

MOOC


Objectif IPv6 !

vers l'internet nouvelle génération

Document Compagnon

Séquence 3

Les mécanismes de gestion d'un réseau IPv6

Le contenu de ce document d'accompagnement du MOOC IPv6 est publié sous
Licence Creative Commons **CC BY-SA 4.0 International**. 

Licence Creative Commons CC BY-SA 4.0 International



Attribution - Partage dans les Mêmes Conditions 4.0 International (CC BY-SA 4.0)

Avertissement Ce résumé n'indique que certaines des dispositions clé de la licence. Ce n'est pas une licence, il n'a pas de valeur juridique. Vous devez lire attentivement tous les termes et conditions de la licence avant d'utiliser le matériel licencié.

Creative Commons n'est pas un cabinet d'avocat et n'est pas un service de conseil juridique. Distribuer, afficher et faire un lien vers le résumé ou la licence ne constitue pas une relation client-avocat ou tout autre type de relation entre vous et Creative Commons.

Clause C'est un résumé (et non pas un substitut) de la licence.

<http://creativecommons.org/licenses/by-sa/4.0/legalcode>

Vous êtes autorisé à :

- **Partager** — copier, distribuer et communiquer le matériel par tous moyens et sous tous formats
- **Adapter** — remixer, transformer et créer à partir du matériel
- pour toute utilisation, y compris commerciale.

L'Offrant ne peut retirer les autorisations concédées par la licence tant que vous appliquez les termes de cette licence.

Selon les conditions suivantes :

Attribution — You must give **appropriate credit**, provide a link to the license, and **indicate if changes were made**. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

Partage dans les Mêmes Conditions — Dans le cas où vous effectuez un remix, que vous transformez, ou créez à partir du matériel composant l'Oeuvre originale, vous devez diffuser l'Oeuvre modifiée dans les même conditions, c'est à dire avec **la même licence** avec laquelle l'Oeuvre originale a été diffusée.

No additional restrictions — Vous n'êtes pas autorisé à appliquer des conditions légales ou des **mesures techniques** qui restreindraient légalement autrui à utiliser l'Oeuvre dans les conditions décrites par la licence.

Notes: Vous n'êtes pas dans l'obligation de respecter la licence pour les éléments ou matériel appartenant au domaine public ou dans le cas où l'utilisation que vous souhaitez faire est couverte par une **exception**.

Aucune garantie n'est donnée. Il se peut que la licence ne vous donne pas toutes les permissions nécessaires pour votre utilisation. Par exemple, certains droits comme **les droits moraux, le droit des données personnelles et le droit à l'image** sont susceptibles de limiter votre utilisation.

Les informations détaillées sont disponibles aux URL suivantes :

- <http://creativecommons.org/licenses/by-sa/4.0/deed.fr>
- http://fr.wikipedia.org/wiki/Creative_Commons

Les auteurs



Bruno Stévant

Bruno STEVANT est enseignant chercheur à l'IMT Atlantique. Il intervient dans l'enseignement et sur les projets de recherche autour d'IPv6 depuis plus de 10 ans. Il est secrétaire et responsable des activités de formation de l'association G6, association pour la promotion et le déploiement d'IPv6 en France.



Jacques Landru

Enseignant chercheur au département Informatique et Réseaux à l'IMT Lille Douai, Jacques est responsable de l'UV de spécialisation ARES (Architecture des RESeaux) à la fois dans le mode traditionnel présentiel que dans sa forme à distance dans le cadre du cursus diplômant TutTelNet.



Jean-Pierre Rioual

Ingénieur Conseil Réseaux – EURÊKOM. Fort de 30 années d'expérience dans le domaine des réseaux, il intervient auprès des entreprises pour des missions d'expertise sur leurs réseaux de transmission de données (intégration, mesures, optimisation, administration), conçoit et anime des actions de formation "réseaux".

**Pascal Anelli**

Pascal ANELLI est enseignant-chercheur à l'Université de la Réunion. Il enseigne les réseaux depuis plus de 20 ans. Il est membre du G6 depuis sa création. A ce titre, il est un des contributeurs du livre IPv6. En 1996, il a participé au développement d'une version de la pile IPv6 pour Linux.

**Joël Grouffaud**

Joël GROUFFAUD est professeur agrégé de mathématiques. Il est chef du département Réseaux et Télécommunications de l'IUT de la Réunion, une composante de l'université de La Réunion. Au sein du département, il enseigne les réseaux et IPv6. Il anime l'académie Cisco (formations CCNA) de La Réunion.

**Pierre Ugo TOURNOUX**

Pierre Ugo TOURNOUX est enseignant chercheur à l'Université de la Réunion. Il est responsable des enseignements d'administration réseau, de routage et des réseaux sans fil dans lesquels il intègre IPv6 depuis de nombreuses années.

Remerciements à :

- Vincent Lerouvillois, pour son travail de relecture attentive ;
- Bruno Di Gennaro (Association G6) ;
- Bruno Joachim (Association G6) pour sa contribution à l'activité « Contrôler la configuration réseau par DHCPv6 » ;
- Richard Lorion (Université de la Réunion) pour sa contribution à l'activité « Etablir la connectivité IPv6 tunnels pour IPv6 ».

Tables des activités

Les auteurs	5
Introduction	11
Activité 31 : Contrôler le fonctionnement du réseau par ICMPv6	13
Introduction.....	13
Format général d'un message ICMPv6.....	13
Test d'accessibilité d'un nœud.....	15
Rapport d'erreur.....	17
Découverte des voisins.....	18
Format des messages mis en œuvre.....	19
Message Sollicitation d'un voisin.....	19
Message Annonce d'un voisin.....	20
Fonctionnement de la résolution d'adresse IP.....	21
Fonctionnement de la détection d'adresse dupliquée.....	24
Conclusion.....	26
Références bibliographiques.....	26
Pour aller plus loin.....	26
Activité 32: Configurer automatiquement les paramètres réseau	27
Principe de l'auto-configuration.....	27
Mécanismes mis en œuvre.....	27
La création de l'adresse "lien-local".....	28
Découverte des paramètres communs au réseau.....	29
L'auto-configuration "sans état" pour une adresse IP routable.....	31
L'auto-configuration "avec état" de l'adresse IP routable.....	33
La configuration de la table de routage.....	33
La découverte des serveurs DNS.....	34
Exemple de configuration automatique.....	34
Conclusion.....	38
Références bibliographiques.....	39
Pour aller plus loin.....	39
Activité 33 : Faire correspondre adresse et nom de domaine	41
Introduction.....	41
Concepts de base du DNS.....	41
Nommage « à plat ».....	41
Caractéristiques du système de noms de domaine.....	42
Principe de fonctionnement du service DNS.....	46
Les serveurs de noms.....	48
Serveurs de noms primaires et secondaires.....	48
Serveur DNS récursif (<i>caching name server</i>).....	52
Relais DNS (<i>forwarder</i>).....	53
Serveurs DNS à rôles multiples.....	53
Spécifications du service de nommage.....	53
Spécifications du résolveur.....	53
Spécifications des ressources IPv6.....	54
Nommage direct : enregistrement AAAA.....	54
Nommage inverse : enregistrement PTR.....	55
Découverte de la liste de serveurs DNS récursifs.....	57
Principe des trois propositions : RA, DHCPv6, anycast.....	58

Extension de l'autoconfiguration "sans état" pour le DNS.....	58
Extension de la configuration "à état", DHCPv6.....	59
Utilisation d'adresses anycast réservées.....	59
Mises en œuvre d'un serveur de noms.....	59
Références bibliographiques.....	60
Pour aller plus loin.....	60
Activité 34 : Contrôler la configuration réseau par DHCPv6.....	61
Introduction.....	61
Principe de fonctionnement du protocole DHCPv6.....	61
Présentation générale du protocole DHCPv6.....	61
Communication en DHCPv6.....	62
Les entités du protocole.....	62
Gestion centralisée des ressources allouées.....	63
Principe de l'allocation d'adresse IPv6 à un client en l'absence de relais.....	63
Recherche des serveurs DHCPv6 par le client : fonctionnement de la pile de communication.....	64
Principe de l'allocation d'adresse IPv6 à un client en présence d'un relais DHCPv6.....	65
Libération de l'adresse IPv6 par un client DHCPv6.....	66
Fonctions des messages du protocole DHCPv6.....	67
Messages échangés entre client et serveur.....	68
Messages de gestion des ressources allouées.....	68
Messages échangés entre relais et serveur.....	69
Tableau récapitulatif des messages DHCPv6.....	69
Extension du protocole DHCPv6 [RFC 6422].....	70
Structure des messages DHCPv6.....	71
Structure des messages émis par les serveurs et clients DHCPv6.....	71
Structure des messages échangés entre relais et serveur DHCPv6.....	72
Message DHCPv6 RELAY-FORWARD.....	73
Message DHCPv6 RELAY-REPLY.....	73
Types de DUID : DHCPv6 Unique IDentifier.....	73
Association d'identités.....	74
Allocation des adresses non temporaires.....	75
Allocation des adresses temporaires.....	75
Options du protocole DHCPv6.....	75
Délégation de préfixe à états.....	77
Applications de la délégation de préfixe.....	77
Renumérotation des réseaux.....	78
Renumérotation passive.....	78
Renumérotation active.....	78
Principe de l'allocation.....	79
Principe de l'allocation de préfixe à états avec relais.....	79
Conclusion.....	80
Références bibliographiques.....	81
Pour aller plus loin.....	81
Conclusion.....	83
ANNEXE 1 Activité 31 : Contrôler le fonctionnement du réseau par ICMPv6.....	85
Introduction.....	85
Messages ICMPv6 de rapport d'erreur.....	85
Destination inaccessible.....	85

Paquet trop grand.....	86
Délai expiré.....	86
Erreur de paramètre.....	87
Attention au filtrage d'ICMPv6.....	88
Gestion de groupes multicast sur le lien local.....	88
Format des messages pour MLD.....	89
Principe de MLD.....	90
Fonctions autres et expérimentales.....	91
Options véhiculées par les messages ICMPv6.....	92
Adresse physique de la source/cible.....	94
Information sur le préfixe.....	94
En-tête redirigée.....	95
MTU.....	95
Référence bibliographique.....	96
ANNEXE 2 Activité 33 : Faire correspondre adresse et nom de domaine.....	97
Options DNS des RA.....	97
Option de liste de serveurs DNS récurifs (RDNSS).....	97
Option de liste de domaines recherchés (DNSSL).....	97
Options DNS du protocole DHCPv6.....	98
Option serveur de nom récurif de DHCPv6.....	98
Option liste de suffixes de nom de domaine.....	99
Mises en œuvre du service DNS.....	99
Logiciels DNS supportant IPv6.....	100
Principe de configuration d'un serveur DNS.....	100
Définition des fichiers de zone.....	101
Types d'enregistrement de ressource DNS.....	102
Configuration de serveur DNS.....	103
Réseau virtualisé utilisé pour générer ces exemples.....	103
Fichier de configuration d'un serveur BIND9.....	104
Exemple de contenu du fichier <i>/etc/bind9/named.conf</i>	104
Configuration du fonctionnement du serveur.....	105
Contenu du fichier <i>named.conf.options</i>	105
Exemple de configuration locale du serveur de noms BIND9.....	106
Exemple de contenu du fichier <i>named.conf.local</i>	106
Contenu du fichier <i>named.conf.default-zones</i>	107
Fichier de zone DNS pour la résolution directe (nom - adresse).....	107
Fichier de zone DNS inverse en IPv6.....	108
Fichier <i>db.131.tpt.example.com.rev</i>	108
Fichier <i>db.132.tpt.example.com.rev</i>	108
Fichier <i>db.133.tpt.example.com.rev</i>	109
Clients du service de nommage.....	109
Exemple de fichier de configuration <i>/etc/resolv.conf</i> d'un serveur de noms.....	109
Exemple de fichier de configuration <i>/etc/resolv.conf</i> d'une machine.....	109
Outils de vérification de la configuration DNS.....	110
Exemples d'interrogation d'un serveur DNS avec <i>dig</i> : résolution directe.....	110
Exemple d'interrogation d'un serveur DNS avec la commande <i>host</i> : résolution directe.....	111
.....	111
Exemple d'interrogation d'un serveur DNS avec la commande <i>dig</i> : résolution inverse.....	111
.....	111
Exemple d'interrogation d'un serveur DNS avec la commande <i>host</i> : résolution inverse.....	111
.....	111

Recommandations opérationnelles pour l'intégration d'IPv6.....	112
Deux impossibilités d'accéder au service de nommage et leurs remèdes.....	113
Premier scénario : client IPv4 et serveur IPv6.....	113
Second scénario : client IPv6 et serveur IPv4.....	113
Taille limitée des messages DNS en UDP, extension EDNS.0.....	113
Glue IPv6.....	114
Publication des enregistrements AAAA dans le DNS.....	115
Pour aller plus loin : mises à jour dynamiques du DNS.....	116
Conclusion.....	116
ANNEXE 3 Activité 34 : Format DHCPv6.....	119
Structure des options du protocole DHCPv6.....	119
Option d'identification du client.....	119
Option identification du serveur (<i>Server Identification Option</i>).....	119
Option association d'identité pour les adresses non temporaires.....	120
Option d'association d'identité pour les adresses temporaires.....	120
Option d'adresse d'association d'identités.....	121
Option de demande d'options.....	121
Option de priorité (du serveur).....	121
Option "temps écoulé" (depuis le début d'un échange).....	122
Option "message relayé".....	122
Option d'authentification.....	123
Option d'utilisation de l'adresse individuelle du serveur.....	123
Option de code d'état.....	124
Option de Validation rapide.....	124
Option classe d'utilisateur.....	124
Option de classe de constructeur.....	125
Option d'information spécifique d'un constructeur.....	125
Option d'identification d'interface.....	126
Option de message de reconfiguration.....	127
Option d'acceptation de reconfiguration.....	127
Extension du protocole DHCPv6 : options spécifiques des relais.....	127
Codes d'état du protocole DHCPv6.....	128
Structure des identifiants DUID du protocole DHCPv6.....	129
DUID construit à partir de l'adresse physique + horodate (DUID-LLT).....	129
DUID dérivé du numéro d'entreprise affecté par un constructeur (DUID-EN).....	129
DUID dérivé de l'adresse physique de l'équipement (DUID-LL).....	130
Options pour la délégation de préfixes (RFC 8415).....	131
Structure de l'option d'association d'identités pour la délégation de préfixes.....	131
Option de préfixe d'association d'identités pour la délégation de préfixe.....	131

Introduction

La séquence précédente vous a montré qu'IPv6 constitue un retour aux fondamentaux du protocole IP : transporter des données d'un point à un autre de l'Internet, avec une intervention minimale des équipements intermédiaires, intervention réduite, la plupart du temps, à la fonction d'acheminement (*forwarding*) des paquets.

Cette nouvelle séquence va se concentrer sur les mécanismes qui ont été spécifiés autour du protocole IPv6 pour contrôler la connectivité offerte. La première activité va vous présenter le protocole **ICMPv6** qui s'attache, à l'échelle de l'Internet, de contrôler le fonctionnement du protocole IP et l'envoi de rapport d'erreur si besoin. La seconde activité s'attachera à décrire la **configuration automatique** d'un nouveau noeud au sein du réseau local. La troisième activité abordera la **configuration automatique avec état**. Enfin, dans la dernière activité, vous étudierez le fonctionnement du **système de nommage**. Ce système est essentiel pour utiliser des noms à la place des adresses IP. Les machines et services Internet désignés par des noms présentent l'avantage d'être facilement mémorisables par les utilisateurs.

Activité 31 : Contrôler le fonctionnement du réseau par ICMPv6

Vous suivez une activité de découverte



Introduction

Le bon fonctionnement de la couche réseau est supervisé par le protocole ICMPv6 (*Internet Message Control Protocol*) [RFC 4443]. Tout comme ICMP pour IPv4, ICMPv6 s'appuie sur IPv6 pour réaliser ces fonctions. La famille des protocoles ICMP donne les informations sur l'état de marche du réseau. Elle rapporte également les erreurs quand un paquet ne peut être traité correctement. On distingue 3 fonctions propres à ICMP :

- le signalement d'erreur en cours d'acheminement d'un paquet ;
- le test d'accessibilité d'un nœud ;
- la configuration automatique des équipements.

Terminologie

Un équipement connecté au réseau est dénommé nœud. Si ce nœud est un équipement terminal, on parle d'hôte. Le sous-réseau qui correspond à un réseau local sous-jacent se qualifie, en IPv6, de lien. Tous les nœuds attachés au même lien sont des voisins.

À la différence d'ICMP pour IPv4, qui comporte également ces trois fonctions, ICMPv6 intègre les fonctions de gestion des groupes de multicast (*Multicast Listener Discovery* (MLD)) et de résolution d'adresse IP en adresse physique. En IPv4, ces fonctions étaient assurées par des protocoles annexes (la gestion des groupes était du ressort de IGMP (*Internet Group Management Protocol*), et la résolution d'adresse, du protocole ARP (*Address Resolution Protocol*). La résolution d'adresse en IPv6 s'effectue par la procédure de découverte des voisins (*Neighbor Discovery Protocol*(NDP)). La notion de voisinage est définie par la connectivité au lien. Deux nœuds connectés sur le même lien sont des voisins. Ils partagent le même préfixe réseau. Un lien est, par exemple, un domaine de diffusion Ethernet bordé par au moins un routeur. ICMPv6 comporte aussi des fonctions pour la mobilité IP. Il en ressort qu'ICMPv6 est bien plus complet que son prédécesseur. Non seulement ICMPv6 rapporte les erreurs, mais pour les mécanismes d'autoconfiguration ou de découverte du voisinage, il est un élément indispensable dans le service de connectivité offert par la couche de **réseau**.

Format général d'un message ICMPv6

Les messages ICMPv6 sont encapsulés directement dans un paquet IPv6. Le protocole se voit attribuer le numéro 58 pour être représenté dans l'en-tête IPv6 comme prochain en-tête (champ *Next Header*). Le format général des messages ICMPv6 est donné par la figure 1. L'en-tête des messages ICMPv6 comporte 3 champs :

1. Le champ Type : il indique la nature du message ICMPv6 et donc, le format spécifique du message. Les messages ICMPv6 forment 2 groupes : un groupe pour les messages

d'informations et un autre pour les messages d'erreurs. Les groupes sont identifiés par le bit de poids fort de ce champ. Les messages d'erreurs ont ce bit à zéro et donc, le champ Type prendra, pour ces messages, une valeur comprise entre 0 et 127. Les messages d'informations sont identifiés par un champ Type dont la valeur est comprise entre 128 et 255.

2. Le champ Code s'interprète dans le contexte du type de message. Il est utilisé pour ajouter une granularité plus fine au type.
3. Le champ Checksum sert à vérifier l'intégrité du message ICMP. Il porte aussi bien sur l'en-tête du message que sur sa charge utile.

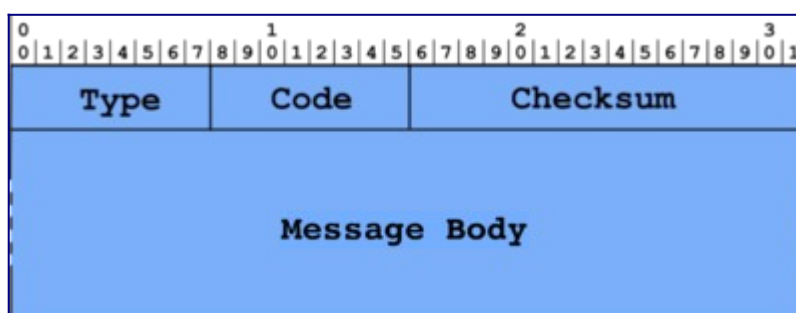


Figure 1 : Format d'un message ICMPv6.

La charge utile du message ICMPv6 (*message body*) est relative au contexte fonctionnel :

- dans le cas des messages de compte rendu d'erreurs, elle contient le paquet IPv6 ayant provoqué l'erreur. Pour éviter d'avoir à fragmenter, la longueur du message ICMPv6 est limitée à 1 280 octets. Par conséquent, le contenu du paquet IPv6 renvoyé peut être tronqué ;
- pour les messages de découverte du voisinage, la charge utile est composée de différentes options selon qu'il s'agisse d'une sollicitation de voisin ou d'une annonce de voisin ;
- les messages de test d'accessibilité embarquent des données de taille et de contenu quelconque.

Les messages sont spécifiés dans différents RFC. Cependant, la liste complète des types de messages et les différents paramètres associés sont regroupés par l'IANA (*Internet Assigned Number Authority*)[1]. Le tableau 1 présente les valeurs et les codes définis dans le [RFC 4443](#). Ce qu'il faut noter, c'est que les valeurs de type inférieures à 128 sont pour les messages d'erreurs et qu'à partir de 128, ce sont des messages d'informations.

Type	Code	Signification
Message de compte rendu d'erreur		
1		Destination inaccessible :
	0	Aucune route vers la destination.
	1	La communication avec la destination est administrativement interdite.

2	Hors portée de l'adresse source.
3	L'adresse est inaccessible.
4	Le numéro de port est inaccessible.
5	L'adresse source est filtrée par un <i>firewall</i> .
6	L'adresse destination est rejetée.
2	Paquet trop grand.
3	Délai expiré :
0	Limite du nombre de sauts atteinte.
1	Temps de réassemblage dépassé.
4	Erreur de paramètre :
0	Champ d'en-tête erroné.
1	Champ d'en-tête suivant non reconnu.
2	Option non reconnue.
Message d'information	
128	Demande d'écho
129	Réponse d'écho

Tableau 1 : Messages ICMPv6 décrit dans le [RFC 4443](#).

Test d'accessibilité d'un nœud

Le test d'accessibilité vise à vérifier qu'un nœud est joignable par l'adresse IPv6 de son interface. Autrement dit, depuis un nœud, on vérifie qu'il existe une connectivité entre deux nœuds de l'Internet. Ce test est couramment utilisé, notamment à l'aide de l'outil nommé ping. Le principe de fonctionnement est le même que pour IPv4. Un message de demande d'écho (*echo request*), message ICMPv6 type 128 est envoyé vers le nœud dont on veut tester la connectivité. Ce dernier répond par le message "réponse d'écho" (*echo reply*), message ICMPv6 de type 129. Le format de ces deux messages est présenté par la figure 2.

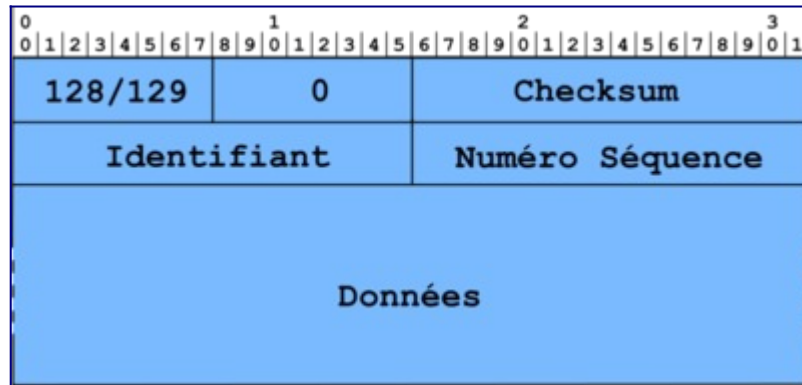


Figure 2 : Format du message d'écho.

Les réponses sont identifiées par le champ `identifiant`. Ainsi, la réponse est rapprochée de la requête. Ceci est particulièrement utile quand plusieurs commandes `ping` sont exécutées simultanément sur la machine. Le champ `numéro de séquence` complète le mécanisme de rapprochement de la réponse à la requête et va servir à mesurer la durée d'aller et retour dans le cas où les demandes sont émises en continu et que le délai de propagation est élevé. La réponse doit toujours contenir les mêmes valeurs que la requête pour ces deux champs. Le champ `données` permet d'augmenter la taille du message (et donc la durée de transmission) pour les mesures.

Pour illustrer le test d'accessibilité, observons l'échange suivant : Le nœud nommé "Uma" teste la connectivité du nœud "Ganesha" via la commande `ping6`. La commande entrée sur Uma est la suivante :

```

uma# ping6 ganesha
trying to get source for ganesha
source should be 2001:db8:12:3:a00:20ff:fe0a:aa6d
PING ganesha (2001:db8:12:3::3): 56 data bytes
64 bytes from 2001:db8:12:3::3: icmp6_seq=0 ttl=255 time=5.121 ms
    
```

Les messages ICMPv6 obtenus sont les suivants.

- Le nœud "Uma", qui est l'initiateur, envoie un message *ICMPv6 Demande d'écho*. La trace 1 montre un paquet IPv6 contenant un message *ICMPv6 Demande d'écho* (en bleu dans la trace).

```

Version : 6           Classe : 0x00           Label : 000000 Longueur : 64 octets
(0x0040)
Protocole : 58 (0x3a, ICMPv6)
Nombre de sauts : 64 (0x40)
Source : 2001:db8:12:3:a00:20ff:fe0a:aa6d (Uma)
Desti. : 2001:db8:12:3::3 (Ganesha)
ICMPv6
Type : 128 (0x80, Demande d'écho)           Code : 0 Checksum : 0xcfe8
Identificateur : 0x0e02 Numéro de séquence : 0x0001
Données : b6e0 f056 ...

0000:  6000 0000 0040 3a40 2001 0db8 0012 0003
0010:  0a00 20ff fe0a aa6d 2001 0db8 0012 0003
0020:  0000 0000 0000 0003|80|00|cfe8|0e02|0001|
0030:  b6e0 f056 d947 0700 0809 0a0b 0c0d 0e0f
    
```



```
0040: 1011 1213 1415 1617 1819 1a1b 1c1d 1e1f
0050: 2021
```

Trace 1 : Message ICMPv6 Demande d'écho.

- Le destinataire du message "Demande d'écho", qui est Ganesha sur la figure 10, acquitte ce message en retournant un message *ICMPv6 Réponse d'écho* (voir la trace 2).

```
Version : 6           Classe : 0x00           Label : 000000 Longueur : 64 octets
(0x0040)
Protocole : 58 (0x3a, ICMPv6)
Nombre de sauts : 64 (0x40)
Source : 2001:db8:12:3::3 (Ganesha)
Desti. : 2001:db8:12:3:a00:20ff:fe0a:aa6d (Uma)
ICMPv6
Type : 129 (0x81, Réponse d'écho)           Code : 0 Checksum : 0xcee8
Identificateur : 0x0e02 Numéro de séquence : 0x0001
Données : b6e0 f056 ...

0000: 6000 0000 0040 3a40 2001 0db8 0012 0003
0010: 0000 0000 0000 0003 2001 0db8 0012 0003
0020: 0a00 20ff fe0a aa6d|81|00|cee8|0e02|0001|
0030: b6e0 f056 d947 0700 0809 0a0b 0c0d 0e0f
0040: 1011 1213 1415 1617 1819 1a1b 1c1d 1e1f
0050: 2021
```

Trace 2 : Message ICMPv6 Réponse d'écho.

Rapport d'erreur

ICMPv6 permet de signaler, à l'émetteur d'un paquet, un problème dans son acheminement ou dans sa réception, par des messages de rapport d'erreur. Lorsqu'une machine émet un paquet, si une erreur est détectée par le destinataire ou par tout routeur intermédiaire le long du chemin vers le destinataire, alors l'élément qui détecte l'erreur renvoie à l'émetteur un rapport sous la forme d'un message ICMPv6. Le type du message et son code définissent précisément l'erreur détectée. L'extrait, jusqu'à concurrence de 1280 octets du paquet en défaut, est incluse pour permettre l'analyse de l'erreur. L'exemple ci-dessous illustre un message ICMP *Paquet trop grand*, généré par un routeur intermédiaire dès qu'un datagramme ne peut être retransmis en raison de la limitation de la MTU sur son interface de sortie.

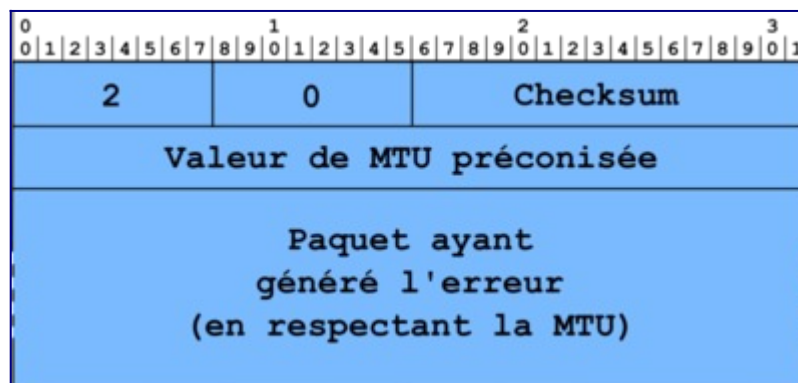


Figure 4 : Format du message ICMPv6 Paquet trop grand.

Différentes erreurs peuvent être signalées par ICMPv6. Les cas les plus courants sont :

- destination inaccessible (type 1), la raison est précisée par le champ Code ;
- paquet trop grand (type 2) ;
- délai expiré (type 3) ;
- erreur de paramètre (type 4).

Nota : une description de ces messages ICMPv6 est disponible dans le document annexe à cette activité : [MOOC:Annexe_Compagnon_Act31](#).

Découverte des voisins

La découverte des voisins ou NDP (*Neighbor Discovery Protocol*) est décrite par le [RFC 4861](#). Ce RFC, paru en 2007, est la troisième et dernière version du protocole. On parle de protocole car les messages utilisés par NDP sont encapsulés dans les paquets IPv6, de la même manière qu'ICMPv6. En fait, on peut voir NDP comme un sous-protocole d'ICMPv6. NDP vise à gérer les interactions entre un nœud et ses voisins. Les voisins sont les nœuds qui partagent une même connectivité physique. Dans la terminologie IPv6, on parle de lien. Avec NDP, un nœud est capable de dialoguer avec les nœuds connectés au même support (hôtes et routeurs). Il ne s'agit pas, pour un nœud, de connaître exactement la liste de tous les autres nœuds connectés sur le lien, mais uniquement de gérer ceux avec qui il dialogue.

Le protocole utilise cinq types de messages ICMPv6, comme le montre le tableau 3. Nous allons, dans la suite de ce paragraphe, nous intéresser à deux fonctions de NDP :

- la détermination de l'adresse physique d'un nœud à partir de son adresse IP ;
- la détection d'adresses IP dupliquées.

Ces fonctions sont réalisées à travers deux messages ICMPv6 : "sollicitation de voisin" (*Neighbor Solicitation* ou NS) et "annonce d'un voisin" (*Neighbor Advertisement* ou NA). La fonction de découverte du routeur et d'auto-configuration sera présentée dans une autre activité.

Type	Code	Signification
Découverte de voisins		
133		Sollicitation du routeur
134		Annonce du routeur
135		Sollicitation d'un voisin
136		Annonce d'un voisin
137		Redirection
Découverte de voisins inverse (RFC 3122)		
141		Sollicitation
142		Annonce

Découverte de voisins sécurisée (SEND, RFC 3971)	
148	Sollicitation de chemin de certification
149	Annonce de chemin de certification

Tableau 3 : Messages ICMPv6 pour les interactions entre voisins

Format des messages mis en œuvre

Avant d'étudier la procédure, nous allons présenter le format des messages impliqués.

Les messages ICMPv6 pour NDP sont encapsulés dans des paquets IPv6. Il est intéressant de souligner que le champ nombre de sauts de l'en-tête IPv6 contient la valeur 255. Cette valeur peut sembler trop grande pour des datagrammes qui ne doivent pas être routés hors du lien physique. En fait, si un nœud reçoit un datagramme avec une valeur plus petite, cela signifie que l'information provient d'un autre réseau et qu'elle a déjà traversé un routeur. Les datagrammes ayant une valeur différente de 255 doivent être ignorés par le récepteur.

Message Sollicitation d'un voisin

Le message de la figure 8 sert à demander des informations d'un nœud voisin, c'est-à-dire situé sur le même lien physique (ou connecté via des ponts). Le message peut lui être explicitement envoyé, ou émis sur une adresse multicast. Dans le cas de la détermination de l'adresse physique, il a la même fonction qu'une requête ARP du protocole IPv4.

Le champ adresse source du paquet IPv6 contient, soit l'adresse locale au lien, soit une adresse globale, soit l'adresse non spécifiée.

Le champ adresse destination contient, soit l'adresse de multicast sollicité correspondant à l'adresse recherchée, soit l'adresse d'un nœud dans le cas d'une détection d'inaccessibilité des voisins (*Neighbor Unreachability Detection* NUD).

Le champ adresse de la cible contient l'adresse IPv6 du nœud recherché.

Le champ option contient, en général, l'adresse physique de la source.

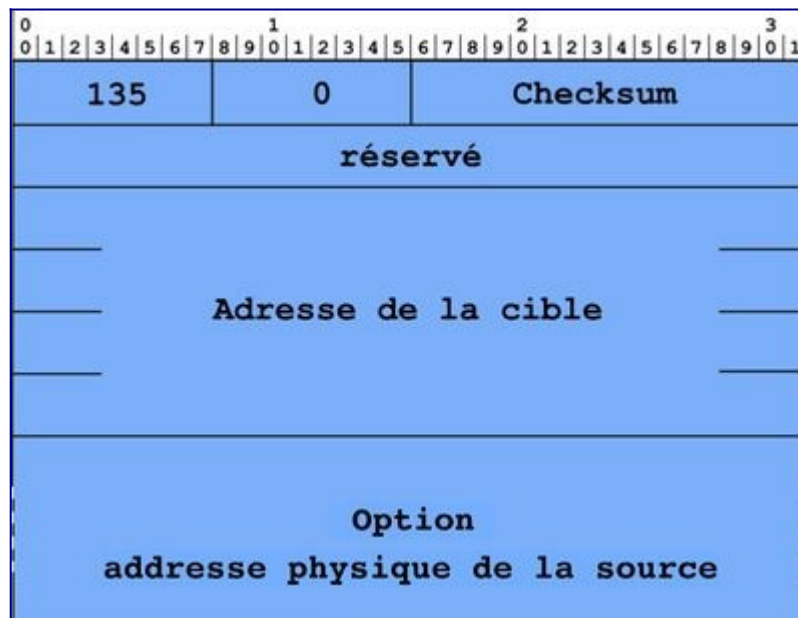


Figure 8 : Format du message de Sollicitation d'un voisin.

Message Annonce d'un voisin

Le message de la figure 9 est émis en réponse à une sollicitation, mais il peut aussi être émis spontanément pour propager une information de changement d'adresse physique, ou de statut routeur. Dans le cas de la détermination d'adresse physique, il correspond à la réponse ARP du protocole IPv4. L'adresse de la cible, dans ce cas-là, correspond à l'adresse de la source de ce message.

Les champs de ce message ont la signification suivante :

- le bit R est mis à 1 si l'émetteur est un routeur ;
- le bit S mis à 1 indique que cette annonce est émise en réponse à une sollicitation ;
- le bit 0 mis à 1 indique que cette annonce doit effacer les informations précédentes qui se trouvent dans les caches des autres nœuds, en particulier la table contenant les adresses physiques ;
- le champ adresse de la cible reprend l'adresse de la cible de la sollicitation auquel ce message répond (le bit S vaut 1 dans ce cas). Si le message d'annonce de voisin est envoyé sans sollicitation, il s'agit, pour l'émetteur, d'indiquer une nouvelle adresse "lien-local". Le champ adresse de la cible contient alors cette nouvelle adresse "lien-local" ;
- l'option adresse physique de la cible contient l'adresse physique de l'émetteur.

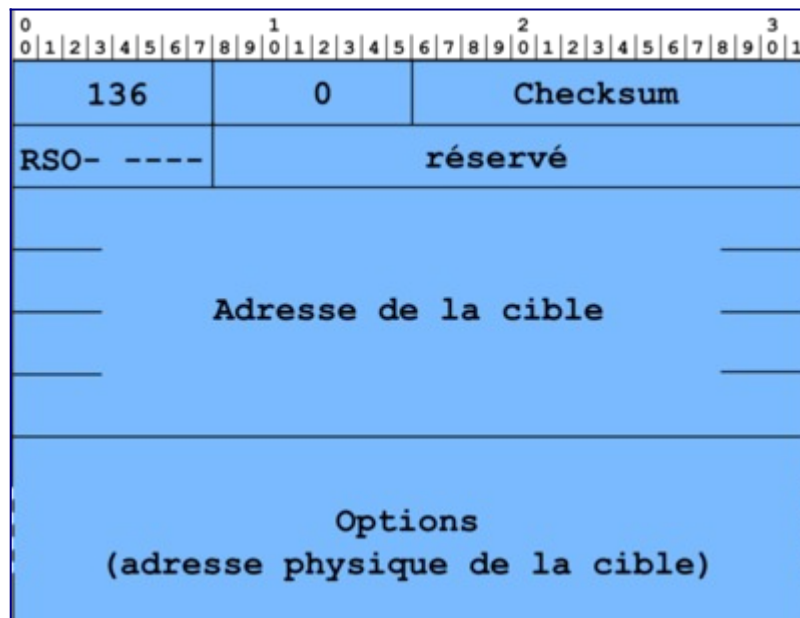


Figure 9 : Format du message d'annonce d'un voisin.

Fonctionnement de la résolution d'adresse IP

La résolution d'adresse est la procédure par laquelle l'adresse IP d'un voisin est mise en correspondance avec son adresse physique. C'est la même fonction qu'ARP en IPv4. Les messages utilisés seront NS et NA dont nous venons de voir le format. Pour illustrer le fonctionnement de la résolution d'adresse par NDP, nous prenons l'exemple indiqué par la figure 10 dans lequel les deux nœuds sont sur le même lien. Sur la figure, les adresses physiques, dites MAC, et IPv6 sont indiquées. Pour chaque niveau d'adresse, les adresses multicast, en plus des adresses unicast, sont indiquées.

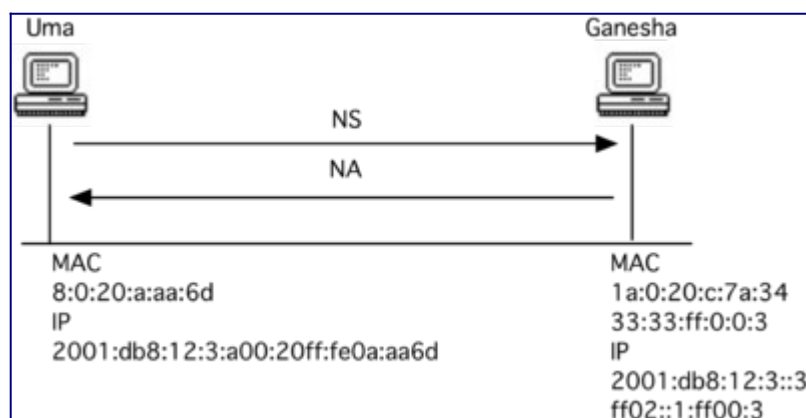


Figure 10 : Lien utilisé comme exemple pour la résolution d'adresse IPv6.

Le nœud Uma essaye de tester la connectivité avec Ganesha via la commande ping6. La commande entrée sur Uma est la suivante :

```

uma# ping6 ganesha
trying to get source for ganesha
source should be 2001:db8:12:3:a00:20ff:fe0a:aa6d
PING ganesha (2001:db8:12:3::3): 56 data bytes
64 bytes from 2001:db8:12:3::3: icmp6_seq=0 ttl=255 time=5.121 ms
    
```

Commande ping6

La commande ping6 est l'équivalent de la commande ping d'IPv4 mais, comme son numéro l'indique, en utilisant le protocole IPv6. La commande ping, dans certains OS, comporte une option -6 qui rend cette commande équivalente à la commande ping6.

Avant de pouvoir émettre un paquet IPv6 sur le réseau, l'émetteur a besoin de connaître l'adresse physique du nœud destinataire. Dans notre exemple, le nœud destinataire est le destinataire final, autrement dit le récepteur. Dans d'autre situation, le destinataire est le nœud destinataire de la transmission comme le routeur du lien (*Next hop*). L'émetteur utilise le protocole de découverte des voisins pour découvrir l'adresse physique. Par conséquent, il commence la résolution par l'émission d'un message de sollicitation d'un voisin (NS), comme le montre la trace 3.

```
Ethernet Src : 8:0:20:a:aa:6d Dst : 33:33:ff:0:0:3 Type : 0x86dd
IPv6
Version : 6 Classe : 0xf0 Label : 000000
Longueur : 32 octets (0x0020) Protocole : 58 (0x3a, ICMPv6)
Nombre de sauts : 255 (0xff)
Source : 2001:db8:12:3:a00:20ff:fe0a:aa6d (uma)
Desti. : ff02::1:ff00:3 (multicast sollicité associé à 2001:db8:12:3::3)
ICMPv6
Type : 135 (0x87, Sollicitation de voisin) Code : 0 Checksum : 0x4d7f
Cible : 2001:db8:12:3::3 (ganesha)
Option :
Type : 1 (Adresse physique source) Lg : 8 octets (0x01) : 08-00-20-0a-aa-6d

0000: 6f 00 00 00 00 20 3a ff 20 01 0d b8 00 12 00 03
0010: 0a 00 20 ff fe 0a aa 6d ff 02 00 00 00 00 00 00
0020: 00 00 00 01 ff 00 00 03|87|00|4d 7f|00 00 00 00|
0030: 20 01 0d b8 00 12 00 03 00 00 00 00 00 00 00 03|
0040: 01|01|08 00 20 0a aa 6d
```

Trace 3: Message ICMPv6 Sollicitation de voisin(NS).

Dans l'en-tête IPv6, l'adresse de la source est l'adresse globale de l'interface d'émission d'Uma. On aurait pu penser que l'émetteur utiliserait l'adresse locale au lien comme adresse de source. L'utilisation de l'adresse source globale, comme on le verra par la suite, permet au destinataire de remplir directement sa table de correspondance avec l'adresse physique associée à l'adresse IPv6 de l'émetteur puisque le destinataire trouvera dans l'option du message NS l'adresse physique de l'émetteur.

L'adresse de destination est l'adresse de multicast sollicitée associée à l'adresse recherchée, et l'adresse Ethernet de destination est l'adresse associée à cette adresse multicast [RFC 2464].

Le message NS apparaît en bleu dans la trace. Le format du message est représenté par la figure 8. Ce message NS contient, dans le champ cible, l'adresse IPv6 du nœud pour laquelle l'adresse physique est recherchée. Dans notre cas, il s'agit de l'adresse de Ganesha. On peut remarquer que les trois derniers octets correspondent au groupe de multicast de l'adresse de destination dans l'en-tête IPv6. Le champ option contient l'adresse physique de l'émetteur de la requête, à savoir celle d'Uma.

