MOOC
Objectif IPv6 !
vers l'internet nouvelle génération

Document Compagnon[1]

Séquence 3

Les mécanismes de gestion d'un réseau IPv6

Par **- Bruno Stévant** Télécom Bretagne / G6

- Bruno Joachim Télécom Paritech / G6

Avec la contribution de Nabil Benamar, Université de Meknès, Maroc.

Licence Creative Nommons 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 l**es 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

Tables des activités

Introduction 9

Activité 31: Contrôler le fonctionnement du réseau par ICMPv6 11
Fonctions de gestion du niveau IP 11
Format générique d'un message ICMPv6 11
Test d'accessibilité entre équipements (ping) 14
Rapport d'erreur 15
Destination inaccessible 15
Paquet trop grand 16
Délai expiré 16
Erreur de paramètre 17
Pourquoi filtrer avec précaution ICMPv6 17
Gestion des abonnements sur le lien-local: MLD 18
Pour aller plus loin: Fonctionnement du Protocole MLD 19
Messages de recensement et rapports d'abonnement périodiques MLD 19
Rapports d'abonnements MLD non-sollicités 20
Découverte des voisins 20
Détails des messages mis en oeuvre 20
Message Sollicitation d'un voisin 20
Message Annonce d'un voisin 21
Fonctionnement de la résolution d'adresse physique 22
Sollicitation du voisin 23
Annonce du voisin 24
Fonctionnement de la détection d'adresse dupliquée 25
Conclusion 26

Activité 32: Configurer automatiquement les paramètres réseau 27
Principe de l'auto-configuration 27
Mécanismes mis en oeuvre 27
La création de l'adresse lien-local 28
La fourniture des paramètres communs au réseau 28
L'auto-configuration sans état de l'adresse IP 30
L'auto-configuration avec état de l'adresse IP 32
La configuration de la table de routage 32
La découverte de la liste de serveurs DNS récursifs 33
Exemple de configuration automatique 34
Conclusion 40

Activité 33: Contrôler la configuration réseau par DHCPv6 41
Introduction 41
Principe de fonctionnement du protocole DHCPv6 41
Présentation générale du protocole DHCPv6 41
Pile de communication utilisée par DHCPv6 42
Présentation des entités du protocole DHCPv6 42
Gestion centralisée des ressources allouées 43
Fonctions des Messages du protocole DHCPv6 43
Messages échangés entre client et serveur 44
Messages de gestion des ressources allouées 44
Messages échangés entre relais et serveur 45
Tableau récapitulatif des messages DHCPv6 45

Pour en savoir plus: extension du protocole DHCPv6 [RFC 6422] 47
Structure des messages DHCPv6 47
Structure des messages émis par les serveurs et clients DHCPv6 47
Structure des messages échangés entre relais et serveur DHCPv6 48
Message DHCPv6 RELAY-FORWARD 50
Message DHCPv6 RELAY-REPLY 50
Types de DUID: DHCPv6 Unique IDentifier 50
DUID construit à partir de l'adresse physique + horodate (DUID-LLT) 51
DUID dérivé du numéro d'entreprise affecté par un constructeur (DUID-EN) 52
DUID dérivé de l'adresse physique de l'équipement (DUID-LL) 52
Association d'identités 53
Allocation des adresses non temporaires 53
Options du protcole DHCPv6 54
Principe de l'allocation d'adresse IPv6 à un client en l'absence de relais 55
Recherche des serveurs DHCPv6 par le client: fonctionnement de la pile de communication 56
Principe de l'allocation d'adresse IPv6 à un client en présence d'un relais DHCPv6 57
Libération de l'adresse IPv6 par le client DHCPv6 avec présence d'un relais 59
Délégation de préfixe à états 60
Applications de la délégation de préfixe 61
Renumérotation des réseaux 61
Renumérotation passive 62
Renumérotation active 62
Structure de l'option d'association d'identités pour la délégation de préfixes (RFC 3633 , RFC 7550) 62
Option de préfixe d'association d'identités pour la délégation de préfixe 63
Principe de l'allocation de préfixe à états sans relais 65
Principe de l'allocation de préfixe à états avec relais 66
Conclusion 66
Annexe 1. Structure des options du protocole DHCPv6 67
Option d'identification du client 68
Option identification du serveur (Server Identification Option) 68
Option association d'identité pour les adresses non temporaires 69
Option d'association d'identité pour les adresses temporaires 69
Option d'adresse d'association d'identités 70
Option de demande d'options 71
Option de priorité (du serveur) 71
Option temps écoulé (depuis le début d'un échange) 72
Option message relayé 72
Option d'authentification 73
Option d'utilisation de l'adresse individuelle du serveur 74
Option de code d'état 75
Option de Validation rapide 75
Option classe d'utilisateur 76
Option de classe de constructeur 76
Option d'information spécifique d'un constructeur 77
Option d'identification d'interface 78
Option de message de reconfiguration 79
Option d'acceptation de reconfiguration 79
Extension du protocole DHCPv6: options spécifiques des relais 80
Annexe 2. Codes d'état du protocole DHCPv6 80

Activité 34: Faire correspondre adresse et nom de domaine 83
Introduction 83
Concepts de base du DNS 83
Nommage «à plat» 83
Caractéristiques du système de noms de domaine 84
Principe de fonctionnement du service DNS 87
Serveurs de noms primaires et secondaires 91
Serveur DNS récursif (caching name server) 96
Relais DNS (forwarder) 96
Serveurs DNS à rôles multiples 97
Spécifications du service de nommage 97
Nommage direct: enregistrement AAAA 98
Nommage inverse: enregistrement PTR 99
Découverte de la liste de serveurs DNS récursifs 101
Extension de l'autoconfiguration sans état pour le DNS 102
Option de liste de serveurs DNS récursifs (RDNSS) 102
Option de liste de domaine recherchés (DNSSL) 103
Extension de la configuration à états, DHCPv6 103
Option serveur de nom récursif de DHCPv6 103
Option liste de suffixes de nom de domaine 104
Utilisation d'adresses anycast réservées 105
Mises en œuvre du service DNS 106
Logiciels DNS supportant IPv6 106
Présentation du principe de configuration d'un serveur DNS 107
Définition des fichiers de zone 108
Types d'enregistrement de ressource DNS 109
Configuration de serveur DNS 109
Réseau virtualisé utilisé pour générer ces exemples 109
Fichier de configuration d'un serveur BIND9 110
Exemple de contenu du fichier /etc/bind9/named.conf 111
Configuration du fonctionnement du serveur 111
Contenu du fichier named.conf.options 111
Exemple de configuration locale du serveur de noms BIND9 112
Exemple de contenu du fichier named.conf.local 113
Contenu du fichier named.conf.default-zones 115
Fichier de zone DNS pour la résolution directe (nom - adresse) 116
Fichier de zone DNS inverse en IPv6 117
Fichier db.131.tpt.example.com.rev 117
Fichier db.132.tpt.example.com.rev 118
Fichier db.133.tpt.example.com.rev 118
Clients du service de nommage 119
Exemple de fichier de configuration /etc/resolv.conf d'un serveur de noms 119
Exemple de fichier de configuration /etc/resolv.conf d'une machine 119
Outils de vérification de la configurations DNS 119
Exemples d'interrogation d'un serveur DNS avec dig: résolution directe 120
Exemple d'interrogation d'un serveur DNS avec la commande host: résolution directe 121
Exemple d'interrogation d'un serveur DNS avec la commande dig: résolution inverse 121
Exemple d'interrogation d'un serveur DNS avec la commande host: résolution inverse 122
Recommandations opérationnelles pour l'intégration d'IPv6 123

Deux impossibilités d'accéder au service de nommage et leurs remèdes 124
Premier scénario: client IPv4 et serveur IPv6 124
Second scénario: client IPv6 et serveur IPv4 124
Taille limitée des messages DNS en UDP, extension EDNS.0 125
Glue IPv6 126
Publication des enregistrements AAAA dans le DNS 126
Pour aller plus loin: mises à jour dynamiques du DNS 127
Conclusion 128

Conclusion 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 du réseau, avec une intervention minimale des équipements intermédiaires, intervention réduite la plus part du temps à la fonction de routage.

Cette séquence va se concentrer sur les mécanismes qui ont été spécifiés autour du protocole IPv6 pour assurer le bon fonctionnement d'un réseau IP. La première activité va vous présenter le protocole **ICMPv6** qui permet de faire, à l'échelle de l'Internet, le contrôle du fonctionnement et l'envoi de rapport d'erreur si besoin. La seconde activité décrira les mécanismes définis pour assurer la **configuration automatique** des paramètres réseau au niveau du réseau local. La troisième activité abordera la **configuration automatique avec état** . Enfin pour la dernière activité, vous pourrez étudier le fonctionnement du **système de nommage** , essentiel pour qu'utilisateur et administrateur puisse désigner machines et services par des noms plutôt que par des adresses IP.

