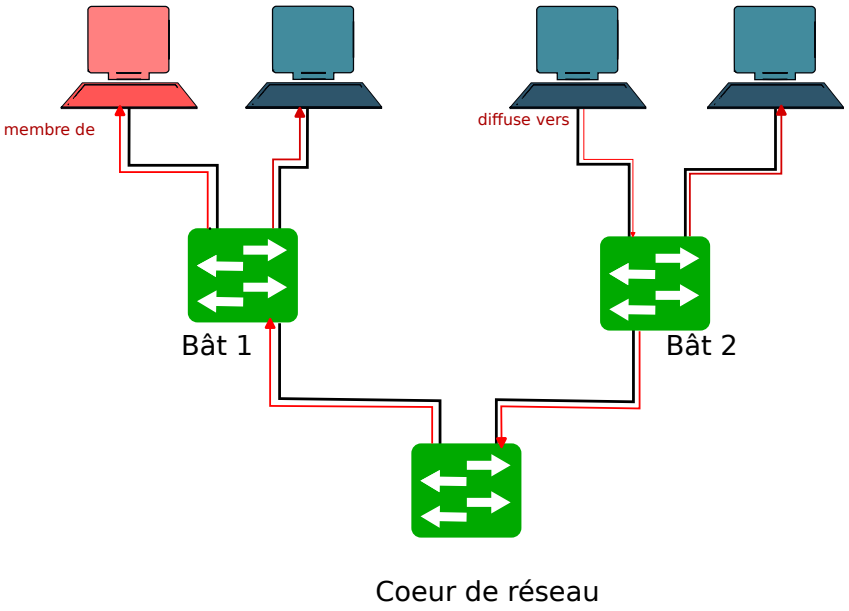

Figure 1: Réseau sans interception de MLD

La figure 2 illustre la diffusion d'un paquet multicast vers un groupe (rouge) après que les commutateurs ont appris sur quel port se trouve le nœud membre du groupe (marqué par des points rouges)

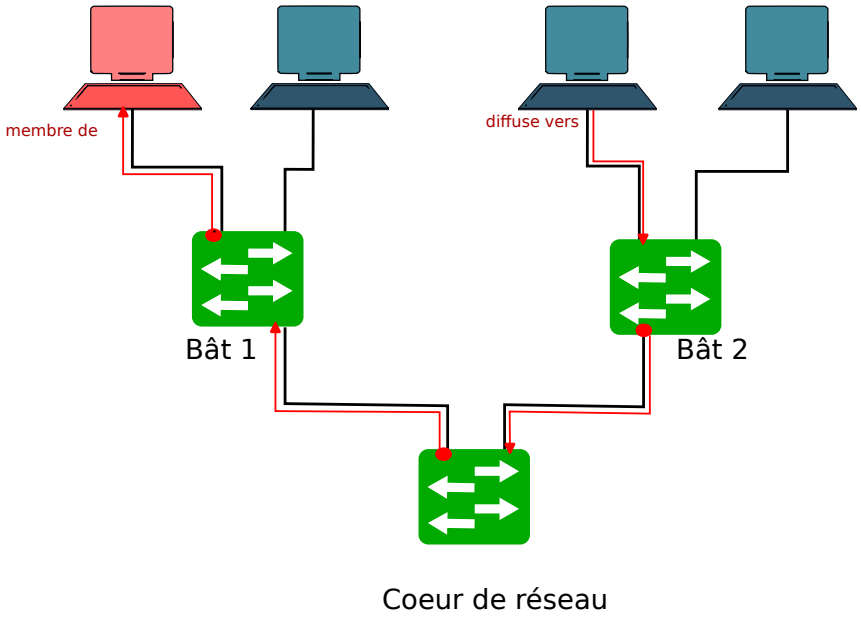

Figure 2: Réseau après apprentissage par interception MLD

1.4.2 Observation de MLD

L'observation du protocole MLD avec un outil comme tcpdump ou wireshark nécessite de prendre en compte la présence des entêtes d'extension dans les messages. Or le filtre « icmpv6 » de la bibliothèque PCAP ne capture pas ces paquets (le filtre recherche l'identifiant du protocole ICMP à un offset fixe par rapport au début du paquet IP; lorsque des extensions sont présentes, l'identifiant de protocole se trouve à une position plus lointaine par rapport au début du protocole).

Pour observer le trafic MLD il est donc nécessaire de capturer tout le trafic IPv6 sur une interface, puis de décoder tous les paquets afin d'identifier les entêtes d'extension et de trouver le début des paquets ICMPv6 pour les identifier correctement.