Le nœud Ganesha, qui écoute les groupes multicast dont le ou les groupes multicast sollicités associés à ses adresses, reçoit le message NS. Il reconnaît dans le champ *Cible* une de ses adresses IPv6. Il y répond par un message NA dont le format est rappelé par la figure 9. La trace 4 montre la réponse émise.

```
Ethernet Src : 1a:0:20:c:7a:34 Dst : 8:0:20:a:aa:6d Type : 0x86dd
IPv6
Version : 6 Classe : 0xf0 Label : 000000
Longueur : 32 octets (0x20) Protocole : 58 (0x3a, ICMPv6)
Nombre de sauts : 255 (0xff)
Source : fe80::1800:20ff:fe0c:7a34 (ganesha, lien-local)
Desti. : 2001:db8:12:3:0a00:20ff:fe0a:aa6d (uma)
ICMPv6
Type : 136 (0x88, Annonce de voisin) Code : 0 Checksum : 0xd7fb
Bits (0x7) R = 1, S = 1, O = 1
Cible : 2001:db8:12:3::3 (ganesha)
Option :
Type : 2 (Adresse physique cible) Lg : 8 octets (0x01) : 1a-00-20-0c-7a-34
```

Trace 4 : Message ICMPv6 Annonce de voisin(NA).

L'adresse source utilisée par Ganesha est celle de portée locale au lien. Le bit R indique que le nœud qui répond a une fonction de routeur. Le bit S indique que ce message est une réponse à une demande explicite (le message précédent). Le bit O indique que cette réponse doit remplacer toute valeur connue précédemment. Le champ *Cible* rappelle l'adresse IPv6. Le champ *Option* donne l'adresse physique recherchée.

L'adresse physique ainsi obtenue est ensuite enregistrée dans une table de correspondance du nœud émetteur, appelée *cache des voisins*. De cette manière, l'émetteur n'a pas besoin de redemander l'adresse physique d'un même destinataire à chaque paquet. Ce cache est maintenu à jour par une procédure de détection d'injoignabilité (*Neighbor Unreachability Detection* (NUD)), reposant sur les mêmes messages.

Un fois la résolution d'adresse terminée, les messages ICMPv6 pour le test d'accessibilité peuvent être échangés. Ces messages "Demande d'écho" et "Réponse d'écho" ont été présentés précédemment dans le paragraphe "Test d'accessibilité d'un nœud".

Tant que la commande ping6 n'est pas arrêtée, les échanges de messages d'écho s'effectuent alors à intervalle de temps régulier. Au bout d'un certain temps, et périodiquement, les nœuds vérifieront que leur voisin est toujours correct en utilisant la procédure NUD. Le voisin a pu tomber en panne ou être remplacé avec changement d'adresse Ethernet. Aussi, de temps en temps, chaque nœud va émettre un message NS. Une réponse NA (avec le bit S) confirmera que le voisin (ici le correspondant) est toujours valide. Nous montrons par les traces 5 et 6 un échange NUD. Il s'agit du nœud Ganesha qui lance une vérification de la validité du nœud Uma.

```
IPv6
Version : 6 Classe : 0x00 Label : 000000
Longueur : 32 octets (0x20) Protocole : 58 (0x3a, ICMPv6)
Nombre de sauts : 255 (0xff)
Source : fe80::1800:20ff:fe0c:7a34 (ganesha, lien-local)
```



```

Desti. : 2001:db8:12:3:a00:20ff:fe0a:aa6d (uma)
ICMPv6
Type : 135 (0x87, Sollicitation de voisin) Code : 0 Checksum : 0x1116
Cible : 2001:db8:12:3:a00:20ff:fe0a:aa6d (ganesha)
Option :
Type : 1 (Adresse physique source) Lg : 8 octets (0x01) : 1a-00-20-0c-7a-34

0000: 6000 0000 0020 3aff fe80 0000 0000 0000
0010: 1800 20ff fe0c 7a34 2001 0db8 0012 0003
0020: 0a00 20ff fe0a aa6d|8700 1116 0000 0000
0030: 2001 0db8 0012 0003 0a00 20ff fe0a aa6d
0040: 0101 1a00 200c 7a34
    
```

Trace 5 : Message ICMPv6 Sollicitation de voisin(NS).

On remarque que le message de sollicitation est envoyé par une communication unicast avec l'adresse IPv6 qui est enregistrée dans les tables de correspondance. Si une réponse n'arrive pas, le nœud émetteur effacera l'entrée de son cache "Résolution de voisin". Tout trafic ultérieur reprendra l'enquête de résolution au début, avec utilisation de l'adresse multicast sollicitée.

La réception du message "Annonce voisin" (NA) par Ganesha apporte la confirmation que son voisin est toujours accessible. Ce dernier, qui est Uma, indique son adresse dans le champ cible du message d'annonce de voisin.

```

IPv6
Version : 6 Classe : 0x00 Label : 000000
Longueur : 24 octets (0x18) Protocole : 58 (0x3a, ICMPv6)
Nombre de sauts : 255 (0xff)
Source : 2001:db8:12:3:a00:20ff:fe0a:aa6d (uma)
Desti. : fe80::1800:20ff:fe0c:7a34 (ganesha, lien-local)
ICMPv6
Type : 136 (0x88, Annonce de voisin) Code : 0 Checksum : 0x855f
Bits (0x4) R = 0, S = 1, O = 0
Cible : 2001:db8:12:3:a00:20ff:fe0a:aa6d (uma)

0000: 6000 0000 0018 3aff 2001 0db8 0012 0003
0010: 0a00 20ff fe0a aa6d fe80 0000 0000 0000
0020: 1800 20ff fe0c 7a34|8800 855f 4000 0000
0030: 2001 0db8 0012 0003 0a00 20ff fe0a aa6d
    
```

Trace 6 : Message ICMPv6 Annonce de voisin(NA).

Fonctionnement de la détection d'adresse dupliquée

Pour vérifier l'unicité des adresses lien-local ou unicast qui viennent d'être configurées manuellement ou automatiquement sur leurs interfaces, les nœuds doivent exécuter un algorithme de "Détection d'Adresse Dupliquée" (DAD) avant de les utiliser [[RFC 4862](#)]. L'algorithme utilise les messages ICMPv6 "Sollicitation d'un voisin" et "Annonce d'un voisin". Si une adresse déjà en service est découverte, elle ne pourra être attribuée à l'interface. L'auto-configuration s'arrête et une intervention humaine devient obligatoire.

Pourquoi arrêter l'auto-configuration en cas d'échec de la DAD ?

Lorsque la DAD échoue, cela veut dire que l'unicité de l'adresse n'est plus. Dans le [RFC 4429](#), il est proposé d'anticiper une réponse négative du DAD (i.e. pas d'adresse dupliquée) afin d'utiliser l'adresse de manière anticipée. Dans ce RFC, on trouve, en Annexe A, une étude de la probabilité d'une collision d'adresses. La conclusion est qu'une collision est plus probablement due à une erreur de configuration du réseau qu'à une rencontre probabiliste malheureuse. L'intervention manuelle de l'administrateur est alors, dans ces cas, souhaitable pour pouvoir corriger l'erreur. Un mécanisme de résolution automatique de collision d'adresses n'enlèverait pas l'erreur.

Une adresse est qualifiée de "provisoire" pendant l'exécution de l'algorithme DAD et ce jusqu'à la confirmation de son unicité. Une adresse provisoire ne peut servir pour les communications. Elle ne peut être utilisée dans un en-tête de paquet IPv6. On ne peut que la trouver dans le champ cible des messages de sollicitation et d'annonce d'un voisin. L'algorithme DAD consiste à envoyer un message "sollicitation d'un voisin" avec, dans le champ adresse de la cible, l'adresse provisoire. Afin de distinguer l'algorithme DAD de celui de découverte des voisins, le paquet IPv6 contenant un message de sollicitation d'un voisin a comme adresse de source l'adresse indéterminée. Trois cas se présentent :

1. Un message "Annonce d'un voisin" est reçu : l'adresse provisoire est utilisée comme adresse valide par un autre nœud. L'adresse provisoire n'est pas unique et ne peut être retenue.
2. Un message "Sollicitation d'un voisin" est reçu dans le cadre d'une procédure DAD : l'adresse provisoire est également une adresse provisoire pour un autre nœud. L'adresse provisoire ne peut être utilisée par aucun des nœuds.
3. Rien n'est reçu au bout d'une seconde (valeur par défaut) : l'adresse provisoire est unique. Elle passe de l'état "provisoire" à celui de "valide" et elle est assignée à l'interface.

La trace affichée par la commande `tcpdump` ci-dessous illustre le premier cas. Un hôte active son interface réseau, et après avoir constitué une adresse unicast par auto-configuration, effectue une DAD (nous verrons le détail de la construction de l'adresse dans la prochaine activité). Comme l'adresse provisoire n'est pas unique, un message NA est reçu pour signaler que cette adresse est une adresse valide d'un autre nœud sur le lien.

```
1 IP6 :: > ff02::1:ff02:202: ICMP6, neighbor solicitation, who has
2001:db8:b:2:fd:c8ff:fe02:202
2 IP6 2001:db8:b:2:fd:c8ff:fe02:202 > ff02::1: ICMP6, neighbor
advertisement, tgt is 2001:db8:b:2:fd:c8ff:fe02:202
```

Nota : le format de la trace consiste ici en un numéro de ligne, le protocole, l'adresse de la source, l'adresse de destination et le message ICMPv6.

À noter que la procédure DAD n'offre pas une fiabilité absolue ; notamment lorsque le lien est coupé.

Conclusion

Cette activité a présenté les fonctions assurées par le protocole ICMPv6. Ce protocole est en effet indispensable au bon fonctionnement de la couche réseau. Par ce protocole, les défauts qui peuvent affecter la connectivité IPv6 peuvent être connus et, donc, corrigés.

A l'échelle de l'Internet, le bon acheminement d'un paquet IPv6 est contrôlé et les défauts sont notifiés à l'émetteur par des messages ICMPv6. Ainsi, en cas de non livraison d'un paquet, les informations fournies par ICMPv6 vont servir à découvrir le problème et à adapter la communication si besoin est. A l'échelle du lien, c'est à travers le protocole ICMPv6 que les nœuds se découvrent entre eux, vérifient l'unicité de leurs adresses et gèrent les abonnements aux groupes multicast.

Références bibliographiques

1. ↑ IANA. [Internet Control Message Protocol version 6 \(ICMPv6\) Parameters](#)

Pour aller plus loin

RFC et leur analyse par S. Bortzmeyer :

- [RFC 1191](#) Path MTU Discovery
- [RFC 2464](#) Transmission of IPv6 Packets over Ethernet Networks
- [RFC 2710](#) Multicast Listener Discovery (MLD) for IPv6
- [RFC 2894](#) Router Renumbering for IPv6
- [RFC 3122](#) Extensions to IPv6 Neighbor Discovery for Inverse Discovery Specification
- [RFC 3971](#) SEcure Neighbor Discovery (SEND) [Analyse](#)
- [RFC 3810](#) Multicast Listener Discovery Version 2 (MLDv2) for IPv6
- [RFC 4065](#) Instructions for Seamoby and Experimental Mobility Protocol IANA Allocations
- [RFC 4429](#) Optimistic Duplicate Address Detection (DAD) for IPv6
- [RFC 4443](#) Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification [Analyse](#)
- [RFC 4620](#) IPv6 Node Information Queries
- [RFC 4861](#) Neighbor Discovery for IP version 6 (IPv6) [Analyse](#)
- [RFC 4862](#) IPv6 Stateless Address Autoconfiguration [Analyse](#)
- [RFC 4890](#) Recommendations for Filtering ICMPv6 Messages in Firewalls
- [RFC 6275](#) Mobility Support in IPv6
- [RFC 8201](#) Path MTU Discovery for IP version 6 [Analyse](#)

Activité 32: Configurer automatiquement les paramètres réseau

Vous suivez une activité de découverte 

Principe de l'auto-configuration

La précédente activité a présenté le mécanisme de découverte des voisins afin qu'un nœud connecté à un lien puisse récupérer automatiquement les adresses des autres nœuds du même lien. C'est la même philosophie qui est mise en œuvre dans la configuration automatique (ou auto-configuration) des paramètres d'une interface réseau. L'objectif de ce mécanisme est de réduire au maximum l'intervention humaine dans ce processus pour :

- que l'utilisateur possède une connectivité opérationnelle dès le branchement de l'interface réseau de son terminal ;
- que l'administrateur puisse centraliser la configuration sur un seul équipement. C'est ce dernier qui se chargera de propager la configuration aux hôtes.

Route par défaut

La route par défaut agrège l'ensemble des adresses qui ne sont pas sur le réseau local. Elle dirige le trafic vers le routeur qui a la connectivité Internet. Dans un réseau de distribution qui connecte des utilisateurs, cette route commence par le routeur connecté sur le même lien que l'hôte. Dans un réseau routier, la route par défaut correspond au panneau "Toutes Directions".

L'auto-configuration vise à fournir les informations pour que l'interface de communication au réseau d'un hôte soit opérationnelle. Il s'agit au minimum des éléments suivants :

- les informations pour déterminer l'adresse IP, ou les informations indiquant la méthode pour l'obtenir ;
- la longueur du préfixe IP du réseau ;
- l'adresse du routeur local à utiliser pour la route par défaut ;
- le serveur de noms à utiliser.

L'administrateur renseigne les informations communes pour un lien sur un nœud. Les hôtes récupèrent ces informations pour déterminer la configuration spécifique qui sera appliquée à leur interface réseau. La connexion au réseau et, dans la plupart des cas, à l'Internet, sera alors effective. L'hôte sera alors en mesure de recevoir et d'émettre des paquets IP.

L'auto-configuration est un mécanisme prévu pour les hôtes. Les nœuds intermédiaires dans l'infrastructure, comme les routeurs, étant des équipements "gérés", ils ne sont pas censés utiliser ce mécanisme. Leur configuration est à la charge de l'administrateur.

Mécanismes mis en œuvre

Avec ou sans état

Le terme 'sans état' désigne une méthode ne nécessitant pas un serveur comme DHCPv6.

Par opposition, lorsqu'un serveur dirige la configuration, la méthode est qualifiée de 'avec état'. Dans la méthode 'sans état', un hôte commence directement la procédure sans avoir recours aux informations d'un serveur comme c'est le cas avec DHCP.

L'auto-configuration se déroule en plusieurs étapes mettant en œuvre différents mécanismes :

- la toute première étape consiste à créer l'adresse "lien-local". Une fois l'unicité de cette adresse vérifiée, le nouveau nœud est en mesure de communiquer avec les autres nœuds du lien (ses voisins) ;
- le nouveau nœud doit ensuite acquérir les informations communes au lien, ainsi que la politique de configuration de l'adresse IP. Ces informations sont transmises par le routeur. S'il y a un routeur sur le lien, la machine doit appliquer la méthode indiquée par le message d'annonce de routeurs, à savoir :
 - l'auto-configuration "sans état" (*IPv6 Stateless Address Autoconfiguration* (SLAAC)) [[RFC 4862](#)],
 - ou l'auto-configuration "avec état" (par DHCPv6) [[RFC 3315](#)] ;
- les informations transmises par le routeur permettent de plus, au nœud, de configurer sa table de routage.
- enfin, toujours en fonction de la politique de configuration, le nœud va récupérer d'autres informations nécessaires à la configuration dont, notamment, le serveur de noms.

En l'absence de routeur sur le lien, le nœud doit essayer d'acquérir l'adresse unicast globale par la méthode d'auto-configuration "avec état". Si la tentative échoue, c'est terminé. Les communications se feront uniquement sur le lien avec l'adresse "lien-local". Le nœud n'a pas d'adresse avec une portée qui l'autorise à communiquer avec des nœuds autres que ceux de son lien d'attachement.

La création de l'adresse "lien-local"

À l'initialisation de son interface, le nouveau nœud construit un identifiant pour l'interface, qui doit être unique sur le lien. Cet identifiant utilise l'adresse EUI-64. Le principe de base de la création d'adresse unicast IPv6, tel que vu dans la première séquence, est de compléter un préfixe réseau avec l'identifiant. L'adresse "lien-local" est donc créée en prenant le préfixe "lien-local" (fe80::/64) standardisé pour cet usage.

L'adresse ainsi constituée est encore interdite d'usage. Elle possède un état provisoire (*tentative*) car le nœud doit vérifier l'unicité de cette adresse sur le lien au moyen de la procédure de détection d'adresse dupliquée (DAD), présentée dans l'activité précédente. Si le nœud détermine que l'adresse "lien-local" n'est pas unique, l'auto-configuration s'arrête et une intervention manuelle est nécessaire.

Une fois que l'assurance sur l'unicité de l'adresse "lien-local" est obtenue, l'adresse provisoire devient une adresse valide pour l'interface. L'adresse est allouée à l'interface. La première étape de l'auto-configuration est achevée.

Découverte des paramètres communs au réseau

L'objectif du nœud qui configure son interface de communication est maintenant d'allouer une adresse IP routable. C'est avec cette adresse qu'il pourra effectuer des communications "inter-liens". La seconde étape de l'auto-configuration consiste à récupérer les informations communes au lien d'attachement de l'hôte en phase d'auto-configuration. Ces informations sont fixées par l'administrateur et localisées sur le ou les routeurs du lien. Le ou les routeurs se chargeront de propager les informations communes aux systèmes d'extrémité. A noter que les routeurs ne rentrent pas dans le périmètre de l'auto-configuration car ils restent sous la responsabilité de l'administrateur qui aura en charge de les configurer.

Dans cette étape d'auto-configuration, l'hôte vise à obtenir du routeur local les instructions et les informations pour continuer le processus de configuration. Ceci est fait, soit en écoutant les messages d'annonce (RA) émis périodiquement par le routeur, soit en envoyant une requête (RS) au routeur. Ces échanges sont réalisés au moyen de messages ICMPv6 :

- sollicitation d'un routeur, noté RS (*Router Solicitation*) (voir la figure 1). Ce message ICMPv6 est identifié par le champ type de valeur 133 ;
- annonce de routeur, noté RA (*Router Advertisement*) (voir la figure 2). Le message ICMPv6 d'annonce de routeur est identifié par le champ type de valeur 134.

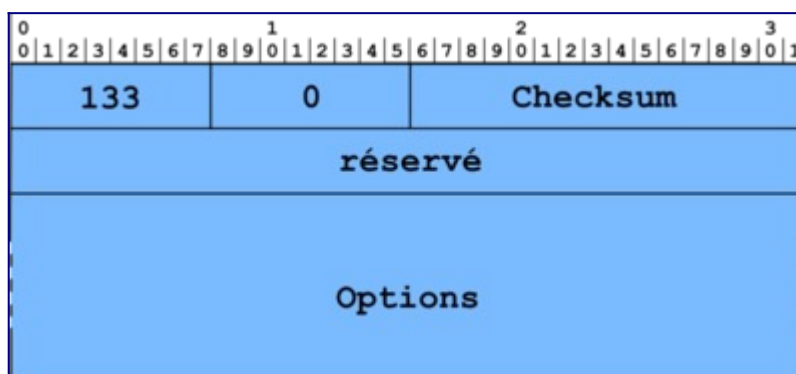


Figure 1 : Format du message de sollicitation d'un routeur.

La sollicitation d'un routeur forme une requête émise par le nœud. Le message RS est envoyé à destination de l'adresse IPv6 de multicast réservée aux routeurs sur le même lien `ff02::2`. Le champ `option` contient normalement l'adresse physique du nœud demandeur.

Un routeur émet périodiquement le message RA, ou il l'émet en réponse à un message de sollicitation (RS) d'un nœud. Le champ `adresse source` dans le paquet IPv6 contient l'adresse locale au lien du routeur. La destination du message RA est soit le nœud qui a émis la sollicitation, soit le groupe multicast de tous les nœuds du lien identifié par l'adresse `ff02::1`. Le message RA est primordial dans le fonctionnement d'un réseau IPv6, car en plus de délivrer les informations nécessaires à l'auto-configuration, il notifie régulièrement auprès des nœuds la présence du ou des routeurs afin de confirmer la connectivité "inter-lien".

Nota : ces messages peuvent être la source de nombreux problèmes lorsqu'il sont envoyés par des équipements configurés par défaut avec maladresse ou intentionnellement avec de mauvaises informations (tentative de détournement de trafic par des routeurs usurpateurs par

exemple) comme le note le [RFC 6104](#). Lors de tentatives d'usurpation, ces messages sont même qualifiés de RAcailles[1].

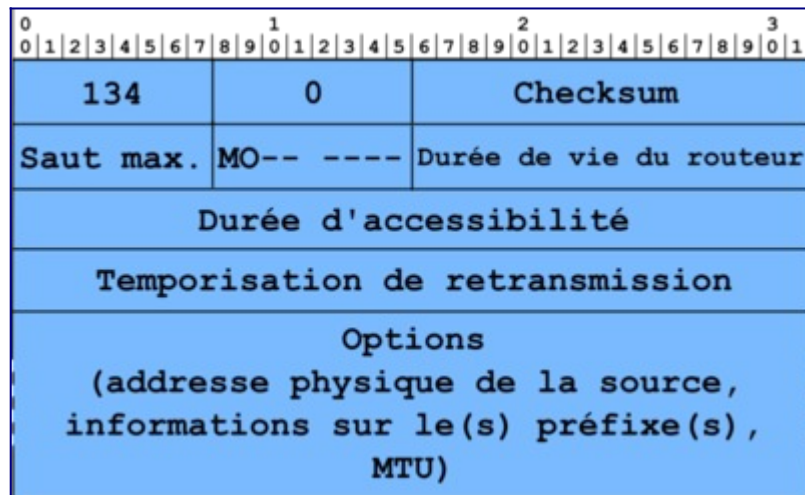


Figure 2 : Format du message d'annonce de routeur.

Le message RA contient un ensemble d'informations propres au routeur et à la politique de configuration du réseau. Parmi les informations propres au routeur, nous avons les champs suivants :

- durée de vie du routeur : il donne, en secondes, la période pendant laquelle le routeur exercera les fonctions de routeur par défaut ;
- durée d'accessibilité : ce champ indique la durée, en millisecondes, pendant laquelle une information de ce message contenue dans le cache d'un nœud peut être considérée comme valide ; par exemple, la durée de validité d'une entrée dans la table de correspondance entre adresse IPv6 et adresse physique. Au bout de cette période, la procédure de découverte de non-joignabilité (NUD) est entreprise pour vérifier la pertinence de l'information ;
- temporisation de retransmission : ce champ donne, en millisecondes, la période entre deux émissions non sollicitées du message RA. Il sert aux autres nœuds à détecter une inaccessibilité du routeur.

Communiquée au nœud qui se configure, la politique de configuration indique les mécanismes d'auto-configuration à utiliser. Cette politique est définie par deux bits du message d'annonce de routeur :

- le bit M (*Managed address configuration*) mis à 1, indique que le nœud doit explicitement demander son adresse auprès d'un serveur d'adresses et donc, utiliser la configuration "avec état" de l'adresse IP. Si ce bit est à 0, alors le mécanisme de configuration "sans état" doit être utilisé pour construire une adresse IPv6 ;
- le bit O (*Other stateful configuration*) mis à 1, indique que le nœud doit interroger le serveur de configuration pour obtenir des paramètres autres que l'adresse. Si ce bit est à 0, les paramètres de configuration sont inclus dans le message d'annonce de routeur au moyen d'options spécifiques.

L'auto-configuration "sans état" pour une adresse IP routable

Le principe de base de l'auto-configuration "sans état" de l'adresse IP est qu'un nœud génère son adresse IPv6 à partir d'informations locales (son identifiant d'interface) et de préfixes éventuellement reçus du routeur. Le routeur communique au nœud les informations sur le préfixe utilisé sur son lien au moyen d'une option incluse dans le message ICMPv6 d'annonce de routeur [RFC 4861]. Cette option est présentée par la figure 3.

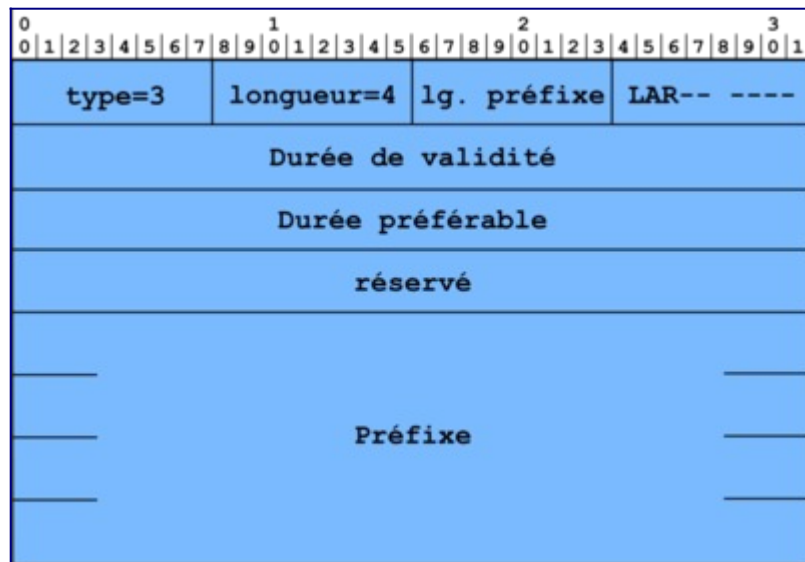


Figure 3 : Format de l'option d'information sur le préfixe.

L'option information sur le préfixe est composée par les champs suivants :

- type, de valeur 3, identifie cette option ;
- longueur indique le nombre de mots de 64 bits de l'option. Dans le cas de cette option d'information, la longueur vaut 4 ;
- lg. préfixe indique combien de bits sont significatifs pour le préfixe annoncé ;
- le bit L (*On link*) signifie, quand il est à 1, que le préfixe indique que les autres nœuds partageant le même préfixe sont sur le même lien. L'émetteur peut donc directement les joindre. Dans le cas contraire, le nœud émet le paquet vers le routeur. Si ce dernier sait que l'émetteur peut joindre directement le destinataire, il notifiera l'émetteur par un message ICMPv6 d'indication de redirection ;
- le bit A (*Autonomous address-configuration*) indique, quand il est à 1, que le préfixe annoncé peut être utilisé pour construire l'adresse IP du nœud ;
- durée de validité indique, en secondes, la durée pendant laquelle le préfixe est valide ;
- durée préférable indique la durée de préférence, en secondes, pour une adresse construite avec l'auto-configuration "sans état". Pour ce champ et celui de durée de validité, une valeur de 0xffff ffff représente une durée infinie ;
- réservé est là uniquement pour aligner le préfixe (le champ suivant) sur une frontière de mot de 64 bits ;
- préfixe contient la valeur de préfixe annoncé sur le lien. Pour maintenir un alignement sur 64 bits pour le reste des données du paquet, ce champ a une longueur fixe de 128

bits.

Interprétation du bit L

L'interprétation du bit L à 0 (signifiant implicitement "off-link") a fait l'objet de précisions complémentaires aux définitions originales du [RFC 4861](#) (Neighbor Discovery in IPv6). Notamment dans le [RFC 4943](#) (IPv6 Neighbor Discovery On-Link Assumption Considered Harmful) de 2007 puis dans le [RFC 5942](#) (IPv6 Subnet Model : The Relationship between Links and Subnet Prefixes) de 2010. Il s'agissait de clarifier le comportement du nœud pour la procédure de découverte du voisinage dans des situations particulières, notamment lorsque la liste des routeurs par défaut est vide.

Sur les réseaux locaux s'appuyant sur des protocoles de niveau 2 en mode diffusion (cas des réseaux Ethernet), le bit L est généralement à 1 (on-link). Par contre, sur les réseaux mobiles ou pour les protocoles d'itinérance tel que MIPv6 ([RFC 6275](#)) la situation de localisation d'un nœud, relativement au lien, peut varier en fonction de son état d'itinérance (cas des téléphones mobiles changeant de cellule par exemple) et nécessiter l'usage d'un intermédiaire (un routeur ou un proxy PMIPv6 [RFC 5213](#), [RFC 6543](#) et [RFC 7864](#)), pour la découverte du voisinage. L'état "off-link" d'une adresse est alors significatif.

Au passage, la précaution de forcer le champ "Hop-limit" à la valeur maximale à 255 pour les paquets de découverte de voisins, protège des nœuds hors lien qui émettraient accidentellement, ou par malice, des messages ND.

L'introduction, dans les infrastructures de type "cloud" de nouveaux protocoles, tels que VXLAN, qui permettent d'étendre les domaines de diffusion sur plusieurs centres de données (*data centers*) distants ajoutent de nouvelles situations où le mode hors lien peut être significatif.

Comme pour la création de l'adresse "lien-local", l'adresse IP unicast routable est obtenue en concaténant le préfixe avec l'identifiant de l'interface. L'adresse IP unicast routable est une adresse utilisable pour des communications non limitées au lien d'attachement du nœud. Une adresse routable est soit une adresse de type ULA soit une adresse tirée du plan d'adressage global (GUA). Le préfixe de l'adresse routable provient ici du message d'annonce de routeurs, et plus précisément de l'option « information sur le préfixe ». Pour construire son adresse, le nœud est ensuite libre de choisir l'identifiant d'interface créé à partir de l'adresse MAC [[RFC 4291](#)] ou généré selon un autre principe, comme le tirage aléatoire [[RFC 4941](#)]. Profitant de la souplesse offerte par IPv6, le nœud peut de plus créer autant d'adresses qu'il souhaite.

Les valeurs de durée préférable et de durée de validité contrôlent le cycle de vie des adresses créées. Une fois la durée préférable écoulée, l'adresse concernée passera de l'état préféré à l'état déprécié comme le montre la figure 4. Le temps écoulé se mesure à partir de la réception du message d'annonce d'un routeur. Et, lorsque le temps, indiqué par la durée de validité, sera écoulé, l'adresse passera à l'état invalide. Des messages d'annonces avec des valeurs spécifiques peuvent permettre, par exemple, de contrôler l'utilisation par les nœuds d'adresses construites à partir de certains préfixes. Les champs de durée peuvent servir dans la renumérotation lors du passage d'un fournisseur d'accès à un autre ; c'est-à-dire d'un préfixe à un autre.

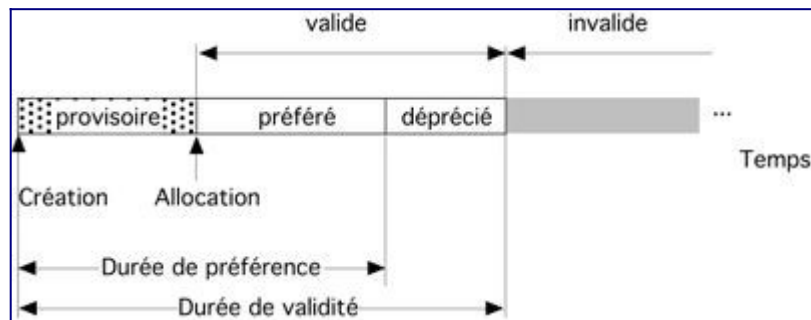


Figure 4 : Durée associée aux états d'une adresse auto-configurée.

L'auto-configuration "avec état" de l'adresse IP routable

Cette méthode de configuration d'adresse, qui sera présentée dans l'activité suivante, repose sur la présence d'un serveur d'adresses contenant une base d'adresses IP disponibles sur le réseau. Le nœud va solliciter le serveur en utilisant le protocole DHCPv6.

Un nœud recevant un message d'annonce de routeur est donc supposé initier un dialogue avec un serveur DHCPv6 si ce message présente le bit M avec pour valeur 1 (voir la figure 2). Mais ce comportement, tel que prévu dans les standards, n'est pas entièrement mis en œuvre dans les systèmes d'exploitation actuels, et il est très souvent nécessaire d'explicitier l'usage de DHCPv6 au nœud, alors que cette information est fournie par le réseau.

La configuration de la table de routage

En IPv6, seuls les routeurs utilisent des protocoles de routage pour remplir leur table de routage. Le routage des autres nœuds repose sur la notion de route directe (ou locale) et de route par défaut.

La route locale, c'est-à-dire la route vers les adresses du même lien, est définie à partir des informations présentes dans l'option concernant le préfixe réseau. En partant du champ préfixe et de sa longueur, le nœud peut déterminer les bits communs aux adresses IP connectées au même lien. L'acheminement des paquets à destination de ces adresses ne nécessitera pas de routeur. Le nœud destinataire est localisé sur le même lien. Le nœud émetteur effectue alors une remise directe en utilisant l'adresse de niveau liaison (par exemple adresse Ethernet) découverte par la résolution d'adresse.

La route par défaut passe à travers le routeur local du lien. Elle est configurée grâce à l'adresse "lien-local" contenue dans le champ source du paquet IPv6 contenant le message ICMPv6 RA. L'adresse physique du routeur est de plus contenue dans une des options du message. L'émetteur d'un paquet vers un nœud à l'extérieur du lien utilisera donc cette adresse comme premier saut pour l'acheminement du paquet.

Cependant, lorsqu'il y a plusieurs routeurs et donc plusieurs routes possibles sur un lien, le message d'annonce de routeur peut contenir l'option d'information de route [RFC 4191]. Ce message va pouvoir indiquer la ou les routes desservies par le routeur.

La découverte des serveurs DNS

L'auto-configuration IPv6 "sans état", telle que spécifiée dans le [RFC 4862](#), n'a pas prévu de mécanisme de découverte automatique des serveurs DNS. En revanche, il était prévu que ces informations complémentaires soient fournies par l'auto-configuration "avec état". Les spécifications du protocole d'auto-configuration "avec état" par DHCPv6 ont pris du temps (six ans environ) pour être publiées dans le [RFC 3315](#). En l'absence d'information de configuration sur les serveurs DNS locaux, un hôte n'est pas capable de résoudre des noms de domaines en adresses IPv6. Pour ne pas freiner le déploiement d'IPv6, trois propositions ont émergé de l'IETF pour mettre en oeuvre un mécanisme de découverte automatique du DNS. Ces propositions ont été produites par le groupe de travail DHC (*Dynamic Host Configuration*) et DNSOP (*Domain Name System Operations*). Les co-auteurs des trois propositions ont conjointement rédigé un document synthétique [\[RFC 4339\]](#) décrivant, pour chaque technique, le fonctionnement ainsi que les scénarios d'utilisation. Ce document donne également des recommandations pratiques quant à la solution ou à la combinaison de solutions à adopter en fonction de l'environnement technique dans lequel se trouvent les équipements à configurer.

Une des propositions s'appuie sur des adresses anycast bien connues (*Well-known anycast addresses*). L'idée est que les demandes au service DNS seraient émises par les clients vers une adresse anycast. Le réseau routerait alors la requête jusqu'au serveur DNS le plus proche. Cette proposition semble avoir été abandonnée et n'a pas été reprise ailleurs.

La première proposition mise en oeuvre repose sur le protocole DHCP et l'option *DHCPv6 DNS Recursive Name Server* spécifiée dans le [RFC 3646](#). Un serveur DHCPv6 dit "sans état" [\[RFC 3736\]](#) n'alloue pas d'adresses IPv6 mais informe simplement les clients des différents paramètres à utiliser : serveur DNS, serveur NTP, serveur d'impression... Depuis, le protocole DHCPv6 pour serveur "avec état" a été développé [\[RFC 3315\]](#). En plus des informations de configuration, il alloue les adresses IPv6. Nous verrons son fonctionnement dans la prochaine activité de ce cours.

La seconde proposition, appelée ND RDNSS (*Neighbor Discovery Recursive DNS Server*), a été développée sur la base des messages ICMPv6 d'annonce de routeur [\[RFC 8106\]](#). ND RDNSS consiste à ajouter à l'annonce du routeur une option pour l'information du DNS.

La disponibilité des mécanismes DHCPv6 et ND RDNSS dépend des systèmes d'exploitation [\[2\]](#).

Exemple de configuration automatique

Par un exemple, nous allons illustrer les différentes étapes de l'auto-configuration et les messages échangés entre le nœud et le routeur du lien comme montré par la figure 5.

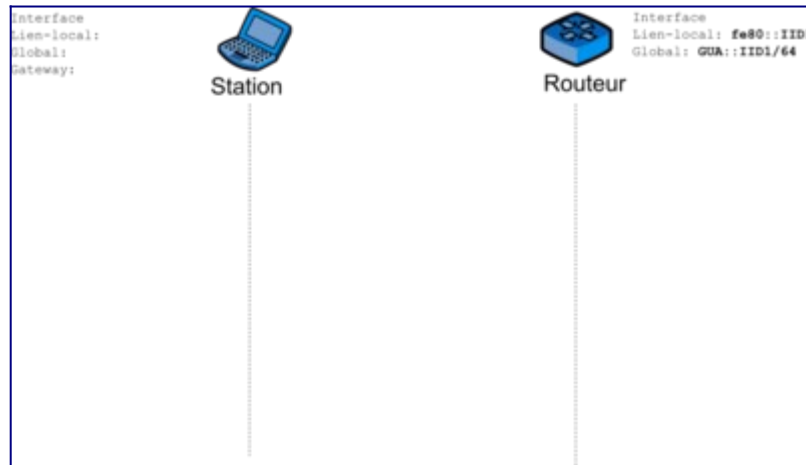


Figure 5 : Configuration de l'exemple.

Préalablement à l'attachement du nœud au lien, le routeur local est configuré avec le préfixe IPv6 à utiliser sur ce lien. Le nœud, à l'activation de son interface de communication au réseau, crée une adresse "lien-local" provisoire à partir de l'adresse matérielle de son interface. Afin de vérifier si cette adresse est unique, le nœud débute la procédure de détection d'adresse dupliquée (DAD) (voir la figure 6).

Création d'un identifiant d'interface

La méthode de création à partir de l'adresse physique MAC est expliquée dans l'activité "Utilisation des adresses sur une interface réseau" de la séquence 1. En résumé, cela consiste à inverser la valeur du 7^e bit de l'octet de poids fort de l'adresse physique et d'ajouter au milieu de l'adresse MAC, soit après le 3^e octet, la valeur 0xFFFE.

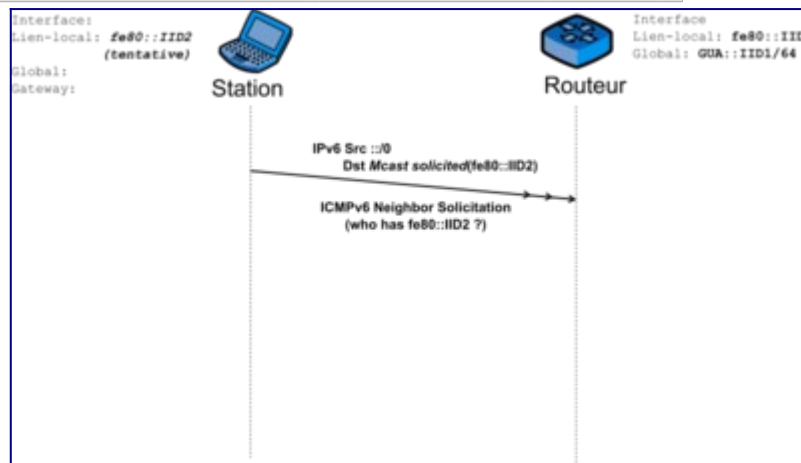


Figure 6 : Procédure DAD.

Comme décrit dans l'activité précédente, la station émet un message de sollicitation d'un voisin (NS) à l'adresse "multicast sollicitée" associée à son adresse provisoire. L'adresse de source du message est indéterminée car l'état de l'adresse provisoire ne permet pas de l'utiliser pour les communications. L'adresse dont l'unicité est vérifiée est placée dans le champ cible. La capture ci-dessous montre le message ICMPv6 NS émis.

```
Ethernet Src : 8:0:20:a:aa:6d Dst : 33:33:ff:a:aa:6d Type : 0x86dd
IPv6
```

```
Version : 6 Priorité : 0xf0 Label: 000000
Longueur : 24 octets (0x0018) Protocole : 58 (0x3a, ICMPv6)
Nombre de sauts : 255 (0x0ff)
Source : ::
Desti. : ff02::1:ff0a:aa6d (multicast sollicité associé à l'adresse cible)
ICMPv6
Type : 135 (0x87, Sollicitation d'un voisin) Code : 0 Checksum : 0xfe37
cible : fe80::0a00:20ff:fe0a:aa6d (lien-local)

0000: 6f 00 00 00 00 18 3a ff 00 00 00 00 00 00 00
0010: 00 00 00 00 00 00 00 00 ff 02 00 00 00 00 00
0020: 00 00 00 01 ff 0a aa 6d|87 00 fe 37 00 00 00
0030: fe 80 00 00 00 00 00 00 0a 00 20 ff fe 0a aa 6d
```

Si une réponse est reçue sous forme d'un message d'annonce d'un voisin, le mécanisme d'auto-configuration échoue et une intervention humaine est nécessaire. Si aucune réponse n'est reçue à ce message dans les 2 secondes suivant sa diffusion, la station considère son adresse "lien-local" comme unique. L'adresse perd son état provisoire et devient valide.

Cette première étape terminée, la station possède donc une adresse "lien-local" pour communiquer avec les nœuds présents sur le même lien (ses voisins). Elle va chercher maintenant à obtenir les informations de configuration afin de pouvoir communiquer avec des nœuds en dehors du réseau. La station émet pour cela un message ICMPv6 RS à destination de tous les routeurs du lien en utilisant l'adresse multicast correspondante : ff02::2 comme indiqué par la figure 7.

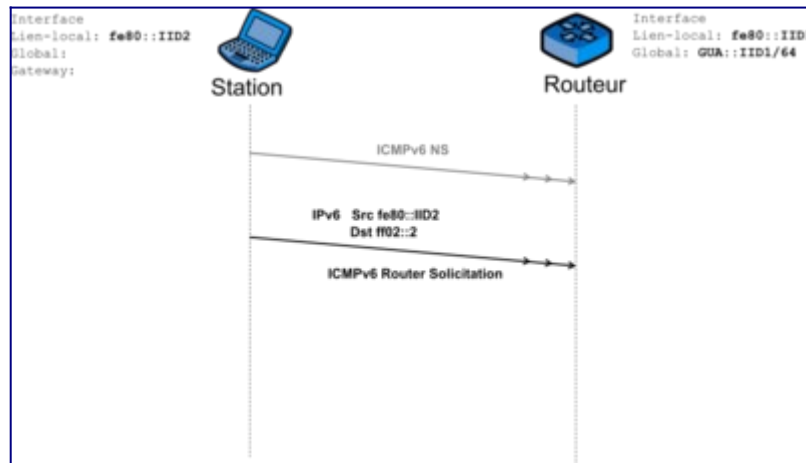


Figure 7 : Demande du préfixe IPv6 pour une adresse IPv6 non local.

Le message ainsi émis est présenté ci-dessous sous sa forme capturée :

```
Ethernet Src : 8:0:20:a:aa:6d Dst : 33:33:0:0:0:2 Type : 0x86dd
IPv6
Version : 6 Priorité : 0xf0 Label: 000000
Longueur : 16 octets (0x0010) Protocole : 58 (0x3a, ICMPv6)
Nombre de sauts : 255 (0x0ff)
Source : fe80::a00:20ff:fe0a:aa6d (lien-local)
Desti. : ff02::2 (multicast, tous les routeurs du lien)
ICMPv6
Type : 133 (0x85, Sollicitation du routeur) Code : 0 Checksum : 0xd63e
Option :
Type : 1 (Adresse physique source) Lg : 8 octets (0x01) : 08-00-20-0a-aa-
```

6d

```
0000: 6f 00 00 00 00 10 3a ff fe 80 00 00 00 00 00
0010: 0a 00 20 ff fe 0a aa 6d ff 02 00 00 00 00 00
0020: 00 00 00 00 00 00 00 02|85 00 d6 3e 00 00 00 00|
0030: 01 01 08 00 20 0a aa 6d
```

Si un routeur est présent, un message ICMPv6 RA est reçu par la station se configurant. Elle y trouve les instructions d'auto-configuration par les bits M et O, ainsi que les informations sur le ou les préfixes du lien. La figure 8 montre la réception du message d'annonce du routeur.