Activité 31: Contrôler le fonctionnement du réseau par ICMPv6

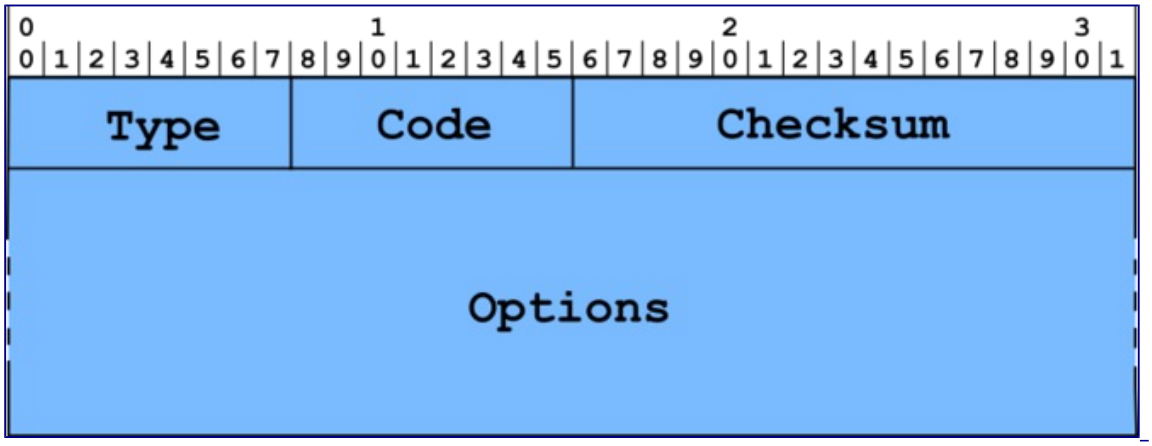
Fonctions de gestion du niveau IP

Même si le protocole IPv6 a été conçu pour être le plus simple possible, le bon fonctionnement d'un réseau IP nécessite un ensemble d'interactions entre les équipements, interactions relevant de la gestion du réseau et non du transport d'informations. Ces interactions permettent notamment:

- le signalement d'erreur en cours d'acheminement d'un paquet (équipement inaccessible, durée de vie expirée, etc.)
- le test d'accessibilité de bout en bout à travers le réseau (ping)
- La configuration automatique des équipements (redirection, découverte des voisins et routeurs)

Afin d'assurer les échanges pour réaliser ces fonctions, la famille de protocole ICMP (Internet Control Management Protocol) a été spécifiée. ICMPv6 (RFC 4443) réalise ces fonctions sur la base d'IPv6. De plus ICMPv6 intègre les fonctions de gestion des groupes de multicast au niveau du réseau local (assuré par le protocole IGMP en IPv4).

Format générique 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'entête IPv6 comme prochaine entête (champ Next Header). Le format générique des paquets ICMPv6 est donné figure Format générique d'un message ICMP:

- Le champ `type` (voir tableau Valeurs des champs type et code d'ICMPv6) code la nature du message ICMPv6. Les valeurs de ce champs sont classée pour distinguer les messages d'erreurs des autres messages d'information (ping, configuration automatique).
- Le champ `code` précise la cause du message ICMPv6.

- Le champ `checksum` permet de vérifier l'intégrité du paquet ICMP (rendu obligatoire pour tout protocole transporté au dessus d'IPv6).
- Le champ `options` est divisé en plusieurs champs selon le `type` et le `code` du message.

Les messages ICMPv6 de compte rendu d'erreur contiennent dans la partie données le paquet IPv6 ayant provoqué l'erreur. Pour éviter des problèmes de fragmentation puisqu'il est difficilement envisageable de mettre en œuvre la découverte du MTU, la longueur du message ICMPv6 est limitée à 1 280 octet. Par conséquent le contenu du paquet IPv6 renvoyé peut être tronqué.

Valeurs des champs type et code d'ICMPv6

type	code	nature
		Gestion des erreurs
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 firewall
	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
		Information
128		Demande d'écho
129		Réponse d'écho
		Gestion des groupes multicast (MLD, RFC 2710)

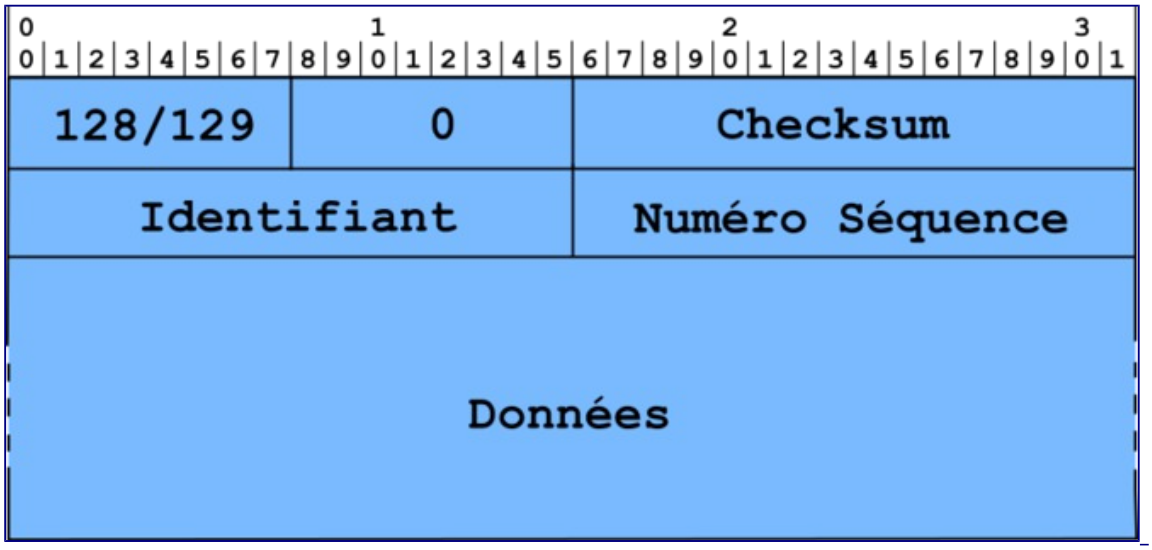
130		Requête d'abonnement
131		Rapport d'abonnement
132		Fin d'abonnement
	Découverte de voisins (RFC 2461)	
133		Sollicitation du routeur
134		Annonce du routeur
135		Sollicitation d'un voisin
136		Annonce d'un voisin
137		Redirection
	Renumérotation des routeurs (expérimental, RFC 2894)	
138		Renumérotation des routeurs:
	0	* Commande
	1	* Résultat
	255	* Remise à zéro du numéro de séquence
	Recherche d'information sur un noeud (expérimental)	
139		Demande d'information
140		Réponse
	Découverte de voisins inverse (RFC 3122)	
141		Sollicitation
142		Annonce
	Gestion des groupes multicast (MLDv2, RFC 3810)	
143		Rapport d'abonnement MLDv2
	Mobilité (RFC 3775)	
144		Découverte d'agent mère (requête)
145		Découverte d'agent mère (réponse)
146		Sollicitation de préfixe mobile
147		Annonce de préfixe mobile
	Découverte de voisins sécurisée (SEND, RFC 3971)	
148		Sollicitation de chemin de certification

149	Annonce de chemin de certification

Mobilité (expérimental)

150	Protocoles de mobilité expérimentaux, tels que Seamoby

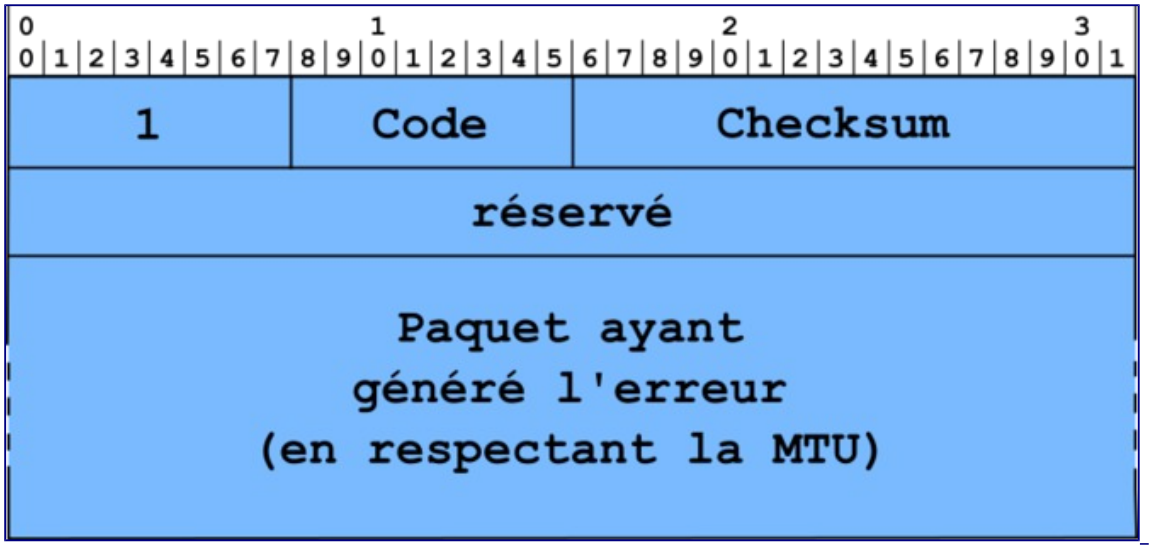
Test d'accessibilité entre équipements (ping)



Ces deux messages servent en particulier à la commande ping permettant de tester l'accessibilité d'une machine. Le principe de fonctionnement est le même que pour IPv4, une requête (type 128) est envoyée vers l'équipement dont on veut tester le fonctionnement, celui-ci répond par le message réponse d'écho (type 129). Le champ identificateur permet de distinguer les réponses dans le cas où plusieurs commandes ping seraient lancées simultanément sur la machine. Le champ numéro de séquence permet d'associer la réponse à une requête pour mesurer le temps d'aller et retour dans le cas où les demandes sont émises en continu et que le délai de propagation est élevé. Le champ données permet d'augmenter la taille du message pour les mesures.