1.5 MLD Snooping

Les commutateurs sur le chemin vont intercepter le trafic MLD entre le routeur par défaut et les nœuds pour apprendre quel nœud sur quel port est intéressé par un trafic multicast donné.

1.5.1 Commutateurs

En l'absence de routeur multicast sur le réseau local, la plupart des commutateurs qui implémentent l'interception MLD savent également se comporter comme un routeur du point de vue de MLD et peuvent envoyer les requêtes *MLD query* périodiquement pour recevoir les *MLD report* assurant ainsi le bon fonctionnement du mécanisme.

1.5.2 Bridges sous Linux

Le noyau Linux dispose de plusieurs mécanismes pour créer des ponts (bridges) ou commutateurs virtuels. Un des mécanismes les plus utilisés, notamment dans les solutions de virtualisation telles que libvirt ou Proxmox est *bridge(8)*. Il est également utilisé sur de nombreux points d'accès Wifi basés sur Linux, en particulier ceux basés sur OpenWRT.

Ce sous-système dispose d'un ensemble de paramètres qui permettent de contrôler comment sont traités les paquets multicast. En particulier il propose un mécanisme d'apprentissage des adresses multicast appelé *multicast snooping* qui, comme IGMP ou MLD apprend les groupes auxquels sont abonnés les membres du bridge pour optimiser les réémissions. Ce mécanisme est actif par défaut,

mais en l'absence de routeur multicast générant des requêtes IGMP ou MLD, il arrive que des paquets multicast ne soient pas correctement réémis et cela va provoquer une perte de la connectivité IPv6. C'est pourquoi on trouve sur les sites internet de question/réponses de nombreux conseils visant à désactiver `multicast_snooping`, notamment sur les petits réseaux de type SOHO (Small Office / Home Office) parce qu'ils n'ont pas de routeur multicast configuré.

1.5.3 Open vSwitch

Le composant *Open vSwitch* permet de réaliser sur Linux des commutateurs bien plus riches fonctionnellement que le simple bridge. Open vSwitch implémente IGMP et MLD snooping de manière proche à ce que l'on trouve sur un vrai commutateur, avec la possibilité de configurer l'envoi de requêtes MLD.

1.5.4 VMware

Les hyperviseurs VMware disposent de leur propre mécanisme de réseau virtuel entre les machines virtuelles et le réseau physique, qui peut filtrer les paquets multicast et espionner IGMP et MLD. Entre les versions 6 et 7 de vSphere, VMware a modifié le mode de filtrage, passant d'un mode basique (le multicast snooping du bridge de Linux) à un mode qui implémente entièrement IGMP et MLD snooping (à la mode OpenVswitch).

1.5.5 Contrôleurs et points d'accès Wifi

Les contrôleurs Wifi et / ou les points d'accès qu'ils contrôlent disposent également de mécanismes de filtrage des paquets multicast, destinés à limiter le trafic radio pour des raisons d'efficacité. Selon les constructeurs, les options permettant un traitement efficace et correct des paquets multicast nécessaires au bon fonctionnement d'IPv6 diffèrent.

2 Outils

2.1 Multicast en général

Les outils permettant de tester les protocoles multicast en IPv6 ne sont pas très nombreux. Certains outils génériques comme netcat peuvent écouter ou envoyer des paquets multicast.

2.2 Lister les groupes multicast auxquels est abonné un nœud

Sous Linux la commande *iproute2* `ip maddr show` (ou `netstat -n -g` si iproute2 n'est pas disponible) liste les groupes multicast auxquels le nœud est abonné.

Dans les cas des bridges linux `bridge mdb show` permet d'afficher la table des groupes multicast par membre du bridge. Cette commande permet de vérifier le bon fonctionnement du *multicast snooping* en s'assurant que l'ensemble des groupes multicast d'un nœud sont biens présents sur l'interface du bridge à laquelle il est connecté.

2.3 NDPMon

Ndpmon [4] est un outil de surveillance du sous-protocole NDP. Il permet d'apprendre la liste des nœuds d'un segment ainsi que le routeur par défaut. Il génère des alertes lorsqu'un nœud envoie un *Router Advertisement* qui n'était pas attendu (routeur pirate).

ndpmon a été développé au début des années 2000 au LORIA. Il n'est plus activement maintenu. Cependant l'auteur a entrepris de reprendre le code en le modernisant et en corrigeant quelques bugs. Cette version est disponible ici : https://github.com/mherrb/ndpmon

Dans un contexte où MLD snooping est actif sur les commutateurs, ndpmon peut ne pas voir certaines réponses NA parce que l'adresse multicast utilisée en destination n'est pas diffusée sur le port du commutateur où se trouve l'hôte où tourne ndpmon.