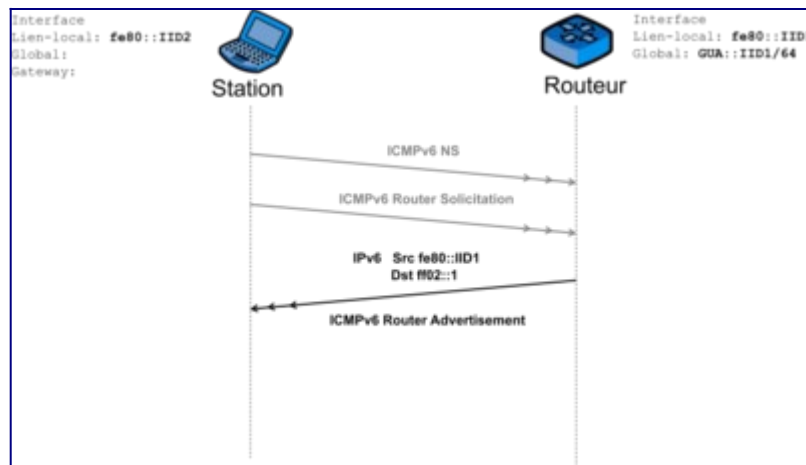


Figure 8 : Réception d'un message RA.

Le message "Annonce de routeur" est émis vers le groupe de tous les nœuds IPv6 du lien. Le drapeau A étant positionné, le préfixe annoncé peut alors servir à la construction d'une adresse IPv6 routable. La durée de validité de cette adresse n'est pas limitée. La station se construit donc l'adresse 2001:db8:12:3:a00:20ff:fe0a:aa6d à partir du préfixe et de l'identifiant d'interface.

```
Ethernet Src : 1a:0:20:c:7a:34 Dst : 33:33:0:0:0:1 Type : 0x86dd
IPv6
Version : 6 Priorité : 0xf0 Label: 000000
Longueur : 56 octets (0x0038) Protocole : 58 (0x3a, ICMPv6)
Nombre de sauts : 255 (0xff)
Source : fe80::1800:20ff:fe0c:7a34 (routeur, lien-local)
Desti. : ff02::1 (multicast, tous les noeuds du lien)
ICMPv6
Type : 134 (0x86, Annonce de routeurs) Code : 0 Checksum : 0x773c
Nombre de sauts : 0 (non précisé) Gestion d'adresse : 0 (Pas de DHCP)
Validité : 6000 secondes (0x1770) Timers : 0, 0 (non précisés)
Options :
Type : 1 (Adresse physique source) Lg : 8 octets (0x01) : 1a-00-20-0c-7a-34
Type : 3 (Information sur le préfixe) Lg : 32 octets (0x04)
Drapeaux : L=1, A=1
Durée de validité : -1 (infinie)
Durée préférable : -1 (infinie)
Préfixe : 2001:db8:12:3::/64

0000: 6f 00 00 00 00 38 3a ff fe 80 00 00 00 00 00
0010: 18 00 20 ff fe 0c 7a 34 ff 02 00 00 00 00 00
0020: 00 00 00 00 00 00 00 01|86 00 77 3c 00 00 17 70
```

```
0030: 00 00 00 00 00 00 00 00 | 01 01 1a 00 20 0c 7a 34 |
0040: 03 04 40 c0 ff ff ff ff ff ff ff ff 00 00 00 00
0050: 20 01 0d b8 00 12 00 03 00 00 00 00 00 00 00 00
```

De la même façon que l'unicité de l'adresse "lien-local" a été vérifiée, la station utilise le mécanisme DAD pour vérifier l'unicité de l'adresse "unicast globale" construite à partir du préfixe communiqué. Cette procédure est schématisée par la figure 9.

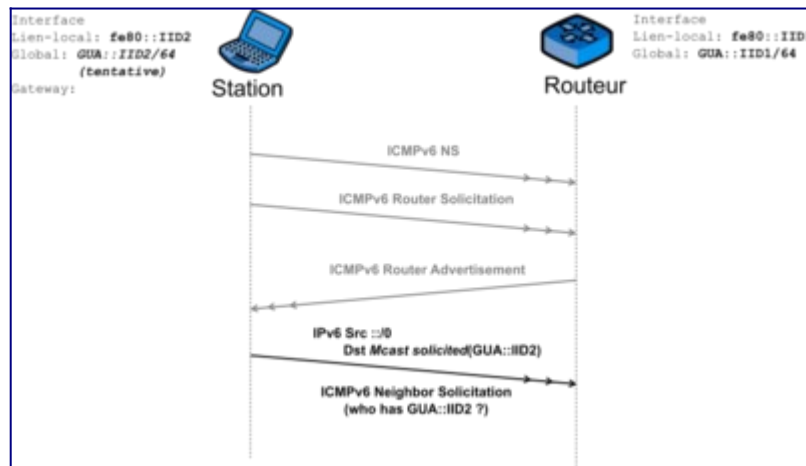


Figure 9 : Détection d'adresse dupliquée.

Une fois l'unicité de cette adresse vérifiée, la station configure dans sa table de routage l'adresse "lien-local" du routeur comme routeur par défaut. La configuration de l'interface réseau de la station et les messages échangés à l'issue de la phase d'auto-configuration sont indiqués par la figure 10. La station est désormais capable de communiquer avec des nœuds situés au-delà de son lien routeur. D'autres informations, comme notamment le DNS à utiliser, peuvent être communiquées dans le message d'annonce de routeur.

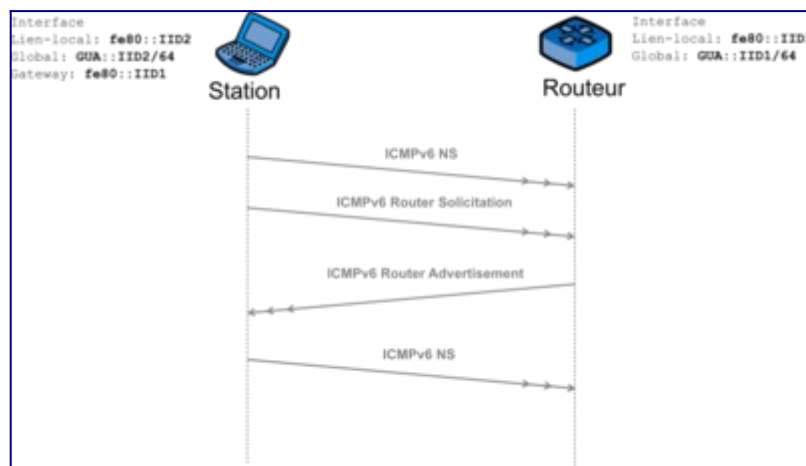


Figure 10 : Les adresses allouées.

Conclusion

L'auto-configuration "sans état" des paramètres réseau IPv6 permet une connectivité

fonctionnelle de l'interface réseau d'un hôte dès son branchement. Ce mécanisme ne nécessite aucune intervention humaine et l'automatisation évite certaines erreurs humaines dans la configuration. Les paramètres de configuration sont centralisés sur le routeur du réseau qui devient l'équipement indispensable à la configuration d'un réseau IPv6. Les stations sont ensuite autonomes pour récupérer ces paramètres et décider de leur adresse IPv6 afin de se configurer.

L'auto-configuration "sans état" a d'autres avantages par rapport à des méthodes manuelles ou reposant sur un serveur, en particulier dans le cas des équipements mobiles qui se déplacent de réseau en réseau. Ceux-ci peuvent récupérer une adresse valide sans avoir à connaître préalablement les informations du réseau visité. Le routeur sur le réseau visité va indiquer les instructions pour l'auto-configuration. La renumérotation d'un réseau et de ses nœuds peut être facilitée par la configuration automatique.

Cependant, la configuration automatique n'est pas adaptée à tous les cas. En effet, pour certaines stations, l'administrateur voudra plus finement maîtriser leurs adresses, comme par exemple pour les serveurs. Le mécanisme DHCPv6, décrit dans l'activité suivante, peut être utilisé à cette fin.

Références bibliographiques

1. ↑ Bortzmeyer, S. (2013). Article. [Rogue IPv6 Router Advertisement Problem Statement](#)
2. ↑ APNIC. [Comparison of IPv6 support in operating systems](#)

Pour aller plus loin

RFC et leur analyse par S. Bortzmeyer :

- [RFC 3315](#) Dynamic Host Configuration Protocol for IPv6 (DHCPv6)
- [RFC 3646](#) DNS Configuration options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6) [Analyse](#)
- [RFC 3736](#) Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6
- [RFC 4191](#) Default Router Preferences and More-Specific Routes
- [RFC 4291](#) IP Version 6 Addressing Architecture [Analyse](#)
- [RFC 4339](#) IPv6 Host Configuration of DNS Server Information Approaches
- [RFC 4861](#) Neighbor Discovery for IP version 6 (IPv6) [Analyse](#)
- [RFC 4862](#) IPv6 Stateless Address Autoconfiguration [Analyse](#)
- [RFC 4941](#) Privacy Extensions for Stateless Address Autoconfiguration in IPv6 [Analyse](#)
- [RFC 4943](#) IPv6 Neighbor Discovery On-Link Assumption Considered Harmful
- [RFC 5942](#) P6 Subnet Model : The Relationship between Links and Subnet Prefixes
- [RFC 6104](#) Rogue IPv6 Router Advertisement Problem Statement [Analyse](#)
- [RFC 6275](#) Mobility Support in IPv6
- [RFC 8106](#) IPv6 Router Advertisement Options for DNS Configuration [Analyse](#)

Activité 33 : Faire correspondre adresse et nom de domaine

Vous suivez une activité de découverte



Introduction

Cette activité introduit le système de nommage communément appelé DNS (*Domain Name System*). Nous présenterons les spécifications pour IPv6, les principes de sa mise en œuvre et les recommandations opérationnelles pour l'intégration d'IPv6. Cette activité commence par poser la problématique à résoudre et les principes généraux retenus pour la résolution de noms. Les spécifications du protocole s'attachent à traiter la résolution de noms et la résolution inverse ainsi que les ressources propres à IPv6. Les principes de mise en œuvre du service DNS expliquent la configuration d'un service DNS autonome en IPv6. Enfin, les recommandations opérationnelles pour l'intégration d'IPv6 décrivent les nouveaux problèmes induits par IPv6 et leurs réponses pour y faire face. Le lecteur pourra se reporter aux nombreux ouvrages traitant des principes et des éléments de configuration du DNS[1].

Concepts de base du DNS

Le DNS est un système de base de données hiérarchique et distribué. Il gère les correspondances directes entre les noms de machines (FQDN : *Fully Qualified Domain Name*) et les adresses IP (IPv4 et/ou IPv6), et les correspondances inverses entre les adresses IP (IPv4 et/ou IPv6) et les noms de machines. Le DNS gère également d'autres informations : par exemple, les informations relatives aux agents de transfert de courrier (*Mail eXchanger*, MX) ou encore celles relatives aux serveurs de noms (*Name Servers*, NS) et, plus généralement, d'autres informations utiles pour les applications TCP/IP.

Aujourd'hui, les utilisateurs font principalement référence aux noms de machines. Ces noms logiques sont plus faciles à mémoriser que les adresses, et souvent, reflètent la fonction de la machine. Ainsi, *www.tpt.example.com* ou *ftp.tpt.example.com* représentent respectivement les noms des serveurs Web et FTP de la société *tpt.example.com*.

Une application qui s'exécute sur un équipement A, et qui souhaite communiquer avec une autre application s'exécutant sur un équipement distant B dont elle ne connaît que le nom, a besoin d'en obtenir l'adresse IP. Sans cette adresse, la communication ne peut en général pas avoir lieu : les machines utilisent le protocole IP pour communiquer et ce protocole n'utilise que les adresses IP. A l'instar d'un répertoire téléphonique, le DNS est un annuaire global assurant la correspondance entre les noms logiques de machines et leurs références IP, essentiellement leurs adresses, mais d'autres informations techniques peuvent également être référencées.

Nommage « à plat »

Aux débuts de l'Internet, les adresses IPv4 en usage sont peu nombreuses. Il est donc relativement facile de les stocker dans un fichier centralisé : le fichier *hosts.txt* ([RFC 608](#)). Les

noms doivent aussi être uniques. Un nom utilisé dans une organisation ne peut alors pas l'être dans une autre organisation. Chaque responsable de site transmet ses modifications, ajouts et suppressions à un centre de gestion chargé de mettre à jour le fichier central. Chacun de ces responsables peut alors télécharger ce fichier, via FTP par exemple, pour mettre à jour les informations de nommage stockées localement (par exemple, le fichier `/etc/hosts` pour les systèmes Unix). Un équipement disposant localement d'une version à jour du fichier de nommage peut ainsi communiquer avec toutes les machines connues dans ce fichier. Dès le début des années 80, la croissance exponentielle du nombre de noms et d'adresses IP utilisées et le besoin de plus en plus fréquent de renuméroter les équipements ont rendu le choix des noms, leur mise à jour, et la mémorisation des adresses dans ce fichier central de plus en plus difficile, voire impossible dans des délais raisonnables. Ce système a donc été abandonné au profit du système de nommage.

Caractéristiques du système de noms de domaine

Paul Mockapetris, de l'Université de Californie, conçoit le système de nommage DNS en 1983. Il en écrit la première mise en œuvre à la demande de Jon Postel. Jon Postel est un informaticien américain, un des principaux contributeurs à la création de l'Internet. Il a été l'éditeur des RFC (*Request For Comments*). Il est notamment célèbre pour être l'auteur de cette phrase : «*Be liberal in what you accept, and conservative in what you send*».

Le DNS est initialement un service de résolution, de mise à jour et d'enregistrement des correspondances directes "nom-adresse" et des correspondances inverses "adresse-nom". Il fournit aux utilisateurs, quelle que soit leur localisation, l'adresse IP associée à un nom de domaine. Il distribue, de plus, la responsabilité de la mise à jour des informations de nommage sur chaque site et met en place un système coopératif d'accès aux informations de nommage. Petit à petit, le DNS s'est imposé comme infrastructure critique pour l'ensemble des applications TCP/IP classiques comme le mail, le web, le transfert de fichier et la connexion à distance. Ce système est donc hiérarchique, réparti, robuste et extensible.

- **Hiérarchique.** Le système de nommage est hiérarchique, pour garantir l'unicité des noms. Le système de nommage hiérarchique utilise une structure d'arbre (cf. figure 1). Un arbre est un graphe sans cycle, c'est-à-dire un ensemble de nœuds reliés par des arcs tel qu'il n'existe qu'un seul chemin reliant la racine de l'arbre à chacune de ses feuilles. Un arbre, à son plus haut niveau, se compose d'une racine et d'un ensemble de nœud « fils ». Chaque fils, dans l'arbre, est relié à son père par un arc. Chaque fils, au second niveau, possède à son tour ses propres fils. Et ainsi de suite jusqu'aux feuilles de l'arbre. Une feuille de l'arbre est un nœud qui n'a pas de fils. Le nommage hiérarchique associe un nom à chaque nœud d'un arbre : l'arbre de nommage. Un domaine correspond à un nœud dans l'arbre de nommage. Chaque nœud, sauf la racine, a un nom. Le nom d'un domaine est alors défini comme la succession des noms des nœuds qui, dans l'arbre de nommage, conduisent de ce nœud à la racine de l'arbre de nommage. Comme un arbre ne contient pas de cycle, chaque nœud n'est accessible que par un seul chemin. Par conséquent, dans un arbre de nommage, les noms de domaines

sont uniques.

Arbres informatiques

Les arbres informatiques sont couramment représentés avec la racine positionnée en haut et les feuilles (nœuds sans fils) en bas. Différentes méthodes algorithmiques permettent un parcours efficace de ces structures de données.

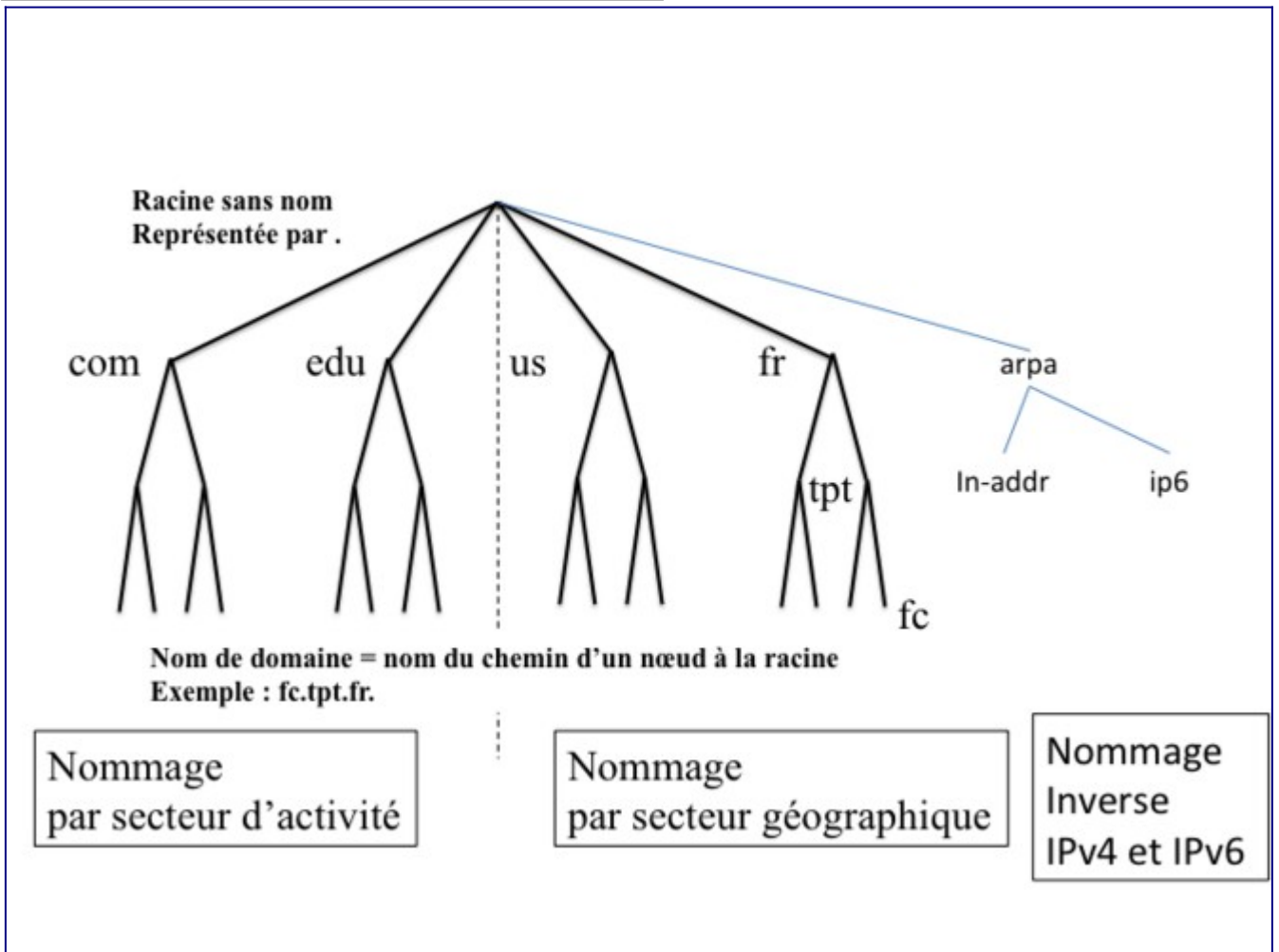


Figure 1 : Arbre de nommage.

Les nœuds du premier niveau (*les fils de la racine*) sont couramment dénommés *Top Level Domain* (TLD). Le nommage se fait, soit en fonction du secteur d'activité, soit en fonction du code pays (ISO). Deux sous-arbres, sous le TLD réservé "arpa" sont dédiés à la résolution inverse : *in-addr* pour IPv4 et *ip6* pour IPv6.

- **Réparti.** Nul n'est mieux placé que le responsable du nommage dans un domaine (de responsabilité administrative), par exemple celui d'une société, pour gérer les ajouts, modifications, suppressions dans le sous-arbre de nommage de cette société. Chaque responsable du nommage gère le nommage dans sa société. Il produit donc une base locale de nommage. Reste ensuite à partager ces informations pour les mettre à disposition des utilisateurs du réseau.
- **Robuste.** Aujourd'hui, tout le fonctionnement de l'internet dépend du bon fonctionnement du système de nommage. D'un point de vue pratique, s'il n'existe qu'un seul serveur

officiel pour un domaine, le service de nommage devient indisponible si ce serveur tombe en panne ou est arrêté. C'est pourquoi, au moins deux serveurs, situés sur des sites géographiquement distincts et indépendants, sont nécessaires pour chaque zone de nommage (zone DNS). Ceci assure à la fois une meilleure disponibilité et un meilleur équilibrage de charge.

- **Disponibilité.** La probabilité d'occurrence simultanée d'une panne catastrophique (avec perte des données) sur les deux sites est faible, plus faible en tout cas que s'il n'y a qu'un seul serveur. Si un des deux serveurs tombe en panne, l'autre continue de fournir le service. Cette probabilité de panne est encore réduite s'il existe plus de deux sites hébergeant des serveurs de noms secondaires.
- **Équilibrage de charge.** Lorsque ces deux serveurs sont opérationnels, un client peut, par exemple, interroger simultanément les deux serveurs pour déterminer celui des deux qui est le moins sollicité, et utiliser préférentiellement ses services. En cas de non réponse du serveur choisi, le client peut interroger l'autre serveur pour obtenir les réponses à ses questions. En pratique, les demandes des différents clients se répartissent sur les différents serveurs de noms. Et si deux serveurs ne peuvent supporter la charge, il suffit d'en ajouter d'autres.
- **Extensible.** La structure d'arbre est extensible (*scalable*) (cf. figure 2). Pour ajouter un nom, il suffit, dans l'arbre de nommage, entre la racine et les feuilles, d'ajouter un nœud et toute sa descendance, et de relier ce nœud à un père en vérifiant que ce père n'a pas deux fils de même nom.

Ainsi, si l'on considère une nouvelle société dont le nom de domaine est *société1.com*, déclarer cette société dans le système de nommage revient à ajouter un fils : *société1* sous le nœud père *com*, lequel est lui-même fils de «.» (point), la racine (sans nom) de l'arbre de nommage.

Extensibilité des arbres

Les structures de données arborescentes ont cette capacité de pouvoir être étendues sans limite théorique et sans modification de leur structure. Les espaces de nommage de taille quelconque (potentiellement arbitrairement grands) sont généralement construits sous forme arborescente. Le DNS en est une illustration concrète, la structure et le protocole n'ont pas été modifiés lors de l'explosion des noms de domaine consécutive à la banalisation de l'Internet depuis les années 1990. D'autres espaces de nommage sont bâtis sur le même principe : arborescence d'annuaires LDAP, référencement d'objets de l'IETF sous forme d'Object Identifier (OID) pour les protocoles SNMP ou LDAP, pour ne citer que des exemples informatiques et réseaux

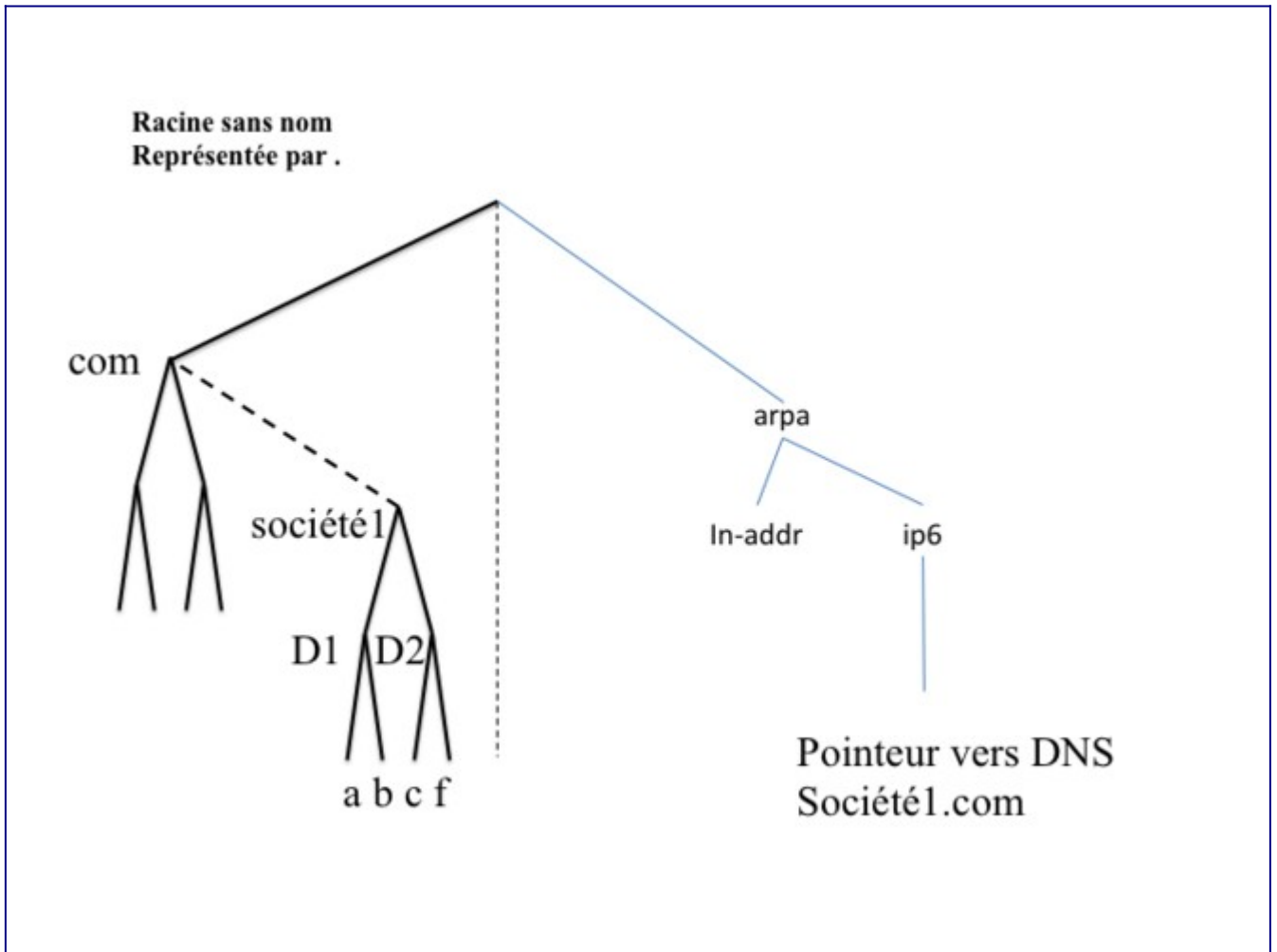


Figure 2 : Extension de l'arbre de nommage.

L'idée, simple mais géniale, a été de concevoir un système client-serveur pour cela, concrètement basée sur une arborescence de serveurs. Un serveur DNS est associé à chaque nœud de l'arbre de nommage. En fait, pour des raisons administratives, l'espace de nommage est partitionné en zones, correspondant à des "sous-arbres". Selon le principe de délégation de responsabilité administrative, chaque zone est autonome et responsable de son étendue de nommage.

Chaque zone commence au niveau d'un nœud (un domaine) et s'arrête aux nœuds de l'arbre de nommage qui correspondent à d'autres zones. Une zone correspond donc à l'ensemble des domaines (nœuds de l'arbre de nommage) relevant d'une même responsabilité administrative. Un serveur de nommage officiel gère les données d'une zone. Si, comme c'est possible dans certains cas, l'arbre de nommage est très profond, nous verrons que plusieurs serveurs DNS distincts peuvent être regroupés sur une même machine physique. Un serveur DNS peut gérer officiellement plusieurs zones en étant primaire pour une zone et secondaire pour différentes autres zones par exemple. Ces regroupements réduisent la profondeur de la hiérarchie de serveurs DNS, ce qui permet d'en accélérer le balayage (cf. figure 3). Les serveurs DNS sont reliés les uns aux autres par un chaînage double : chaque père connaît chacun de ses fils, et chaque fils connaît son père.

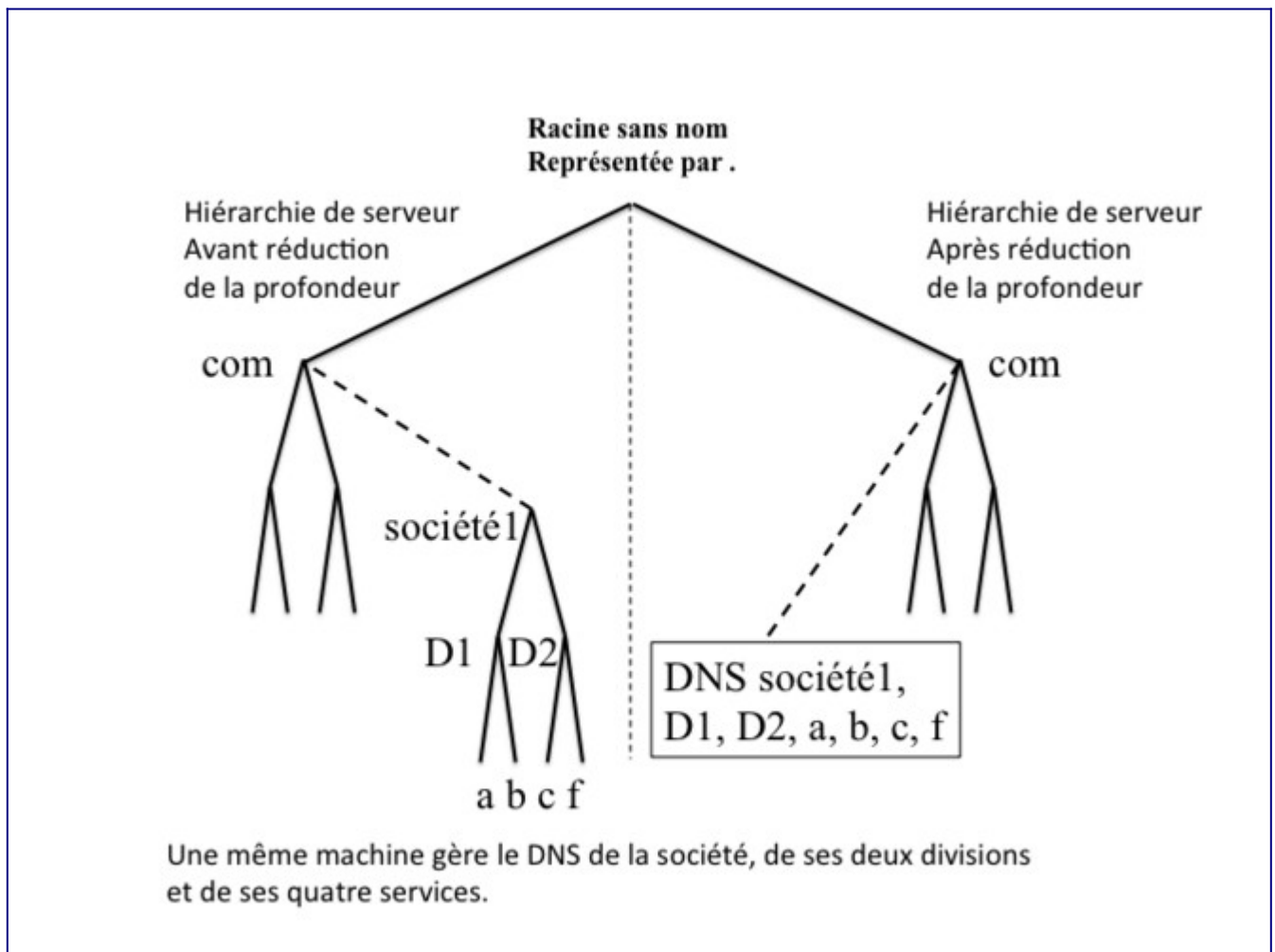


Figure 3 : Réduction de la profondeur de la hiérarchie de serveurs : avant après.

Les clients du service de nommage ne se trouvent qu'au niveau des feuilles de l'arbre de nommage. Plus précisément, il n'y a qu'un client du service de nommage par machine, le résolveur. Cela signifie que toutes les applications qui s'exécutent sur une machine et qui doivent résoudre un nom sollicitent le seul et unique client DNS de cette machine, le résolveur.

Principe de fonctionnement du service DNS

Chacune des applications d'une machine s'adresse au résolveur unique de cette machine (*stub resolver*) et lui demande des informations associées à des noms de domaines, comme des adresses IP, des relais de messagerie (enregistrement de type MX) ou des serveurs de noms (enregistrement de type NS). Le résolveur est une application commune à toutes les applications d'une machine. Il est souvent implémenté sous la forme d'une bibliothèque de procédures (*Au niveau des systèmes d'exploitation des machines, le résolveur DNS est généralement nativement implanté dans le code de mise en œuvre de la pile IP*). Pour l'utiliser, les programmes d'application invoquent les procédures de la bibliothèque (*selon le mécanisme des appels système*).

Initialement, le résolveur de la machine locale interrogeait successivement chacun des serveurs (résolution itérative) jusqu'à ce qu'il s'adresse au serveur officiel du domaine concerné. Afin

d'accélérer la réponse aux requêtes suivantes, le résolveur conservait dans un cache les informations de nommage. Aujourd'hui, pour optimiser davantage le fonctionnement du système de nommage, les résolveurs fonctionnent en mode récursif. Ils s'adressent à un serveur DNS local et lui demandent de leur fournir les informations de nommage demandées. Ils ne gèrent alors plus de cache local. Ce dernier est mutualisé au niveau du serveur DNS local. Les informations mises en cache bénéficieront à l'ensemble des utilisateurs.

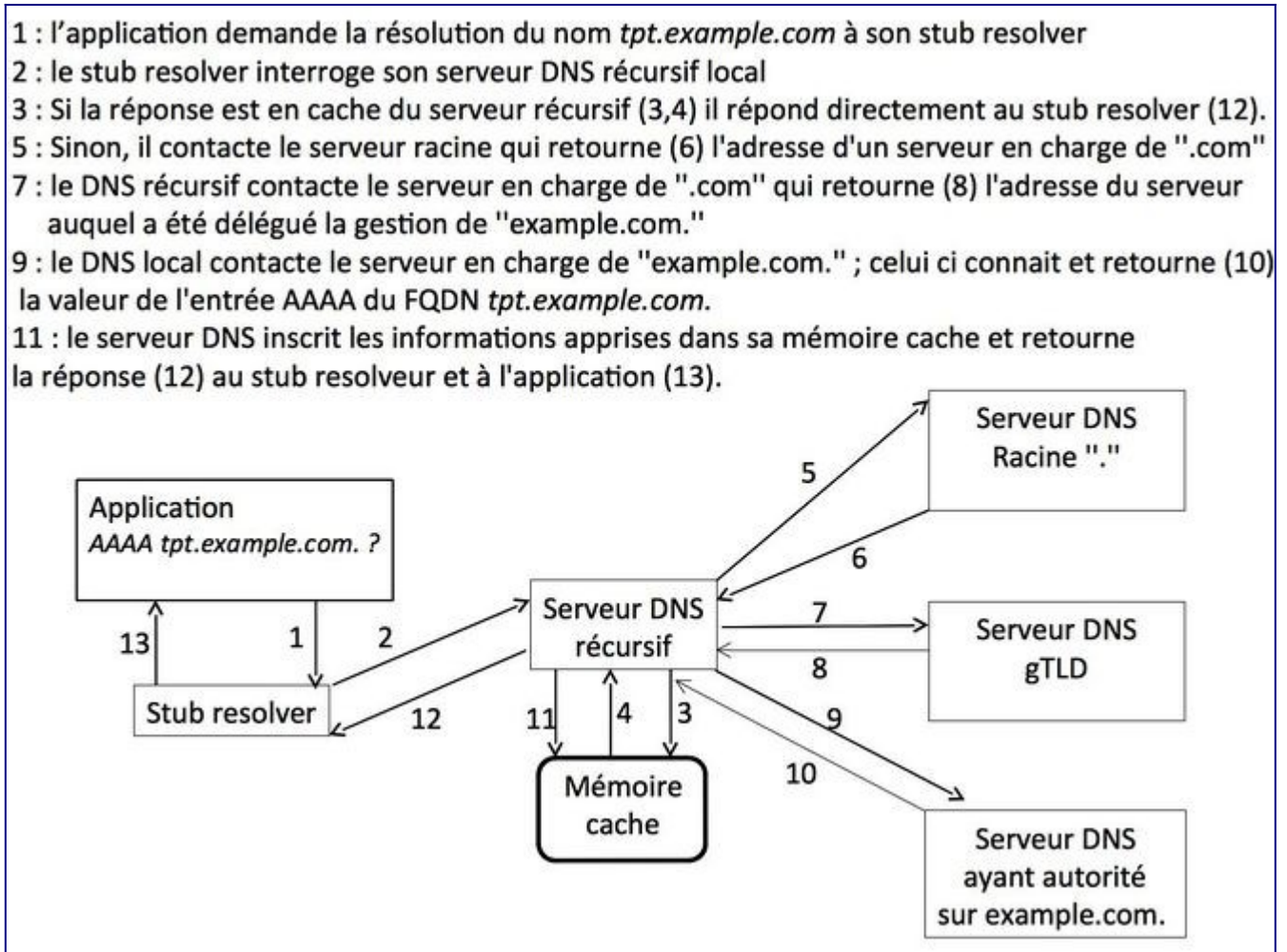


Figure 4 : Relations entre les applications d'une machine : le résolveur et le serveur DNS local.

Le serveur DNS local supporte la récursivité, c'est-à-dire qu'il accepte des demandes de résolution récursives de la part de ses clients. Le serveur DNS récursif local résout ensuite la requête de manière itérative.

On notera que toutes les résolutions itératives démarrent par la racine et que cette dernière pointe vers les serveurs des TLD. Pour des raisons évidentes de répartition de charge, les serveurs racines sont répliqués. Leurs noms et adresses sont enregistrés dans le fichier *db.root*. Le serveur DNS local enregistre le contenu de ce fichier dans une partie réservée de la mémoire cache lorsqu'il démarre. Il dispose ainsi des noms et adresses de chacun des serveurs DNS racine. Un serveur racine connaît chacun de ses fils dans l'arbre de nommage du

DNS, c'est-à-dire les serveurs en charge des TLD. Il ne dispose localement d'aucune information de nommage. Il n'enregistre pas non plus d'information de nommage dans une mémoire cache. En revanche, en fonction du nom de domaine à résoudre, il sait lequel de ses fils, soit gère la correspondance, soit sait qui la gère. Il fournit donc cette information au serveur DNS local. Notre serveur DNS local s'adresse donc successivement au serveur DNS fils (le serveur administrativement responsable du TLD), puis au serveur DNS petit-fils du serveur DNS racine. Il finit par adresser sa demande au serveur DNS ayant autorité sur les informations de nommage recherchées. Le serveur DNS ayant autorité fournit donc ces informations de nommage au serveur DNS local. Celui-ci les enregistre dans sa mémoire cache et les transmet au résolveur à l'origine de la demande. Le résolveur fournit les informations de nommage à l'application à l'origine de la demande. Un exemple de résolution d'adresse est présenté par la figure 4. L'application demande la résolution du FQDN *tpt.example.com.* à son résolveur, lequel contacte le serveur récursif local. Celui-ci contacte le serveur racine puis un serveur en charge du TLD *.com.* et enfin le serveur en charge du domaine *example.com.*. La réponse est alors mise en cache de manière à accélérer la résolution des requêtes ultérieures.

Notez que le serveur DNS local, à chaque étape de la résolution itérative, enregistre dans sa mémoire cache les nom et adresse de chaque serveur DNS interrogé ainsi que les réponses des différents serveurs DNS officiels. Il mutualise donc les informations de nommage pour toutes les machines qui utilisent ses services. Le serveur DNS local, si un résolveur lui pose une question déjà posée par un autre résolveur, fournit immédiatement la réponse à partir de sa mémoire cache lorsque cette information est valide et s'y trouve. Si la question concerne un serveur DNS ayant autorité sur un domaine déjà connu, le serveur DNS local contacte directement le serveur DNS concerné. Notez cependant que les informations enregistrées dans la mémoire cache du serveur DNS local ont une durée de vie limitée. Lorsque les informations de nommage présentes dans la mémoire cache ne sont plus valides, le serveur DNS local ne peut les utiliser pour fournir des réponses aux applications. Il redemande alors directement cette information au serveur DNS officiel du domaine concerné.

Les serveurs de noms

L'arborescence des serveurs de noms est composée de plusieurs types de serveurs fonctionnels répartis sur le réseau internet.

Serveurs de noms primaires et secondaires

Gestion des données de zone

À l'origine, les données administratives d'une zone étaient gérées par l'administrateur dans de simples fichiers texte. Aujourd'hui, les fournisseurs d'accès à Internet ainsi que les prestataires du service DNS, administrant des zones dont le contenu est volumineux, ont délaissés les fichiers à plat au profit de systèmes de bases de données ou d'annuaires LDAP.

Le système DNS distingue, pour une zone donnée, deux types de serveurs de noms : primaire et secondaire. Notez tout d'abord que les serveurs de noms primaire et secondaire pour une zone donnée sont tous des serveurs officiels pour cette zone. Le serveur DNS primaire est le

serveur sur lequel l'administrateur du réseau effectue les mises à jour des informations de nommage. Il dispose de fichiers de nommage (les données de zone) enregistrés dans une mémoire locale non volatile. Un serveur DNS primaire peut, par défaut, synchroniser au plus 10 serveurs DNS secondaires. Le numéro de version de chacun des fichiers de zone du serveur DNS primaire change, soit à chaque modification faite par l'administrateur du réseau, soit à l'expiration d'un certain délai en cas de mise à jour dynamique lorsque les mises à jour sont nombreuses.

Les serveurs DNS secondaires sont des serveurs de nommage qui acquièrent leurs informations de nommage, soit depuis le serveur DNS primaire, soit depuis un autre serveur DNS secondaire déjà synchronisé, à l'aide d'un protocole de transfert de fichier, par exemple. Notez qu'un serveur DNS secondaire est synchronisé si le numéro de version de chacun de ses fichiers de zone est identique à ceux de chacun des fichiers de zone du serveur DNS primaire. L'administrateur du réseau ne gère les mises à jour du système de nommage qu'au niveau des fichiers de zone du serveur DNS primaire. Il incrémente le numéro de version d'un fichier de zone à chaque modification (c.f. figure 5). Il déclenche la prise en compte des modifications en redémarrant le serveur DNS primaire ou en le réinitialisant (cf. figure 6).