Rapport d'erreur

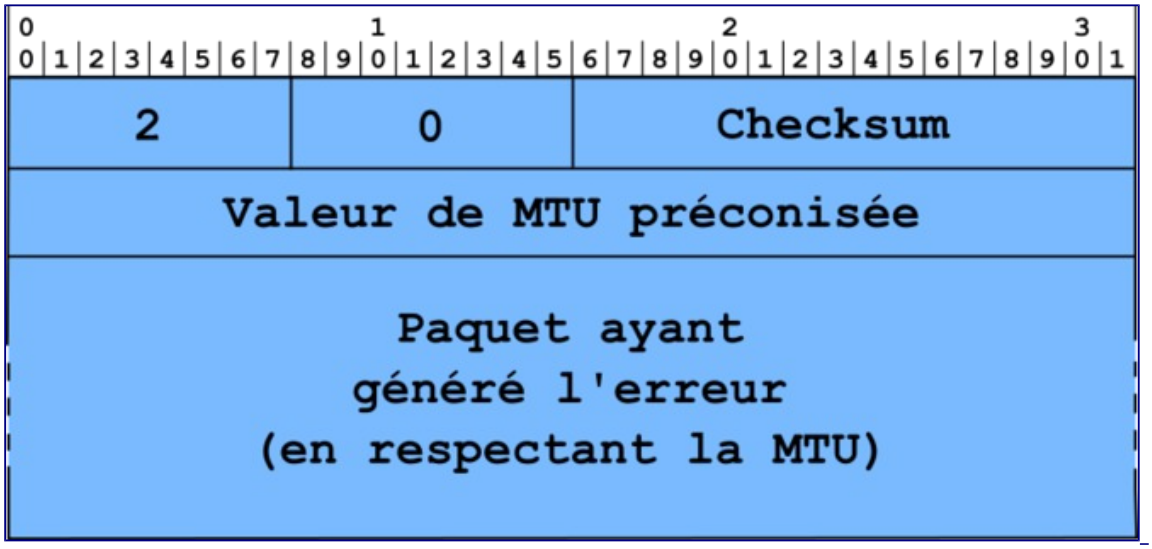
Destination inaccessible



Ce message est émis par un routeur intermédiaire quand le paquet ne peut pas être transmis parce que soit:

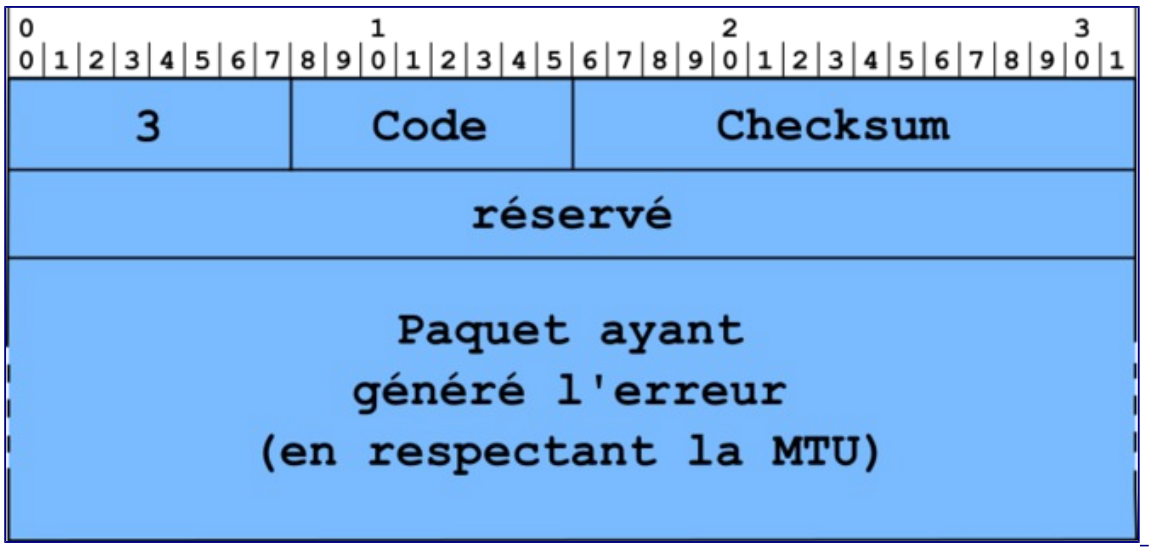
- le routeur ne trouve pas dans ses tables la route vers la destination (code = 0);
- le franchissement d'un équipement de type firewall est interdit ("raison administrative", code = 1);
- l'adresse destination ne peut être atteinte avec l'adresse source fournie, par exemple si le message est adressé 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 a un rejet du paquet (code = 6).

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 1981 , 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.

Délai expiré

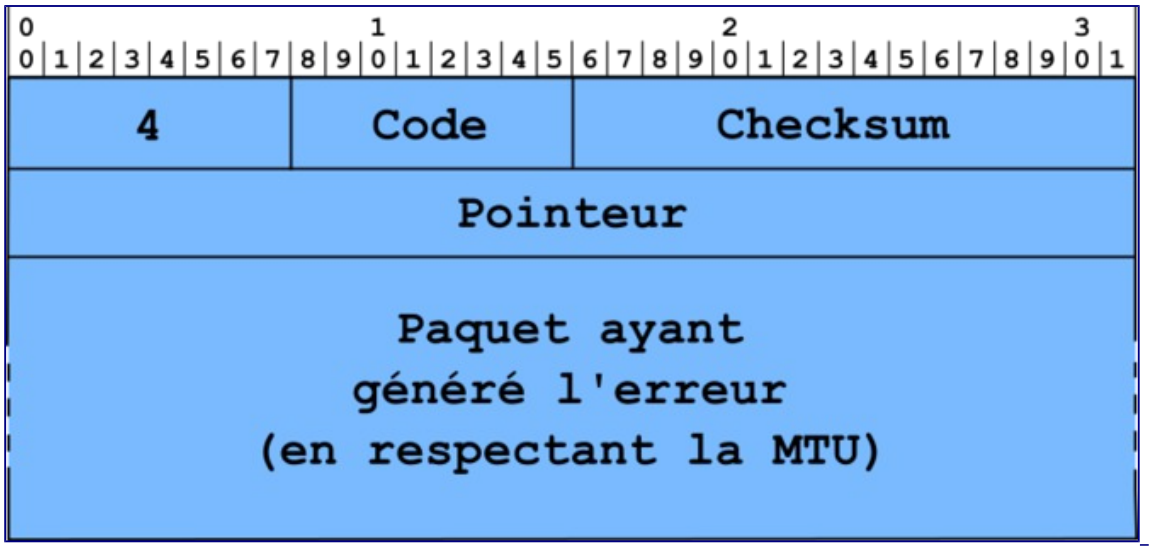


Ce message indique que le paquet a été rejeté par le routeur:

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

Ce message sert aussi à la commande traceroute pour déterminer le chemin pris par les paquets.

Erreur de paramètre



Ce message est émis par un nœud ayant détecté une erreur de syntaxe dans l'en-tête du paquet IP ou des extensions. 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.

Pourquoi filtrer avec précaution 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 Paquet trop grand) car cela peut avoir des conséquences néfastes sur le bon fonctionnement du réseau. Supposons que le MTU entre un client et un serveur soit de 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 commande HTTP courte (GET /). Le serveur va répondre en envoyant une page complète, si le paquet est trop grand, le routeur va rejeter le paquet et envoyer au serveur un message indiquant que le paquet et trop grand. Si le pare-feux du serveur le filtre, le serveur ne pourra jamais adapter la taille du paquet. On se trouve dans une situation où certains paquets passent (ouvertures de connexion, ping, sessions SSH,...) et d'autres sont bloquées.

Le RFC 4890 donne les bonnes pratiques pour filtrer correctement les paquets IPv6 en entrée d'un réseau.

Gestion des abonnements sur le lien-local: MLD

Pour offrir un service de distribution multicast, deux composants sont nécessaires: un protocole de gestion de groupe multicast et un protocole de construction d'arbre multicast. Le protocole de gestion de groupe multicast réalise la signalisation entre l'hôte et son routeur d'accès à l'Internet. En IPv6, ce protocole est MLD (*Multicast Listener Discovery*). Il est utilisé par un routeur de bordure IPv6 pour découvrir la présence de récepteurs multicast sur ses liens directement attachés, ainsi que les adresses multicast concernées.

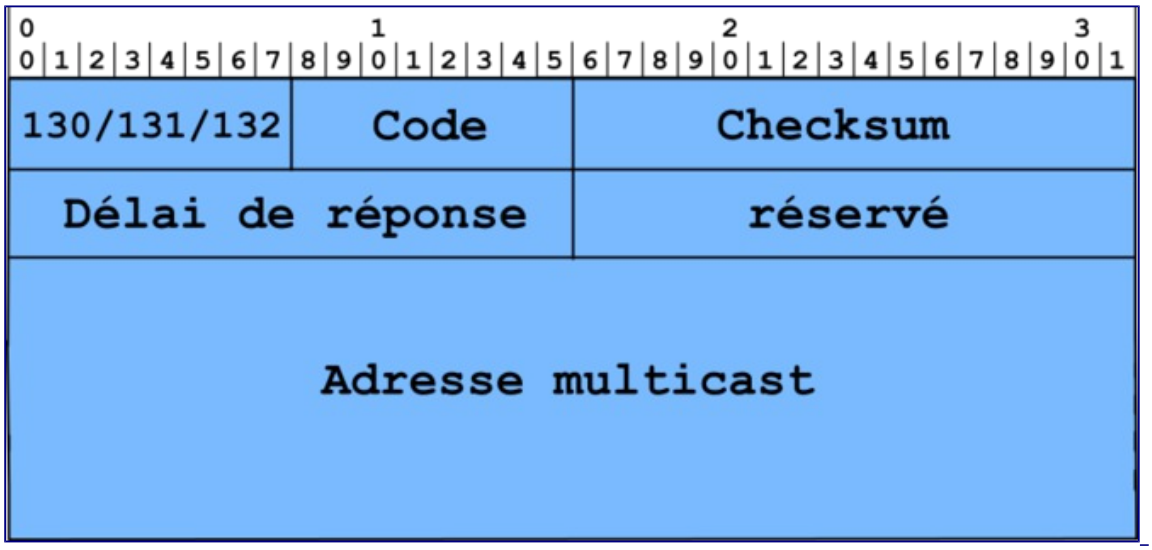
MLD est un protocole asymétrique qui spécifie un comportement différent pour les hôtes et les routeurs multicast. Toutefois, pour les adresses multicast sur lesquelles un routeur lui-même écoute, il doit exécuter les deux parties du protocole et répondre à ses propres messages.