Une solution serait de faire envoyer des *MLD queries* par la sonde ndpmon pour, comme sur un commutateur, demander des rapports aux nœuds, afin de connaître tous les groupes multicast utilisés et d'abonner la sonde à tous ces groupes.

2.4 RA guard

La plupart des commutateurs proposent une fonction de protection contre la diffusion de RA pirates. En déclarant quelles sont les adresses MAC des routeurs sur le lien, le commutateur filtrera (et générera une alarme) les paquets RA provenant d'autres nœuds que ceux déclarés.

2.5 MLDmon

Comme il est difficile de filtrer le trafic MLD avec un filtre Pcap, l'auteur a développé un prototype de moniteur de paquets MLD en langage Rust : mldmon. L'outil capture tous les paquets ICMPv6, mais n'analyse que les paquets de type MLD, même en présence d'extensions IP, pour faciliter l'observation du trafic MLD.
Cet outil permet de vérifier quels paquets sont diffusés par les commutateurs qui espionnent également MLD. Il est disponible à l'adresse https://gitlab.laas.fr/matthieu/mldwatch.

2.6 Outils d'« attaque »

Il est parfois utile, pour tester une configuration de pare-feu ou de commutateur de disposer d'outils pour générer du trafic anormal [5]. Parmi les outils classiques on peut citer :

– scapy : la boite à outils en python pour manipuler des paquets réseau. Il dispose d'un module IPv6 pour construire ou analyser les paquets ICMPv6 https://scapy.net/

– THC-IPv6 : une collection d'outils tous prêts pour générer différents types de paquets malicieux, correspondant à des types de vulnérabilités connues [6]. THC-IPv6 ne fonctionne que sur Linux https://github.com/vanhauser-thc/thc-ipv6

– SI6 toolkit : une autre collection d'outils qui fonctionne également sous *BSD. https://www.si6networks.com/research/tools/ipv6toolkit/

Tous ces outils sont également disponibles pré-compilés dans la distribution Kali Linux.

3 Conclusion

En général, le passage à l'adressage IPv6 se fait sans grosses difficultés. Cependant l'utilisation intensive de protocoles multicast nécessite d'avoir une infrastructure de niveau 2 capable de traiter ces protocoles efficacement.
Sur des petits réseaux, il est envisageable de ne pas activer MLD snooping sur les commutateurs, ce qui revient à un fonctionnement équivalent à IPv4 ou l'essentiel de ce trafic sera diffusé sur tous les segments.

Au-delà d'une certaine taille, MLD snooping permet de bénéficier pleinement des avantages liés à l'utilisation de multicast par IPv6 en réduisant le trafic sur chaque brin réseau. Mais cela nécessite de disposer d'outils pour vérifier le bon fonctionnement de l'ensemble et identifier les problèmes.

Glossaire

ARP : Address Resolution Protocol

DAD: Duplicate Address Detection

DHCP : Dynamic Host Configuration Protocol

ICMP : Internet Control Message Procotol

IGMP : Internet Group Management Protocol

MLD : Multicast Listener Discovery

NA : Neighbor Advertisement

ND : Neighbor Discovery

NDP : Neighbor Discovery Protocol

PCAP : Packet Capture

RA : Router Advertisement

RS : Router Solicitation

SOHO : Small Office / Home Office : réseau simple avec un seul routeur qui ne dispose que d'une seule adresse IP publique vers l'extérieur et fait de la traduction d'adresses depuis le réseau local.

Bibliographie

[1] T. Martin, IPv6 Multicast Primer, Spring 2014 https://txv6tf.wordpress.com/wp-content/uploads/2016/03/2014-martin-ipv6-multicast-primer.pdf

[2] B. Stevant, J Landru, J.P. Rioual, P. Anelli, J. Grouffaud, P.U. Tournoux, MOOC Objectif IPv6 ! Document Compagnon, Séquence 1. Avril 2017.

[3] G. Wollman, The network nightmare that ate my week Septembre 2014, https://blog.bimajority.org/2014/09/05/the-network-nightmare-that-ate-my-week/

[4] F. Beck, T. Cholez, O. Festor and I. Chrisment, Monitoring the Neighbor Discovery Protocol, The Second International Workshop on IPv6 Today - Technology and Deployment – Ipv6TD 2007, Mar 2007, Guadeloupe/French Caribbean, Guadeloupe. inria-00153558

[5] S. Bortzmeyer, Le cours « Hacking IPv6 », juin 2013, https://www.bortzmeyer.org/hacking-ipv6.html

[6] M. Heuse, IPv6 Vulnerabilities, Failures - and a Future?, Novembre 2011 https://www.mh-sec.de/downloads/mh-ipv6_vulnerabilities.pdf