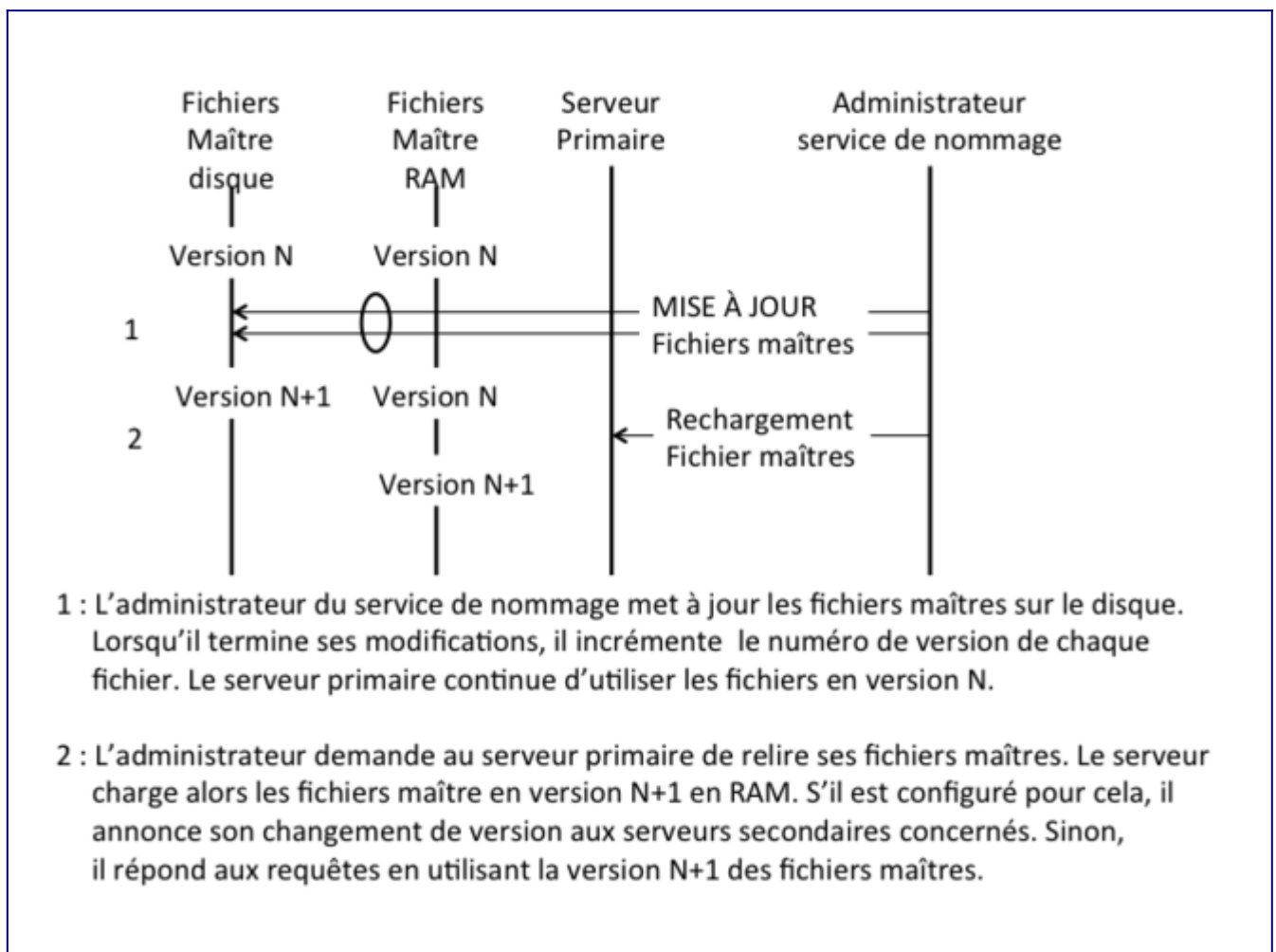


Figure 5 : Mise à jour d'un fichier de zone du serveur DNS primaire par l'administrateur du réseau.

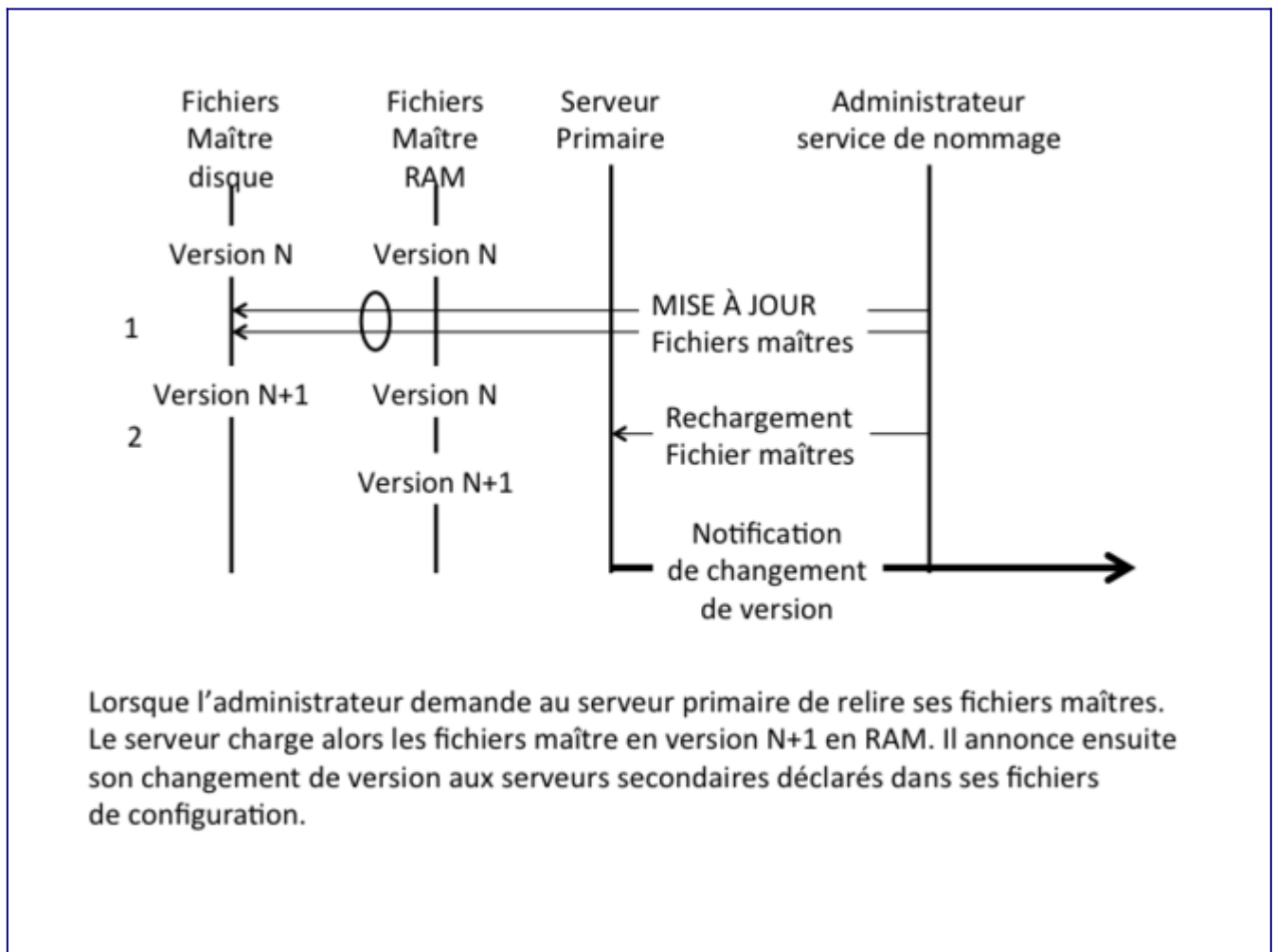


Figure 6 : Mise à jour d'un fichier de zone et réinitialisation du serveur DNS primaire par l'administrateur du réseau.

Redémarrage et réinitialisation d'un serveur DNS

Lorsque l'administrateur redémarre le serveur DNS primaire, celui-ci relit son fichier de configuration et ses fichiers de zone et les charge en mémoire RAM (*Random Access Memory*). Il n'utilise ensuite que les informations disponibles en RAM. Lorsque l'administrateur réinitialise le serveur DNS, celui-ci ne relit que ses fichiers de zone et les charge en mémoire RAM. Il n'utilise ensuite que les informations disponibles en RAM.

Il configure le mode de déclenchement de la synchronisation des serveurs DNS secondaires, soit à l'initiative du serveur DNS primaire (notification), soit à l'initiative des serveurs DNS secondaires (interrogation).

Synchronisation par notification : lorsque la synchronisation se fait à l'initiative du serveur DNS primaire, ce dernier envoie le nouveau numéro de version de ses fichiers de zone à tous les serveurs DNS secondaires. Tous les serveurs DNS secondaires tentent alors de se synchroniser. La synchronisation peut s'effectuer à partir du seul serveur DNS primaire ou également s'effectuer à partir de serveurs DNS secondaires déjà synchronisés.

Synchronisation par interrogation : lorsque la synchronisation se fait à l'initiative des serveurs DNS secondaires, chaque serveur DNS secondaire vérifie périodiquement le numéro

de version de la base de nommage du serveur DNS primaire. Si ce numéro de version de la base de nommage du serveur DNS primaire n'a pas changé, le serveur DNS attend le temps fixé par la période de scrutation avant de revérifier le numéro de version de la base de nommage du serveur DNS primaire. Si le numéro de version de la base de nommage du serveur DNS primaire est plus élevé que le sien, le serveur DNS secondaire tente de démarrer une synchronisation de sa base de nommage. Si sa tentative échoue, il attend pendant un certain temps, à l'expiration duquel il tente à nouveau de se synchroniser.

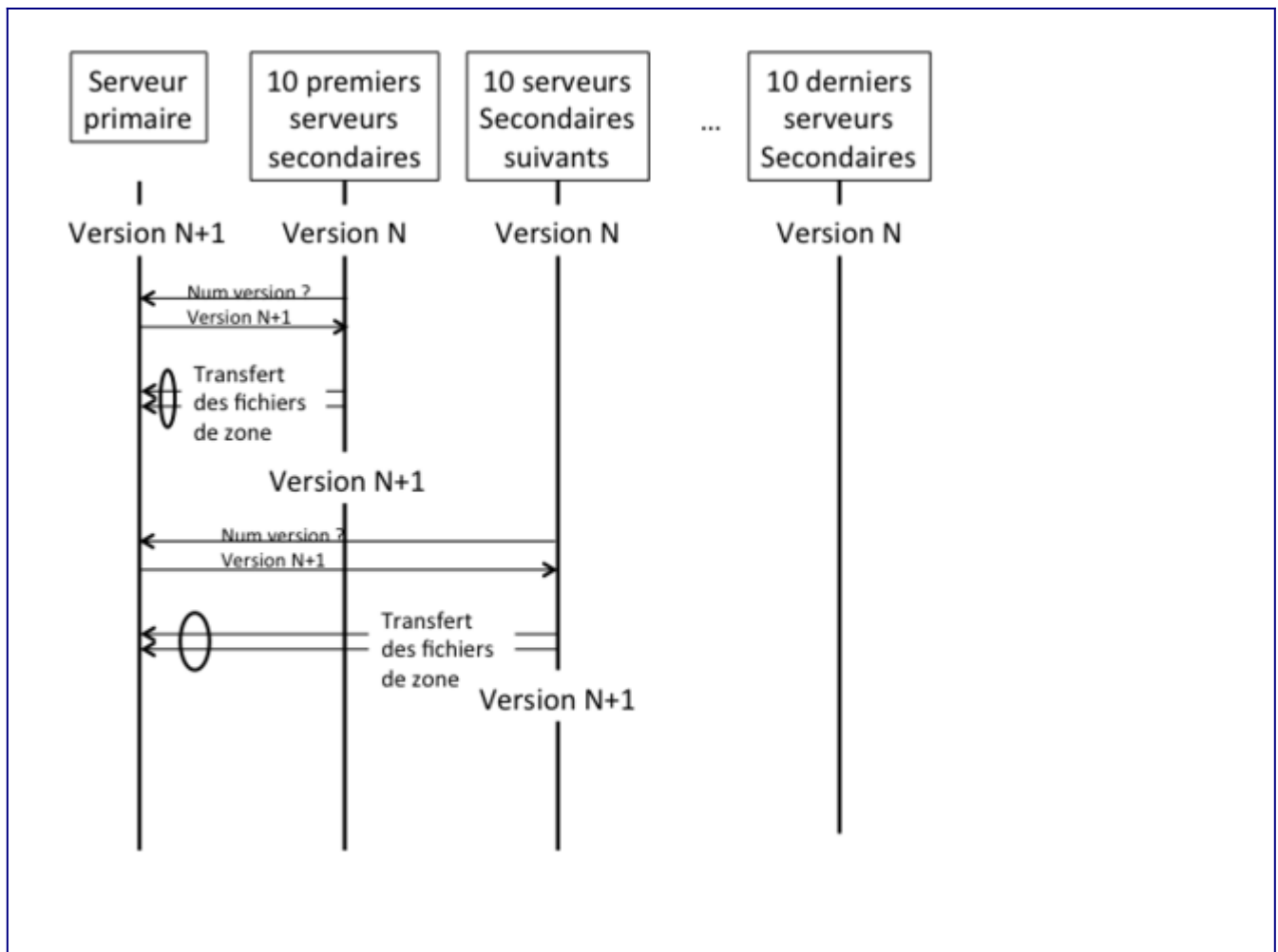


Figure 7 : Transfert des fichiers de zones mises à jour sur le serveur primaire vers les serveurs secondaires.

Ainsi, les serveurs qui le peuvent (10 maximum) se synchronisent immédiatement. Les autres attendent pendant une durée au minimum égale au temps de synchronisation de la première vague puis, tentent à nouveau de se synchroniser (cf. figure 7). Notez, qu'ici encore, l'administrateur du réseau peut optimiser le délai de synchronisation en configurant de façon appropriée les serveurs DNS secondaires pour qu'ils se synchronisent à partir du serveur DNS primaire et des serveurs DNS secondaires déjà synchronisés. Il suffit pour cela de définir les

serveurs DNS secondaires qui se synchronisent immédiatement, ceux qui se synchronisent dans un deuxième, un troisième, et éventuellement dans un quatrième temps (cf. figure 8).

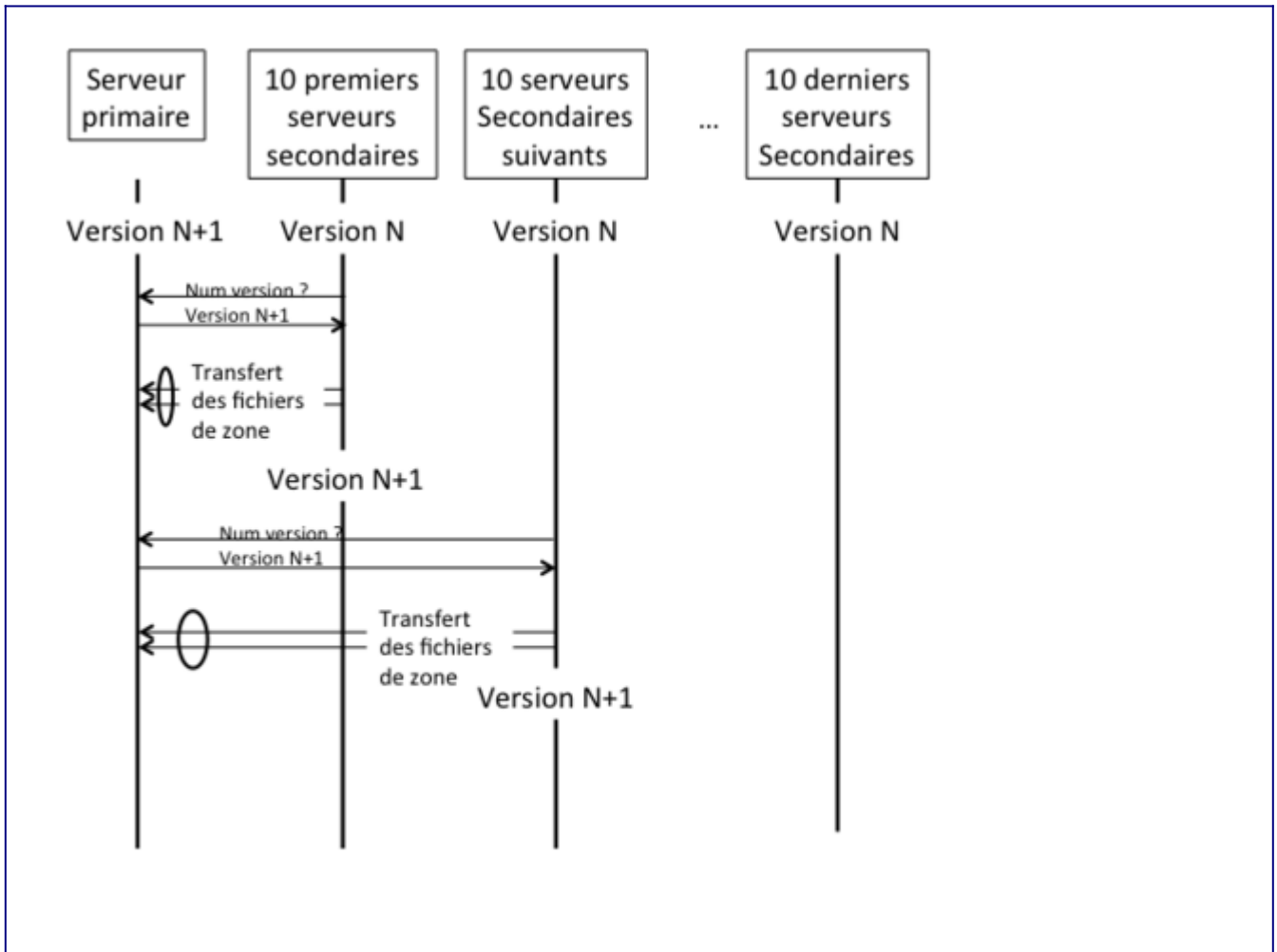


Figure 8 : Optimisation du transfert des fichiers de zones via l'utilisation de serveurs secondaires déjà synchronisés.

Notez qu'un serveur DNS secondaire peut, selon son mode de configuration, stocker localement et sur une mémoire non volatile, une copie des fichiers de nommage. S'il enregistre localement, et dans une mémoire non volatile, une copie de ses fichiers de zone, il peut d'une part, démarrer de façon autonome en cas de panne, catastrophique ou non, du serveur DNS primaire, et d'autre part, très facilement être transformé, si nécessaire, en serveur DNS primaire. Cette bonne pratique est recommandée par l'IETF car elle contribue à la réplication des fichiers de zone.

Serveur DNS récursif (caching name server)

Les résolveurs sont en général incapables d'effectuer la totalité du processus de résolution d'adresse (cf. schéma de principe de la résolution de la figure 4). Ils sont incapables d'interroger directement les serveurs DNS officiels. Ils s'appuient sur un serveur DNS local pour effectuer la résolution. De tels serveurs sont appelés serveurs DNS récursifs ou serveurs DNS "cache". Ces deux termes sont synonymes. Un serveur DNS récursif, pour améliorer les performances,

enregistre les résultats obtenus dans sa mémoire cache. Une durée de vie associée à chaque enregistrement de ressource contrôle la durée de validité d'une information de nommage dans la mémoire cache.

Relais DNS (*forwarder*)

Un relais DNS peut ne pas effectuer l'intégralité de la recherche lui-même. Il achemine tout ou partie des demandes d'information de nommage reçues et qu'il ne sait pas satisfaire, à partir des données de sa mémoire cache, vers un autre serveur DNS récursif. Ce serveur est dit "relais DNS" (*forwarder*). Il peut y avoir un ou plusieurs relais DNS. Chacun est interrogé à tour de rôle jusqu'à épuisement des serveurs de la liste ou obtention de la réponse.

Les relais DNS servent, par exemple, lorsque vous ne souhaitez pas que tous les serveurs DNS d'un site interagissent directement avec les serveurs de l'Internet. Ainsi, un exemple typique implique plusieurs serveurs DNS internes et un pare-feu d'accès à Internet. Les serveurs de nommage incapables d'acheminer leurs messages à travers le pare-feu les adressent aux serveurs DNS capables de le faire (serveurs DNS généralement en zone dite *démilitarisée* (*DMZ*) de "l'autre côté" du pare-feu et autorisés à accéder à l'Internet public). Et ces serveurs DNS interrogent alors les serveurs DNS de l'Internet pour le compte des serveurs DNS internes. Les serveurs de relais sont également utiles dans le cas où le délai entre le réseau local et l'Internet est significatif. Dans ce cas, on déploiera un serveur relais dans le réseau local, lequel contactera un serveur récursif déployé ailleurs sur l'Internet.

Serveurs DNS à rôles multiples

Un serveur DNS BIND (*BIND Berkeley Internet Name Domain server est l'implémentation logicielle de référence d'un serveur DNS*) peut simultanément se comporter comme un serveur ayant autorité en qualité de serveur primaire pour certaines zones, secondaire pour d'autres, et se comporter comme serveur DNS récursif pour un certain nombre de clients.

Les fonctions des serveurs DNS officiels et récursifs sont habituellement activées sur des machines distinctes. Un serveur DNS ne fournissant qu'un service DNS officiel fonctionnera avec la récursivité désactivée, ce qui est à la fois plus fiable et plus sûr. Un serveur DNS non officiel et qui ne fournit que des services de nommage récursifs à des clients locaux n'a pas besoin d'être accessible depuis l'Internet. Il peut donc fonctionner derrière un pare-feu. Un serveur DNS peut cependant être configuré comme serveur officiel pour tous les utilisateurs et n'accepter les requêtes récursives que pour les utilisateurs du réseau interne.

Spécifications du service de nommage

Spécifications du résolveur

Rappelons que pour les applications communicantes (une session web par exemple), une phase pendant laquelle le client DNS local, appelé *stub resolver*, interroge son serveur DNS récursif (ou cache), précède l'établissement effectif de la communication. Le service DNS fonctionne au niveau de la couche application de la pile TCP/IP. Il s'applique de manière

analogue aux réseaux IPv6 et aux réseaux IPv4.

Singularité du service DNS

Notez que le DNS est le seul service de l'internet pour lequel le client doit absolument être configuré avec l'adresse IP d'au moins un serveur DNS. C'est généralement l'adresse d'un serveur DNS local. Les adresses IPv6 étant quatre fois plus grande que celle de IPv4, il est d'autant moins probable que les utilisateurs puissent les retenir, ce qui rend le DNS d'autant plus indispensable.

Pour les machines Unix, par exemple, le fichier de configuration du client DNS, */etc/resolv.conf*, fournit l'adresse IP d'un ou plusieurs serveurs de noms. Le résolveur, lorsqu'il démarre, lit ce fichier de configuration. Il dispose donc de l'adresse d'un ou plusieurs serveurs DNS à interroger, ce qui lui permet d'initialiser sa recherche d'information de nommage pour le compte des applications locales. Dans la pratique, ce fichier est renseigné soit manuellement par l'administrateur de la machine soit, plus généralement, automatiquement lors de la procédure d'auto-configuration avec ou sans état selon les principes détaillés ci-dessous dans le paragraphe intitulé "Découverte de la liste des serveurs DNS récursifs".

Spécifications des ressources IPv6

Les ressources logiques de la base de données répartie du DNS sont gérées sous forme d'enregistrements de ressource communément appelées *Ressource Record* ou RR. Différents types de RR ont été spécifiés tels que ceux déjà évoqués dans ce document : RR de type A pour une correspondance d'adresse IPv4, RR de type NS pour un serveur de domaine, ou encore RR de type MX (*Mail eXchanger*) pour une correspondance entre un domaine et son ou ses relais de messagerie.

Afin de supporter le nouveau schéma d'adressage d'IPv6, deux extensions DNS ont été définies ([RFC 3596](#)) : l'enregistrement de ressource de type AAAA, et un nouveau sous-domaine dédié à la résolution inverse (adresse-nom) en IPv6 : *ip6.arpa*.

- Le RR de type AAAA (*prononcé « quad A »*) enregistre les correspondances nom - adresse IPv6. Le code réservé de ce nouveau type d'enregistrement de ressources vaut 28.
- Le nouveau sous-domaine *ip6.arpa* est dédié à la résolution DNS inverse en IPv6 (correspondance "adresse IPv6" vers "nom"). La résolution DNS inverse utilise, pour IPv6, la notion de quartet (*nibble*). *Rappel : un quartet correspond à un chiffre hexadécimal. Comme nous l'avons vu en séquence 1, une adresse IPv6 est composée de 32 quartets.*

Nommage direct : enregistrement AAAA

Le nouveau type d'enregistrement AAAA, défini pour IPv6, établit la correspondance entre un nom de domaine et ses adresses IPv6. Une machine ayant plusieurs adresses IPv6 globales a, en principe, autant d'enregistrements AAAA publiés dans le DNS. (*De façon analogue, la correspondance entre un nom de domaine et ses adresses IPv4 est réalisée en associant au nom en question un ou plusieurs enregistrements DNS de type A. Chaque RR type A contient la*

valeur d'une adresse IPv4. Une machine a autant d'enregistrements de type A qu'elle a d'adresses IPv4 (machine multi-domiciliée ou routeur, par exemple).

Une requête DNS de type AAAA concernant un FQDN (*Fully Qualified Domain Name*) renvoie dans ce cas tous les enregistrements AAAA publiés dans le DNS et correspondant à ce FQDN. Notez que toutes les adresses n'ont cependant pas leur place dans le DNS. Ce sujet sera traité au chapitre **Publication des enregistrements AAAA dans le DNS**. Le format textuel d'un enregistrement AAAA tel qu'il apparaît dans le fichier de zone DNS est le suivant :

```
[ttl] IN AAAA
```

L'adresse est écrite suivant la représentation classique des adresses IPv6 ([RFC 4291](#)) (représentation hexadécimale pointée, l'usage de la notation canonique ([RFC 5952](#)) est probablement une bonne pratique mais n'est cependant pas obligatoire). Par exemple, l'adresse IPv6 de la machine *ns3.nic.fr* est publiée dans le fichier de zone *nic.fr* comme suit :

```
ns3.nic.fr. IN AAAA 2001:3006:3006:1::1:1
```

Notez que toutes les adresses IPv4 ou IPv6 correspondant à un nom de domaine donné (c'est le cas d'un réseau configuré en double pile de communication, *dual-stack*), doivent cohabiter dans le même fichier de zone renseignant le nom de domaine en question. Ainsi, les adresses de *ns3.nic.fr* sont publiées dans le fichier de zone *nic.fr* comme suit :

```
$ORIGIN nic.fr.
ns3 IN A 192.134.0.49
    IN AAAA 2001:db8:3006:1::1:1
```

Cependant, il faut rester vigilant avec une telle configuration puisque certains résolveurs recherchent prioritairement un enregistrement AAAA avant un enregistrement A, même si l'hôte exécutant le résolveur n'a qu'une connexion IPv6 limitée (une simple adresse locale au lien). Dans ce cas, cet hôte attend l'expiration du délai d'attente d'établissement de la session IPv6 avant de revenir à l'utilisation d'IPv4.

Nommage inverse : enregistrement PTR

Trouver le nom de domaine associé à une adresse est un problème quasi insoluble. Néanmoins, une astuce permet de résoudre élégamment ce problème. Il suffit de présenter les adresses comme des noms (succession des noms de domaines conduisant, dans l'arbre de nommage, d'une feuille à la racine de l'arborescence).

C'est-à-dire que, pour IPv4, il suffit d'écrire l'adresse IP en miroir : au lieu de commencer l'écriture d'une adresse par les octets de poids fort, on commence par celle des octets de poids faible. Ainsi l'adresse IPv4 192.168.1.150 pourrait être référencée sous le nom 150.1.16.193.in-addr.arpa dans le DNS inverse. Pour IPv6, on considère une adresse IPv6 comme une succession de chiffres hexadécimaux (32 quartets par adresse IPv6) séparés par des «.». Une adresse IPv6 est donc transformée en un nom de domaine publié dans le sous-arbre de nommage réservé à la résolution inverse pour IPv6 (ip6.arpa) de la manière suivante : les 32 demi-octets formant l'adresse IPv6 sont séparés par le caractère '.' et concaténés dans l'ordre

- la première concerne l'ajout d'options dans les annonces de routeur ;
- la seconde concerne l'ajout d'options spécifiques dans DHCPv6 ;
- la troisième concerne l'utilisation d'adresses anycast réservées, spécifiques des serveurs DNS récursifs.

Les co-auteurs de ces trois propositions ont rédigé conjointement un document synthétique ([RFC 4339](#)).

Ce document décrit le fonctionnement ainsi que les scénarios d'utilisation de chaque technique. Il donne également des recommandations pratiques quant à la solution ou à la combinaison de solutions à adopter en fonction de l'environnement technique dans lequel se trouvent les équipements à configurer.

Principe des trois propositions : RA, DHCPv6, anycast

1. **RA** : le mécanisme à base d'annonce de routeur (RA) est spécifié dans le [RFC 8106](#). Cette proposition étend l'autoconfiguration "sans état" ([RFC 4862](#)). Elle définit de nouvelles options. Ces options enrichissent les annonces de routeurs ([RFC 4861](#)) en y ajoutant, sous la forme d'options, les informations relatives au DNS. Cette extension est en cours de standardisation à ce jour.
2. **DHCPv6** : le mécanisme à base de DHCPv6 propose deux solutions légèrement différentes. Elles proposent toutes les deux d'utiliser la même option « DHCPv6 DNS Recursive Name Server » spécifiée dans le [RFC 3646](#). La première utilise un serveur DHCPv6 "à état" ([RFC 3315](#)). Celui-ci annonce l'adresse des serveurs de noms récursifs dans des options (ce serveur alloue dynamiquement les adresses IPv6 et les paramètres de configuration du réseau, en particulier les informations de configuration du service de nommage des clients). La seconde propose une utilisation dans le DHCPv6 "sans état" ou serveur DHCPv6-lite ([RFC 3736](#)). Celui-ci n'alloue pas d'adresses IPv6, mais informe simplement les clients des différents paramètres à utiliser (DNS récursif, serveur NTP, serveur d'impression...). Dans les deux cas, si un hôte est configuré à la fois avec DHCPv4 (pour IPv4) et avec DHCPv6 (pour IPv6), l'administrateur du réseau doit définir une politique d'arbitrage par un client lorsque les deux listes de serveurs DNS récursifs obtenues par IPv4 et IPv6 sont incohérentes.
3. **Anycast** : mécanisme à base d'adresses anycast réservées (*Well-known anycast addresses*). Ce mécanisme utilise des adresses IPv4 et IPv6 anycast qui seraient connues par tous les clients et préconfigurées automatiquement par le logiciel d'installation du système d'exploitation de l'équipement. Cette proposition semble avoir été abandonnée. Elle pose de réels problèmes de fonctionnement avec TCP et avec les applications qui gèrent des états au-dessus d'UDP.

Extension de l'autoconfiguration "sans état" pour le DNS

Le [RFC 4862](#) spécifie l'autoconfiguration IPv6 "sans état". Il ne prévoit pas de mécanisme de découverte automatique de la liste des serveurs DNS récursifs. Le [RFC 8106](#) définit deux options d'annonce de routeur : une option qui fournit une liste de serveurs DNS récursifs

(RDNSS : *Recursive DNS Server*) et une option pour définir la liste des noms de domaines recherchés (*DNSSEC DNS Search List*). Avec ces deux options, les machines IPv6 peuvent configurer complètement leur accès au service DNS pour utiliser les services de l'internet. Ces options fournissent les informations nécessaires pour configurer le fichier *resolv.conf*.

L'autoconfiguration, avec configuration complète du service DNS, sert dans les réseaux dépourvus de serveur DHCPv6 ou pour des machines IPv6 dépourvues de client DHCPv6. Elle fonctionne sur tout réseau supportant la découverte des voisins, (sous réserve que l'OS des machines supporte ces options spécifiques). Les configurations du réseau et du service DNS sont alors simultanées. L'administrateur du réseau configure manuellement les annonces des routeurs pour cette autoconfiguration.

La représentation détaillée des options ICMPv6 relatives à RDNSS et DNSSEC est reportée à l'annexe 1 de cette activité.

Extension de la configuration "à état", DHCPv6

Le [RFC 3315](#) spécifie le protocole d'autoconfiguration "à état", DHCPv6 : Dynamic Host Configuration Protocol version 6. Ce protocole fournit également les informations de configuration de l'accès au service DNS d'une machine IPv6. La représentation détaillée des options ICMPv6 relatives à RDNSS et DNSSEC est reportée à l'annexe 2 de cette activité.

Utilisation d'adresses anycast réservées

Une troisième solution est basée sur les adresses anycast réservées. Elle définit plusieurs adresses réservées dans les fichiers de configuration du résolveur d'une machine IPv6. Le [RFC 1546](#) présente plusieurs pistes. Aucun mécanisme de transport ou protocole n'est donc nécessaire. Cette solution s'appuie sur le routage normal des datagrammes et, selon les cas, un filtrage peut être nécessaire en périphérie du réseau.

Ce service est utilisable lorsque les machines IPv6 souhaitent localiser un hôte supportant un service, sans s'intéresser au serveur qui, lorsqu'il y en a plusieurs, rend le service. Le principe est le suivant : une machine envoie un datagramme vers une adresse anycast. L'interconnexion de réseau assure la remise du datagramme à au plus un serveur et, de préférence, à un seul des serveurs répondant à cette adresse anycast. Lorsque des serveurs sont répliqués, une machine peut, par exemple, accéder à la réplique la plus proche. Un certain nombre de questions se posent dans le cas de services "sans état" et "avec état", notamment lorsque plusieurs serveurs sont susceptibles de répondre.

Mises en œuvre d'un serveur de noms

La mise en œuvre d'un service de nommage dépasse le cadre de cette présentation. Vous trouverez, en annexe 3 de cette activité, un exemple détaillé de mise en œuvre d'un serveur Bind9.

Références bibliographiques

1. ↑ Zitrax [Livre sur les principes du DNS et les éléments de configuration de bind](#)

Pour aller plus loin

RFC et leur analyse par S. Bortzmeyer :

- [RFC 608](#) Host Names On-line
- [RFC 1034](#) Domain Names - Concepts And Facilities [Analyse](#)
- [RFC 1035](#) Domain Names - Implementation And Specification [Analyse](#)
- [RFC 1546](#) Host Anycasting Service
- [RFC 1912](#) Common DNS Operational and Configuration Errors
- [RFC 1918](#) Address Allocation for Private Internets [Analyse](#)
- [RFC 3315](#) Dynamic Host Configuration Protocol for IPv6 (DHCPv6) [Analyse](#)
- [RFC 3596](#) DNS Extensions to Support IP Version 6
- [RFC 3646](#) DNS Configuration options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6) [Analyse](#)
- [RFC 3736](#) Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6
- [RFC 3901](#) DNS IPv6 Transport Operational Guidelines
- [RFC 4291](#) IP Version 6 Addressing Architecture [Analyse](#)
- [RFC 4339](#) IPv6 Host Configuration of DNS Server Information Approaches
- [RFC 4472](#) Operational Considerations and Issues with IPv6 DNS [Analyse](#)
- [RFC 4861](#) Neighbor Discovery for IP version 6 (IPv6) [Analyse](#)
- [RFC 4862](#) IPv6 Stateless Address Autoconfiguration [Analyse](#)
- [RFC 8106](#) IPv6 Router Advertisement Options for DNS Configuration [Analyse](#)
- [RFC 6891](#) Extension Mechanisms for DNS (EDNS(0)) [Analyse](#)

Activité 34 : Contrôler la configuration réseau par DHCPv6

Vous suivez une activité d'approfondissement



Introduction

L'autoconfiguration "à état" utilise un serveur pour allouer des adresses IPv6 ou des paramètres de configuration à des nœuds IPv6. Elle réduit les efforts de configuration des nœuds IPv6, tout comme l'autoconfiguration "sans état". Elle offre, à la différence de l'autoconfiguration "sans état", une information de configuration plus riche et un meilleur contrôle de l'affectation des paramètres de configuration. Elle permet en outre la reconfiguration éventuelle des équipements du réseau.

Les deux techniques d'auto-configuration, "avec état" et "sans état", ne sont pas exclusives et peuvent coexister dans un même environnement. Un nœud peut, par exemple, obtenir son adresse "unicast globale" par auto-configuration "sans état" et obtenir les informations relatives aux serveurs de noms (DNS) par l'autoconfiguration "avec état".

L'autoconfiguration "avec état" permet :

- d'assigner des adresses IPv6 stables et prédictibles à la demande et de manière contrôlée ;
- de provisionner au préalable les adresses à assigner aux nœuds ;
- d'automatiser le mécanisme d'assignement ;
- de centraliser les configurations.

Tout le mécanisme d'autoconfiguration "avec état" est bâti sur le modèle client-serveur. Il utilise le protocole DHCPv6 (*Dynamic Host Configuration Protocol for IPv6*).

Principe de fonctionnement du protocole DHCPv6

Le [RFC 8415](#) définit le principe de fonctionnement du protocole DHCPv6. Ce document spécifie l'architecture de communication, les principes de fonctionnement de chaque entité et le format des messages échangés par ces entités. La mise au point de ce protocole a cependant fait l'objet de nombreux débats au sein du groupe de travail de l'IETF. DHCP est un élément important du fonctionnement d'un réseau. En conséquence, la parution tardive d'un standard finalisé a entraîné un déploiement lent.

Présentation générale du protocole DHCPv6

Le protocole DHCPv6 est un protocole de niveau application. Il fonctionne conformément au modèle client-serveur. Il utilise une communication en mode "non connecté", sous forme d'échanges de type requêtes / réponses. Son architecture fait intervenir quatre types d'entités : les clients, les serveurs, les relais et les interrogateurs (*requestors*). Les clients sollicitent les serveurs pour obtenir des adresses IPv6 ou des paramètres de configuration du réseau. Ils

communiquent directement avec les serveurs DHCPv6 lorsqu'ils se trouvent sur le même lien (au sens de la couche 2 du modèle OSI). Lorsque clients et serveurs ne se trouvent pas sur les mêmes liens, un ou plusieurs relais intermédiaires acheminent les requêtes des clients vers les serveurs. Réciproquement, ils relaient également les réponses des serveurs destinées aux clients. Les administrateurs utilisent les interrogateurs pour obtenir des informations relatives aux paramètres de configuration des clients de leurs serveurs DHCPv6. Enfin, il existe deux types de messages : ceux échangés entre clients et serveurs et ceux échangés, soit entre relais, soit entre relais et serveurs.

Communication en DHCPv6

DHCPv6 utilise le protocole de transport UDP. Les messages UDP sont encapsulés dans des datagrammes IPv6. Les numéros de ports d'écoute utilisés sont 546 pour le client et 547 pour les serveurs ou les relais.

Lorsque le client et le serveur sont sur le même lien, le serveur reçoit la requête du client sur son port 547. Lorsque le client n'est pas sur le même lien que le serveur, un relais reçoit la demande du client sur son port 547. Le relais réexpédie ensuite ce message vers le port 547 du relais suivant ou du serveur.

Le serveur DHCPv6 envoie ses réponses depuis son port 547. Il les envoie vers le port 546 du client si la remise directe est possible. Sinon, le serveur envoie sa réponse au premier relais du chemin de retour, sur le port 547.

En fonction des indications du serveur DHCPv6, les communications peuvent, au niveau IPv6, se faire en point à point ou en multidiffusion pour la découverte des serveurs DHCPv6. IPv6 s'appuie ensuite sur les fonctions de diffusion générale ou sélective du réseau physique sous-jacent pour assurer le transport effectif des messages vers leur destination. Lorsque le réseau n'est pas diffusant, il fait par exemple appel à un serveur de diffusion.

Les entités du protocole

Le protocole DHCPv6 utilise quatre entités pour fonctionner : le client, le serveur, le relais et l'interrogateur. L'utilisation de la quatrième entité, l'interrogateur, est facultative.

- Le serveur DHCPv6 centralise les paramètres de configuration des équipements du réseau.
- Le client DHCPv6 est une machine candidate à une connectivité globale IPv6. Il demande des informations de configuration du réseau à un serveur DHCPv6 pour activer cette connectivité. Il est en relation directe (c'est-à-dire qu'il est sur le même lien) soit avec un relais DHCPv6, soit avec le serveur DHCPv6. Il émet des messages DHCPv6 au serveur DHCPv6.
- Les relais sont transparents. Le client ignore l'existence des relais DHCPv6 et a l'impression de communiquer directement avec le serveur DHCPv6. Ce sont des équipements reliés à plusieurs liens. Ils interceptent le trafic des clients DHCPv6 pour l'acheminer vers les serveurs DHCPv6 lorsque ces derniers ne se trouvent pas sur le lien

du client. Ils utilisent pour cela des options spécifiques des relais. *Notez que ni les relais, ni le serveur ne modifient les messages du client. Les relais se contentent de les encapsuler dans une option de message de relais avant de les relayer vers le serveur.*

- Les interrogateurs (*requestors*) [[RFC 5007](#)] sont des entités spécifiques. Les administrateurs les utilisent pour demander à un serveur DHCPv6 des informations relatives aux clients. Un administrateur peut ainsi obtenir des informations relatives au bail d'un client ou à la machine qui utilise une adresse à un instant donné, ou encore obtenir les adresses allouées à un client donné. Nous ne détaillerons pas ici leur utilisation.

Gestion centralisée des ressources allouées

Le client, dans la configuration DHCPv6 "sans état" (*stateless*), a configuré ses adresses IPv6 soit de façon manuelle (fichier interface, intervention de l'administrateur), soit à partir d'informations extraites d'annonces de routeurs (autoconfiguration "sans état"). Il a alors besoin, pour communiquer, d'informations supplémentaires telles que l'adresse IPv6 du serveur DNS.

Lorsque le serveur DHCPv6 transmet des informations statiques, ces dernières ne nécessitent pas de conserver un état. Elles ne font donc pas l'objet d'un enregistrement dans le fichier des baux du serveur DHCPv6.

Le serveur DHCPv6, dans la configuration "avec état" (*stateful*), alloue une ou plusieurs adresses IPv6 au client. Ces adresses font l'objet d'un contrat de location temporaire : un bail. Il consigne alors ce contrat de location dans un registre spécial enregistré dans une mémoire non volatile : le fichier des baux (*lease file*). Pour cette raison, ce type de configuration est dit "avec état".

Principe de l'allocation d'adresse IPv6 à un client en l'absence de relais

Un client DHCPv6 utilise le message DHCPv6 SOLICIT pour découvrir les serveurs configurés pour lui fournir des adresses IPv6 ou des paramètres de configuration du réseau. Comme, à priori, le client ignore l'adresse IPv6 du serveur, le client DHCPv6 envoie toujours ce message à l'adresse multicast FF02::1:2 qui identifie le groupe des serveurs et relais DHCPv6 (*ALL_DHCP_Relay_Agents_And_Servers*).

Les serveurs capables d'allouer des adresses au client répondent avec un message DHCPv6 ADVERTISE. Ils font une offre au client DHCPv6. Si plusieurs serveurs DHCPv6 sont disponibles, le client ne collecte leurs réponses que pendant un certain temps. Il sélectionne ensuite l'offre qui satisfait le mieux ses besoins. Il émet alors un message REQUEST destiné au serveur choisi. Il envoie ce message à l'adresse de diffusion sélective *ALL_DHCP_Relay_Agents_And_Servers*. Tous les serveurs qui ont répondu à la demande du client savent ainsi si leur offre a été retenue ou non. Le serveur dont l'offre a été retenue, et lui seul, retourne un message REPLY au client. La figure 1 résume les messages DHCPv6 échangés dans ce cas.