Comme MLD est un sous-protocole d'ICMPv6, les messages MLD sont des messages ICMPv6 particuliers. Ils 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'entête Hop-by-Hop correspondante.

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.

Le format générique d'un message MLD est donné sur la figure Format générique d'un message ICMP pour MLD:



Trois types de messages sont utilisés. Le premier concerne le recensement des des récepteurs multicast (`type` = 130) selon plusieurs méthodes:

- recensement général émis à l'adresse de diffusion générale sur le lien (`FF02::1`)
- recensement spécifique à une adresse multicast, l'adresse de destination est l'adresse multicast du groupe en question

Le second permet d'obtenir un rapport d'abonnement multicast (`type` = 131), l'adresse de destination est l'adresse multicast du groupe en question Enfin le troisième permet à un récepteur d'annoncer une résiliation d'abonnement multicast (`type` = 132), émis à l'adresse du groupe multicast "tous les routeurs du lien local" (`FF02::2`).

Les champs 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
- `inutilisé` : 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

Pour aller plus loin: Fonctionnement du Protocole MLD

Messages de recensement et rapports d'abonnement périodiques MLD

Le routeur envoie régulièrement des messages de recensement général à l'adresse de diffusion générale sur le lien (`FF02::1`). Les hôtes arment un temporisateur pour chaque adresse multicast qui les concerne. Si un temporisateur expire sans que l'hôte ait entendu une réponse d'un de ses voisins concernant la même adresse, il envoie un rapport d'abonnement à l'adresse multicast du groupe. Ce système de temporisateurs permet aux hôtes de surveiller les rapports des autres hôtes sur le lien et d'annuler leurs propres rapports concernant les mêmes adresses. Ainsi la quantité du trafic MLD peut être minimisée.

Rapports d'abonnements MLD non-sollicités

Les changements d'état des hôtes sont notifiés par des messages non-sollicités:

- Pour souscrire à une adresse multicast spécifique, un hôte envoie un rapport

d'abonnement non-sollicité;

- Pour cesser d'écouter sur une adresse multicast, l'hôte 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 l'hôte était le dernier concerné par l'adresse multicast sur le lien;
- La résiliation rapide est aussi une possibilité offerte par MLDv1. L'hôte 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.

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.

Découverte des voisins

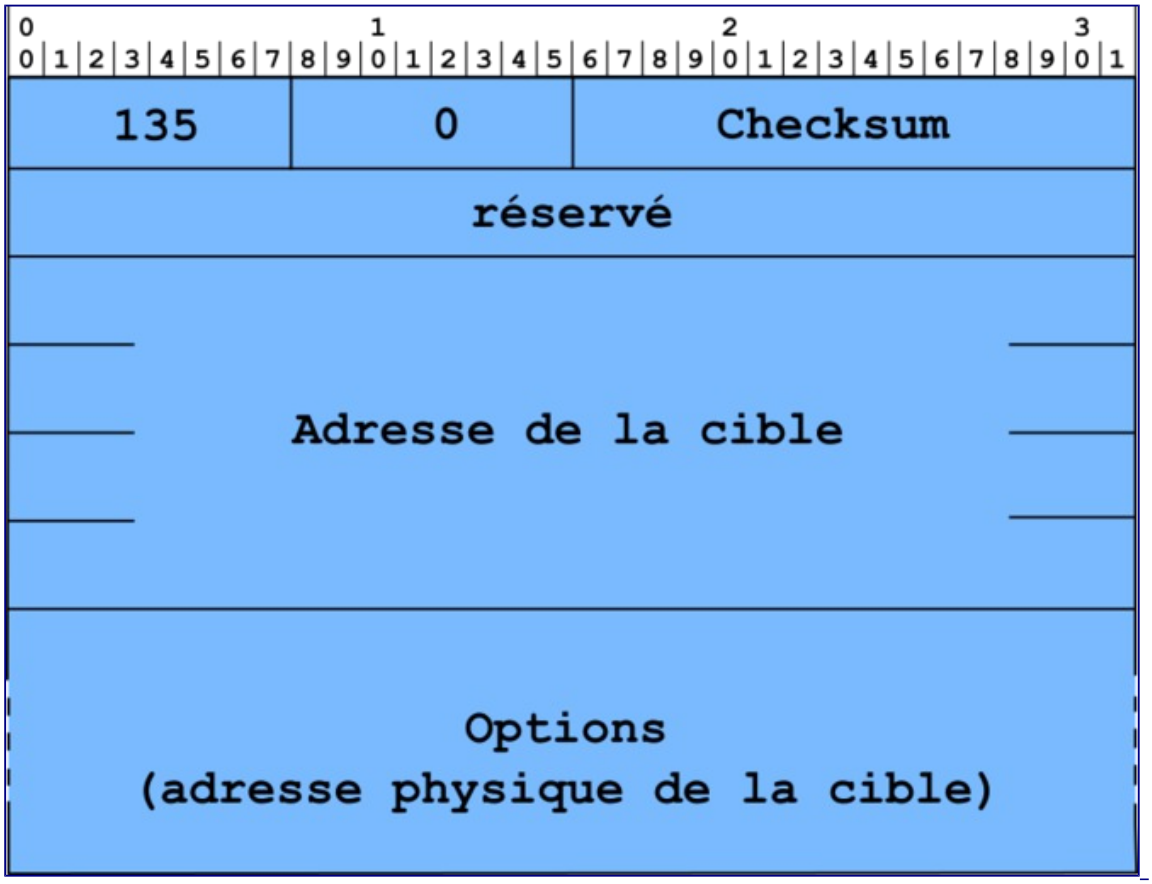
Cette fonction permet à 2 équipements connectés sur le même réseau de se découvrir l'un et l'autre et d'échanger des informations de configuration. La découverte des voisins est principalement mise en oeuvre dans 2 cas d'usage:

- La détermination de l'adresse physique d'un équipement à partir de son adresse IP
- La détection d'adresses IP dupliquées

Cette fonction de découverte des voisins est réalisée en IPv6 à travers 2 messages ICMPv6: Sollicitation d'un voisin (Neighbor Sollicitation ou NS) et Annonce d'un voisin (Neighbor Advertisment ou NA).

Détails des messages mis en oeuvre

Message Sollicitation d'un voisin

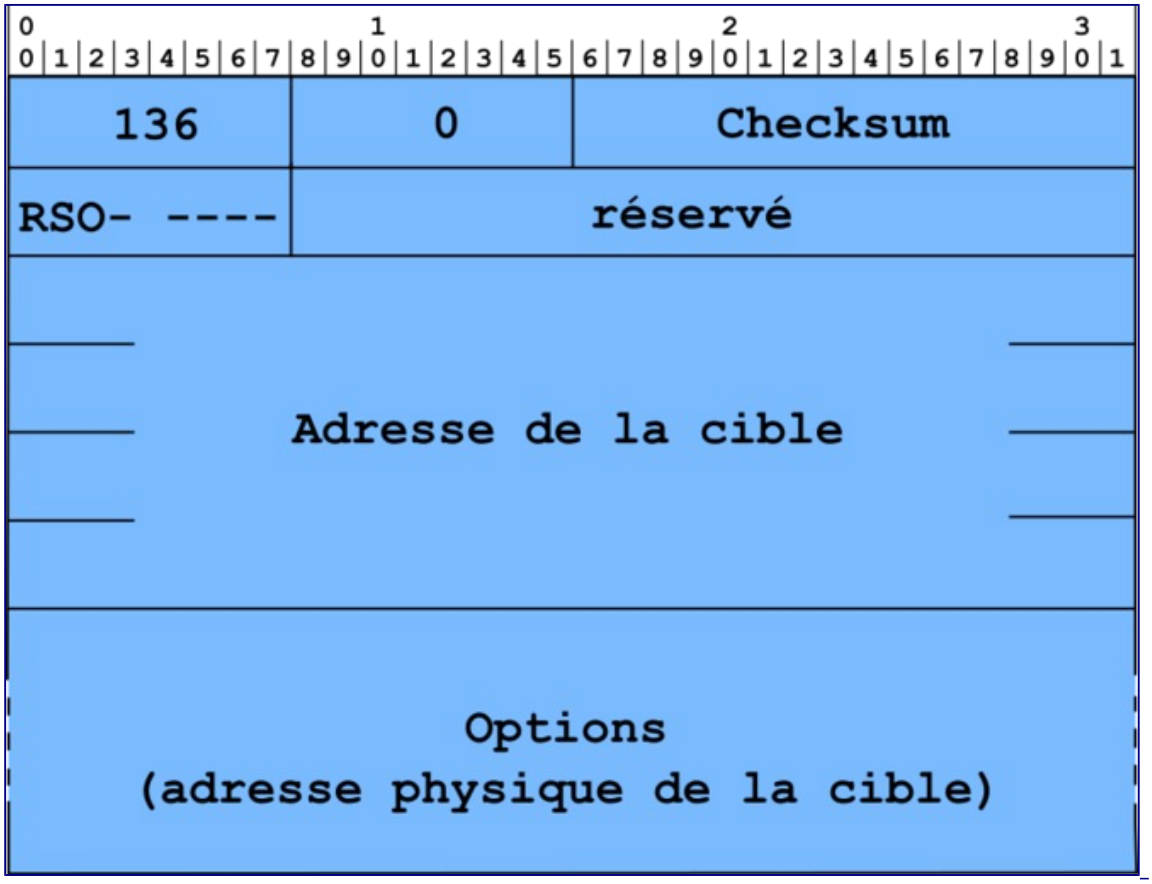


Ce message (cf. figure Format des paquets de sollicitation d'un voisin) permet d'obtenir des informations d'un équipement voisin, c'est-à-dire situé sur le même lien physique (ou connecté via des ponts). Le message peut lui être explicitement envoyé ou émis sur une adresse de diffusion. Dans le cas de la détermination de l'adresse physique, il correspond à la requête ARP du protocole IPv4.