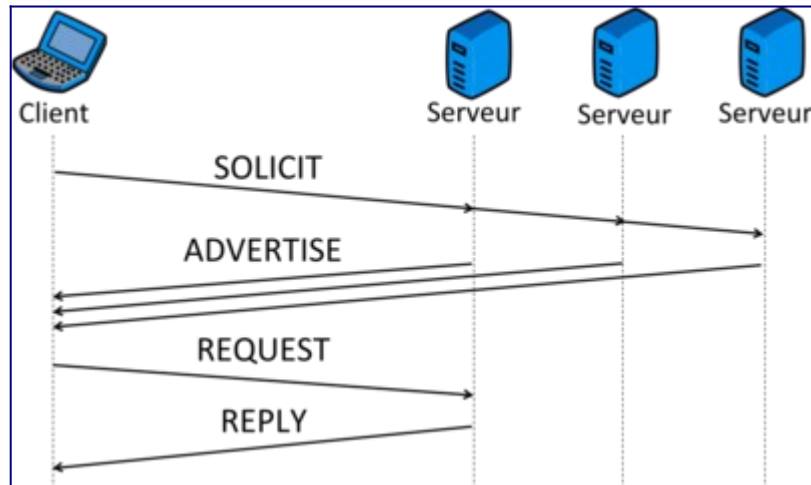


Figure 1 : Dialogue entre client et serveur DHCPv6 présents sur le même lien physique.

Recherche des serveurs DHCPv6 par le client : fonctionnement de la pile de communication

Le client DHCPv6 demande au serveur une adresse IPv6 et un certain nombre de paramètres de configuration du réseau. Il fabrique donc un message DHCPv6 SOLICIT. Il émet ensuite ce message DHCPv6 SOLICIT pour découvrir les serveurs DHCPv6 disponibles.

Il s'adresse localement au protocole UDP sur le port local du client DHCPv6 (546) pour expédier ce message vers le port UDP destination du serveur (547). Comme, à ce stade, le client DHCPv6 ignore l'adresse IPv6 du serveur, il fournit à UDP l'adresse IPv6 de multicast réservée au protocole DHCPv6 comme adresse IPv6 de destination.

UDP ne gère pas les adresses IPv6. Il transmet donc simplement l'adresse IPv6 de destination du message UDP à la couche IPv6.

IPv6 fabrique l'en-tête du datagramme qui transporte le message DHCPv6 encapsulé dans UDP. Si notre client n'a qu'une interface, celle-ci est associée à la route par défaut. Sinon, le client envoie le message depuis l'interface de réseau associée à la route par défaut. L'adresse IPv6 "source" utilisée dans le datagramme IPv6 est l'adresse locale au lien de cette interface.

Notez que l'administrateur du réseau définit l'interface de réseau à utiliser par défaut. Il peut effectuer cette configuration au niveau d'une image disque ou encore au niveau d'un fichier de configuration du client DHCPv6.

L'adresse de destination est une adresse de diffusion sélective. Elle n'est associée à aucune route spécifique. Le trafic destiné à ce groupe emprunte la route par défaut. L'adresse IPv6 "source" utilisée ici est donc l'adresse locale au lien de cette interface.

IPv6 demande ensuite à Ethernet d'expédier ce datagramme. L'adresse IPv6 de diffusion sélective de destination est ensuite associée à l'adresse Ethernet de diffusion sélective spécifique d'IPv6 (selon le mécanisme d'association d'une adresse IPv6 de multicast à une adresse MAC de multicast, tel qu'il est présenté dans l'activité 15 de la séquence 1). Ceci permet d'utiliser, au niveau d'Ethernet, la diffusion sélective et de ne pas recourir, sur le lien, à la diffusion générale ; ce qui dérangerait un nombre potentiellement considérable de machines

sur un réseau IPv6.

Principe de l'allocation d'adresse IPv6 à un client en présence d'un relais DHCPv6

Lorsque le client se trouve sur un lien différent de celui du serveur DHCPv6, ce dernier ignore sur quel lien se trouve le client. Il ne peut alors allouer des adresses correspondant aux liens du client qu'à condition de pouvoir identifier ces liens, et donc d'identifier le ou les préfixes à y utiliser.

Le routeur intermédiaire, entre le client et le serveur DHCPv6, doit supporter une fonction relais DHCPv6. Comme DHCPv6 est un nouveau protocole spécifique d'IPv6, il n'a pas de contrainte de compatibilité ascendante. C'est pourquoi le fonctionnement des relais DHCPv6 est différent de celui des relais DHCPv4.

L'activation de la fonction relais DHCPv6 sur le routeur le transforme en relais DHCPv6. Nous ferons un abus de langage en nommant ce routeur "relais DHCPv6" (nous l'avons déjà fait mais sans le dire...). Notez que, pour un routeur Linux, par exemple, il suffit de configurer un processus relais DHCPv6 et d'activer ce processus pour que le relais soit opérationnel.

Un relais DHCPv6 qui reçoit un message DHCPv6 d'un client l'encapsule dans un message DHCPv6 RELAY-FORWARD. Le message du client est inclus dans l'option "message relayé" du message RELAY-FORWARD que le relais envoie ensuite au serveur DHCPv6, soit en utilisant l'adresse de diffusion sélective réservée, et dans ce cas aucune configuration n'est nécessaire, soit en utilisant l'adresse individuelle (unicast) du serveur DHCPv6. L'administrateur du réseau doit, bien entendu dans ce cas, adapter la configuration du serveur et des relais en fonction du type d'adresse, individuelle ou diffusion sélective, utilisé.

Lorsque le message DHCPv6 d'un client doit traverser plusieurs relais DHCPv6, chaque relais encapsule le message RELAY-FORWARD reçu du relais précédent dans l'option "message relayé" de son propre message RELAY-FORWARD.

Chaque relais traversé identifie (adresse globale ou locale au lien), dans son message RELAY-FORWARD, l'interface sur laquelle il a reçu le message du client ou du relais précédent et l'adresse locale au lien de l'interface par laquelle il réexpédie son message RELAY-FORWARD au serveur ou au relais suivant.

Notez que le message du client est recopié dans l'option "message relayé" du message RELAY-FORWARD du premier relais DHCPv6 traversé. Si le message traverse plusieurs relais, l'option "message relayé" du relais courant contient le message RELAY-FORWARD du relais précédent.

Lorsque le serveur DHCPv6 reçoit le message RELAY-FORWARD du dernier relais DHCPv6, l'en-tête de ce message contient l'adresse IPv6 du dernier relais. Il saura donc où envoyer son message RELAY-REPLY.

Chaque relais intermédiaire procède de la sorte en extrayant le message RELAY-REPLY du relais précédent de l'option "message relayé" du message RELAY-REPLY reçu.

Le chemin inverse n'est par conséquent pas difficile à construire. Le protocole DHCPv6 peut ainsi faire parvenir la réponse du serveur au client.

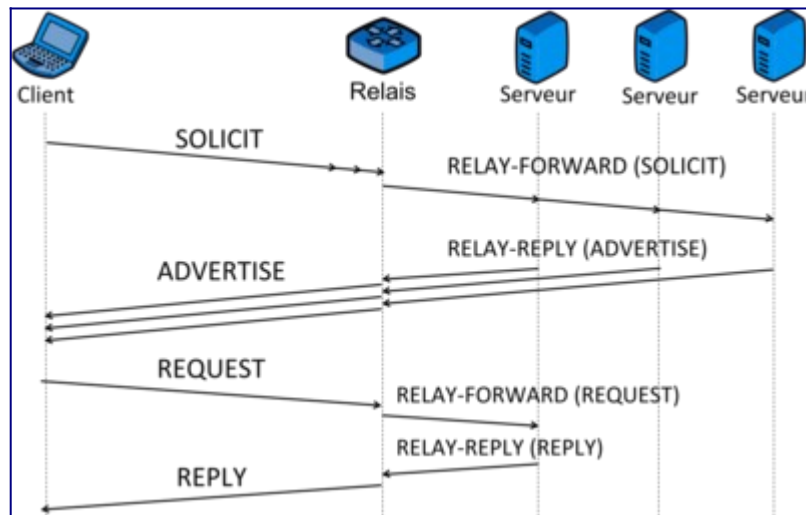


Figure 2 : Dialogue entre client et serveur DHCPv6 non présents sur le même lien physique.

Après la phase d'acquisition de l'adresse IPv6, le client DHCPv6 vérifie que l'adresse IPv6 allouée n'est pas déjà en service (DAD : détection d'adresse dupliquée). Il configure alors ses interfaces de réseau, et l'utilisateur qui travaille sur le client DHCPv6 peut accéder au réseau.

Le processus DHCPv6 client devient alors inactif jusqu'à ce que l'utilisateur qui travaille sur le client DHCPv6 ferme sa session et arrête le client. Il se réactive alors pour libérer (*release*) l'adresse IPv6 allouée.

Libération de l'adresse IPv6 par un client DHCPv6

Le processus d'arrêt normal du client DHCPv6, par échange des messages RELEASE / REPLY inclut la libération de l'adresse IPv6 allouée par le serveur.

La figure 3 ci-dessous présente la libération de l'adresse IPv6 en l'absence de relais :

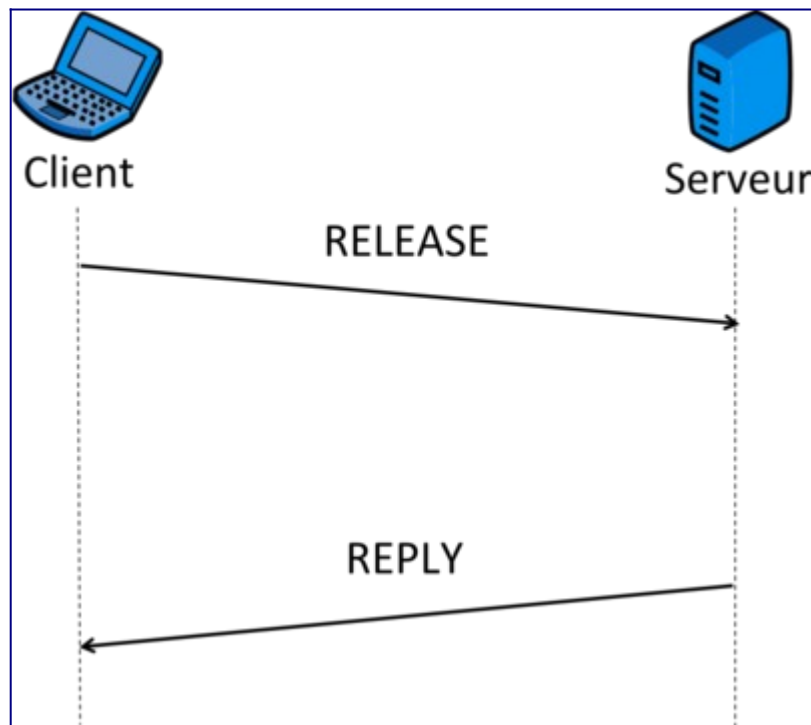


Figure 3 : Libération d'une adresse IPv6 obtenue directement d'un serveur DHCPv6.

La figure 4 ci-dessous présente la libération de l'adresse IPv6 en présence d'un relais :

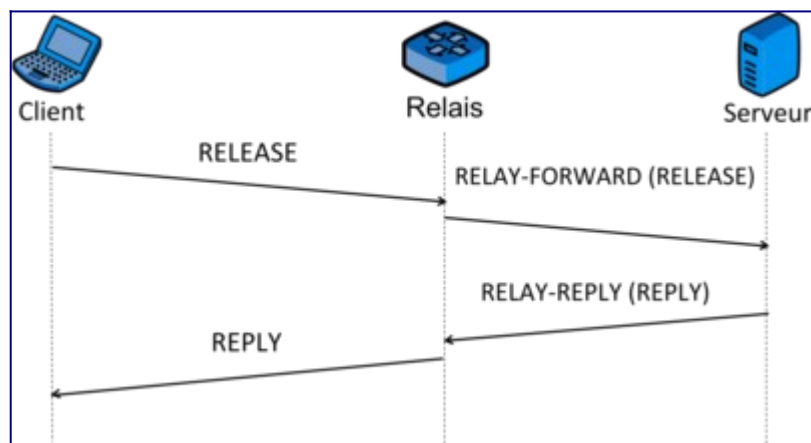


Figure 4 : Libération d'une adresse IPv6 obtenue via un relais DHCPv6.

Fonctions des messages du protocole DHCPv6

Cette partie introduit les messages du protocole DHCPv6. Ce protocole distingue deux types de messages : d'une part, les messages échangés entre client et serveur et, d'autre part, les messages échangés entre serveur et relais. Nous les présentons successivement dans cet ordre.

En général, les messages échangés transportent des identificateurs de transactions et des associations d'identités. Les serveurs DHCPv6 utilisent les identificateurs de transactions pour associer leurs réponses aux demandes correspondantes des clients. L'identificateur de transaction change pour chaque transaction et est globalement unique pour une transaction donnée. Mais les messages associés à une transaction se distinguent notamment par le champ

Type de l'en-tête DHCPv6.

Les associations d'identités permettent aux serveurs et aux clients de s'identifier mutuellement. Elles identifient également les interfaces de réseau concernées par les demandes de paramètres de configuration du réseau des clients ou par les réponses des serveurs. Elles sont également transmises dans des options du protocole DHCPv6.

Messages échangés entre client et serveur

Un client utilise le message SOLICIT (champ Type = 1) pour localiser les serveurs configurés pour allouer des adresses ou des paramètres de configuration du réseau.

Un serveur configuré pour fournir des adresses ou des paramètres de configuration du réseau aux clients annonce sa disponibilité au client DHCPv6 à l'aide d'un message ADVERTISE (champ Type = 2).

Un client utilise ensuite le message REQUEST (champ Type = 3) pour demander des adresses ou des paramètres de configuration au serveur DHCPv6 choisi. Une option *options demandées* contient la liste des paramètres de configuration qu'il demande.

Un serveur utilise le message REPLY (champ Type = 7) pour répondre à un message SOLICIT ou REQUEST reçu d'un client DHCPv6.

Messages de gestion des ressources allouées

Un client utilise le message CONFIRM (champ Type = 4) pour indiquer au serveur qui lui a alloué adresses et paramètres de configuration du réseau et que ces paramètres sont adaptés au lien auquel il est raccordé.

Un client utilise le message RENEW (champ Type = 5) pour prolonger le bail de location des adresses et actualiser des paramètres de configuration auprès du serveur qui les lui a alloués. Le client utilise ce message à la demande explicite du serveur.

Un client utilise le message REBIND (champ Type = 6) pour obtenir un bail de location des adresses et actualiser des paramètres de configuration auprès de tout serveur DHCPv6, si le serveur DHCPv6 auquel il s'est adressé pour renouveler le bail de ses adresses et ses paramètres de configuration du réseau ne répond pas à son message RENEW.

Un serveur utilise le message REPLY (champ Type = 7) pour répondre à un message RENEW ou REBIND reçu d'un client.

Un client utilise le message RELEASE (champ Type = 8) pour indiquer au serveur DHCPv6 qu'il libère des adresses IPv6.

Un client utilise le message DECLINE (champ Type = 9) pour signaler au serveur qu'une ou des adresses allouées par le serveur sont déjà utilisées sur le lien du client. La DAD (détection d'adresses dupliquées) d'IPv6 peut, par exemple, fournir cette information.

Notez que la détection d'adresses dupliquées incombe toujours au client DHCPv6. En effet, le serveur DHCPv6 ne peut effectuer la DAD que lorsqu'il se trouve sur le même réseau que son

client, ce qui n'est pas toujours le cas. Or, la DAD n'est possible que sur un lien auquel on est connecté.

Un serveur utilise le message RECONFIGURE (champ Type = 10) pour signaler au client qu'il a de nouveaux paramètres de configuration du réseau ou les a actualisés. Ce message précise en particulier si le client doit utiliser le message RENEW ou REBIND.

Un client utilise le message INFORMATION-REQUEST (champ Type = 11) pour demander au serveur des paramètres de configuration du réseau, sans demander d'adresse.

Messages échangés entre relais et serveur

Un relais DHCPv6 utilise le message RELAY-FORWARD (champ Type = 12) pour relayer des messages DHCPv6 vers un serveur DHCPv6. Le message relayé est soit le message DHCPv6 du client, soit le message RELAY-FORWARD du relais précédent (sur le chemin reliant le client au serveur DHCPv6). Un relais DHCPv6 ne modifie jamais le message d'un client.

Le message du client DHCPv6 est relayé, sans être modifié, dans une option *message relayé* du message RELAY-FORWARD du premier relais rencontré sur le chemin reliant le client au serveur DHCPv6.

Un serveur DHCPv6 utilise le message RELAY-REPLY (champ Type = 13) pour envoyer un message à un client, via un relais.

Chaque relais qui reçoit un message RELAY-REPLY extrait le message contenu dans l'option "message relayé" et le réexpédie vers le client. Seul le contenu de l'option "message relayé" est donc transmis vers le client.

Le dernier relais extrait le message REPLY destiné au client et contenu dans l'option "message relayé" de ce message RELAY-REPLY pour le lui remettre. Ici encore, le message du client reste inchangé.

Tableau récapitulatif des messages DHCPv6

Le tableau ci-dessous résume le nom, le type, l'émetteur et la fonction des messages DHCPv6 échangés entre client et serveur.

Message DHCPv6			
Type		Emetteur	Fonction
		r	
SOLICIT	1	Client	Localiser les serveurs configurés pour fournir des adresses ou des paramètres de configuration .
ADVERTISE	2	Serveur	Annoncer la disponibilité du serveur DHCPv6.
REQUEST	3	Client	Demander des adresses ou des paramètres de configuration au serveur choisi.
CONFIRM	4	Client	Indiquer au serveur qui a alloué adresses et

paramètres de configuration que ces paramètres sont adaptés au lien auquel le client est raccordé.

RENEW	5	Client	Prolonger le bail de location des adresses et actualiser des paramètres de configuration auprès du serveur qui les a alloués.
REBIND	6	Client	Obtenir un bail de location des adresses et actualiser des paramètres de configuration auprès de tout serveur en cas de non réponse au message RENEW.
REPLY	7	Serveur	Répondre à un message SOLICIT, REQUEST, REBIND, RELEASE reçu d'un client.
RELEASE	8	Client	Indiquer au serveur que le client n'utilise plus des adresses IPv6.
DECLINE	9	Client	Signaler au serveur qu'une ou des adresses allouées par le serveur sont déjà utilisées sur le lien du client.
RECONFIGURE	10	Serveur	Signaler au client que le serveur a de nouveaux paramètres ou les a actualisés.
INFORMATION-REQUEST	11	Client	Demander des paramètres de configuration au serveur, sans demander d'adresse.
RELAY-FORWARD	12	Relais	Relayer des messages vers un serveur DHCPv6. Le message relayé (celui du client DHCPv6 ou du relais précédent) est placé dans une option de ce message RELAY-FORW.
RELAY-REPLY	13	Serveur	Envoyer, depuis un serveur, un message à un client via un relais . Le relais extrait le message destiné au client ou au relais suivant contenu dans l'option "message relayé" de ce message pour le lui remettre.

Extension du protocole DHCPv6 [\[RFC 6422\]](#)

Notez qu'un mécanisme d'option de relais spécifique permet qu'un relais DHCPv6 communique des paramètres de configuration susceptibles d'intéresser un client DHCPv6 et dont il a connaissance au serveur DHCPv6.

Le serveur DHCPv6 peut ensuite décider ou non, en fonction de la politique définie par l'administrateur du réseau, de communiquer au client tout ou partie des paramètres de configuration du réseau spécifiques issus du relais.

Structure des messages DHCPv6

Le document [RFC 8415](#) décrit l'ensemble des éléments du protocole DHCPv6. À l'instar de nombreux protocoles de l'Internet, le protocole d'échange d'informations est découplé de l'information elle-même. La nature des informations échangées peut donc changer et évoluer rapidement, sans impacter les mécanismes de cet échange. Cette séparation assure la stabilité et l'extensibilité du protocole.

La structure des unités de données du protocole reprend ce découpage : un en-tête de taille fixe pour les informations du protocole lui-même et une charge utile transportée dans des champs d'option pour les informations applicatives.

Pour étendre le protocole, il suffit de définir de nouvelles options et de concevoir leur traitement, en émission et en réception. Les options utilisables par DHCPv6 sont référencées dans un registre maintenu par l'IANA[1]. Dans la terminologie DHCPv6, le terme "message" désigne une unité de données du protocole DHCPv6. Chaque type de message DHCPv6 (client-serveur ou relais-serveur) a un format d'en-tête identique. De ce point de vue, DHCPv6 reprend les principes de simplification du processus de développement du protocole qui ont guidé la conception du format du segment TCP : un seul format pour l'ensemble des fonctions de TCP.

Structure des messages émis par les serveurs et clients DHCPv6

La structure générale des messages échangés entre client et serveur DHCPv6 est la suivante : un champ type *Type-msg*, un champ identificateur de transaction *ID-transaction*, et une liste variable d'options, *Option list* (voir la figure 5).

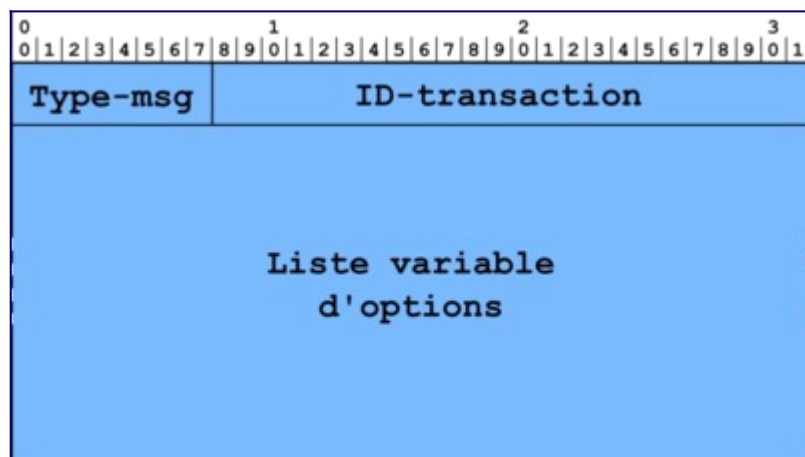


Figure 5 : Format des messages échangés entre clients et serveurs DHCPv6.

Type-msg : le champ type de message identifie la nature du message DHCPv6. Il est codé sur un octet.

Id-transaction : l'identificateur de transaction identifie un échange (question/réponse). Il est spécifique aux messages participant à une transaction, et est globalement unique. Il permet d'associer les réponses aux requêtes correspondantes. En effet, la couche transport UDP ne garantit pas le séquençement des réponses lorsque plusieurs requêtes successives ont été émises à destination d'un serveur. Il est codé sur 3 octets.

Option list : la liste des options du message est de taille variable. Elle correspond à une succession d'options rangées séquentiellement, selon la sémantique du message, et uniquement alignées sur des frontières d'octets. Il n'y a pas de bourrage entre deux options consécutives. Elles transportent soit les adresses IPv6, soit les paramètres de configuration du réseau (hors adresse IPv6) nécessaires au fonctionnement du réseau.

Pour en savoir plus sur les options, reportez-vous à l'annexe 1 *Options du protocole DHCPv6* de cette activité.

Structure des messages échangés entre relais et serveur DHCPv6

La figure 6 présente la structure des messages échangés entre relais et serveur.

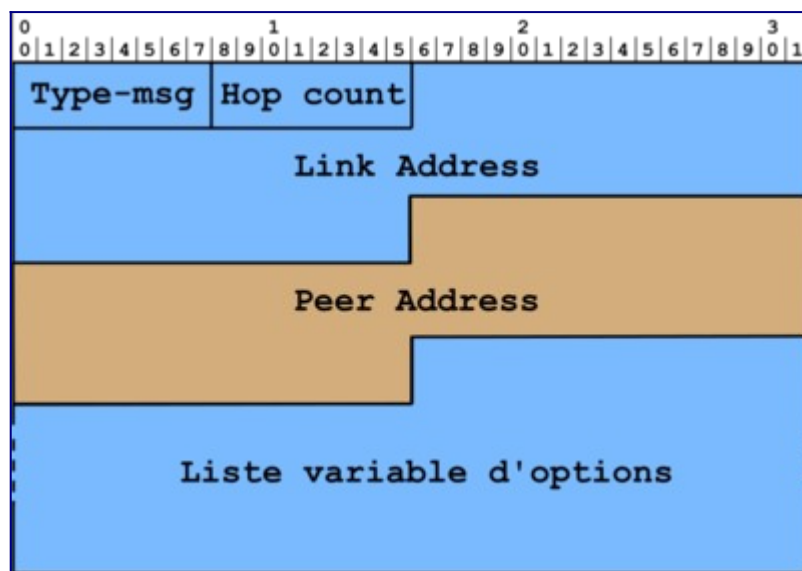


Figure 6 : Format des messages échangés entre relais et serveurs DHCPv6.

Les messages utilisés pour la communication entre serveur et relais sont différents des messages utilisés pour la communication entre client et serveur. Un message RELAY-FORWARD transite d'un relais vers un serveur. Un message RELAY-REPLY transite du serveur vers le client.

Type-msg : le type du message identifie le type du message DHCPv6.

Hop-count : le nombre de sauts identifie soit le nombre de relais déjà traversés pour atteindre le serveur, soit le nombre de relais restant à traverser pour atteindre le client.

Link-address : l'adresse de lien est une adresse unicast (globale ou locale) qui sera utilisée par le serveur pour identifier le lien sur lequel est localisé le client. C'est l'adresse unicast (globale ou locale) du relais du côté du client.

Peer-address : l'adresse du pair est l'adresse du client ou du relais depuis laquelle le message à relayer a été reçu. Elle est extraite de l'adresse source du paquet du message reçu. Elle permet d'identifier l'interface du relais derrière laquelle se trouve le client. Elle sera utilisée comme adresse de destination du paquet contenant le message RELAY-REPLY.

Ainsi, même en présence de plusieurs relais DHCPv6, le serveur sait auquel des relais

s'adresser pour répondre à un client donné. Chacun des relais, lorsqu'il faut en traverser plusieurs pour atteindre le client, sait à qui transmettre le message RELAY-REPLY reçu. Le champ Peer-address de ce message contient l'adresse locale au lien du relais suivant ou, pour le dernier relais, l'adresse locale au lien du client. Le dernier relais peut donc envoyer au client la réponse du serveur.

Message DHCPv6 RELAY-FORWARD

Type-msg : le champ type de ce message vaut 12.

Hop-count : le nombre de sauts indique le nombre de relais traversés par ce message pour atteindre le serveur.

Link-address : l'adresse de lien, est une adresse unicast (globale ou locale) qui sera utilisée par le serveur pour identifier le lien sur lequel est localisé le client. C'est l'adresse unicast (globale ou locale) du relais du côté du client.

Peer-address : l'adresse du pair est l'adresse du client ou du relais depuis laquelle le message à relayer a été reçu. Elle est extraite de l'adresse source du paquet du message reçu. Elle permet d'identifier l'interface du relais derrière laquelle se trouve le client. Elle sera utilisée comme adresse de destination du paquet contenant le message RELAY-REPLY.

Option list : la liste d'options de ce message contient obligatoirement une option de message relayé (Relay Message Option) et éventuellement d'autres options ajoutées par le relais.

Notez qu'en aucun cas le relais ne modifie le message DHCPv6 du client.

Message DHCPv6 RELAY-REPLY

Le serveur envoie ce message au premier relais sur le chemin du retour vers le client demandeur.

Type-msg : le champ type de ce message vaut 13.

Hop-count : le nombre de sauts indique le nombre de relais que ce message traversera pour atteindre le client.

Link-address et Peer-address : les adresses du lien et du pair sont recopiées à partir du message RELAY-FORWARD précédent.

Option list : la liste d'options doit obligatoirement contenir une option de message relayé (*Relay Message option*). Cette option transporte la réponse du serveur DHCPv6 destinée au client DHCPv6.

Types de DUID : DHCPv6 Unique Identifier

Afin de connaître l'état des ressources gérées (représentées par les paramètres de configuration), le serveur DHCP gère une liste d'associations entre le paramètre attribué et le client. Comme l'adresse unicast du client est une ressource sous le contrôle du serveur, celle-ci

ne peut pas être utilisée pour identifier un client. Le serveur référence donc le client par un identifiant unique à usage exclusif de DHCP : le DUID (*DHCP Unique Identifier*).

Chaque station génère son identifiant. Cet identifiant doit être permanent et avoir une grande durée de vie. Une station peut, par exemple, et à un instant donné, générer un DUID à partir de l'adresse MAC d'une de ses cartes réseau. Elle le conservera alors comme identifiant, même en cas de remplacement ultérieur de cette carte réseau.

Les clients utilisent les DUID pour identifier les serveurs quand ils en ont besoin ; par exemple, pour mémoriser l'identité du serveur qui leur a alloué des adresses IPv6 ou des paramètres de configuration du réseau. Le contenu des DUID n'est pas interprété mais uniquement utilisé pour des comparaisons ou pour vérifier l'identité du correspondant. Le DUID concerne la machine (client ou serveur) et non une de ses interfaces.

Le [RFC 8415](#) définit trois types d'identificateurs uniques DHCPv6 (DUID). Les DUID peuvent donc être générés selon trois méthodes, repérées par le champ `type` de DUID dont les valeurs respectives sont :

- 1 : **DUID-LLT** (*Link-Layer address plus Time*) résultant de la combinaison d'une adresse physique et d'une horodate ;
- 2 : **DUID-EN** (*Vendor-assigned unique ID based on Enterprise Number*) dérivé d'un numéro de constructeur ou d'un numéro unique affecté par un constructeur ;
- 3 : **DUID-LL** (*Link-Layer address*) dérivé de l'adresse MAC d'une interface de réseau.

Le type de DUID est codé sur 2 octets. Un nombre variable d'octets suit, et constitue l'identificateur. La longueur maximale d'un identificateur est 128 octets.

Le DUID est lui-même une structure de données qui, selon le mode de construction, contient des types de valeurs différents (*la structure détaillée des différents type de DUID est présentée en annexe 3 de cette séquence*).

Association d'identités

Une association d'identités IA (*Identity Association*) permet qu'un serveur ou un client identifie, groupe ou gère un ensemble d'adresses IPv6 associées. Chaque association se compose d'un identificateur d'association et des informations de configuration associées. Ces informations sont enregistrées dans des options de l'association.

Un client associe au moins une association d'identités, IA, à chacune des interfaces de réseau pour laquelle il requiert une adresse IPv6.

Cette IA reste affectée en permanence à l'interface. Elle simplifie le format des messages DHCPv6, la gestion de la durée de vie des adresses IPv6 ou encore la renumérotation du réseau IPv6.

Les informations de configuration correspondent à une ou plusieurs adresses IPv6 et à leurs temporisations associées, T1 et T2, où :

- T1 représente la durée de vie de l'adresse dans l'état préféré ;

- T2 représente la durée de validité de l'adresse IPv6.

Un serveur DHCPv6 peut allouer deux types d'adresses IPv6 :

- des adresses non temporaires ;
- des adresses temporaires.

Allocation des adresses non temporaires

Le serveur choisit les adresses d'un client en fonction du lien du client, du DUID du client, des options fournies par le client, et des informations fournies par le relais DHCPv6.

Les adresses allouées font l'objet d'une écriture dans le fichier des baux.

Allocation des adresses temporaires

DHCPv6 gère les adresses temporaires comme les adresses non temporaires : une association d'identités pour adresse temporaire ne contient au plus qu'une seule adresse temporaire. Ici encore, l'allocation d'adresse fait l'objet d'une écriture dans le fichier des baux.

Le serveur DHCPv6, s'il est configuré pour cela, effectue des mises à jour dynamiques sécurisées du service de noms de domaines.

Options du protocole DHCPv6

Chaque option est codée en format TLV : type, longueur, valeur ; à savoir :

- le type de l'option : un champ `type` d'option identifie chaque option d'un paquet DHCPv6. Il permet l'interprétation des données transportées. Certaines options peuvent en contenir d'autres ou être structurées en plusieurs champs (voir annexe 1 : options du protocole DHCPv6) ;
- la longueur, en octets, du champ `valeur` du paramètre qui suit ;
- le champ `valeur` du paramètre de configuration.

Le champ `type` d'option est toujours codé sur 2 octets. Le champ `longueur` est codé sur 2 octets. Il est toujours présent, même en l'absence de valeur ou pour une information de longueur fixe. Il exclut le champ `type` de l'option.

Le tableau qui suit présente les options du protocole DHCPv6, leur code et leur définition. L'annexe 1 présente leur structure.

Options de DHCPv6

Désignation	Code	Définition
OPTION_CLIENT ID	1	Identification du client
OPTION_SERVER ID	2	Identification du serveur
OPTION_IA_NA	3	Association d'identités pour les options d'adresse non temporaire

OPTION_IA_TA	4	Association d'identités pour les options d'adresse temporaire
OPTION_IAADDR	5	Adresse associée à IA_NA ou IA_TA
OPTION_ORO	6	Identifie une liste d'options dans les messages échangés entre un client
OPTION_PREFERENCE	7	Annonce au client la priorité du serveur DHCPv6 et comment gérer cette priorité.
OPTION_ELAPSED_TIME	8	Temps écoulé depuis le démarrage d'un échange pour la machine qui tente d'achever sa configuration.
OPTION_RELAY_MSG	9	Transporte un message DHCPv6 relayé dans des messages <i>relay-forw</i> ou <i>relay-repl</i>
OPTION_AUTH	11	Transporte les informations d'authentification de l'identité et du contenu des messages DHCPv6.
OPTION_UNICAST	12	Permet au serveur d'indiquer au client qu'il peut utiliser l'adresse individuelle (unicast) du serveur pour échanger avec lui.
OPTION_STATUS_CODE	13	Indique le statut du message DHCPv6 qui transporte cette option.
OPTION_RAPID_COMMIT	14	Permet à un client, dans un message SOLICIT, de demander ce mode de fonctionnement pour réaliser des échanges en deux temps au lieu de quatre. Le serveur doit inclure cette option dans la réponse correspondante (<i>Solicit reply</i>).
OPTION_USER_CLASS	15	Définit la classe d'utilisateur associée à un utilisateur ou à une application.
OPTION_VENDOR_CLASS	16	Identifie le constructeur du matériel utilisé par le client.
OPTION_VENDOR_OPTS	17	Permet que le client et le serveur échangent des informations spécifiques d'un constructeur.
OPTION_INTERFACE_ID	18	Identifie l'interface de réception du message du client DHCPv6.
OPTION_RECONF_MSG	19	Indique, dans un message <i>reconfiguration</i> , si le client doit répondre par un message <i>renew</i> ou <i>information-request</i> .
OPTION_RECONF_ACCEPT	20	Indique à un serveur si le client accepte ou refuse les messages <i>reconfigure</i> ou annonce à un client qu'il peut ou non accepter les messages <i>reconfigure</i> .

Délégation de préfixe à états

La délégation de préfixe à états fait intervenir deux routeurs : un routeur délégataire et un routeur demandeur. Le routeur délégataire alloue les préfixes. Le routeur demandeur demande un ou plusieurs préfixes au routeur délégataire.

La délégation de préfixe à états utilise le protocole DHCPv6 pour déléguer les préfixes. Elle définit deux options : une association d'identités pour l'allocation de préfixes (IA_PD) et une option de préfixe d'association d'identités pour la délégation de préfixes (IA_PD Prefix). Le routeur demandeur émet ses demandes sur l'interface qui donne accès au routeur délégataire.

Le routeur délégataire répond sur l'interface qui donne accès au routeur demandeur. Lorsque ces deux routeurs ne se trouvent pas sur le même réseau, des relais DHCPv6 interviennent, comme dans le cas de l'allocation d'adresses. Leur fonctionnement est inchangé.

La délégation de préfixe à états se fait sans relais lorsque les routeurs délégataire et demandeur sont sur le même lien.

Les options de délégation de préfixe permettent au routeur délégataire de déléguer la gestion d'un ou plusieurs préfixes à un routeur demandeur.

L'association d'identités pour l'allocation de préfixes associe notamment les DUID des routeurs demandeur et délégataire, et les préfixes alloués. L'option de préfixe d'association d'identités pour la délégation de préfixe transporte un préfixe qu'un routeur délégataire a délégué à un routeur demandeur. Cette option peut apparaître plusieurs fois dans une association d'identités (IA_PD).

Notez que la délégation de préfixe à états est indépendante de l'allocation des adresses IPv6.

Applications de la délégation de préfixe

La délégation de préfixe convient pour des situations où le routeur délégataire ignore la topologie du réseau auquel le routeur demandeur donne accès et n'a pas d'autre information à connaître que l'identité du routeur demandeur pour allouer le préfixe. C'est, par exemple, le cas du routeur d'un FAI (Fournisseur d'Accès à Internet) qui alloue un préfixe au routeur d'accès d'un client (CPE : *Customer Premise Equipment*, familièrement dénommé *box*) reliant un réseau interne au réseau du FAI. La figure 7 présente un exemple où la délégation de préfixe à états est possible.

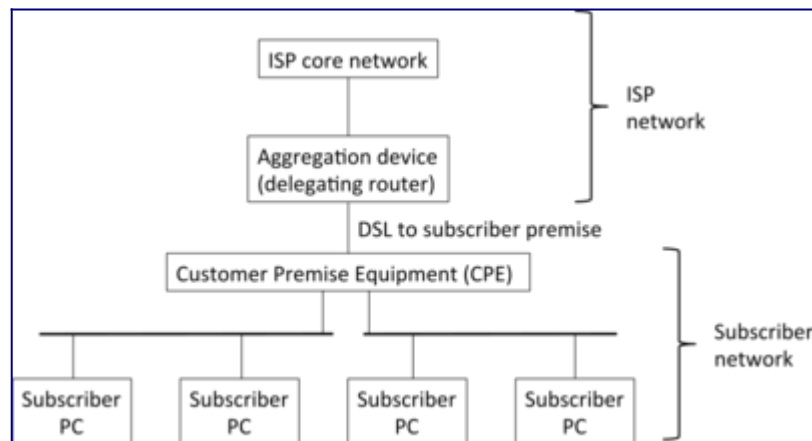


Figure 7 : Exemple de délégation de préfixe à états.

La délégation de préfixe facilite également la renumérotation. Elle permet, par exemple, d'allouer le préfixe qui servira à générer les nouvelles adresses IPv6. Les préfixes sont censés avoir une grande durée de vie. En cas de renumérotation, la cohabitation pendant un certain temps de l'ancien et du nouveau préfixe est fort probable. C'est par exemple le cas pour la renumérotation passive présentée ci-dessous.

Renumérotation des réseaux

La renumérotation peut se faire de deux façons : passive ou active.

Renumérotation passive

Dans la renumérotation passive, chaque machine du réseau dispose de deux adresses IPv6 : une ancienne et une nouvelle. L'ancienne adresse est utilisée par les communications en cours. Ces communications sont préservées aussi longtemps que nécessaire (RENEW). Par contre, les nouvelles communications sont établies à l'aide de la nouvelle adresse. La renumérotation est terminée lorsque la dernière machine du réseau cesse d'utiliser son ancienne adresse.

Renumérotation active

Dans la renumérotation active, chaque machine, comme dans le cas précédent, dispose d'une ancienne adresse et d'une nouvelle.

Le serveur DHCPv6 force les clients à cesser d'utiliser leur ancienne adresse à une date donnée. Le serveur réduit la durée de vie des anciennes adresses en fonction de la date d'échéance cible.

Lorsque la date d'échéance arrive, aucune utilisation d'ancienne adresse n'est possible. Toutes les communications utilisant les anciennes adresses sont coupées. Elles sont, en cas de besoin, rétablies en utilisant les nouvelles adresses.

Ici encore, la délégation de préfixe à états peut faciliter les choses en permettant que les machines autoconfigurent leurs nouvelles adresses.

Notez que l'utilisation du préfixe alloué sur le routeur demandeur est impossible sur le lien

donnant accès au routeur délégataire. Ceci empêche par conséquent l'agrégation des routes d'accès au routeur demandeur et d'accès au réseau qu'il dessert.

Deux autres options [RFC 6603], permettent d'exclure un seul préfixe pour l'affecter au lien qui, sur le routeur demandeur, donne accès au routeur délégataire.

Certains réseaux mobiles doivent pouvoir agréger les routes (vers le routeur demandeur et le réseau interne). Dans ce cas, le routeur demandeur doit utiliser le préfixe du réseau interne de l'interface qui le relie au routeur délégataire. Il utilise alors deux des options du RFC 6603. (l'annexe 4 présente la structure de l'option d'association d'identités pour la délégation de préfixes).

Principe de l'allocation

Le routeur demandeur se comporte comme un client DHCPv6. Il émet un message SOLICIT contenant une association d'identités pour l'allocation de préfixes à états, IA_PD. Le routeur délégataire se comporte comme un serveur DHCPv6. Il alloue les préfixes en fonction de l'identité du routeur demandeur et des options de préfixe indiquées (voir la figure 8).

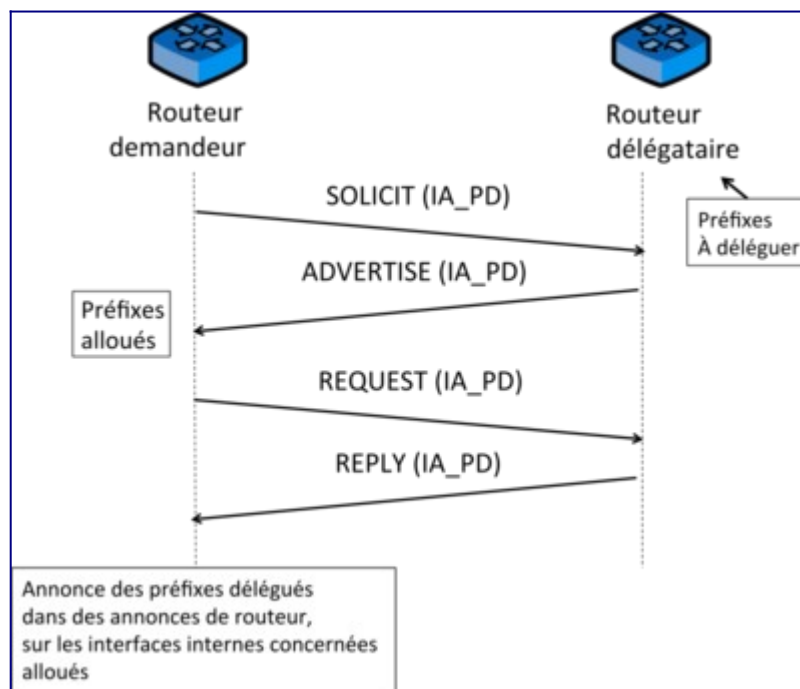


Figure 8 : Allocation de préfixe par un routeur délégataire.

Principe de l'allocation de préfixe à états avec relais

Le relais encapsule le message SOLICIT du client dans l'option "message relayé" de son message RELAY-FORWARD. Il achemine ensuite ce message vers le serveur.

Le serveur renvoie son message RELAY-REPLY au relais.

Le relais extrait le message ADVERTISE de l'option "message relayé" du message RELAY-REPLY du serveur. Il le transmet ensuite au client. Il identifie l'interface d'accès au client grâce à l'adresse du lien incluse dans le champ *Peer-Address* de l'en-tête du message RELAY-REPLY

(voir la figure 9).

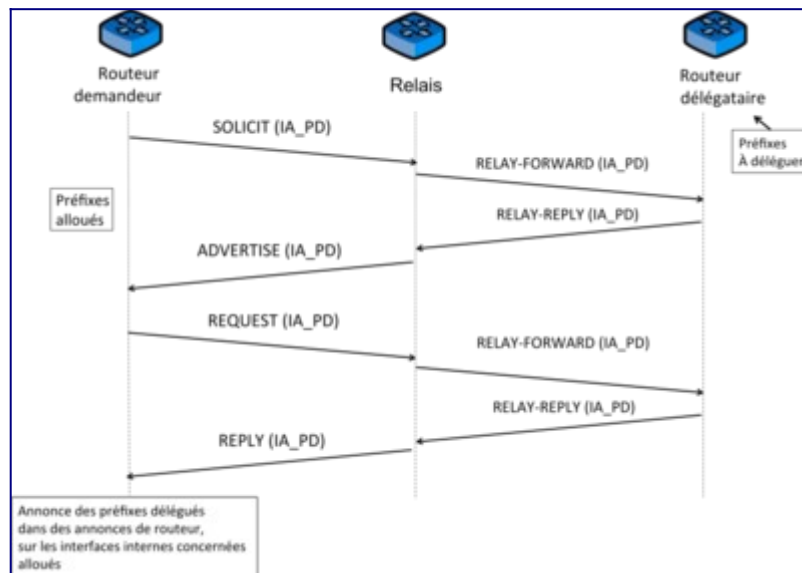


Figure 9 : Allocation de préfixe par un routeur délégataire en présence d'un relais.

Conclusion

DHCPv6 est un protocole de niveau application. Il utilise le protocole de transport UDP et fonctionne en mode client-serveur. Les messages échangés transportent l'identité de l'émetteur (DUID), celle du récepteur, ou les deux, en fonction du sens de transmission du message et de l'avancement de l'échange.

Ce protocole permet qu'un administrateur centralise et gère simplement les paramètres de configuration du réseau, répercute les changements de configuration à l'initiative du serveur DHCPv6 (renumérotation active), ou au contraire, laisse aux clients la possibilité de les prendre en compte lorsqu'ils le souhaitent (renumérotation passive).

Il fonctionne sans relais lorsque le client et le serveur se trouvent sur le même lien. Il fait intervenir des relais lorsque client et serveur sont sur des liens distincts.

Les relais utilisent des messages spécifiques pour communiquer avec les serveurs DHCPv6. Ils encapsulent les messages relayés dans une option de "message relayé". Ainsi, les messages des clients, ceux des serveurs, ou ceux des relais, ne sont jamais modifiés.

Lorsque les relais disposent d'informations locales, des options spécifiques des messages RELAY-FORWARD leur permettent de les communiquer aux serveurs DHCPv6. Les serveurs DHCPv6, en fonction de leur configuration par l'administrateur du réseau, peuvent alors communiquer tout ou partie de ces informations à leurs clients.

Tous les paramètres de configuration du réseau sont transportés dans des options des messages, ce qui fait de DHCPv6 un protocole extensible. Pour étendre le protocole, il suffit d'y ajouter de nouvelles options. Ainsi, initialement, ni la délégation de préfixe ni l'exclusion de préfixe n'existaient. Il a suffi de définir deux options supplémentaires et leur gestion en émission et en réception pour ajouter cette nouvelle fonctionnalité dans DHCPv6.

Références bibliographiques

1. ↑ IANA. Protocol Registries [Dynamic Host Configuration Protocol for IPv6 \(DHCPv6\)](#)

Pour aller plus loin

RFC et leur analyse par S. Bortzmeyer :

- [RFC 5007](#) DHCPv6 Leasequery
- [RFC 6422](#) Relay-Supplied DHCP Options
- [RFC 6603](#) Prefix Exclude Option for DHCPv6-based Prefix Delegation
- [RFC 8415](#) Dynamic Host Configuration Protocol for IPv6 (DHCPv6) [Analyse](#)

Conclusion

Cette séquence a présenté un ensemble de mécanismes associé à IPv6. Ils visent à rendre le service de connectivité d'IPv6 opérant dans différents contextes, à fournir les informations nécessaires à un nœud pour qu'il se configure au sein d'un réseau IPv6, à aider au diagnostic des dysfonctionnements et, enfin, à aider l'utilisateur humain à identifier les nœuds du réseau.

Le protocole ICMPv6 fournit les messages spécifiques à différentes fonctions pour aider au fonctionnement du service de connectivité IPv6. Il fournit aussi la procédure pour superviser le bon fonctionnement du réseau et renvoyer, si besoin, des rapports d'erreur à l'émetteur d'un message problématique. La fonction de découverte des voisins traite principalement de la résolution d'adresse IPv6 en adresse physique pour la transmission des paquets IPv6 sur des supports multi-points. La fonction d'autoconfiguration d'un nœud à un réseau IPv6 se rend "sans état" à l'aide de messages ICMPv6. La version "avec état" repose sur le protocole applicatif DHCPv6. La fonction d'autoconfiguration est de nos jours essentielle. Elle permet à toutes sortes d'équipements de se connecter au réseau. Ces équipements peuvent fonctionner sans nécessité d'intervention humaine pour les configurer. Enfin, le système de nommage (ou DNS) offre une identification textuelle des services de l'Internet et évite à l'utilisateur de manipuler ces objets un peu techniques que sont les adresses IP.

La configuration d'un réseau IPv6 n'est pas très différente de celle d'un réseau IPv4. Les points qui nécessitent une vigilance particulière sont :

- le filtrage d'ICMPv6 pour éviter de perturber les rapports d'anomalies ;
- les annonces de routeurs, garantes du bon fonctionnement du réseau local, ne doivent pas être perturbées ou véhiculer de mauvaises informations ;
- le choix entre autoconfiguration "avec" ou "sans état" doit être réfléchi ;
- l'autoconfiguration "sans état" des hôtes peut demander une mise à jour dynamique du DNS si la politique d'administration demande un enregistrement systématique de tous les postes dans le système de nommage.

L'introduction d'IPv6 pose des problèmes avec l'Internet existant en IPv4. La séquence suivante va maintenant s'intéresser à ces problèmes et aux solutions pour y faire face.

ANNEXE 1 Activité 31 : Contrôler le fonctionnement du réseau par ICMPv6

Introduction

Le contenu de cette annexe complète l'activité 31 en

- donnant un descriptif des messages courants d'erreur signalées par ICMPv6 ;
- introduisant MLD le protocole de gestion des inscriptions aux groupes de *multicast* et les messages ICMPv6 associés ;
- introduisant des fonctions expérimentales ;
- décrivant les informations véhiculées par les messages ICMPv6 sous forme d'options.

Messages ICMPv6 de rapport d'erreur

Chaque cas d'erreur est défini par un message ICMP. Nous allons voir les cas d'erreurs rapportées par ICMPv6.

Destination inaccessible

Un message ICMP *Destination Unreachable* est généré dès qu'un datagramme ne peut être traité. Le format de ce message est présenté par la figure 3.

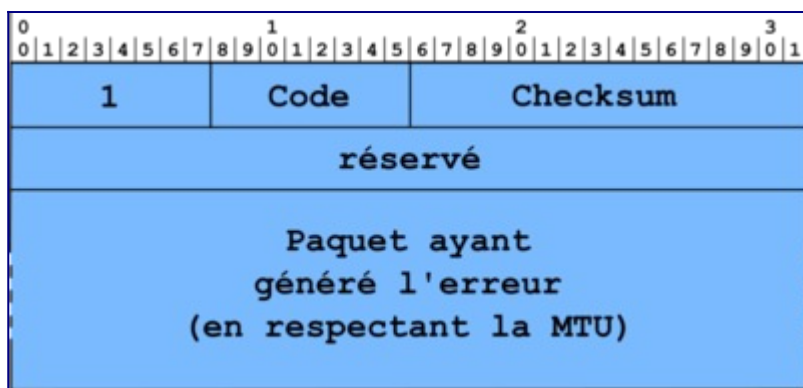


Figure 3 : Format du message de Destination inaccessible.

Ce message est émis par un routeur quand le paquet ne peut pas être transmis pour l'une des raisons suivantes :

- Le routeur ne trouve pas dans ses tables la route vers la destination (code = 0).
- Le franchissement d'un pare-feu (*Firewall*) est interdit ("raison administrative", code = 1).
- L'adresse destination ne peut être atteinte avec l'adresse source fournie ; par exemple, si le message est adressé à un destinataire hors du lien, l'adresse source ne doit pas être une adresse "lien-local" (code = 2).
- Toute autre raison comme, par exemple, la tentative de routage d'une adresse locale au lien (code = 3).
- Le destinataire peut aussi émettre un message ICMPv6 de ce type quand le port destination contenu dans le paquet n'est pas affecté à une application (code = 4).

- Le paquet a été rejeté à cause de son adresse source (code = 5).
- La route vers la destination conduit à un rejet du paquet (code = 6).

Le champ de données contient tout ou partie du datagramme IP qui a occasionné ce message d'erreur.

Paquet trop grand

Si un routeur ne peut pas relayer le datagramme IP car celui-ci a une taille supérieure à la MTU (*Maximum Transmission Unit*) du lien de sortie, il émet le message d'erreur *Packet Too Big*. Les routeurs en IPv6 ne sont plus habilités à effectuer la fragmentation. Le format du message est représenté par la figure 4.

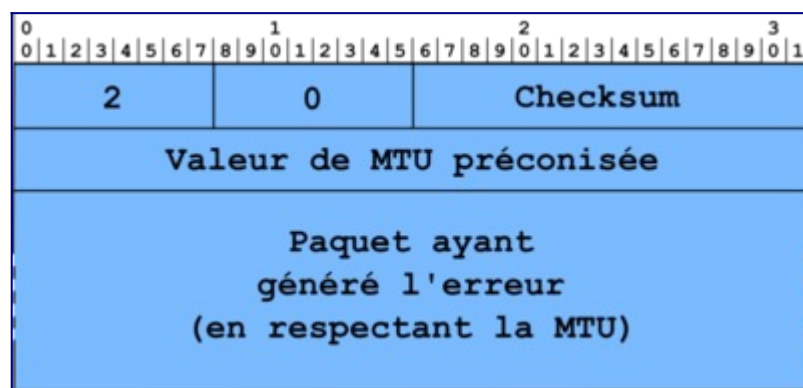


Figure 4 : Format du message ICMPv6 Paquet trop grand.

Ce message ICMPv6 est utilisé par le protocole de découverte de la MTU pour trouver la taille optimale des paquets IPv6 afin qu'ils puissent traverser les routeurs. Ce mécanisme, spécifié par le [RFC 8201](#), est décrit dans la séquence 2. Ce message contient la taille du MTU acceptée par le routeur pour que la source puisse efficacement adapter la taille des données. Ce champ manquait cruellement dans les spécifications initiales de IPv4, ce qui compliquait la découverte de la taille maximale des paquets utilisables sur l'ensemble du chemin. Pour IPv4, le [RFC 1191](#) proposait déjà une modification du comportement des routeurs pour y inclure cette information. A noter que le [RFC 4443](#) indique que ce message n'est pas produit dans le cas d'une communication multicast.

Délai expiré

Quand un routeur relaie un datagramme, il décrémente la valeur du nombre de sauts (*Hop Limit*) de 1. Ce champ, dans l'en-tête IPv6, limite la durée de vie d'un paquet dans le réseau. La durée de vie est exprimée en nombre de routeurs traversés (ou sauts). Si un routeur reçoit un datagramme avec un nombre de sauts de 1, la décrémentation amène la valeur de ce champ à zéro. Le routeur supprime le datagramme et en avertit la source à l'aide du message Délai expiré (*Time Exceeded*) (voir la figure 5). Le signalement de cette erreur peut indiquer une boucle dans le réseau au niveau du routage ou que l'émetteur a positionné une valeur de nombre de sauts trop faible. Le dernier cas peut provenir d'un paquet émis par la commande `traceroute`. Cette commande vise à déterminer le chemin pris par les datagrammes [1].

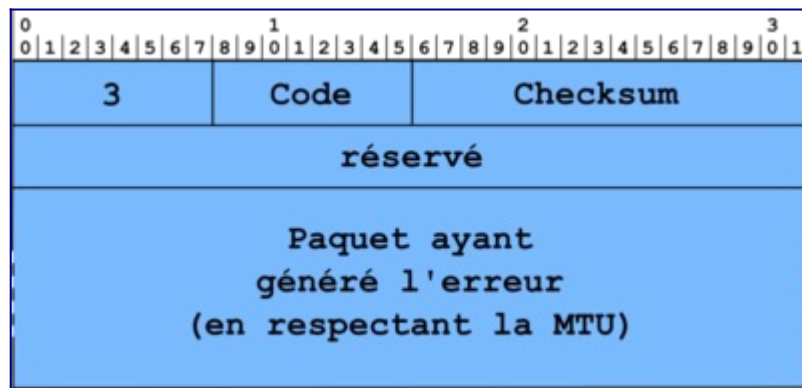


Figure 5 : Format du message de délai expiré.

Le message ICMPv6 Délai expiré indique que le paquet a été rejeté :

- soit par un routeur intermédiaire parce que le champ nombre de sauts a atteint 0 (code = 0) ;
- soit par la destination parce qu'un fragment s'est perdu et le temps alloué au réassemblage a été dépassé (code = 1).

Erreur de paramètre

Le message *Parameter problem* est émis quand un paquet IPv6 ne peut être traité par un noeud du fait d'un problème dans l'en-tête du paquet ou dans ses extensions d'en-tête. Le paquet IPv6 est supprimé mais le noeud envoie à la source le message ICMPv6 dont le format est présenté par la figure 6.

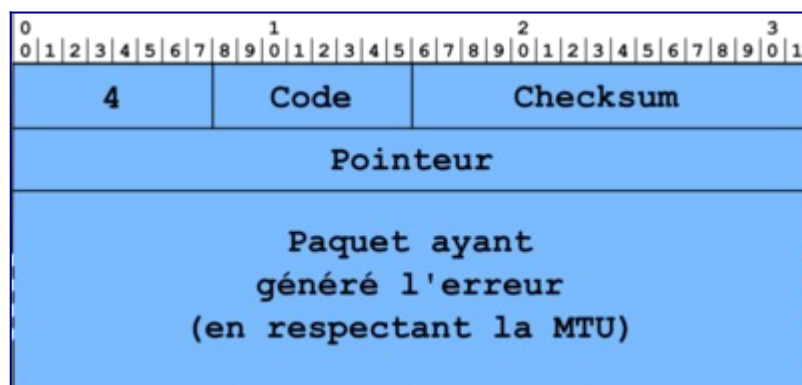


Figure 6 : Format du message Erreur de paramètre.

Le champ type prend la valeur 4. Le champ code révèle la cause de l'erreur :

- la syntaxe de l'en-tête n'est pas correcte (code = 0) ;
- le numéro en-tête suivant n'est pas reconnu (code = 1) ;
- une option de l'extension (par exemple "proche en proche" ou "destination") n'est pas reconnue et le codage des deux bits de poids fort oblige à rejeter le paquet (code = 2).

Le champ pointeur indique l'octet où l'erreur est survenue dans le paquet retourné. Le message contient tout ou partie du paquet IPv6 qui a occasionné le message lui-même.

Attention au filtrage d'ICMPv6

Contrairement à une pratique couramment répandue en IPv4, il ne faut jamais filtrer l'ensemble des messages ICMPv6 en entrée d'un réseau (en particulier le message "Paquet trop grand"). Le filtrage peut introduire des effets néfastes sur le fonctionnement du réseau[2]. Supposons que la MTU entre un client et un serveur soit limitée à 1480 octets à cause d'un tunnel IPv6 dans IPv4 et qu'un serveur filtre les messages ICMPv6 (en particulier "Paquet trop grand"). Le client et le serveur vont échanger des petits paquets pour ouvrir la connexion TCP (SYN, SYN ACK, ACK), puis le client va envoyer une requête HTTP qui est elle-même de petite taille. Le serveur va répondre en envoyant une page complète. Si le paquet a une longueur de 1500 octets, il va être détecté trop grand par le routeur à l'entrée du tunnel. Ce dernier va rejeter le paquet et envoyer au serveur un message ICMPv6 indiquant que le paquet est trop grand. Si le pare-feu du serveur le filtre, le serveur ne connaîtra jamais la taille de la MTU adaptée au chemin qui mène à ce client. Il ne pourra pas adapter la taille de ses paquets. Tous les paquets de 1500 octets seront inexorablement supprimés. Le client devient alors quasiment injoignable et c'est le service de connectivité de la couche "réseau" qui est cassé. Dans cette situation, on se trouve dans une situation où certains paquets passent (ouvertures de connexion, ping, sessions SSH...) et d'autres sont bloqués. Diagnostiquer ce type d'erreur est particulièrement délicat.

Comme ICMPv6 est essentiel au fonctionnement d'IPv6, le [RFC 4890](#) donne les bonnes pratiques pour filtrer correctement les messages ICMPv6. L'idée est de laisser passer les messages ICMPv6 qui sont indispensables au fonctionnement du réseau et de jeter ceux qui introduisent un risque d'insécurité.

Gestion de groupes multicast sur le lien local

Pour offrir un service de distribution multicast, deux composants sont nécessaires : un protocole de gestion de groupes multicast et un protocole de routage multicast[3]. Le protocole de gestion de groupes multicast réalise la signalisation entre l'hôte et son routeur local. Le protocole de routage multicast vise à échanger les informations entre les routeurs afin qu'un arbre de distribution multicast soit construit.

En IPv6, MLD (*Multicast Listener Discovery*) sert, pour un hôte, à indiquer les groupes auxquels il souhaite souscrire. MLD est donc un protocole de gestion de groupes. Ainsi, un routeur de bordure IPv6 va pouvoir découvrir la présence de récepteurs multicast (qualifiés de *listeners*) sur ses liens directement attachés, ainsi que les adresses multicast concernées. MLD est un protocole asymétrique qui spécifie un comportement différent pour les hôtes (les *listeners*) et les routeurs. Toutefois, pour les adresses multicast sur lesquelles un routeur lui-même est récepteur, il doit exécuter les deux parties du protocole. Ceci implique notamment de répondre à ses propres messages de demande. En effet, les routeurs doivent constituer une liste des adresses multicast pour lesquelles il a un ou plusieurs récepteurs sur leur lien local. Aussi, un des récepteurs sur un lien envoie un message de rapport d'abonnement aux groupes auxquels il souhaite recevoir les messages. L'objectif est, par des communications multicast sur le lien, que le routeur local arrive à faire la liste complète des groupes multicast pour lesquels il doit

relayer le trafic localement.

MLD est une fonction d'ICMPv6 ; aussi, les messages MLD sont des messages ICMPv6. Les messages pour MLD sont envoyés avec :

- une adresse source IPv6 lien-local,
- le champ nombre de sauts fixé à 1,
- l'option IPv6 Router Alert activée en ajoutant l'extension d'en-tête *Hop-by-Hop* correspondant.

Cette dernière option est nécessaire afin de contraindre les routeurs à examiner les messages MLD envoyés à des adresses multicast par lesquelles les routeurs ne sont pas intéressés. La version d'origine du protocole MLD [[RFC 2710](#)] (que nous appellerons également MLDv1) présente les mêmes fonctionnalités que le protocole IGMPv2 en IPv4. MLDv2 a été proposé par le [RFC 3810](#) dans lequel, en plus du groupe, le récepteur peut indiquer la source. MLDV2 est une adaptation de IGMPv3 d'IPv4 à IPv6.

Format des messages pour MLD

Le format générique d'un message MLD est donné par la figure 7. Les différents types de messages ICMPv6 pour MLD sont indiqués par le tableau 2. On distingue trois types de messages pour MLD.

1. Le premier type (type = 130) concerne le recensement des récepteurs multicast selon plusieurs méthodes : (i) recensement général émis à l'adresse de diffusion générale sur le lien (FF02::1), (ii) recensement spécifique pour une adresse multicast ; l'adresse de destination est l'adresse multicast du groupe en question. Le message de requête d'abonnement (*Multicast Listener Query*) est émis par le routeur.
2. Le second type de message (type = 131) vise à obtenir un rapport d'abonnement multicast (*Multicast Listener Report*). Ce message est émis par le récepteur multicast. L'adresse de destination est l'adresse multicast du groupe en question. Avec MLDv2, le rapport d'abonnement à un groupe multicast a été complété par la possibilité de limiter la réception au trafic émis par certaines sources. Le trafic des sources non indiquées est alors non reçu. Cette restriction sur la source s'effectue par un message spécifique (type = 143).
3. Enfin, le troisième type de message (type = 132) va servir à un récepteur pour annoncer une résiliation d'abonnement (*Multicast Listener Done*) à un groupe. Ce message est émis à l'adresse du groupe multicast "tous les routeurs du lien local" (FF02::2).

Les champs des messages pour MLD ont la signification suivante :

- Type : prend la valeur 130, 131 ou 132.
- Code : mis à zéro par l'émetteur et ignoré par les récepteurs.
- Checksum : celui du protocole ICMPv6 standard, couvrant tout le message MLD auquel s'ajoutent les champs du pseudo-en-tête IPv6.
- Délai maximal de réponse :
 - utilisé seulement dans les messages de recensement, il exprime le retard maximal

- autorisé (en millisecondes) pour l'arrivée des rapports d'abonnement ;
- dans les messages de rapport ou de résiliation d'abonnement, ce champ est mis à zéro par l'émetteur et ignoré par les récepteurs.
- réservé : champ non utilisé et mis à zéro par l'émetteur et ignoré par les récepteurs.
- Adresse multicast :
 - pour un message de recensement général, ce champ est mis à zéro ;
 - pour un message de recensement spécifique, il contient l'adresse multicast en question ;
 - pour les messages de rapport et de résiliation d'abonnement, le champ contient l'adresse multicast sur laquelle l'hôte souhaite écouter ou cesser d'écouter.

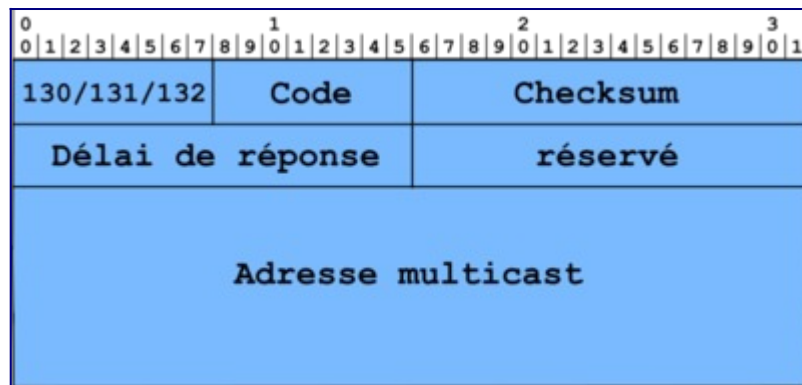


Figure 7 : Format générique d'un message ICMPv6 pour MLD.

Type	Code	Signification
Gestion des groupes multicast		
130		Requête d'abonnement
131		Rapport d'abonnement
132		Fin d'abonnement
143		Rapport d'abonnement MLDv2

Tableau 2 : Messages ICMPv6 pour MLD

Principe de MLD

Le routeur envoie régulièrement des messages de recensement général à l'adresse de multicast FF02::1. Cette adresse équivaut à l'adresse de diffusion sur un lien. Pour éviter que le routeur reçoive plusieurs réponses pour un même groupe, les récepteurs ne répondent pas immédiatement. Pour cela, les récepteurs arment un temporisateur pour chaque adresse multicast qui les concerne. Si le récepteur entend une réponse équivalente à la sienne, il désarme le temporisateur. Sinon, à expiration du temporisateur, le récepteur envoie un rapport d'abonnement à l'adresse multicast du groupe. Avec ce système de temporisateurs, les récepteurs peuvent surveiller les rapports des autres récepteurs sur le lien et ainsi minimiser le trafic MLD.

Les changements d'état des récepteurs sont notifiés par des messages non sollicités. Un message non sollicité est un message émis à l'initiative d'un récepteur d'un groupe multicast ; contrairement au recensement, où c'est le routeur local qui prend l'initiative de l'échange. Les récepteurs peuvent envoyer des messages non sollicités pour les cas suivants :

- Pour souscrire à une adresse multicast spécifique ;
- Pour une résiliation rapide : le récepteur envoie un message de résiliation d'abonnement à l'adresse multicast de "tous les routeurs du lien local" (FF02::2). Le routeur répond avec un message de recensement spécifique à l'adresse en question. S'il n'y a plus de récepteur pour répondre à ce recensement, le routeur efface l'adresse multicast de sa table de routage.

Pour cesser d'écouter sur une adresse multicast, le récepteur peut simplement ne plus répondre aux messages de recensement du routeur. S'il est le seul récepteur de cette adresse multicast sur le lien, après un certain temps l'état du routeur concernant cette adresse expire. Le routeur arrêtera de faire suivre les paquets multicast envoyés à l'adresse en question, s'il s'avère que le récepteur était le dernier concerné par l'adresse multicast sur le lien;

A noter qu'il est possible d'avoir plusieurs routeurs multicast sur le même lien local. Dans ce cas, un mécanisme d'élection est utilisé pour choisir le routeur recenseur. Celui-ci sera le seul responsable pour l'envoi des messages de recensement.

Fonctions autres et expérimentales

Pour être complet, nous pouvons signaler que les messages ICMPv6 servent aussi pour des fonctions expérimentales. Le tableau 4 indique les types de messages associés à ces fonctions. Nous ne détaillerons pas ici ces fonctions, limitées à des usages très spécifiques. Le lecteur curieux est invité à consulter les RFC associés.

Type	Code	Signification
Renumérotation des routeurs (expérimental, RFC 2894)		
138		Renumérotation des routeurs :
	0	Commande
	1	Résultat
	255	Remise à zéro du numéro de séquence
Recherche d'information sur un noeud (expérimental, RFC 4620)		
139		Demande d'information
140		Réponse
Mobilité (RFC 6275)		

144	Découverte d'agent mère (requête)
145	Découverte d'agent mère (réponse)
146	Sollicitation de préfixe mobile
147	Annonce de préfixe mobile
Mobilité (expérimental, RFC 4065)	
150	Protocoles de mobilité expérimentaux, tels que Seamoby

Tableau 4 : Fonctions expérimentales s'appuyant sur ICMPv6

Options véhiculées par les messages ICMPv6

L'intérêt du protocole de découverte des voisins est d'unifier différents protocoles qui existent dans IPv4. En particulier, la plupart des informations à transporter utilise un format commun sous la forme d'options. Le format commun des options simplifie la mise en oeuvre du protocole. Une option se décrit en mot de 64 bits et comporte les champs type, longueur, données.

Les différentes fonctionnalités de découverte des voisins utilisent 5 messages : 2 pour le dialogue entre un équipement et un routeur, 2 pour le dialogue entre voisins et un dernier pour la redirection. Chacun de ces messages peut contenir des options. Le tableau 1 présente l'utilisation des options définies dans le [RFC 4861](#) dans les messages de découverte de voisin.

	Sollicitation du routeur	Annonce du routeur	Sollicitation d'un voisin	Annonce d'un voisin	Indication de redirection
Adresse physique de la source	présent	présent	présent		
Adresse physique de la cible				présent	présent
Information sur le préfixe		≥ 1			
En-tête redirigée					présent
MTU		possible			

Tableau 1: Utilisation des options dans les messages de découverte de voisin.

En plus des cinq options générales décrites dans le tableau 1, il existe d'autres options spécifiques pour la mobilité et les réseaux NBMA (*Non Broadcast Multiple Access*) comme le montre le tableau 2. La liste complète des options pour NDP est gérée par l'IANA et se retrouve

sur une page web[4].

type	description	Message
Basic Neighbor Discovery options [RFC 4861]		
1	Source Link-layer Address (SLLAO)	RS/RA/NS
2	Target Link-layer Address	NA/Redirect
3	Prefix Information (PIO)	RA
4	Redirected Header	Redirect
5	MTU	RA
NBMA (unused) [RFC 2491]		
6	NBMA Shortcut Limit Option	NS
Mobile IP [RFC 3775]		
7	Advertisement Interval Option	RA
8	Home Agent Information Option	RA
9	Source Address List	
10	Target Address List	
SEND [RFC 3971]		
11	CGA option	
12	RSA Signature option	
13	Timestamp option	
14	Nonce option	
15	Trust Anchor option	
16	Certificate option	
Mobility options		
17	IP Address/Prefix Option [RFC 5568]	
18	New Router Prefix Information Option [RFC 4068]	
19	Link-layer Address Option [RFC 5568]	
20	Neighbor Advertisement Acknowledgment Option [RFC 5568]	
23	MAP Option [RFC 4140]	
SLAAC optimization		

24	Route Information Option	[RFC 4191]
25	Recursive DNS Server Option	[RFC 5006] RA
26	RA Flags Extension Option	[RFC 5175]
Fast Mobility options		
27	Handover Key Request Option	[RFC 5269]
28	Handover Key Reply Option	[RFC 5269]
29	Handover Assist Information Option	[RFC 5271]
30	Mobile Node Identifier Option	[RFC 5271]
6LoWPAN [RFC 6775]		
33	Address Registration (ARO)	
34	6LoWPAN Context (6CO)	
35	Authoritative Border Router (ABRO)	
157	Duplicate Address Request (DAR)	
158	Duplicate Address Confirmation (DAC)	
Inverse Neighbor Discovery [RFC 3122]		
138	CARD Request option	[RFC 4065]
139	CARD Reply option	[RFC 4065]

Tableau 2: Identification des options de *Neighbor Discovery*.

Adresse physique de la source/cible

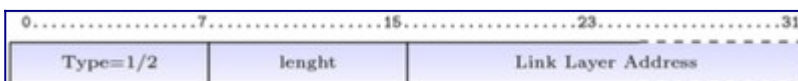


Figure : Format de l'option adresse physique source/cible.

La figure Format de l'option adresse physique source/cible donne le format de ces options. Le type 1 est réservé à l'adresse physique de la source et le type 2 à l'adresse de la cible.

Le champ «longueur» est la taille en mots de 64 bits de l'option. Dans le cas d'une adresse MAC, d'une longueur de 6 octets, il contient donc la valeur 1.

Le [RFC 2464](#) définit le format pour les adresses MAC-48 utilisés dans les réseaux Ethernet et Wi-Fi. Le [RFC 4944](#) définit le format pour les MAC-16 et MAC-64 utilisés dans les réseaux de capteurs reposant sur la norme IEEE 802.15.4.

Information sur le préfixe

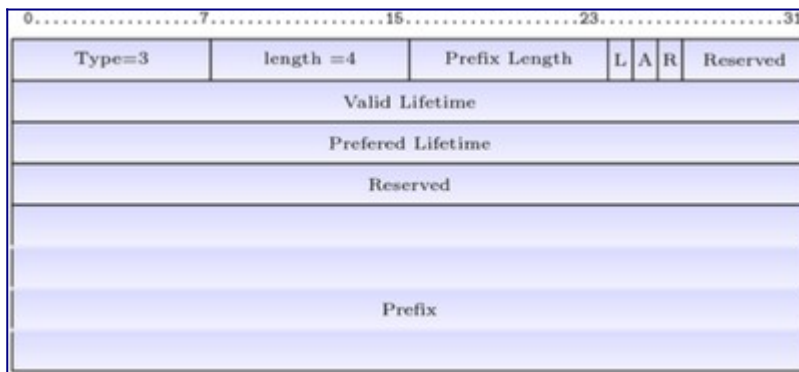


Figure : Format de l'option information sur le préfixe.

Cette option contient les informations sur le préfixe pour permettre une configuration automatique des équipements. Cette option sera présentée en détail dans l'activité d'auto-configuration.

En-tête redirigée

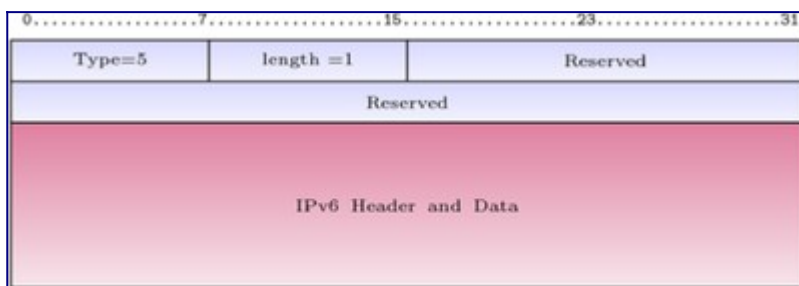


Figure : Format de l'option en-tête redirigée.

Cette option est utilisée par le message d'indication de redirection. Elle permet d'encapsuler les premiers octets du paquet IPv6 qui a provoqué l'émission de ce message comme dans le cas des messages ICMPv6 d'erreur.

Le type vaut 4 et la taille de cette option ne doit pas conduire à un paquet IPv6 dépassant 1280 octets (cf. figure Format de l'option en-tête redirigée). Par contre le paquet doit contenir le maximum d'information possible.

MTU

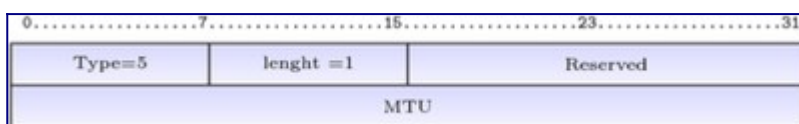


Figure : Format de l'option MTU.

Cette option permet d'informer les équipements sur la taille maximale des données pouvant être émises sur le lien. La figure Format de l'option MTU donne le format de cette option. Il n'est pas nécessaire de diffuser cette information si l'équipement utilise toujours la taille maximale permise. Par exemple, sur les réseaux Ethernet, les équipements utiliseront la valeur 1 500. Par contre pour les réseaux anneau à jeton ou FDDI, il est souvent nécessaire de préciser si les équipements doivent utiliser la valeur maximale permise ou une valeur inférieure pour autoriser

l'utilisation de ponts.

Le champ type vaut 5 et le champ longueur 1.

Référence bibliographique

1. ↑ Wikipédia [Principe de l'utilitaire traceroute](#)
2. ↑ Kline, E. and Townsley, M.; Google IPv6 Center. [ICMPv6 is Non-Optional](#)
3. ↑ Sébastien LOYE. (2005). Techniques de l'ingénieur. ref TE7527. Le multicast IP : principes et protocoles
4. ↑ IANA. [IPv6 Neighbor Discovery Option Formats](#)

ANNEXE 2 Activité 33 : Faire correspondre adresse et nom de domaine

Options DNS des RA

Option de liste de serveurs DNS récurrents (RDNSS)

Cette option d'annonce de routeur contient l'adresse IPv6 d'un ou plusieurs serveurs DNS récurrents (cf. figure 10).

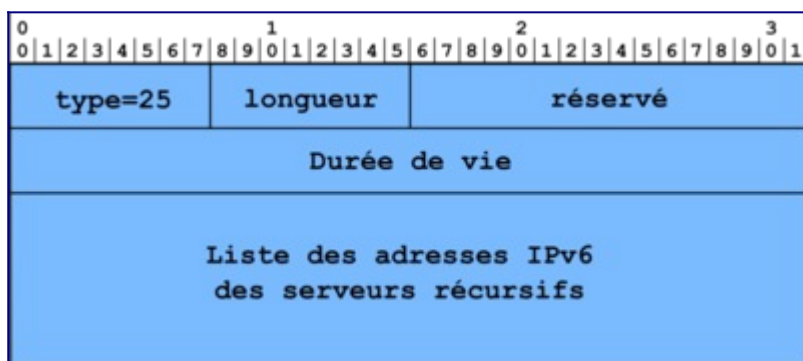


Figure 10 : Format d'une option RDNSS de la [RFC 8106](#).

1. Le champ type a pour valeur 25.
2. Le champ longueur indique la longueur totale de l'option. Les champs type et longueur sont inclus (en multiples de 8 octets). Ce champ permet à l'utilisateur de calculer facilement le nombre d'adresses de serveurs DNS récurrents.
3. Le champ durée de vie indique la durée de vie maximum (en secondes) des adresses associées. Les valeurs de ce champ permettent que la machine sache si elle peut utiliser ces adresses, si leur durée de vie est infinie, si elle doit les rafraîchir ou si elle ne peut plus les utiliser.
4. Le champ adresses contient les adresses IPv6 des serveurs DNS récurrents, codées sur 128 bits.

Option de liste de domaines recherchés (DNSSL)

L'option DNSSL contient un ou plusieurs suffixes de noms de domaines (cf. figure 11). Tous ces suffixes ont la même durée de vie. Certains suffixes peuvent avoir des durées de vies différentes s'ils sont contenus dans des options DNSSL différentes.

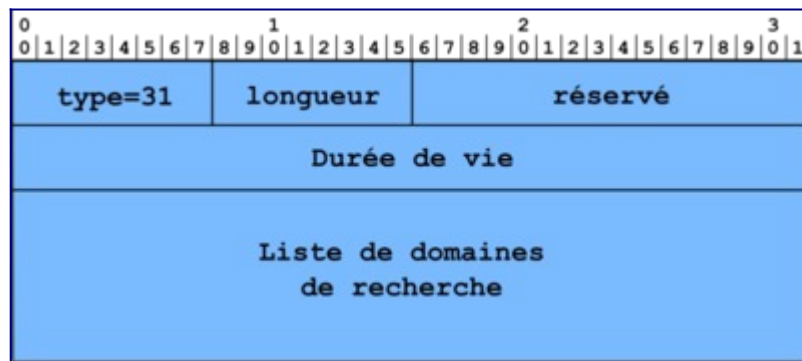


Figure 11 : Format d'une option DNSSEC prévu par la [RFC 8106](#).

1. Le champ type a pour valeur 31.
2. Le champ length indique la longueur totale de l'option, champs type et longueur inclus (en multiples de 8 octets). Le récepteur de cette option utilise ce champ pour calculer le nombre d'adresses de serveurs DNS récursifs.
3. Le champ lifetime indique la durée de vie maximum, en seconde, des suffixes associés. Les valeurs de ce champ permettent que la machine sache si elle peut utiliser ces adresses, si leur durée de vie est infinie, si elle doit les rafraîchir ou si elle ne peut plus les utiliser.
4. Le champ noms de domaines contient la liste des noms de domaines à utiliser pour effectuer les résolutions directes.

Pour simplifier les choses, les noms de domaines ne sont pas compressés. Les bits excédentaires sont mis à 0.

Options DNS du protocole DHCPv6

Option serveur de nom récursif de DHCPv6

L'option de serveur DNS récursif de DHCPv6 fournit, par ordre de préférence, une liste d'adresses IPv6 de serveurs DNS récursifs à une machine IPv6. La structure de l'option est la suivante (cf. figure 12) :

1. Le champ OPTION_DNS_SERVERS vaut le code 23.
2. Le champ longueur représente la longueur de l'option et elle est exprimée en multiple de 16 octets. La valeur du champ indique le nombre d'adresses de serveurs DNS récursifs contenu dans l'option.
3. Le champ DNS-recursive-name-server contient l'adresse IPv6 d'un serveur DNS récursif. Il peut apparaître plusieurs fois.

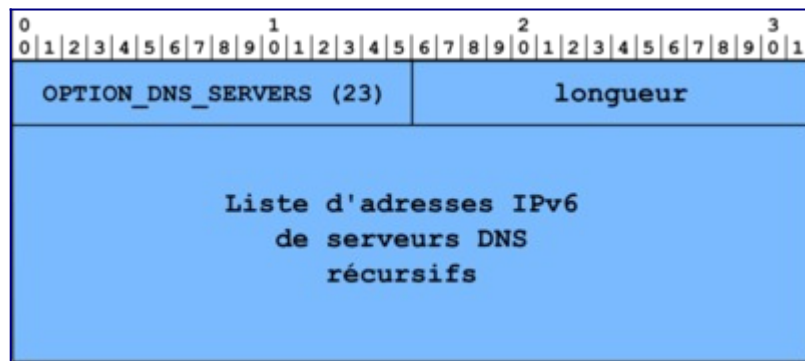


Figure 12 : format de l'option de DHCPv6 spécifiant la liste des serveurs DNS récursifs ([RFC 8415](#)).

Option liste de suffixes de nom de domaine

Le [RFC 8415](#) prévoit également une option spécifiant la liste des suffixes de noms de domaines (cf. figure 13).

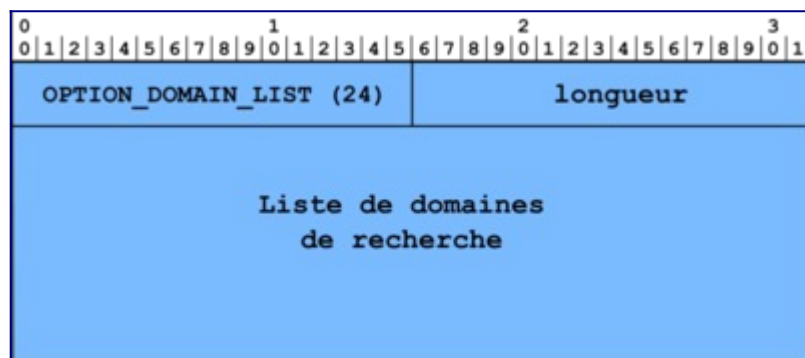


Figure 13 : format de l'option de DHCPv6 spécifiant la liste des suffixes de nom de domaine ([RFC 8415](#)).

1. Le code de l'option OPTION_DOMAIN_LIST vaut 24.
2. Le champ Longueur donne la longueur de l'option en octets.
3. Le champ Searchlist contient la liste de suffixes de noms de domaines.

Les noms de domaines ne sont pas compressés par souci de simplification. Ces deux options ne peuvent apparaître que dans les messages DHCPv6 : SOLICIT, ADVERTISE, REQUEST, RENEW, REBIND, INFORMATION-REQUEST et REPLY.

Mises en œuvre du service DNS

Cette partie présente les principaux logiciels supportant IPv6. Elle renvoie vers une liste plus complète de logiciels. Elle détaille ensuite comment configurer un service de nommage autonome en IPv6. Elle donne également des exemples de fichiers de configuration.

Logiciels DNS supportant IPv6

De nombreux logiciels DNS existent aujourd'hui, mais cette section ne les liste pas de manière exhaustive. Pour avoir une idée plus claire du nombre et de la diversité de ces logiciels, le lecteur peut se référer à la comparaison des logiciels DNS sur Wikipedia. Par ailleurs, certaines distributions logicielles comportent l'implémentation du client et du serveur. D'autres n'incluent que l'implémentation du client ou que celle du serveur. Dans leurs versions récentes, la plupart de ces logiciels DNS supportent complètement IPv6, c'est-à-dire à la fois au niveau de la base de nommage (enregistrements AAAA et PTR) et au niveau du transport IPv6 des messages DNS. Néanmoins, certains ne supportent encore IPv6 qu'au niveau de la base de nommage.