Le champ adresse source du paquet IPv6 contient soit l'adresse locale au lien adresse lien-local, soit une adresse globale, soit l'adresse non spécifiée. Le champ destination contient soit l'adresse de multicast sollicité correspondant à l'adresse recherchée, soit l'adresse de l'équipement (dans le cas d'une détection d'inaccessibilité des voisins, NUD)

Le champ adresse de la cible contient l'adresse IPv6 de l'équipement cherché. Le champ option contient en général l'adresse physique de la source.

Message Annonce d'un voisin



Ce message (cf. figure Format des paquets d'annonce d'un voisin) 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 pour le protocole IPv4.

- Le bit R mis à 1 si l'émetteur est un routeur. Ce bit est utilisé pour permettre la détection d'un routeur qui redevient un équipement ordinaire.
- 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 équipements, en particulier la table contenant les adresses physiques.
- Le champ adresse de la cible contient, si le bit S est à 1, la valeur du champ adresse de la cible de la sollicitation auquel ce message répond. Si le bit S est à 0, ce champ contient l'adresse IPv6 lien-local de l'équipement émetteur.
- L'option adresse physique de la cible contient l'adresse physique de l'émetteur.

Fonctionnement de la résolution d'adresse physique

Nous allons étudier le cas où un noeud cherche à joindre un autre noeud situé sur le même réseau.

```
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
```

Sollicitation du voisin

Avant de pouvoir émettre un paquet IPv6 sur le réseau, l'émetteur a besoin de connaître l'adresse physique de l'équipement destinataire ou du routeur par défaut. Il utilise pour cela le protocole de découverte des voisins et émet une trame de sollicitation d'un voisin.

```
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
```

Dans l'en-tête IPv6, l'adresse de la source est l'adresse globale de l'interface d'émission. On aurait pu penser que l'émetteur utilisait l'adresse locale au lien comme adresse de la 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 entre adresse IPv6 et adresse physique, puisque ce dernier trouvera dans la suite du datagramme l'adresse physique de l'émetteur.

L'adresse de destination est l'adresse de multicast sollicité associée à l'adresse recherchée et l'adresse Ethernet de destination est l'adresse associée (cf. RFC 2464).

L'en-tête ICMPv6 contient dans le champ cible l'adresse IPv6 de la machine dont l'adresse physique est recherchée. On peut remarquer que les trois derniers octets correspondent au groupe de multicast de l'en-tête IPv6. Le champ option contient l'adresse physique de l'émetteur de la requête.

Annonce du voisin

La machine ganesha, qui écoute sur tous les groupes multicast sollicité associés à ses

adresses, reçoit le message de sollicitation de voisin, reconnaît dans la cible une de ses adresses IPv6, et répond.

```
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
```

L'adresse source utilisée est locale au lien. Le bit R indique que l'équipement 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'information est ensuite enregistrée dans un cache du système de l'équipement émetteur, appelé 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 périodiquement grâce à un protocole de découverte de non-joignabilité des voisins (Neighbor Unreachability Discovery), basé sur ces mêmes messages.

Fonctionnement de la détection d'adresse dupliquée

Pour vérifier l'unicité des adresses lien-local ou unicast qui viennent d'être configurée manuellement ou automatique sur leurs interfaces, les machines doivent exécuter un algorithme de Détection d'Adresse Dupliquée (DAD) avant de les utiliser. 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'autoconfiguration s'arrête et une intervention humaine devient obligatoire. 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 est assignée à une interface uniquement pour recevoir les messages de sollicitation et d'annonce d'un voisin. Les autres messages reçus sont ignorés. 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:

- Un message annonce d'un voisin est reçu: l'adresse provisoire est utilisée comme adresse valide par une autre machine. L'adresse provisoire n'est pas unique et ne peut être retenue.

- Un message sollicitation d'un voisin est reçu dans le cadre d'une procédure DAD; l'adresse provisoire est également une adresse provisoire pour une autre machine. L'adresse provisoire ne peut être utilisée par aucune des machines.
- Rien n'est reçu au bout d'une seconde (valeur par défaut): l'adresse provisoire est unique, elle passe de l'état de provisoire à celle de valide et elle est assignée à l'interface.

A noter que cet algorithme 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 crucial au bon fonctionnement de la couche réseau. A l'échelle du lien, c'est à travers le protocole ICMPv6 que les noeuds se découvrent entre eux, vérifient l'unicité de leurs adresses et gèrent les abonnements aux groupes multicast. A l'échelle de l'Internet, ICMPv6 permet de retourner à l'émetteur d'un paquet des informations en cas de non-livraison du paquet.

Activité 32: Configurer automatiquement les paramètres réseau

Principe de l'auto-configuration

La précédente activité a présenté le mécanisme de découverte des voisins permettant à une station connectée à un réseau de récupérer automatiquement les adresses des autres stations du même réseau. C'est la même philosophie qui est mise en oeuvre pour permettre la configuration automatique des paramètres d'une interface réseau.

L'objectif de ce mécanisme est de réduire au maximum l'intervention humaine dans ce processus afin de permettre:

- à l'utilisateur d'avoir une connexion au réseau fonctionnelle dès le branchement de l'interface réseau
- à l'administrateur de centraliser la configuration sur un seul équipement qui se chargera de les propager aux stations.

Les informations nécessaires à la configuration d'une interface sont au minimum:

- des informations permettant de déterminer l'adresse IP ou la méthode permettant de l'obtenir
- la largeur du préfixe IP du réseau afin de déterminer les adresses IP appartenant au même réseau
- l'adresse de la passerelle à utiliser pour joindre les adresses qui ne sont pas sur ce réseau
- le serveur de noms à utiliser sur ce réseau

L'administrateur renseigne les informations communes pour un réseau sur un équipement. Les stations se configurant récupèrent ces informations pour déterminer leur configuration spécifique qui sera appliquée sur l'interface. La connexion au réseau sera alors effective pour l'utilisateur.

Ce mécanisme est donc prévu pour les équipements terminaux. Les équipements intermédiaires dans l'infrastructure, comme les routeurs, ne sont pas censés utiliser ce mécanisme et leur configuration est à la charge de l'administrateur.

Mécanismes mis en oeuvre

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

- La toute première étape consiste à créer l'adresse lien-local. Une fois l'unicité de cette adresse vérifiée, la machine est en mesure de communiquer avec les autres machines du lien.
- La machine doit ensuite acquérir les informations commune au réseau, 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,
- l'auto-configuration avec état.

Note: En l'absence de routeur sur le lien, la machine 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. La machine n'a pas une adresse avec une portée qui l'autorise à communiquer avec des machines autres que celles du lien.

- Les informations transmises par le routeur permettent de plus à la station de configurer sa table de routage.
- Enfin, toujours en fonction de la politique de configuration, la station va récupérer d'autres informations nécessaire à la configuration, dont notamment le serveur de noms.

La création de l'adresse lien-local

À l'initialisation de son interface, la station 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 car la machine doit vérifier l'unicité de cette adresse sur le lien au moyen de la procédure de détection d'adresse dupliquée, présentée dans l'activité précédente. Si la station détermine l'adresse lien-local n'est pas unique, l'auto-configuration s'arrête et une intervention manuelle est nécessaire.

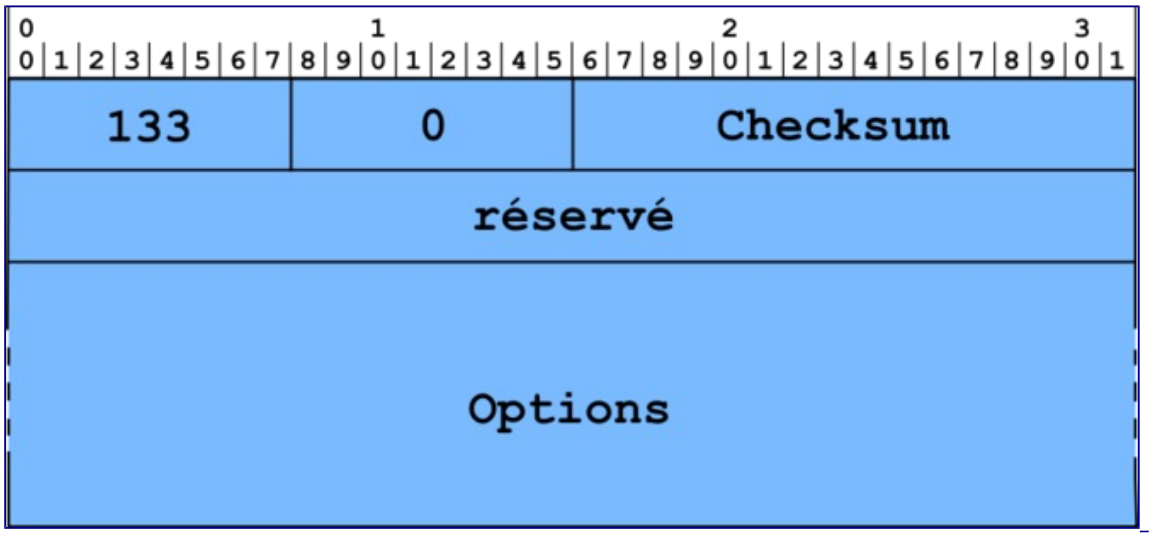
Une fois que l'assurance sur l'unicité de l'adresse lien-local est obtenue, l'adresse provisoire devient une adresse valide pour l'interface. La première phase de l'auto-configuration est achevée.