Par exemple, l'ISC : *Internet Systems Consortium* développe la distribution BIND9 (*Berkley Internet Name Domain*). Cette distribution représente la référence de fait dans le domaine. En effet, il s'agit d'une pile logicielle complète : client, serveur et outils. Il intègre toutes les extensions DNS récentes (IPv6, DNSSEC...). Les distributions BIND 9 présentent l'avantage d'être disponibles en code source et en format binaire pour la quasi-totalité des plates-formes (Unix, MS Windows, Apple...). Ainsi, la distribution BIND9 a été choisie comme base pour les exemples de fichiers de configuration.

Notez que les logiciels DNS développés par les NLnetLabs sont aussi des logiciels libres et qu'ils présentent en outre l'avantage d'être dédiés à une seule fonction, à savoir : serveur DNS récursif ou officiel uniquement. Ainsi, de plus en plus d'opérateurs DNS utilisent aujourd'hui le serveur récursif NSD comme serveur DNS officiel (sans récursion) et Unbound comme serveur DNS récursif pour l'une et/ou l'autre de deux raisons : les performances et la diversité générique. Les performances sont reconnues par des tests comparant, d'un côté, NSD et BIND, et de l'autre, Unbound et BIND montrent la supériorité respective des premiers sur les seconds). La diversité générique concerne la diversité des plates-formes logicielles supportant ces serveurs DNS.

Principe de configuration d'un serveur DNS

Cette partie présente le principe de configuration d'un service DNS autonome. Elle précise également les modifications à effectuer pour relier ce service DNS au service de nommage de l'Internet. Pour configurer un service de nommage, il faut successivement installer le paquetage du serveur de nommage sur les machines "serveur", configurer un serveur DNS primaire, configurer au moins un serveur DNS secondaire et préparer le fichier de configuration des clients du service de nommage.

La configuration du serveur DNS primaire comprend la configuration des options de fonctionnement du serveur, la configuration du fichier de résolution directe et la configuration des fichiers de résolution inverse. Deux outils vérifient la configuration du serveur. Le premier, *named-checkconf*, vérifie l'absence d'erreur dans le fichier de configuration du serveur. Le second, *named-checkzone*, vérifie l'absence d'erreur dans les fichiers de zone du serveur. Il utilise le nom de la zone et le fichier de zone correspondant. En cas d'erreur, ces outils signalent et localisent les erreurs. Ils facilitent donc la mise au point du service. Il faut également déclarer, au niveau du serveur DNS primaire, les serveurs DNS secondaires

autorisés à se synchroniser.

La configuration du serveur DNS secondaire comprend la configuration des options de fonctionnement du serveur, la déclaration du statut (secondaire) du serveur, la déclaration du ou des serveurs primaires qui fournissent les fichiers de zone. L'outil *named-checkconf* vérifie les fichiers de configuration du serveurs DNS secondaire. Notez qu'un serveur DNS secondaire peut se synchroniser, soit à partir du serveur DNS primaire, soit à partir d'un serveur DNS secondaire déjà synchronisé.

L'analyse du fichier journal (*/var/log/syslog* par exemple, sur un système Linux) donne des indications précieuses sur les erreurs d'exécution relatives au service de nommage ou leur absence.

La configuration des clients s'effectue au niveau du fichier (*/etc/resolv.conf* pour les systèmes Linux, par exemple). Le fichier *resolv.conf* contient la déclaration du domaine, jusqu'à trois adresses de serveurs DNS, et une liste de noms de domaines recherchés.

Il faut ensuite vérifier le bon fonctionnement des serveurs primaire et secondaires à l'aide d'un client. La vérification se fait à l'aide des outils *dig* ou *host*, utilisables en ligne de commande. Ces outils utilisent, par défaut, les informations contenues dans le fichier *resolv.conf*. Notez que l'outil *nslookup* n'est plus maintenu. Son utilisation est désormais déconseillée. Nous ne présentons donc pas ici son utilisation.

Définition des fichiers de zone

Les fichiers de zone contiennent principalement des enregistrements de ressources (*RR resource record*). Notez que les recherches ignorent la casse des caractères. Cependant, le DNS conserve la casse des caractères. Les commentaires commencent avec un « ; », et se terminent à la fin de la ligne. Les fichiers de zones sont plus faciles à lire s'ils sont documentés. L'ordre des enregistrements n'a aucune importance. Les enregistrements de ressources doivent commencer dans la première colonne d'une ligne.

La première étape de la configuration d'un serveur DNS primaire correspond à la conversion de la table des machines (fichier *hosts*) en son équivalent pour le DNS : fichier de résolution directe (nom-adresse). Un outil écrit en langage Perl, *h2n*, effectue automatiquement cette conversion à partir du fichier */etc/hosts* pour une machine Linux.

La seconde étape correspond à la production des fichiers de résolution inverse. Il y en a un par lien (fichiers de résolution inverse, adresse-nom). Dans le cas d'IPv6, un outil, *ipcalc*, disponible sous la forme d'un paquet Linux, assure la conversion d'une adresse IPv6 en quartets. Un quartet correspond à un chiffre hexadécimal. Il sert pour la résolution inverse des noms en IPv6.

Le serveur DNS primaire a un fichier de résolution inverse pour l'adresse de boucle locale. Chaque serveur, primaire ou secondaire, est maître pour cette zone. En effet, personne n'a reçu la délégation pour le réseau 127/24, ni pour ::1/128. Chaque serveur doit donc en être responsable.

Le fichier de configuration du serveur de nommage, *named.conf*, relie les domaines dont le

serveur a la responsabilité administrative à leur fichier de zone respectif.

Un serveur DNS doit également connaître les adresses des serveurs racines. Il utilise les informations du fichier *db.cache* pour interroger les serveurs et leur demander une liste à jour des correspondances nom-adresse des serveurs racines. Le serveur enregistre cette liste dans un emplacement spécial de sa mémoire cache normale. Il n'est donc plus nécessaire de leur associer une durée de vie. Pour obtenir les adresses des serveurs racine, établissez une session ftp anonyme avec la machine *ftp.rs.internic.net* et rapatriez le fichier *db.cache* du répertoire *domain*. Ce fichier change de temps en temps. Il est donc nécessaire, périodiquement, d'en rapatrier localement une version à jour.

Dans le cas d'un service de nommage autonome, le serveur DNS primaire sert également de serveur racine. Nous utilisons dans ce cas un fichier *db.fakeroot* au lieu du fichier *db.cache*.

Types d'enregistrement de ressource DNS

Les principaux enregistrements de ressources du DNS sont de deux types : ceux relatifs à la zone et ceux relatifs aux machines.

Les enregistrements relatifs à la zone sont : SOA, NS et MX.

- L'enregistrement de ressource SOA (*Start Of Authority*) indique qui est le serveur DNS primaire officiel de la zone. Il n'y en a qu'un par zone. La syntaxe de l'enregistrement SOA est la suivante : SOA, nom du serveur DNS primaire officiel, adresse mail de l'administrateur du service de noms, numéro de série, délai de rafraîchissement, délai avant nouvel essai, délai d'expiration de l'information, durée maximum de conservation d'une réponse négative dans le cache d'un serveur de nommage.
- L'enregistrement de ressource NS (*Name Server*) désigne un serveur DNS officiel pour la zone. Il y a autant d'enregistrements NS que de serveurs DNS officiels pour une zone donnée. Notez que certains serveurs DNS officiels de la zone peuvent ne pas être déclarés dans les fichiers de zone. il s'agit de serveurs DNS furtifs.
- L'enregistrement de ressource MX (*Mail eXchanger*) désigne un agent de transfert ou un serveur de courrier officiel pour un domaine donné.

Les principaux enregistrements relatifs aux machines de la zone sont : A, AAAA, PTR et CNAME.

- L'enregistrement de ressource A définit une correspondance nom-adresse IPv4.
- L'enregistrement de ressource AAAA définit une correspondance nom-adresse IPv6.
- L'enregistrement de ressource PTR définit une correspondance inverse, adresse-nom. Les pointeurs ne désignent que le nom canonique d'une machine.
- L'enregistrement de ressource CNAME définit une correspondance entre le nom canonique d'une ressource (A ou AAAA) et un nom secondaire surnom (alias) d'une machine.

Configuration de serveur DNS

Même si les logiciels DNS utilisés interfonctionnent, la syntaxe et les règles de configuration varient considérablement d'une implémentation à l'autre. Dans ce chapitre, nous fournissons des exemples suivant la syntaxe et les règles de configuration de BIND 9. Ce logiciel est aujourd'hui considéré comme mise en oeuvre de référence en matière de DNS.

Réseau virtualisé utilisé pour générer ces exemples

Les exemples de fichiers qui suivent ont été configurés dans un environnement réseau incluant trois machines supportant respectivement un serveur, un relais et un client DNS (cf. figure 14). La machine serveur **s-13-v6** supporte le serveur DNS primaire. Elle est également un routeur. Elle donne accès à un réseau A sur lequel se trouve le relais. Le réseau A sert pour faire de l'autoconfiguration DHCPv6 "à état" sans relais. Elle donne également accès au réseau C. Le réseau C sert pour l'autoconfiguration des adresses IPv6 (sans serveur DHCPv6). Le relais **r-13-v6** supporte un serveur DNS secondaire. Cette machine est également un routeur. Cette machine donne accès au réseau B. Le réseau B sert pour faire de l'autoconfiguration "à état" en présence d'un relais DHCPv6. Le client **c-13-v6** est doté de deux interfaces de réseau. La première est connectée soit au réseau A, soit au réseau B pour faire du DHCPv6, respectivement, sans et avec relais. La seconde est connectée au réseau C pour faire de l'autoconfiguration "sans état".

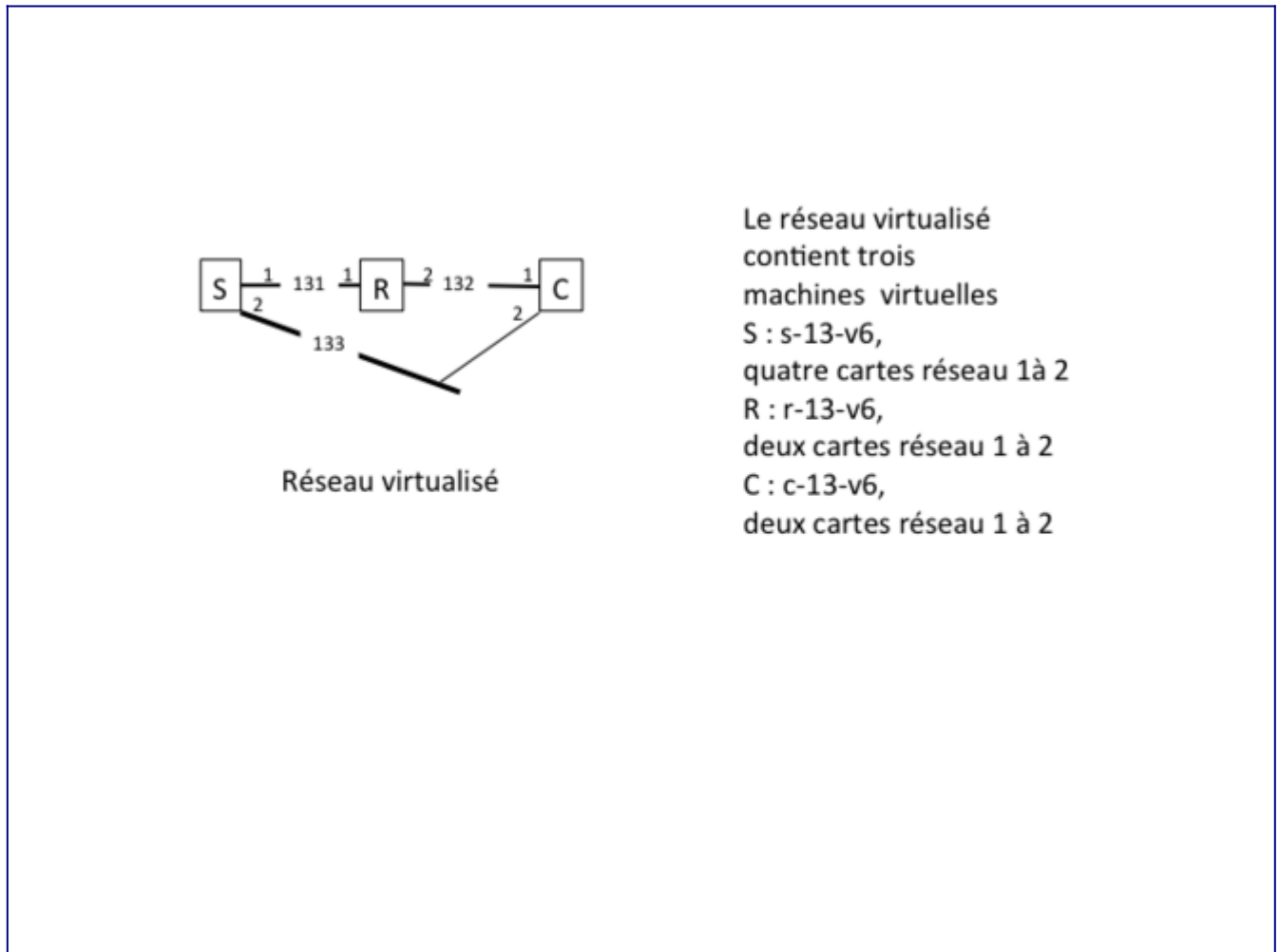


Figure 14 : Réseau virtualisé pour générer ces exemples.

La configuration DNS proposée correspond à un domaine DNS autonome où le serveur DNS primaire fait également fonction de serveur DNS racine.

Fichier de configuration d'un serveur BIND9

La configuration d'un serveur DNS primaire BIND9 concerne quatre aspects : la configuration des options de fonctionnement du serveur, la configuration du fichier de zone pour la résolution directe (nom – adresse), la configuration des fichiers de zone pour la résolution inverse (adresse – nom), et la mise au point du service. Pour tenir compte de cette modularité, le fichier principal de configuration de BIND9 se contente d'inclure d'autres fichiers gérant spécifiquement chacun des aspects précédents. Le fichier de configuration du serveur de nom BIND 9 est, par exemple sous Linux, `/etc/bind9/named.conf`. Ce fichier se contente d'inclure d'autres fichiers. Chacun de ces fichiers contient un ensemble de déclarations relatives à un aspect de la configuration du serveur.

Exemple de contenu du fichier `/etc/bind9/named.conf`

```
// This is the primary configuration file for the BIND DNS server named.
//
// Please read /usr/share/doc/bind9/README.Debian.gz for information on the
```



```
// structure of BIND configuration files in Debian, *BEFORE* you customize
// this configuration file.
//
// If you are just adding zones, please do that in
/etc/bind/named.conf.local
include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
include "/etc/bind/named.conf.default-zones";
```

Configuration du fonctionnement du serveur

Le fichier *named.conf.options* contient, par exemple, différentes options de configuration du fonctionnement du serveur, telles que le répertoire de travail, l'activation de l'écoute des requêtes DNS sur un port (socket) en IPv4 et/ou en IPv6, l'activation ou non du mode récursif, l'affichage ou non du numéro de version du serveur.

Contenu du fichier *named.conf.options*

```
options {
    directory "/var/bind";
    auth-nxdomain no;
    listen-on { any; };
    listen-on-v6 { any; };
    version none;
    allow-query-cache { any; };
    allow-query { any; };
    allow-recursion {
        2001:db8:330f:a0d1::/64;
        2001:db8:330f:a0d2::/64;
        2001:db8:330f:a0d1::/64;
    };
};

include "/etc/bind/rndc-key";
controls {
    inet 127.0.0.1 port 953
    allow {127.0.0.1; ::1; } keys { "rndc-key"; };
};
```

L'option *listen-on* peut avoir plusieurs valeurs possibles. Avec la valeur *any*, le serveur écoute sur toutes les adresses IPv4 opérationnelles. Si une liste d'adresses IPv4 est spécifiée, le serveur écoutera uniquement les requêtes et réponses reçues sur chacune des interfaces configurées avec une de ces adresses. Si la valeur *none* est spécifiée, cela signifie que le serveur ne supporte pas IPv4.

Par défaut, le serveur DNS BIND 9 n'écoute pas les requêtes qui arrivent sur une interface IPv6. Pour changer ce comportement par défaut, il faut utiliser l'option *listen-on-v6*. Si elle vaut *any* le serveur écoute sur toutes les adresses IPv6 opérationnelles. Si une liste d'adresses IPv6 est spécifiée, le serveur écoutera uniquement les requêtes et réponses reçues sur chacune des interfaces configurées avec une de ces adresses. La valeur par défaut est *none*, ce qui signifie que le serveur ne supporte pas IPv6 (valeur par défaut).

Exemple de configuration locale du serveur de noms BIND9

Le fichier *named.conf.local* contient les chemins d'accès aux zones pour lesquelles le serveur DNS est maître officiel (master). Il définit également le chemin d'accès aux données (option *directory*) et le rôle du serveur DNS pour chacune des zones (primaire ou secondaire). Les zones DNS pour lesquelles le serveur DNS (primaire ou secondaire) est officiel sont ensuite déclarées successivement grâce à des rubriques de type "zone". Pour chaque zone, le nom du fichier contenant les enregistrements de chaque zone est précisé. Lorsque le serveur est secondaire pour une zone donnée, l'administrateur du réseau indique (à l'aide de la sous-rubrique *slave*) la liste des adresses IPv4 et/ou IPv6 des serveurs DNS, primaire ou secondaires, à partir desquels ce secondaire peut se synchroniser.

Voici maintenant un extrait du fichier *named.conf.local* de notre serveur DNS autonome.

Exemple de contenu du fichier *named.conf.local*

```
//
// Do any local configuration here
//
// Consider adding the 1918 zones here, if they are not used in your
// organization
//
include "/etc/bind/zones.rfc1918";
//zones primaires
//
//
// Déclaration de la zone tpt.example.com
//
//
zone "tpt.example.com" {
    type master;
    file "/etc/bind/db.tpt.example.com";
    allow-transfer {
        2001:db8:330f:a0d1::197;
        2001:db8:330f:a0d2::197;
    };
};

//
// Déclaration des zones inverses
//
//
// 2001:db8:330f:a0d1::/64
//
zone "1.d.0.a.f.0.3.3.8.b.d.0.1.0.0.2.ip6.arpa." {
    type master;
    file "/etc/bind/db.131.tpt.example.com.rev";
    allow-transfer {
        2001:db8:330f:a0d1::197;
        2001:db8:330f:a0d2::197;
    };
};

//
// 2001:db8:330f:a0d2::/64
//
zone "2.d.0.a.f.0.3.3.8.b.d.0.1.0.0.2.ip6.arpa." {
    type master;
```

```
file "/etc/bind/db.132.tpt.example.com.rev";
allow-transfer {
    2001:db8:330f:a0d1::197;
    2001:db8:330f:a0d2::197;
};

//
// 2001:db8:330f:a0d3::/64
//
zone "3.d.0.a.f.0.3.3.8.b.d.0.1.0.0.2.ip6.arpa." {
    type master;
    file "/etc/bind/db.132.tpt.example.com.rev";
    allow-transfer {
        2001:db8:330f:a0d1::197;
        2001:db8:330f:a0d2::197;
    };
};

//
// Zones secondaires
//
```

Contenu du fichier `named.conf.default-zones`

```
// prime the server with knowledge of the root servers
zone "." {
    type hint;
    file "/etc/bind/db.fakeroot";
};

// be authoritative for the localhost forward and reverse zones, and for
// broadcast zones as per RFC 1912

zone "localhost" {
    type master;
    file "/etc/bind/db.local";
};

zone "127.in-addr.arpa" {
    type master;
    file "/etc/bind/db.127";
};

zone "0.in-addr.arpa" {
    type master;
    file "/etc/bind/db.0";
};

zone "255.in-addr.arpa" {
    type master;
    file "/etc/bind/db.255";
};
```

Fichier de zone DNS pour la résolution directe (nom - adresse)

Voici, à titre d'exemple, un extrait du fichier de résolution directe pour la zone `tpt.example.com`. Il ne fait apparaître que les adresses IPv6. Notez, dans cet exemple, que les adresses IPv6 ont été construites manuellement pour garantir leur pérennité dans le DNS. En effet, rappelons dans ce contexte que les adresses obtenues par auto-configuration dérivent généralement de

l'adresse physique de la carte réseau utilisée ([RFC 4291](#)). Notez également que pour que ces adresses soient automatiquement prises en compte dans le DNS, il faudrait configurer et autoriser la mise à jour dynamique du service de nommage depuis ces machines.

```
$TTL 3h
tpt.example.com.      IN      SOA      s-13-v6.tpt.example.com.    r-
13-v6.tpt.example.com. (
    3          ; numéro de série
    3600       ; refresh (1 heure)
    900        ; nouvel essai (15 minutes)
    3600000    ; expiration (5 semaines jours 16 heures)
    1h)        ; durée de vie minimum (1 heure)
@                IN      NS       s-13-v6.tpt.example.com.
@                IN      NS       r-13-v6.tpt.example.com.

s-13-v6.tpt.example.com.      IN      AAAA     2001:db8:330f:a0d1::217
                                AAAA     2001:db8:330f:a0d1::53
                                AAAA     2001:db8:330f:a0d2::217
                                AAAA     2001:db8:330f:a0d3::217
r-13-v6.tpt.example.com.      IN      AAAA     2001:db8:330f:a0d1::197
                                AAAA     2001:db8:330f:a0d2::197
c-13-v6.tpt.example.com.      IN      AAAA     2001:db8:330f:a0d1::187
                                AAAA     2001:db8:330f:a0d2::187
s13.tpt.example.com.          IN      CNAME    s-13-v6.tpt.example.com.
r13.tpt.example.com.          IN      CNAME    r-13-v6.tpt.example.com.
c13.tpt.example.com.          IN      CNAME    c-13-v6.tpt.example.com.
```

Fichier de zone DNS inverse en IPv6

Voici les fichiers de zone pour la résolution DNS inverse correspondant au préfixe IPv6 d'un lien.

Fichier db.131.tpt.example.com.rev

```
$TTL 3h
;
@                IN      SOA      s-13-v6.tpt.example.com.    root.s- 13-
v6.tpt.example.com. (
    2          ; Numéro de série
    3600       ; rafraîchissement (1 heure)
    900        ; Nouvelle tentative (15 minutes)
    3600000    ; Durée de vie maximale (5 semaines 6 jours
et 16 heures)
    1h)        ; Durée de vie minimale (1 heure)
;
@                IN      NS       s-13-v6.tpt.example.com.
@                IN      NS       r-13-v6.tpt.example.com.
$ORIGIN 1.d.0.a.f.0.3.3.8.b.d.0.1.0.0.2.ip6.arpa.
3.5.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0 IN      PTR      s-13-v6.tpt.example.com.
7.1.2.0.0.0.0.0.0.0.0.0.0.0.0.0 IN      PTR      s-13-v6.tpt.example.com.
7.9.1.0.0.0.0.0.0.0.0.0.0.0.0.0 IN      PTR      r-13-v6.tpt.example.com.
```

Fichier db.132.tpt.example.com.rev

```
$TTL 3h
;
```

```

@                IN          SOA          s-13-v6.tpt.example.com. root.s-13-
v6.tpt.example.com. (
                    2          ; Numéro de série
                    3600       ; rafraîchissement (1 heure)
                    900        ; Nouvelle tentative (15 minutes)
                    3600000    ; Durée de vie maximale (5 semaines 6 jours
;                                     et 16 heures)
                    1h )      ; Durée de vie minimale (1 heure)
;
@                IN          NS          s-13-v6.tpt.example.com.
@                IN          NS          r-13-v6.tpt.example.com.
$ORIGIN 2.d.0.a.f.0.3.3.8.b.d.0.1.0.0.2.ip6.arpa.
7.9.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0 IN PTR r-13-v6.tpt.example.com.

```

Fichier db.133.tpt.example.com.rev

```

$TTL 3h
;
@                IN          SOA          s-13-v6.tpt.example.com. nobody.localhost.
(
                    4          ; Numéro de série
                    3600       ; rafraîchissement (1 heure)
                    900        ; Nouvelle tentative (15 minutes)
                    3600000    ; Durée de vie maximale (5 semaines 6 jours
et 16 heures)
                    1h )      ; Durée de vie minimale (1 heure)
;
@                IN          NS          s-13-v6.tpt.example.com.
@                IN          NS          r-13-v6.tpt.example.com.
$ORIGIN 3.d.0.a.f.0.3.3.8.b.d.0.1.0.0.2.ip6.arpa.
7.1.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0 IN PTR s-13-v6.tpt.example.com.

```

Clients du service de nommage

Un client DNS, un résolveur, se présente souvent sous la forme d'une bibliothèque de nommage. Cette dernière se nomme *libresolv*. Ce client est appelé *resolver*. Nous utilisons le terme résolveur. Rappelons que toutes les applications TCP/IP s'exécutant sur une machine donnée sollicitent ce résolveur. Ce dernier les renseigne sur les ressources DNS nécessaires à l'établissement de leur communication avec des applications distantes.

Exemple de fichier de configuration */etc/resolv.conf* d'un serveur de noms

```

domain tpt.example.com
nameserver ::1
nameserver 2001:db8:330f:a0d1::53
nameserver 2001:db8:330f:a0d1::217
search tpt.example.com

```

Exemple de fichier de configuration */etc/resolv.conf* d'une machine

```

domain tpt.example.com
nameserver 2001:db8:330f:a0d1::197
nameserver 2001:db8:330f:a0d1::53
nameserver 2001:db8:330f:a0d1::217
search tpt.example.com

```

Outils de vérification de la configuration DNS

Outre le résolveur, des outils et commandes dépendent des systèmes d'exploitation existants. Ces outils permettent d'interroger un serveur DNS pour le mettre au point et/ou le dépanner. Les outils *dig* et *host*, par exemple, font partie des distributions BIND9. Nous présentons des exemples de leur utilisation dans la suite de cette partie.

Notez que, lorsque le serveur interrogé n'est pas explicitement renseigné lors de l'invocation de ces commandes, les serveurs par défaut référencés dans le fichier *resolv.conf* sont interrogés. Il peut, par exemple, s'agir de la liste des serveurs récursifs configurée automatiquement (via DHCP, par exemple) ou de celle configurée manuellement dans un fichier de configuration (*/etc/resolv.conf* pour les systèmes Unix ou Linux) ou via une interface graphique de l'équipement (MS Windows et Mac OS). Les mécanismes de découverte de la liste des serveurs DNS récursifs sont décrits plus loin. Voir le chapitre **Découverte de la liste de serveurs DNS récursifs**.

Exemples d'interrogation d'un serveur DNS avec dig : résolution directe

```

root@s-13-v6:/etc/bind# dig @2001:db8:330f:a0d1::53 s-13-v6.tpt.example.com
-t aaaa

; <<>> DiG 9.8.4-rpz2+r1005.12-P1 <<>> @2001:db8:330f:a0d1::53 s-13-
v6.tpt.example.com -t aaaa
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 10043
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 5, AUTHORITY: 2, ADDITIONAL: 2

;; QUESTION SECTION:
;s-13-v6.tpt.example.com.                IN      AAAA

;; ANSWER SECTION:
s-13-v6.tpt.example.com.                10800   IN      AAAA
2001:db8:330f:a0d1::53
s-13-v6.tpt.example.com.                10800   IN      AAAA
2001:db8:330f:a0d1::217
s-13-v6.tpt.example.com.                10800   IN      AAAA
2001:db8:330f:a0d2::217
s-13-v6.tpt.example.com.                10800   IN      AAAA
2001:db8:330f:a0d3::217
s-13-v6.tpt.example.com.                10800   IN      AAAA
2001:db8:330f:a0d4::217

;; AUTHORITY SECTION:
tpt.example.com.                        10800   IN      NS      r-13-
v6.tpt.example.com.
tpt.example.com.                        10800   IN      NS      s-13-
v6.tpt.example.com.

;; ADDITIONAL SECTION:
r-13-v6.tpt.example.com.                10800   IN      AAAA
2001:db8:330f:a0d2::197
r-13-v6.tpt.example.com.                10800   IN      AAAA

```

```
2001:db8:330f:a0d1::197

;; Query time: 0 msec
;; SERVER: 2001:db8:330f:a0d1::53#53(2001:db8:330f:a0d1::53)
;; WHEN: Wed Feb 25 00:55:58 2015
;; MSG SIZE rcvd: 270
```

Exemple d'interrogation d'un serveur DNS avec la commande host : résolution directe

```
root@s-13-v6:/etc/bind# host -t aaaa s-13-v6.tp13.tptfctp.
s-13-v6.tp13.tptfctp has IPv6 address 2001:db8:330f:a0d1::217
s-13-v6.tp13.tptfctp has IPv6 address 2001:db8:330f:a0d2::217
s-13-v6.tp13.tptfctp has IPv6 address 2001:db8:330f:a0d3::217
s-13-v6.tp13.tptfctp has IPv6 address 2001:db8:330f:a0d4::217
s-13-v6.tp13.tptfctp has IPv6 address 2001:db8:330f:a0d1::53
```

Exemple d'interrogation d'un serveur DNS avec la commande dig : résolution inverse

```
root@s-13-v6:/etc/bind# dig @::1 -x 2001:db8:330f:a0d1::217

; <<>> DiG 9.8.4-rpz2+rl005.12-P1 <<>> @::1 -x2001:db8:330f:a0d1::217
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 65205
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 7

;; QUESTION SECTION:
;7.1.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.d.0.a.f.0.3.3.8.b.d.0.1.0.0.2.ip6.arpa.
IN PTR

;; ANSWER SECTION:
7.1.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.d.0.a.f.0.3.3.8.b.d.0.1.0.0.2.ip6.arpa.
10800IN PTR s-13-v6.tp13.tptfctp.

;; AUTHORITY SECTION:
1.d.0.a.f.0.3.3.8.b.d.0.1.0.0.2.ip6.arpa. 10800 IN NS r-13-v6.tp13.tptfctp.
1.d.0.a.f.0.3.3.8.b.d.0.1.0.0.2.ip6.arpa. 10800 IN NS s-13-v6.tp13.tptfctp.

;; ADDITIONAL SECTION:
r-13-v6.tp13.tptfctp. 10800 IN AAAA 2001:db8:330f:a0d2::197
r-13-v6.tp13.tptfctp. 10800 IN AAAA 2001:db8:330f:a0d1::197
s-13-v6.tp13.tptfctp. 10800 IN AAAA 2001:db8:330f:a0d2::217
s-13-v6.tp13.tptfctp. 10800 IN AAAA 2001:db8:330f:a0d3::217
s-13-v6.tp13.tptfctp. 10800 IN AAAA 2001:db8:330f:a0d4::217
s-13-v6.tp13.tptfctp. 10800 IN AAAA 2001:db8:330f:a0d1::53
s-13-v6.tp13.tptfctp. 10800 IN AAAA 2001:db8:330f:a0d1::217

;; Query time: 0 msec
;; SERVER: ::1#53(::1)
;; WHEN: Tue Mar 17 11:31:56 2015
;; MSG SIZE rcvd: 356
```

Exemple d'interrogation d'un serveur DNS avec la commande host : résolution inverse

```
root@r-13-v6:/var/bind# host -t aaaa s-13-v6
s-13-v6.tp13.tptfctp has IPv6 address 2001:660:330f:a0d1::53
```

```
s-13-v6.tp13.tptfctp has IPv6 address 2001:660:330f:a0d1::217
s-13-v6.tp13.tptfctp has IPv6 address 2001:660:330f:a0d2::217
s-13-v6.tp13.tptfctp has IPv6 address 2001:660:330f:a0d3::217
s-13-v6.tp13.tptfctp has IPv6 address 2001:660:330f:a0d4::217
root@r-13-v6:/var/bind# host -t aaaa 2001:660:330f:a0d1::53
3.5.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.d.0.a.f.0.3.3.0.6.6.0.1.0.0.2.ip6.arpa
domain name pointer
s-13-v6.tp13.tptfctp.
root@r-13-v6:/var/bind# host -t aaaa 2001:660:330f:a0d1::197
7.9.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.d.0.a.f.0.3.3.0.6.6.0.1.0.0.2.ip6.arpa
domain name pointer
r-13-v6.tp13.tptfctp.
root@r-13-v6:/var/bind# host -t aaaa 2001:660:330f:a0d2::197
7.9.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.d.0.a.f.0.3.3.0.6.6.0.1.0.0.2.ip6.arpa
domain name pointer
r-13-v6.tp13.tptfctp.
root@r-13-v6:/var/bind# host -t aaaa 2001:660:330f:a0d3::217
7.1.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.3.d.0.a.f.0.3.3.0.6.6.0.1.0.0.2.ip6.arpa
domain name pointer
s-13-v6.tp13.tptfctp.
```

Recommandations opérationnelles pour l'intégration d'IPv6

Le DNS, comme cela a été décrit dans l'introduction de ce chapitre, est à la fois une application TCP/IP et une infrastructure critique. C'est l'application TCP/IP client-serveur qui gère la base de données distribuée à la plus grande échelle qui soit. C'est une application critique parce qu'elle permet à toutes les autres applications TCP/IP classiques (web, mail, ftp...) de fonctionner.

L'intégration progressive d'IPv6 entraîne de nouveaux problèmes opérationnels liés au DNS. Ces problèmes sont dus à la fragmentation de l'espace de nommage. Il convient donc soit de les éviter, soit de trouver les solutions adéquates pour y remédier. À cet effet, les [RFC 3901](#) et [RFC 4472](#) identifient les principaux problèmes et formulent une série de recommandations pratiques pour y faire face. Le chapitre qui suit, **Deux impossibilités d'accéder au service de nommage et remèdes**, résume ces recommandations. Dans un article en ligne, l'auteur revient sur des cas problématiques du déploiement du DNS en IPv6 [\[1\]](#).

Le DNS supporte les enregistrements A et AAAA, et ce, indépendamment de la version d'IP utilisée pour transporter les requêtes et réponses DNS relatives à ces enregistrements. Par ailleurs, en tant qu'application TCP/IP, un serveur DNS utilise les transports UDP sur IPv4 ou IPv6 ou sur les deux à la fois (machine en double pile). Dans tous les cas, le serveur DNS doit satisfaire une requête donnée en renvoyant les informations qu'il a dans sa base de données, indépendamment de la version d'IP qui lui a acheminé cette requête.

Un serveur DNS ne peut pas, *a priori*, savoir si le résolveur initiateur de la requête l'a transmis à son serveur récursif (cache) en utilisant IPv4 ou IPv6. Des serveurs DNS intermédiaires (*cache forwarder*) peuvent, en effet, intervenir dans la chaîne des serveurs interrogés durant le processus de résolution d'une requête DNS. Ces serveurs DNS intermédiaires (*cache forwarder*) n'utilisent pas nécessairement la même version d'IP que leurs clients. Notez en outre, qu'en supposant que le serveur DNS puisse connaître la version d'IP utilisée par le client

qui a initié la requête, il n'a pas à faire d'hypothèse sur l'usage par le client de la réponse DNS renvoyée.

Deux impossibilités d'accéder au service de nommage et leurs remèdes

Cette partie présente deux scénarios où l'accès au DNS est impossible et les remèdes qui permettent d'éviter ces situations. Avant IPv6, le processus de résolution DNS ne faisait intervenir qu'IPv4. Le service était donc garanti pour tous les clients DNS. Avec IPv6, on risque de se trouver confronté à des cas où l'espace de nommage est fragmenté. Dans ce cas, certains fragments de cet espace ne sont accessibles que via IPv4, et d'autres ne sont accessibles que via IPv6. Voici, par exemple, deux scénarios illustrant ce problème de fragmentation de l'espace d'adressage ainsi que la solution recommandée par l'IETF dans chaque scénario : client IPv4 et serveur IPv6, client IPv6 et serveur IPv4.

Premier scénario : client IPv4 et serveur IPv6

Un client ne supportant qu'IPv4 envoie une requête relative à une zone hébergée sur des serveurs DNS ne supportant qu'IPv6. Dans ce cas, le processus de résolution échoue du fait de l'impossibilité d'accéder aux serveurs DNS officiels de cette zone. La recommandation est de faire en sorte que toute zone soit servie par au moins un serveur DNS officiel qui supporte IPv4. Ceci remédie à ce problème.

Second scénario : client IPv6 et serveur IPv4

Un client ne supportant qu'IPv6 envoie une requête relative à une zone hébergée sur des serveurs DNS ne supportant qu'IPv4. Si le serveur récursif interrogé ne supporte pas non plus IPv4, le processus de résolution risque d'échouer du fait de l'impossibilité pour ce serveur DNS récursif de joindre, pour la zone concernée, des serveurs DNS officiels supportant IPv6. La recommandation est de configurer le serveur récursif en le faisant pointer vers un relais DNS fonctionnant en double pile IPv4/IPv6. Ceci remédie à ce problème.

Par exemple, pour une distribution BIND, il suffit d'ajouter l'option : `forwarders { ;}` dans le fichier `named.conf.options`.

Taille limitée des messages DNS en UDP, extension EDNS.0

Les implémentations DNS s'appuient essentiellement sur deux standards de l'IETF : [RFC 1034](#) et [RFC 1035](#). De nombreux autres RFC complémentaires ont été publiés plus tard pour clarifier certains aspects pratiques ou pour apporter de nouvelles extensions répondant à de nouveaux besoins (enregistrements AAAA, SRV, extensions DNSSEC...).

Le DNS, en tant qu'application TCP/IP, doit supporter les deux modes de transport UDP et TCP ([RFC 1035](#)). Le port associé à l'application DNS est le même pour TCP et pour UDP : 53. Le protocole de transport UDP est généralement utilisé pour acheminer les requêtes/réponses DNS. Le protocole de transport TCP est généralement utilisé pour les transferts de zones entre serveur DNS primaire et secondaires.

Lorsque le DNS utilise le protocole de transport UDP, la taille des messages DNS est limitée à 512 octets. Certaines requêtes, trop grandes pour être acheminées par UDP, induisent un acheminement par TCP. Dans ce cas, le client reçoit, dans un premier temps, un message dont la section réponse (*answer section*) est vide et dont le bit TC (*TrunCated*) vaut 1. Ceci signifie implicitement que le client est invité à réinterroger le serveur en utilisant TCP. Notez que ce scénario justifie le fait que le port 53 en TCP ne doit pas être ouvert exclusivement pour des transferts de zones. Notez, par ailleurs, qu'un recours trop fréquent à TCP risque de consommer davantage de ressources, et par conséquent, de dégrader les performances du serveur DNS.

Certains nouveaux types d'enregistrements (AAAA) risquent d'augmenter significativement la taille des réponses DNS. Ceci risque donc d'accroître le nombre de recours à TCP pour satisfaire les requêtes/réponses DNS. Aujourd'hui, ces dépassements sont rares. La plupart des réponses DNS ont une taille qui ne dépasse guère 400 octets. En effet, les sections *answer*, *authority* et *additional*, qui constituent l'essentiel de la réponse DNS, ne contiennent qu'un nombre limité d'enregistrements lorsque cette réponse ne concerne pas directement une zone racine telle que *.com*, *.net*, *.fr*, *.de*.

Face à ce risque, l'IETF a proposé l'extension EDNS.0 du protocole DNS ([RFC 6891](#)). Elle permet qu'un client DNS informe le serveur interrogé qu'il supporte des réponses de taille supérieure à la limite des 512 octets (par exemple, 4096 octets). Ainsi, le support de l'extension du DNS, 'EDNS.0', est fortement recommandé en présence d'IPv6. Cette extension est déjà déployée dans les versions récentes des logiciels DNS. Notez également que le support d'EDNS.0 est aussi indispensable en présence des extensions de sécurité de DNS, DNSSEC.

Le faible taux de pénétration d'EDNS.0 dans les logiciels DNS, surtout les clients, est resté pendant plusieurs années un des principaux motifs du refus de l'IANA/ICANN de publier de nouvelles adresses (IPv4 ou IPv6) pour des serveurs "racine". Depuis le 4 février 2008, l'IANA publie l'adresse IPv6 (enregistrement AAAA) des serveurs "racine" supportant le transport IPv6 dans la zone "racine". La nouvelle version du fichier de démarrage (*db.cache*) de BIND 9 contient également ces adresses. Notez enfin que des informations sur les adresses IPv4 et IPv6 des serveurs de la racine ainsi que sur la répartition géographique de ces serveurs sont publiées sur le site web : [\[\[1\]\]](#).

Glue IPv6

La zone racine publie également les adresses des différents serveurs DNS de chacun des domaines racines (TLD : *Top Level Domain*). Ces adresses, appelées « glue » sont nécessaires au démarrage du processus de résolution des noms.

En effet, rappelons que les serveurs DNS "racine" ne répondent pas eux-mêmes aux requêtes des clients. Leur rôle est de faire le premier aiguillage (*referral*) vers des serveurs DNS "racine" (TLD) : les serveurs DNS qui gèrent les domaines "racine" (TLD). Les informations d'aiguillage incluent la liste des serveurs "racine" qui gèrent officiellement les informations de nommage

d'une zone. Elles incluent également les adresses (glues) de ces serveurs. Sans ces adresses, la résolution ne peut se faire. Le client aurait le nom du serveur, mais pas son adresse et ne pourrait l'obtenir...

En attendant que les serveurs "racine" puissent recevoir des requêtes DNS et répondre en IPv6, les domaines "racine" TLD ont pendant des années milité pour l'introduction des « glues » IPv6 qui leurs sont associées dans la zone racine. L'IANA/ICANN a fini par se convaincre que la publication des adresses IPv6 des serveurs DNS "racine" supportant IPv6 pouvait se faire sans risque pour la stabilité du DNS. L'ICANN/IANA a démarré, en juillet 2004, la publication des adresses IPv6 des domaines "racine" TLD dans la zone racine. Les trois TLD **.fr**, **.jp** et **.kr** ont, les premiers, vu leur glue IPv6 publiée. Aujourd'hui (en 2015), 10 serveurs DNS "racine" fonctionnent en IPv6.

Publication des enregistrements AAAA dans le DNS

On choisit généralement de publier dans le DNS les enregistrements AAAA d'un équipement donné lorsque l'on souhaite que les applications communiquant avec cet équipement découvrent qu'il supporte le transport IPv6. Par exemple, un navigateur supportant IPv6, découvre ainsi, grâce au DNS, qu'il est possible d'accéder en IPv6 au site <http://www.afnic.fr/>. Il peut alors choisir de privilégier la connexion HTTP au serveur en IPv4 ou en IPv6. Or, avec l'intégration progressive d'IPv6, l'adresse IPv6 d'un équipement peut être publiée dans le DNS. Malgré tout, certaines applications s'exécutant sur cet équipement peuvent cependant ne pas supporter IPv6.

La situation suivante risque donc de se produire. L'équipement *foo.tpt.example.com* héberge plusieurs services : web, ftp, mail, DNS. Les serveurs Web et DNS s'exécutant sur *foo.tpt.example.com* supportent IPv6, mais pas les serveurs FTP et mail. Une adresse IPv6 est publiée dans le DNS pour *foo.tpt.example.com*. Un client FTP supportant IPv6 tente d'accéder au serveur de notre équipement : *foo.tpt.example.com*. Le client choisit l'adresse IPv6 associée à *foo.tpt.example.com* comme adresse destination. Sa tentative d'accès au serveur FTP en IPv6 échoue. Selon les implémentations, les clients tentent ou non d'utiliser d'autres adresses IPv6, s'il y en a, et finissent ou non par tenter d'y accéder, en dernier recours, en IPv4.

Notez que, pour pallier ce problème, l'IETF recommande d'associer des noms DNS aux services et non aux équipements. Ainsi, pour notre exemple précédent, il serait judicieux de publier dans le DNS, d'une part, les noms *www.tpt.example.com* et *ns.tpt.example.com* associés à des adresses IPv6, et éventuellement, des adresses IPv4, et d'autre part, les noms *ftp.tpt.example.com* et *mail.tpt.example.com* associés uniquement à des adresses IPv4.

L'enregistrement AAAA pour *foo.tpt.example.com* ne serait alors publié que lorsque l'on aurait la certitude que toutes les applications s'exécutant sur cet équipement supportent IPv6. Par ailleurs, le DNS étant une ressource publique, il est fortement déconseillé (sauf si l'administrateur DNS sait très bien ce qu'il fait !) d'y publier des adresses IPv6 non accessibles depuis l'extérieur, soit à cause d'une portée trop faible (adresse locale au lien, par exemple), soit parce que toutes les communications provenant de l'extérieur du réseau et allant vers ces adresses sont filtrées. Notez que cette règle est déjà appliquée pour les adresses IPv4 privées

([RFC 1918](#)) et que certains logiciels DNS récents supportent aujourd'hui les vues DNS. On parle de *two-face DNS*, de *split-view DNS* ou encore de *split DNS*. Les vues permettent d'exécuter plusieurs serveurs virtuels sur une même machine. Elles permettent que la réponse à une requête DNS dépende de la localisation du client. Par exemple, un client du réseau interne voit les adresses privées des équipements alors que les clients externes ne voient eux que les adresses globales et accessibles depuis l'extérieur.

Pour aller plus loin : mises à jour dynamiques du DNS

Le système de noms de domaines a été initialement conçu pour interroger une base de données statique. Les données pouvaient changer, mais leur fréquence de modification devait rester faible. Toutes les mises à jour se faisaient en éditant les fichiers de zone maîtres (du serveur DNS primaire).

L'opération de mise à jour, UPDATE, permet l'ajout ou la suppression de RR ou d'ensembles de RR dans une zone spécifiée, lorsque certains prérequis sont satisfaits. Cette mise à jour est possible depuis un serveur DHCPv6, par exemple, ou depuis une machine IPv6 (autoconfiguration "sans état"). La mise à jour est atomique, c'est-à-dire qu'elle sera effectuée intégralement avant qu'une autre opération soit effectuée et tous les prérequis doivent au préalable être satisfaits pour que la mise à jour soit possible et qu'elle ait lieu. Aucune condition d'erreur relative aux données ne peut être définie après que les prérequis soient satisfaits. Les prérequis concernent un ensemble de RR ou un seul RR. Ceux-ci peuvent ou non exister. Ils sont spécifiés séparément des opérations de mise à jour.

La mise à jour s'effectue toujours sur le serveur DNS primaire de la zone concernée. Si un client s'adresse à un serveur DNS secondaire, ce dernier relaie la demande de mise à jour vers le serveur DNS primaire (*update forwarding*). Le serveur DNS primaire incrémente le numéro de version de l'enregistrement SOA de la zone concernée, soit après un certain nombre de mises à jour, par exemple 100, soit à l'expiration d'un certain délai, par exemple 5 minutes, en fonction de celle des deux conditions qui est satisfaite la première. Les serveurs DNS secondaires obtiennent une copie des fichiers de zone modifiés par le serveur DNS primaire par transfert de zone. Ceci leur permet de prendre en compte les modifications dynamiques effectuées au niveau du serveur.

Des serveurs tels que DHCP utilisent la mise à jour dynamique pour déclarer les correspondances "nom – adresse" et "adresse – nom" allouées automatiquement aux machines. La structure des messages DNS est modifiée pour les messages de mise à jour du DNS. Certains champs sont ajoutés, d'autres sont surchargés. Ils utilisent alors la procédure *ns_update* du résolveur. Ainsi, la commande *nsupdate* permet, sur un système Linux, les mises à jour dynamiques du DNS en ligne de commande. Pour des raisons évidentes, les mises à jour dynamiques du DNS utilisent des mécanismes de sécurité.

Conclusion

Le système de nommage est l'application client-serveur distribuée qui fonctionne à la plus

grande échelle qui soit. C'est un système de base de données hiérarchique. Il utilise un arbre de nommage pour garantir l'unicité des noms de domaine. Il a été initialement conçu pour stocker des correspondances directes (nom – adresse) et les correspondances inverses (adresse – nom). Mais il peut, plus généralement, stocker tout type d'information ; en particulier, celles concernant les agents de transfert ou serveurs de courrier ou les serveurs de noms.

Ce système privilégie la récupération d'information sur la fraîcheur de l'information remise. Un serveur de nommage fournit une réponse, en fonction des données dont il dispose, sans attendre la fin d'un transfert éventuel de zone. Pour pallier le délai de mise à jour des données de zone du serveur DNS secondaire, un client DNS, un résolveur, peut demander à obtenir des informations du serveur DNS primaire de la zone. Ce serveur est forcément à jour.

Un nom absolu correspond au chemin qui, dans l'arbre de nommage relie une feuille à la racine de l'arbre de nommage. La racine sans nom de l'arbre de nommage est représentée par un « . ». Un domaine est un nœud de l'arbre de nommage.

Le client du système de nommage, le résolveur, est unique pour une machine donnée. Il est réalisé sous forme d'une bibliothèque de procédures. Il s'initialise à partir d'un fichier de configuration ou d'informations fournies par un serveur DHCP ou encore d'options spécifiques des annonces de routeur. Le fichier de configuration du résolveur s'appelle généralement *resolv.conf*.

Le service de nommage est le seul pour lequel l'utilisation de l'adresse IP d'au moins un serveur est obligatoire. L'utilisateur qui souhaite communiquer avec une machine distante fournit généralement le nom de cette machine. Les applications TCP/IP utilisent les procédures de la bibliothèque du résolveur pour obtenir l'adresse IP associée à ce nom. Une fois l'adresse obtenue, elles peuvent établir une session en mode "avec" ou "sans connexion" avec cette machine distante.

Le système de nommage associe une hiérarchie de serveurs de noms à l'arbre de nommage. A chaque nœud de l'arbre correspond un serveur de nommage. Chaque serveur dispose d'un pointeur vers chacun de ses fils et un pointeur vers son père. Chaque père connaît chacun de ses fils. Pour équilibrer la charge, le serveur racine est répliqué.

Les enregistrements de ressources de type A, pour IPv4 et AAAA, pour IPv6, gèrent respectivement les correspondances directes "nom – adresse" respectivement pour IPv4 et pour IPv6. Ils permettent que les utilisateurs manipulent les noms des machines et non leurs adresses. Dans le cas d'IPv6, cela évite que les utilisateurs aient à retenir des adresses IPv6 représentées en notation hexadécimale pointée.

La configuration d'un service de nommage en IPv6 suppose la configuration d'un serveur DNS primaire et d'au moins un serveur DNS secondaire. Ces deux serveurs sont des serveurs DNS officiels pour la zone concernée. Le serveur DNS primaire utilise des fichiers maîtres contenant les informations de nommage direct et indirect. Ces fichiers sont enregistrés dans une mémoire non volatile.

Le fichier de nommage direct, unique pour chaque zone, contient les correspondances "nom-

adresse" IPv4 et IPv6 pour toutes les machines de la zone. Le nommage inverse contient un fichier par lien en IPv6 ou par sous-réseau en IPv4. Les serveurs DNS secondaires peuvent enregistrer, dans une mémoire non volatile, une copie locale des fichiers de zone. L'IETF le recommande fortement. Cette pratique, qui réplique la base de nommage, accélère le démarrage des serveurs DNS secondaires et augmente la robustesse du service en cas de panne catastrophique (ou non) du serveur DNS primaire.

Les outils de vérification de configuration *named-checkconf* et *named-checkzone* vérifient respectivement l'absence d'erreur dans le fichier de configuration de BIND9 et dans les fichiers de zone. L'analyse des fichiers journaux permet de vérifier l'absence d'erreur à l'exécution du service. Le fichier journal est généralement */var/log/syslog* par défaut sur un système Linux. L'utilisateur vérifie le bon fonctionnement de la résolution directe et de la résolution inverse avec les outils *dig* et *host*. ces commandes utilisent par défaut les informations du fichier *resolv.conf*.

Pour éviter la fragmentation de l'espace de nommage due à la coexistence d'IPv4 et d'IPv6, les administrateurs de réseaux doivent configurer au moins un serveur *dual* ou un relais *DNS dual* dans chaque zone.

Les mises à jour dynamiques du système de nommage ont été introduites pour que des services comme DHCP puissent déclarer les correspondances directes et les correspondances inverses des machines auxquelles ils attribuent noms et adresses. Elles utilisent des mécanismes de sécurité pour interdire les modifications non autorisées du service DNS. Les mises à jour atomiques ne sont effectuées que lorsque tous les prérequis d'une mise à jour sont satisfaits. Sinon, elles ne le sont pas.

1. ↑ Evans R. (2015). [Medium On DNS and IPv6](#)

ANNEXE 3 Activité 34 : Format DHCPv6

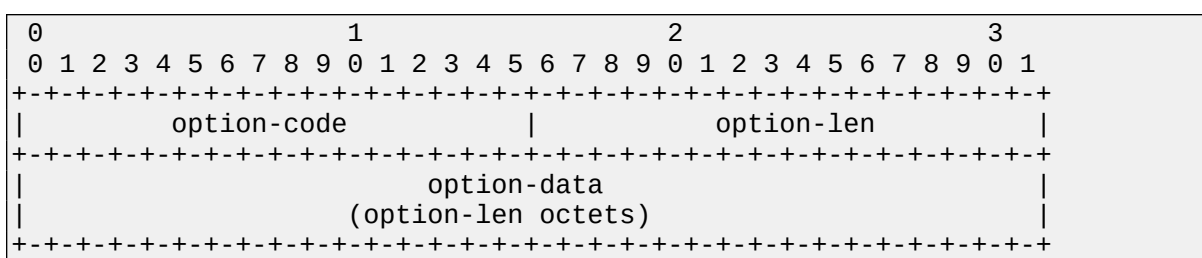
Structure des options du protocole DHCPv6

La structure générale des options est décrite ci-dessous. Elle correspond à un codage TLV : type, longueur, valeur.

Le type ou code est un entier non signé. Il précise quelle est l'option. La longueur de l'option précise la taille en nombre d'octets du champs de données de l'option. Le champ type de l'option en est exclu. Les données de l'option suivent. Dans certains cas, une option peut en contenir d'autres.

La portée des options est définie par encapsulation. Certaines options s'appliquent globalement, d'autres sont spécifiques d'une association d'identités, d'autres encore sont spécifiques d'une adresse, dans une association d'identités.

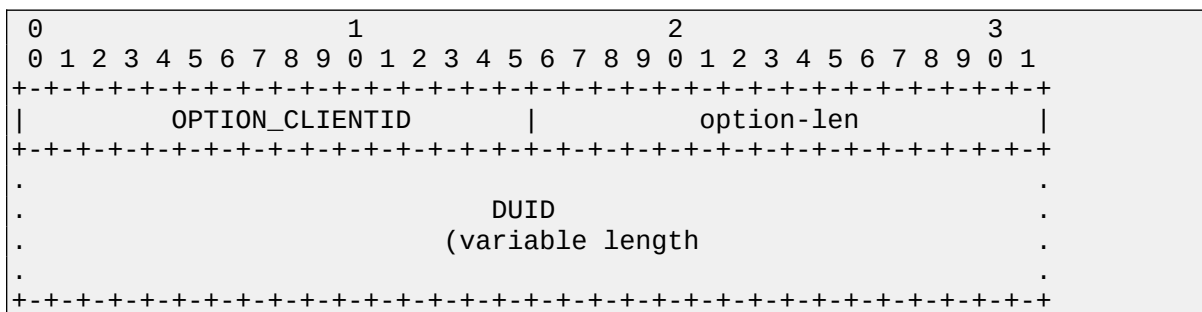
La structure générale d'une option est la suivante :



Option d'identification du client

L'option d'identification du client (*Client Identifier Option*) transporte le DUID (*DHCPv6 User Identification*) du client dans les messages DHCPv6 échangés entre client et serveur.

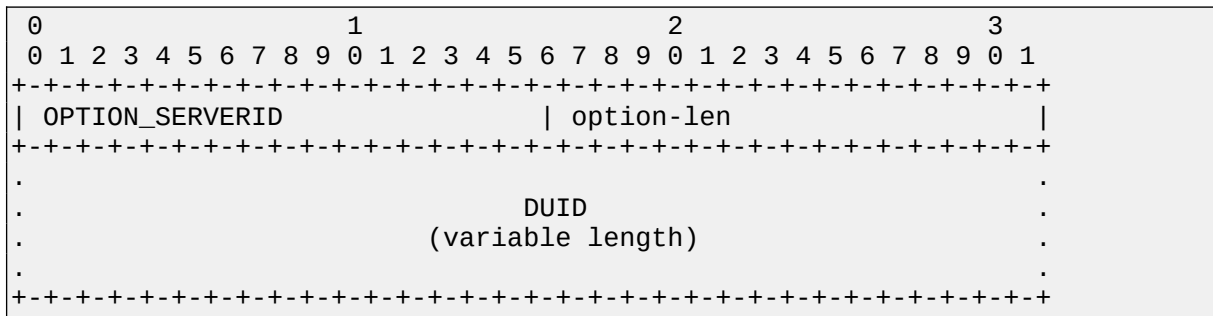
La structure de cette option est la suivante :



Option identification du serveur (*Server Identification Option*)

L'option "identification du serveur" (*Server Identification Option*) transporte le DUID (*DHCPv6 User Identification*) du serveur dans les messages DHCPv6 échangés entre client et serveur.

La structure de cette option est la suivante :

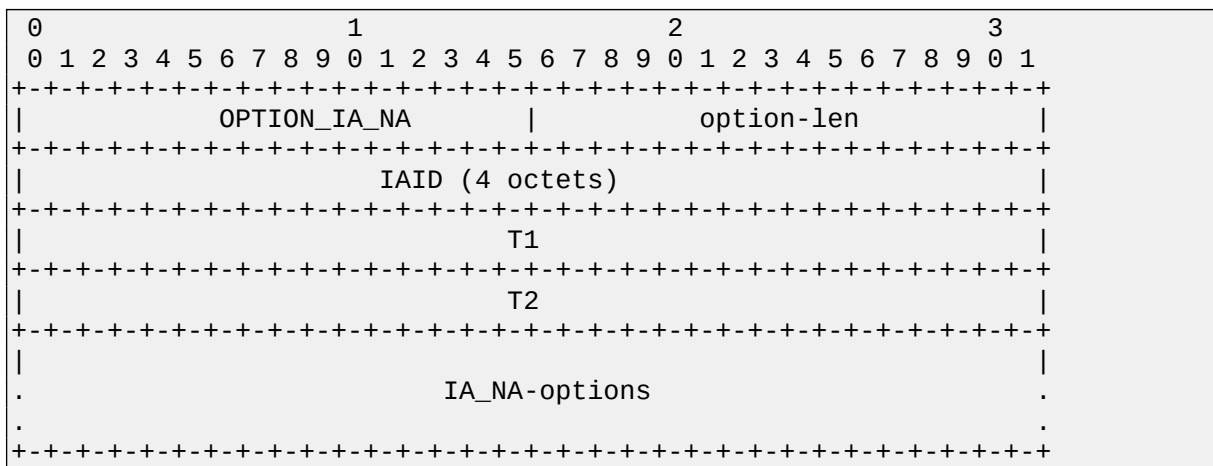


Option association d'identité pour les adresses non temporaires

L'option "association d'identité" pour les adresses non temporaires (option IA_NA : *Identity Association for Non Temporary Addresses*) inclut les paramètres de cette association et les adresses non temporaires associées. Elle apparaît une ou plusieurs fois dans le champ d'options d'un message DHCPv6.

Cette association transporte un identificateur d'IA_NA, les temporisations T1, durée de vie préférée d'une adresse, et T2, durée de vie maximum d'une adresse, et les options de cette association, par exemple la liste des options d'adresse spécifiques de cette association.

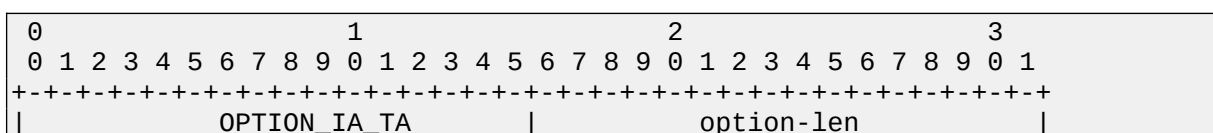
La structure de cette option est la suivante :



Option d'association d'identité pour les adresses temporaires

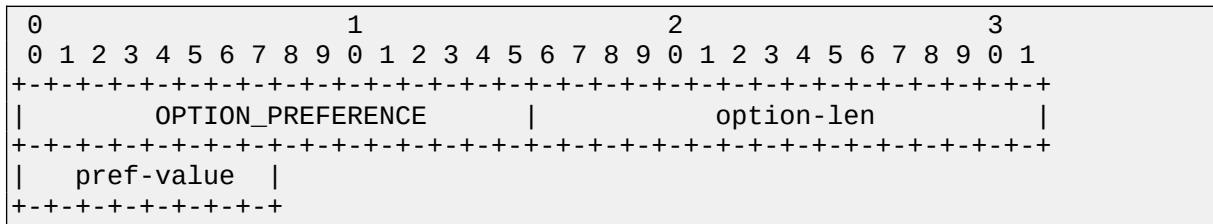
L'option d'association d'identité pour les adresses temporaires (option IA_TA : *Identity Association for Temporary Addresses*) inclut les paramètres de cette association et au plus une adresse temporaire associée par préfixe autorisé sur le lien du client. Elle apparaît une ou plusieurs fois dans le champ d'options d'un message DHCPv6. Une option statut indique l'état de toute opération impliquant cette option.

La structure de cette option est la suivante :



Un client choisit le serveur de priorité la plus élevée. En cas d'égalité des priorités, il choisit le serveur de priorité la plus élevée qui lui propose la meilleure offre. Il peut ne pas choisir l'offre du serveur le plus prioritaire. Le choix repose alors sur l'adéquation de l'offre.

La structure de cette option est la suivante :

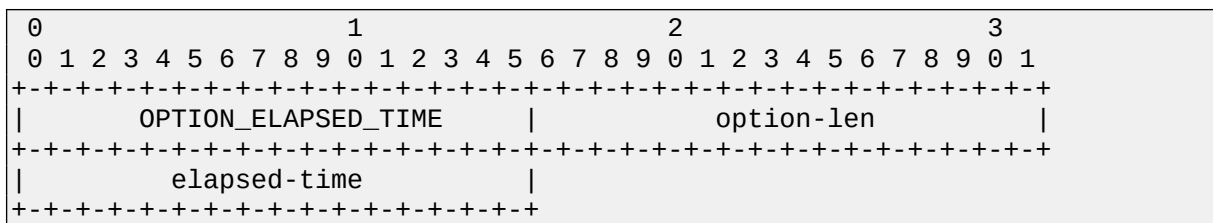


Option "temps écoulé" (depuis le début d'un échange)

L'option "temps écoulé" mesure le temps écoulé (*Elapsed Time Option*) depuis l'émission du premier message d'un échange DHCPv6 inachevé. Cette option vaut 0 dans le premier message d'un échange.

Serveurs et agents utilisent la valeur de cette option pour déterminer leur façon de traiter le message DHCPv6 correspondant. La valeur ffff en hexadécimal (0xffff) représente une durée supérieure à la plus grande durée représentable.

La structure de cette option est la suivante :



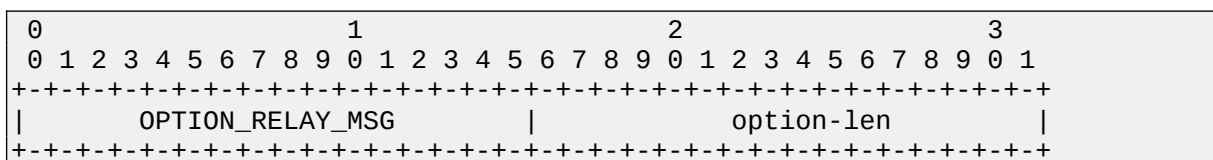
Option "message relayé"

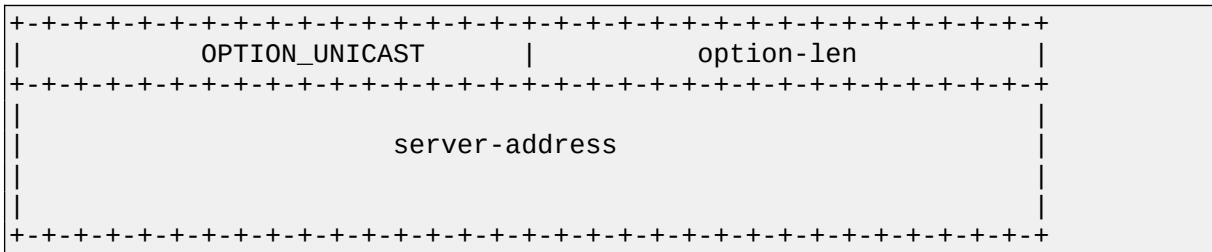
L'option "message relayé" (*RELAY Message Option*) contient le message DHCPv6 relayé dans un message RELAY-FORWARD ou RELAY-REPLY.

Le message relayé, dans le cas d'un message qui transite du client vers le serveur, est soit le message DHCPv6 du client (premier relais), soit le message RELAY-FORWARD du relais précédent (du deuxième relais au dernier).

Le message relayé dans le cas d'un message qui transite du serveur vers le client est, soit le message REPLY du serveur (premier relais), soit le message RELAY-REPLY du relais précédent (du deuxième relais au dernier).

La structure de cette option est la suivante :

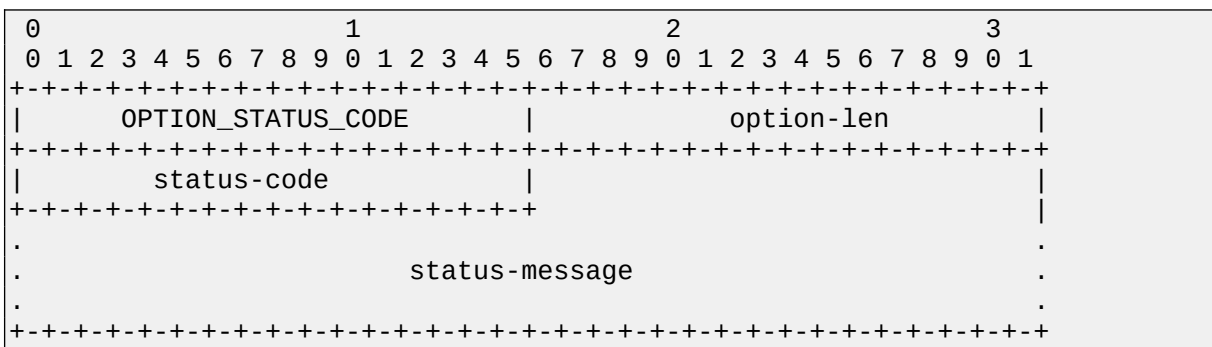




Option de code d'état

L'option code d'état (*Status Code Option*) renvoie une indication d'état relative au message DHCPv6 ou à l'option dans laquelle cette option apparaît. L'omission du code d'état dans un message ou dans une option où son utilisation est possible signifie "succès".

La structure de cette option est la suivante :



L'annexe 2 présente les valeurs des différents codes d'état.

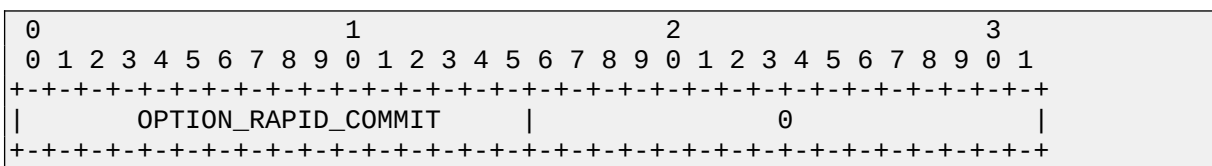
Option de Validation rapide

L'option de validation rapide (*Rapid Commit Option*) indique l'utilisation d'un échange à deux messages pour l'allocation d'adresses IPv6. Le principe de cette allocation est le suivant :

(1) Un client, prêt à utiliser la validation rapide peut inclure cette option dans son message SOLICIT.

(2) Un serveur doit inclure cette option dans le message REPLY qui répond au SOLICIT du client transportant l'option de validation rapide.

La structure de cette option est la suivante :

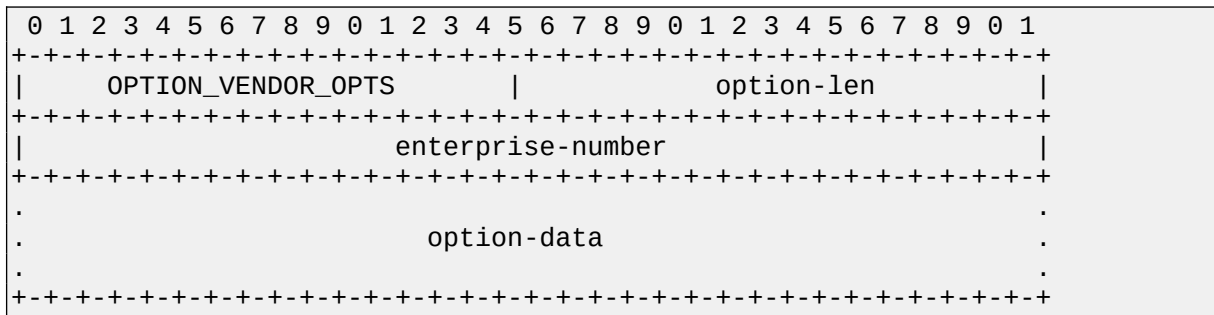


Option classe d'utilisateur

L'option classe d'utilisateur (*User Class Option*) identifie un type ou une classe d'utilisateurs ou d'applications qu'ils représentent. La partie données de cette option contient plusieurs champs non interprétés (*opaque*) par DHCPV6. Ces champs représentent la classe d'utilisateur à

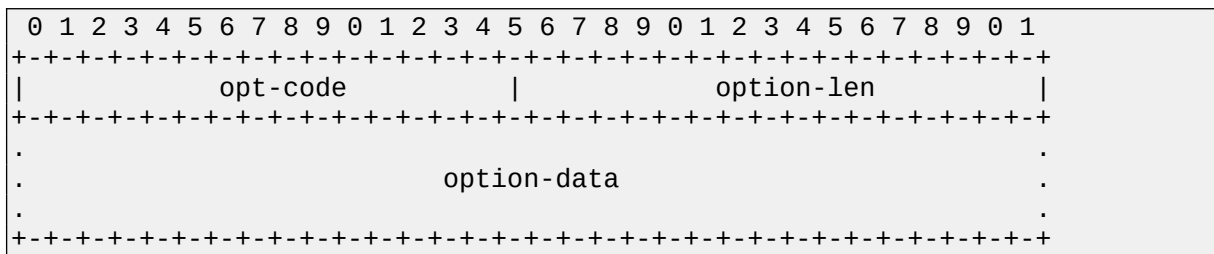
Le numéro d'entreprise identifie le constructeur.

La structure de cette option est la suivante :



La spécification des données échangées dépend du constructeur. Chacune de ces options de données est codée en format TLV. Le constructeur définit leur code. Plusieurs options de données peuvent se succéder dans le champ de données de l'option d'information spécifique d'un constructeur.

La structure de l'option de donnée spécifique d'un constructeur est la suivante :



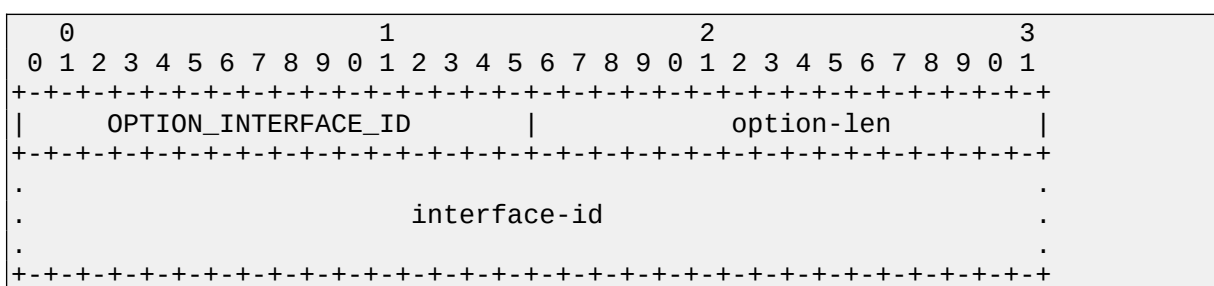
Option d'identification d'interface

L'option d'identification d'interface (*Interface-Id Option*) identifie, sur un relais, l'interface de réception du message d'un client.

Un relais qui reçoit un message incluant une option d'identification d'interface relaie le message reçu sur l'interface identifiée dans l'option.

Les serveurs qui reçoivent cette option dans un message RELAY-FORWARD doivent la recopier dans leur message RELAY-REPLY. cette option est spécifique des messages RELAY-FORWARD et RELAY-REPLY. Ils peuvent également utiliser cette information pour appliquer une politique d'allocation basée sur la correspondance exacte de la valeur de cette option.

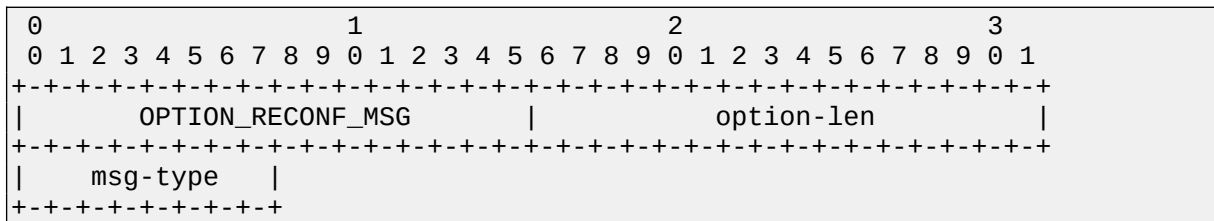
La structure de cette option est la suivante :



Option de message de reconfiguration

L'option de message de reconfiguration (*Reconfigure Message Option*), présente dans un message de reconfiguration issue d'un serveur, indique au client s'il doit répondre à l'aide d'un message RENEW ou INFORMATION-REQUEST. Cette option est spécifique du message de reconfiguration.

La structure de cette option est la suivante :

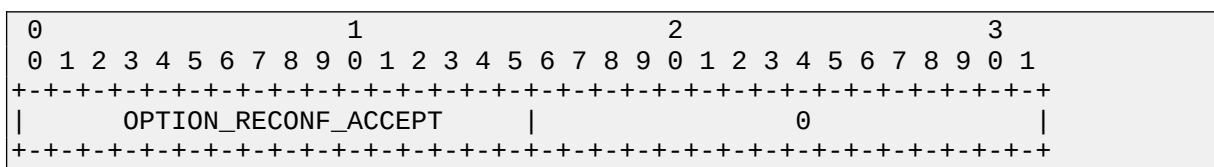


Option d'acceptation de reconfiguration

L'option d'acceptation de reconfiguration (*Reconfigure Accept Option*) annonce au serveur que le client accepte les messages de reconfiguration.

Un serveur utilise cette option pour dire au client s'il doit ou non accepter les messages de reconfiguration. L'absence de cette option indique le refus d'accepter des messages de reconfiguration. La présence de cette option indique au client s'il doit ou non accepter les messages de reconfiguration.

La structure de cette option est la suivante :



Extension du protocole DHCPv6 : options spécifiques des relais

Dans certains cas, les relais DHCPv6 connaissent des informations qui seraient utiles aux clients DHCPv6.

Le protocole DHCPv6 est étendu ([RFC 6422](#)) pour que les relais puissent inclure une option RSSO : RELAY-SUPPLIED OPTIONS OPTION dans les messages RELAY-FORW adressés au serveur DHCPv6.

L'option d'options spécifiques de relais (RELAY-SUPPLIED OPTIONS OPTION) dans les messages RELAY-FORWARD adressés au serveur DHCPv6 contient alors toutes les options correspondant à des paramètres que le relais souhaite porter à la connaissance du client. Cette possibilité n'est effective que pour des paramètres classés RSOO.

Le serveur DHCPv6 qui reçoit un message RELAY-FORWARD contenant une option RSSO enregistre les options classées RSOO fournies par le relais DHCPv6. Il peut ensuite transmettre ces informations aux clients en ajoutant les options de classe RSOO qu'il accepte de transmettre au client.

Structure des identifiants DUID du protocole DHCPv6

DUID construit à partir de l'adresse physique + horodate (DUID-LLT)

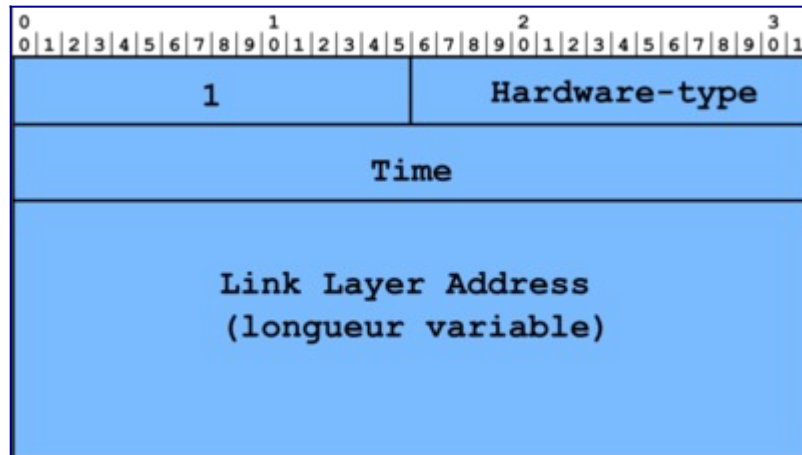


Figure 1 : Format du DUID-LLT.

Msg-type : le champ type (2 octets) vaut 1.

Hardware type : deux octets contiennent le type de réseau physique.

Time : l'horodate est codée sur 4 octets.

Link-layer address : la longueur de l'adresse physique (adresse MAC) varie en fonction du type du réseau physique.

Le choix de l'interface dont on utilise l'adresse physique est indifférent tant que l'identification est unique. Le DUID doit être enregistré dans une mémoire non volatile et doit continuer à être utilisé, même en cas de remplacement ultérieur de l'interface qui a servi à le générer.

Ce type de DUID est recommandé pour les ordinateurs de bureau, les ordinateurs portables, ou plus généralement pour tout équipement doté d'une mémoire non volatile où l'écriture est possible.

DUID dérivé du numéro d'entreprise affecté par un constructeur (DUID-EN)

Un constructeur affecte ce type d'identificateur à un équipement. Le DUID-EN combine le numéro unique affecté à l'entreprise et un identificateur de longueur variable, unique pour l'entreprise et défini par elle. Le numéro d'entreprise est généralement un entier non signé codé sur 32 bits. La figure 2 présente la structure de l'option.

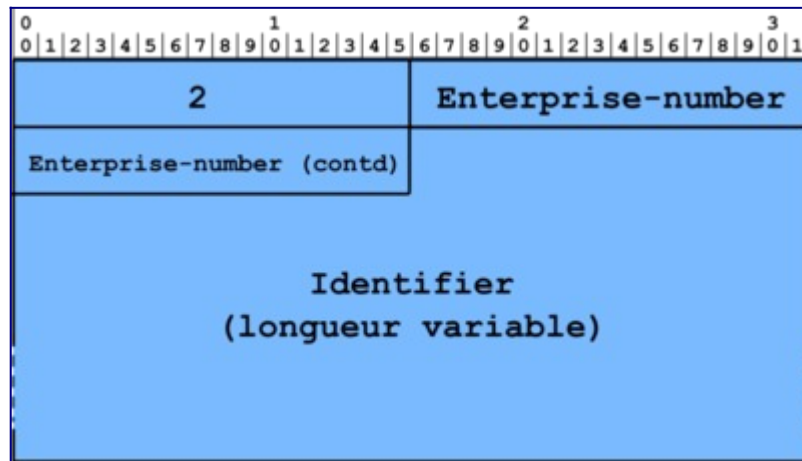


Figure 2 : Format du DUID-EN.

Le constructeur affecte généralement cet identificateur unique à l'équipement lors de sa construction et l'enregistre dans une mémoire non volatile de l'équipement.

DUID dérivé de l'adresse physique de l'équipement (DUID-LL)

Le DUID-LL n'utilise que l'adresse physique de l'équipement. La longueur de l'adresse physique (adresse MAC) varie en fonction du réseau physique. Le choix de l'interface dont on utilise l'adresse physique est indifférent tant que l'identification est unique. Le DUID doit être enregistré dans une mémoire non volatile et doit continuer à être utilisé, même en cas de remplacement ultérieur de l'interface qui a servi à le générer. La figure 3 présente la structure de l'option.

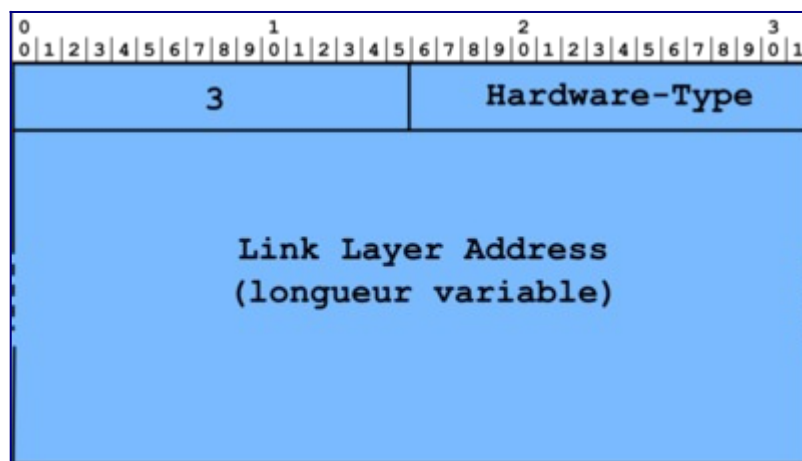


Figure 3 : Format du DUID-LL.

Le constructeur affecte généralement cet identificateur unique à l'équipement lors de sa construction. Il l'enregistre généralement dans une mémoire non volatile de l'équipement.

Ce format est recommandé pour les équipements dépourvus de mémoire de stockage et qui ont une interface de réseau connectée en permanence au réseau (une imprimante réseau, par exemple).

Options pour la délégation de préfixes ([RFC 8415](#))

Structure de l'option d'association d'identités pour la délégation de préfixes

La figure 4 présente la structure de cette option.

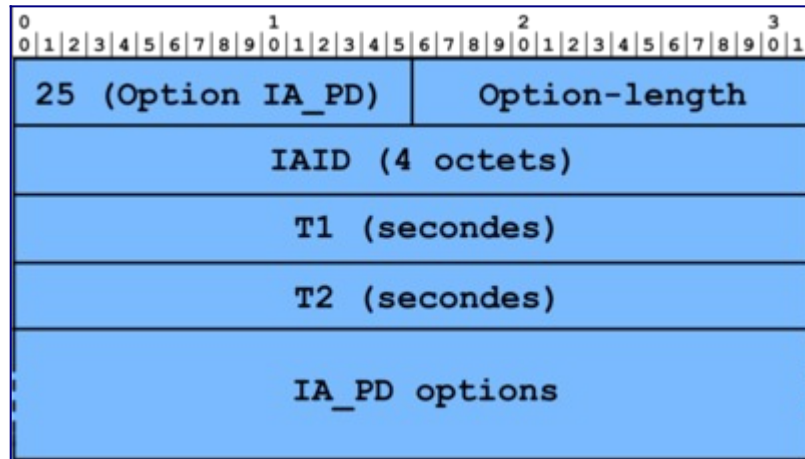


Figure 4 : Format de l'option d'association d'identités pour la délégation de préfixes.

OPTION_IA_PD : le champ *type* de cette option a pour valeur 25.

Option-length : la longueur de l'option est la longueur, en nombre d'octets, de la valeur des options IA_PD options.

IAID : c'est l'identificateur d'association d'identités.

T1, *T2* : les temporisations *T1* et *T2* représentent, en secondes, les durées de vie du préfixe en mode préféré et durée de vie totale.

Option de préfixe d'association d'identités pour la délégation de préfixe

L'option de préfixe d'association d'identités pour la délégation de préfixe (IA_PD Prefix) contient les préfixes associés à une IA_PD. Elle est incluse dans l'option IA_PD. La figure 5 présente la structure de cette option.

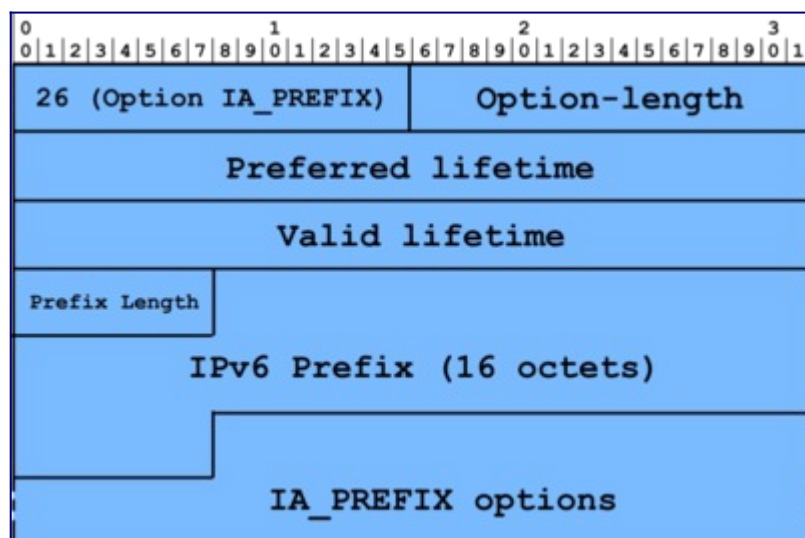


Figure 5 : Format de l'option de préfixe d'associations d'identités pour la délégation de préfixe.

Msg-type : le champ type de cette option vaut 26.

Option-length : le champ longueur du champ option est la longueur en nombre d'octets du champ option de cette option.

Preferred-lifetime, valid lifetime : les durées de vie préférée et totale sont celles du préfixe.

Prefix-length : ce champ donne la longueur en bits du préfixe.

IPv6 prefix : la valeur du préfixe, codée sur 16 octets, donne la valeur du préfixe.

IAprefix-options : liste les options relatives à ce préfixe.