La fourniture des paramètres communs au réseau

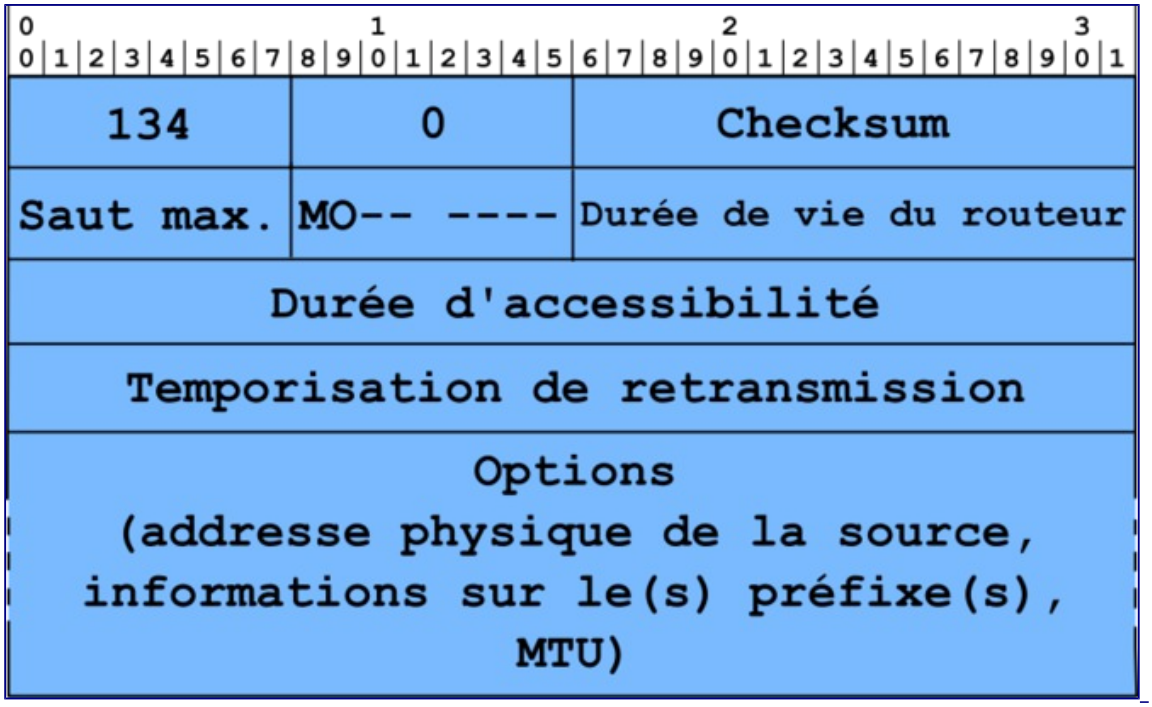
La seconde étape de l'auto-configuration consiste pour la station qui se configure à récupérer les informations communes au réseau qui sont fixées par l'administrateur. Ces informations sont localisées sur le (ou les) routeur(s) du réseau, équipement sous la responsabilité de l'administrateur et qui sera en charge de les propager aux stations.

La transmission de ces informations est assurée par un échange de messages ICMPv6:

- un message de sollicitation d'un routeur (Router Solicitation, ICMP `type` 133) envoyé par la station qui se configure
- un message d'annonce de routeur (Router Advertisment, ICMP `type` 134) envoyé par le routeur en réponse à la sollicitation d'une station, mais aussi périodiquement sur le réseau.



Le message de sollicitation d'un routeur (cf. figure Format des paquets de sollicitation du routeur) est émis par la station qui se configure afin d'obtenir les informations du routeur. Ce message est envoyé à destination de l'adresse IPv6 de multicast réservée aux routeurs sur le même lien `ff02::2` .



Le message d'annonce de routeur (cf. figure Format des paquets d'annonce du routeur) est émis périodiquement par les routeurs ou en réponse à un message de sollicitation d'un routeur par un équipement. Le champ adresse source contient l'adresse locale au lien du routeur, le champ destination contient soit l'adresse de l'équipement qui a émis la sollicitation, soit l'adresse de toutes les stations (`ff02::01`).

Ce dernier message 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 stations de la présence du (ou des) routeur(s) afin de confirmer le bon fonctionnement de leur

connexion.

Ce message contient un ensemble l'information propre au routeur et à la politique de configuration du réseau, puis ensuite un ensemble d'options en fonction de la politique de configuration. Parmi les information propre au routeur, le champ `durée de vie du routeur` donne, en secondes, la période pendant laquelle l'équipement annonçant effectuera les fonctions de routeur par défaut.

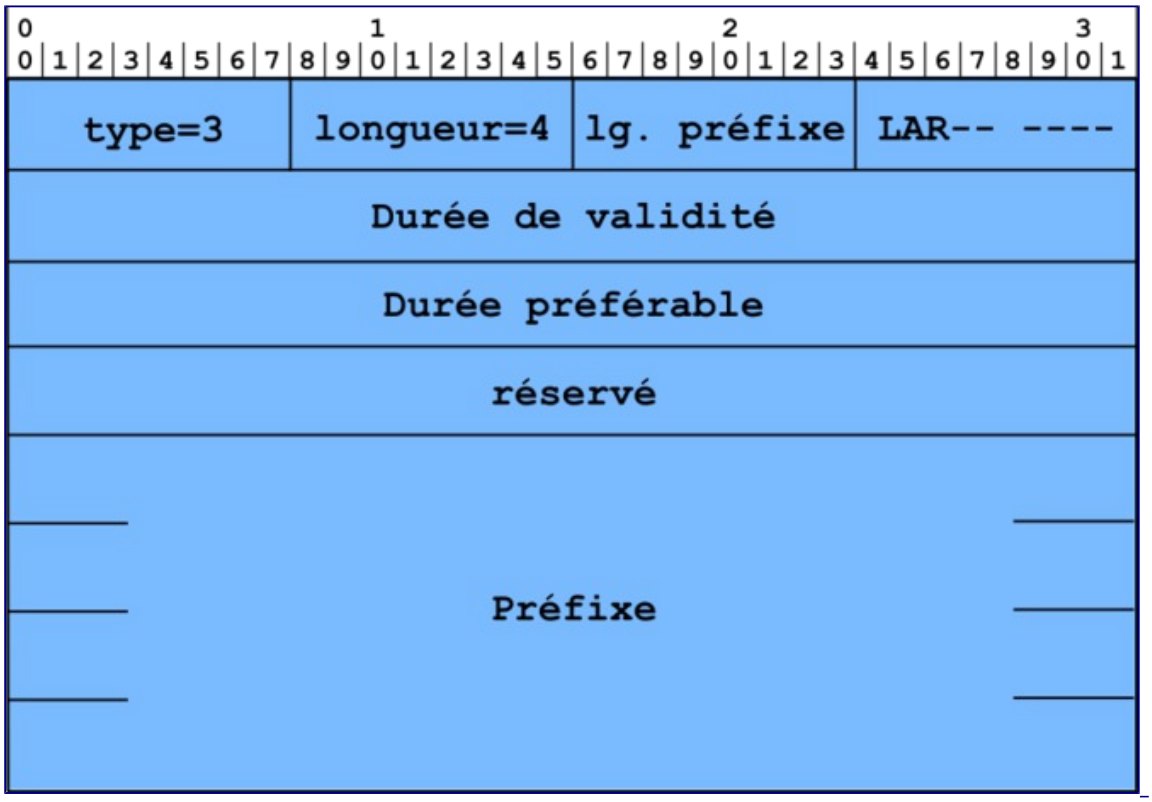
Le champ `durée d'accessibilité` indique la durée en millisecondes pendant laquelle une information de ce message contenue dans le cache de la station peut être considérée comme valide (par exemple, la table de correspondance entre adresse IPv6 et adresse physique). Au bout de cette période, un message de détection d'inaccessibilité (`Neighbor Sollicitation`) est émis pour vérifier la pertinence de l'information.

Le champ `temporisation de retransmission` donne en millisecondes la période entre deux émissions non sollicitées de ce message. Il sert aux autres équipements pour détecter une inaccessibilité du routeur.

La politique de configuration, qui doit indiquer à la station qui se configure les mécanismes à utilsier, est définie par 2 bits du message d'annonce de routeur. Le bit `M` (*Managed address configuration*) mis à 1 indique que l'équipement ne doit pas construire lui-même l'adresse à partir de son identifiant d'interface et des préfixes éventuellement reçus en option du même message. Il 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. Le bit `O` (*Other stateful configuration*) mis à 1 indique que la station doit interroger le serveur de configuration pour obtenir des paramètres autre que l'adresse.

L'auto-configuration sans état de l'adresse IP

Le principe de base de l'auto-configuration sans état de l'adresse IP est qu'une machine génère son adresse IPv6 à partir d'informations locales et d'informations fournies par le routeur. Le routeur fournit à la machine les informations sur le préfixe utilisé sur ce réseau au moyen d'une option incluse dans le message d'annonce de routeur.



Cette option contient les informations sur le préfixe pour permettre une configuration automatique des équipements. Le champ type vaut 3 et le champ longueur vaut 4. La figure Format de l'option information sur le préfixe donne le format de l'option:

- Le champ `lg.préfixe` indique combien de bits sont significatifs pour le préfixe annoncé dans un champ suivant.
- Le bit L indique, quand il est à 1, que le préfixe permet d'indiquer que tous les autres équipements partageant le même préfixe sont sur le même lien. L'émetteur peut donc directement les joindre. Dans le cas contraire, l'équipement émet le paquet vers le routeur. Si ce dernier sait que l'équipement émetteur peut joindre directement le destinataire, il émettra un message ICMPv6 d'indication de redirection.
- Le bit A indique, quand il est à 1, que le préfixe annoncé peut être utilisé pour construire l'adresse de l'équipement.
- Le bit R , indique, quand il est à 1, que le champ préfixe contient l'adresse globale d'un routeur «agent mère». Les bits de poids fort peuvent toujours être utilisés pour construire un préfixe.
- Le champ durée de validité indique en secondes la durée pendant laquelle le préfixe est valide.
- Le champ durée préférable indique la durée en secondes pendant laquelle une adresse construite avec le protocole de configuration sans état demeure «préférable».

Pour ces deux champs, une valeur de `0xffffffff` représente une durée infinie. Ces champs peuvent servir dans la phase de passage d'un fournisseur d'accès à un autre; c'est-à-dire d'un préfixe à un autre.

- Le champ réservé permet d'aligner le préfixe sur une frontière de mot de 64 bits.
- Le champ 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.

Comme pour la création de l'adresse lien-local, l'adresse unicast globale est obtenue en concaténant le préfixe avec l'identifiant de l'interface. Le préfixe provient ici du message d'annonce de routeurs et plus précisément de l'option «information sur le préfixe». Pour construire sont adresse, la station est ensuite libre de choisir l'identifiant d'interface créé à partir de l'adresse MAC ou généré selon un autre principe comme le tirage aléatoire. Profitant de la souplesse offerte par IPv6, la station peut de plus créer autant d'adresses qu'elle souhaite.

Les valeurs de durée préférable et de durée de validité contrôle le cycle de vie des adresses créées. Une fois le temps indiqué par la durée préférable écoulé depuis la réception du message d'annonce d'un routeur, l'adresse créé à partir du préfixe concerné passera de l'état préféré à l'état déprécié. Une fois le temps indiqué par la durée de validité écoulé depuis la réception du message d'annonce d'un routeur, l'adresse passera de l'état préféré ou déprécié à l'état invalide. Des messages d'annonces avec des valeurs spécifiques peuvent permettre par exemple de contrôler l'utilisation par les stations d'adresses construites à partir de certains préfixes.

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

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

Une station recevant un message d'annonce de routeur est donc supposé initier un dialogue avec un serveur DHCPv6 si ce message présent le bit `M` avec pour valeur 1. Mais ce comportement tel que prévu dans les standards n'est pas entièrement mis en oeuvre dans les systèmes d'exploitation actuel, et il est très souvent nécessaire d'expliciter l'usage de DHCPv6 à la station, 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 définir leurs tables de routage. Le routage des autres machines repose sur la notion de route pour le lien et de route par défaut.

La route vers les adresses du même lien est construite à partir des informations présentes dans l'option concernant ce préfixe réseau. En partant du préfixe ainsi que de la longueur du préfixe, la station peut déterminer les bits communs aux adresses IP connectées au même lien. L'acheminement des paquets à destination de ces adresses ne nécessitera pas la passerelle par défaut. Le noeud destinataire étant alors sur le même lien, l'adresse de niveau liaison (par exemple adresse Ethernet) sera récupérée par le protocole de découverte des voisins.

La route par défaut, utilisée pour atteindre le reste de l'Internet à travers le routeur du lien, est

configurée grâce à l'adresse lien-local contenue dans le champs source du message d'annonce de routeur. L'adresse physique de cet équipement est de plus contenue dans une des options du message. La station émettant un paquet vers une machine à l'extérieur du réseau utilisera donc cette adresse comme premier saut pour l'acheminement du paquet.

La découverte de la liste de serveurs DNS récursifs

L'auto-configuration IPv6 sans état, telle que spécifiée par l'IETF dans le RFC 4862 , n'a pas prévu de mécanisme de découverte automatique de la liste des serveurs DNS récursifs (cache). 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 mis longtemps (six ans environ) à passer en RFC (RFC 3315) et le besoin de découverte des serveurs DNS récursifs s'est accentué davantage. En effet, afin de renforcer le déploiement d'IPv6, la communauté IPv6 s'était vite trouvée dans l'urgence de mettre en oeuvre un mécanisme de découverte automatique du DNS avec ou sans DHCPv6 (qui était justement près de sortir).

Trois propositions ont ainsi vu le jour dans le cadre des travaux les groupes «ipv6», «dhc» et «dnsop» de l'IETF. C'est le groupe dnsop qui a pris en charge les débats sur ces propositions. Les co-auteurs de ces 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.

Voici maintenant un résumé des trois propositions:

- **Proposition 1: mécanisme à base de DHCP** : deux solutions légèrement différentes ont été proposées. Elles proposent toutes les deux d'utiliser la même option «DHCPv6 DNS Recursive Name Server» spécifiée dans le RFC 3646 :
  - découverte via un serveur DHCPv6 complet (RFC 3315 : c'est-à-dire qui alloue dynamiquement les adresses IPv6;
  - découverte du DNS via un serveur DHCPv6-lite (RFC 3736): celui-ci n'alloue pas d'adresses IPv6 mais il est simplement chargé d'informer les clients des différents paramètres à utiliser (DNS récursif, serveur NTP, serveur d'impression, ...);
  - Dans les deux cas, si l'équipement est en double pile et s'il est configuré à la fois avec DHCPv4 (pour IPv4) et avec DHCPv6 (pour IPv6), il faut définir une politique d'arbitrage entre les deux listes de serveurs DNS récursifs obtenues si celles-ci sont incohérentes;
- **Proposition 2: mécanisme à base d'annonce de routeur (RA)** : cette proposition, spécifié dans le RFC 6106 et appelé ND RDNSS, apporte un complément à l'auto-configuration sans état (RFC 4862) et consiste à surcharger l'annonce du routeur, en tant que message de la découverte des voisins (RFC 4861) par l'information DNS nécessaire. Ce mécanisme est
- **Proposition 3, mécanisme à base d'adresses anycast bien connues** (*Well-known*

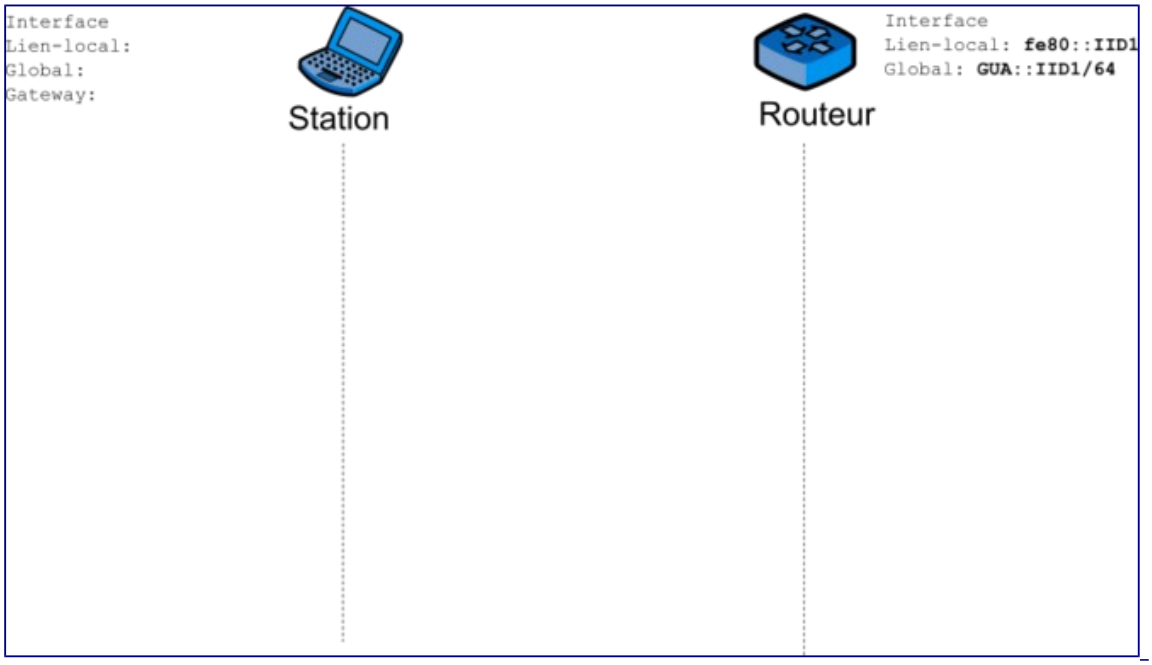
anycast addresses): des adresses IPv4 et IPv6 anycast qui seraient connues par tous les clients et pré-configurées automatiquement par le logiciel d'installation du système d'exploitation de l'équipement. Cette proposition semble avoir été abandonnée et n'a pas été reprise dans un autre document de spécification.

L'information du support des mécanismes DHCPv6 et ND RDNSS dans les différents système d'exploitation est actualisé sur cette page Wikipedia .

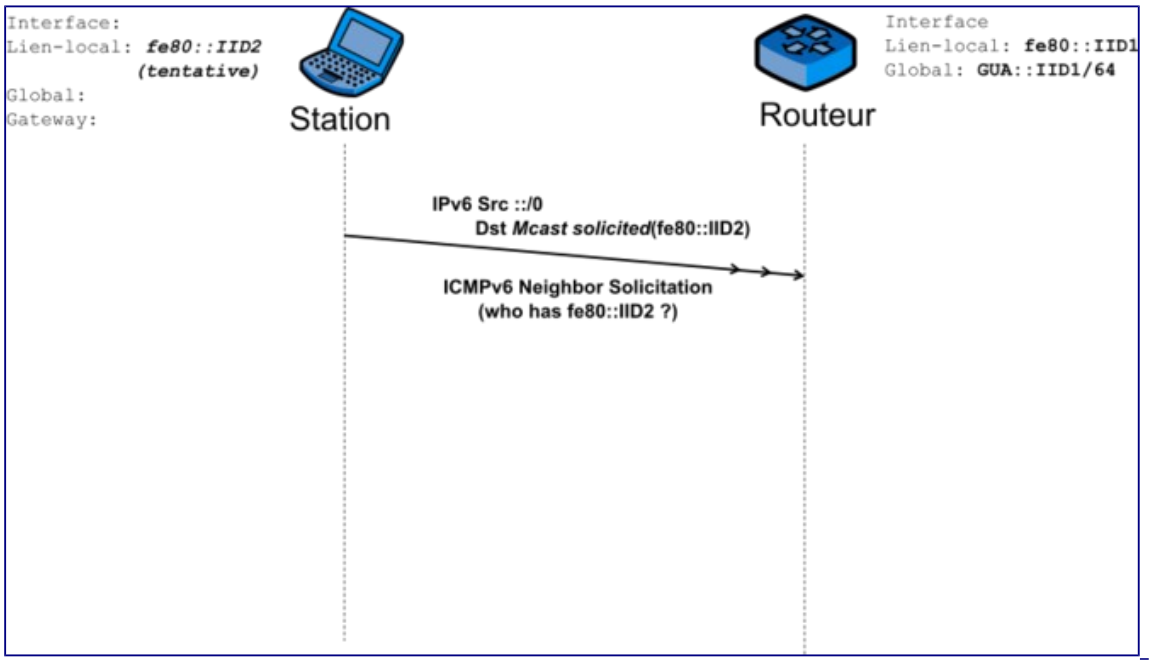
Exemple de configuration automatique

Nous allons illustrer ici les différentes étapes de l'auto-configuration et les messages échangés entre la station et le routeur du lien.

Au préalable au rattachement de la station au réseau, le routeur du lien est configuré avec le préfixe IPv6 à utiliser sur ce lien. La station, à l'activation de l'interface réseau, crée une adresse lien-local provisoire à partir de l'adresse matérielle de celle-ci.



Mais cette adresse est provisoire. Afin de vérifier si cette adresse est unique, la station débute l'algorithme de détection d'adresse dupliquée (DAD), . Comme décrit dans l'activité précédente, elle émet un message de sollicitation d'un voisin à l'adresse multicast sollicité associée à son adresse provisoire. Son adresse de source est indéterminée car son état est encore provisoire pour le moment et ne sert que pour la réception. L'adresse dont l'unicité est vérifiée est placée dans le champ cible.

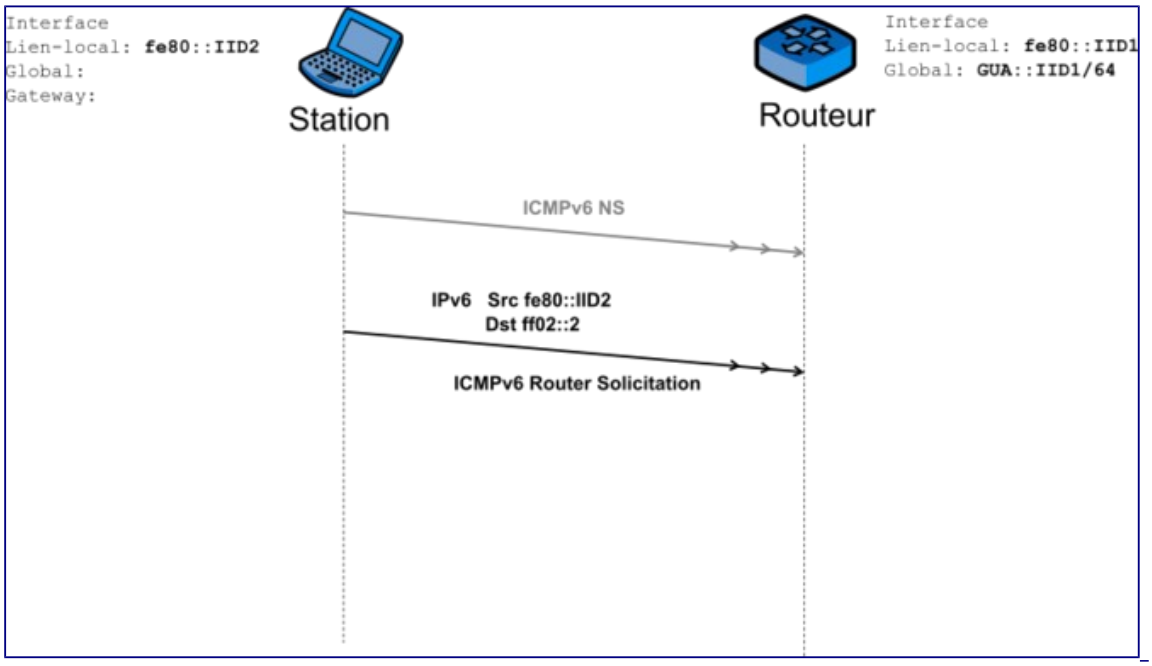



```
Ethernet Src: 8:0:20:a:aa:6d
 Dst: 33:33:ff:a:aa:6d
 Type: 0x86dd
IPv6
 Version: 6 Priorité: 0xf0 Label: 000000
 Longueur: 24 octets (0x0018) Protocole: 58 (0x3a, ICMPv6)
 Nombre de sauts: 255 (0x0ff)
 Source: ::
 Desti.: ff02::1:ff0a:aa6d (multicast sollicité associé à l'adresse cible)
ICMPv6
 Type: 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 00
0010: 00 00 00 00 00 00 00 00 ff 02 00 00 00 00 00 00
0020: 00 00 00 01 ff 0a aa 6d|87 00 fe 37 00 00 00 00
0030: fe 80 00 00 00 00 00 00 0a 00 20 ff fe 0a aa 6d
```

Si aucune réponse n'est donné à ce message dans les 2 secondes suivant sa diffusion, la station considère son adresse lien-local comme unique. 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.

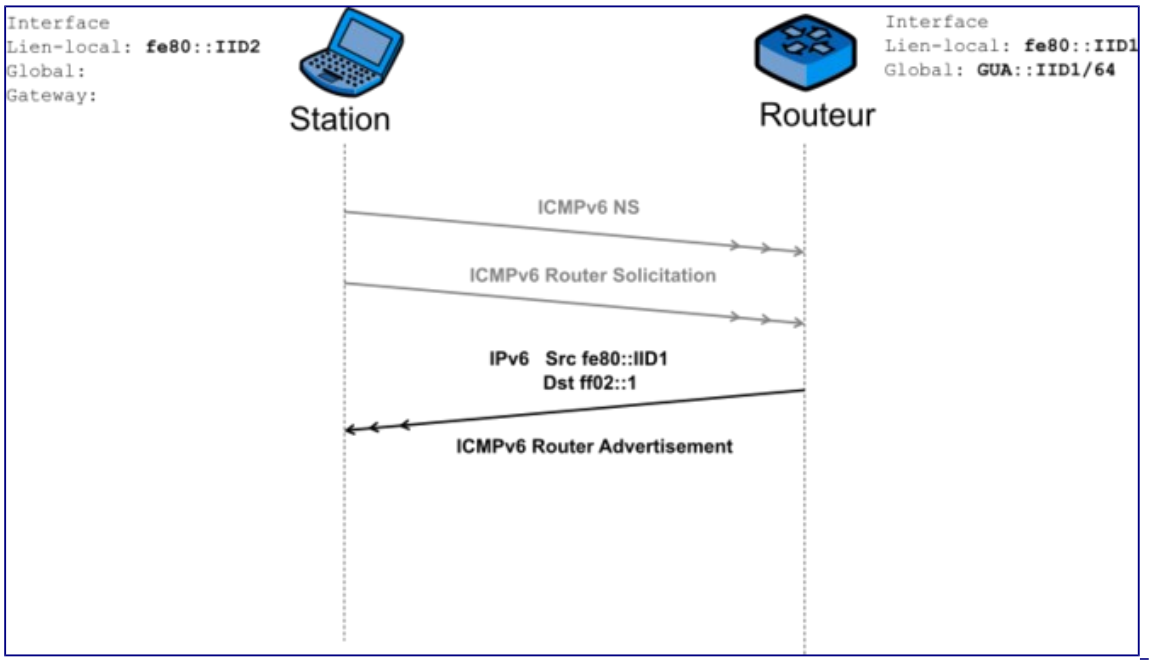
Cette première étape terminée, la station possède donc une adresse lien-local lui permettant de communiquer avec les équipements présents sur le même réseau. Elle va chercher maintenant à obtenir les informations de configuration afin de pouvoir communiquer avec des équipements en dehors du réseau. La station émet pour cela un message de sollicitation de routeur à tous les routeurs du lien en utilisant l'adresse multicast correspondante: `ff02::2` .




```
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 00
0010: 0a 00 20 ff fe 0a aa 6d ff 02 00 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 annonce de routeur est reçu par la machine se configurant. Elle y trouve les bits M , O et les informations sur les préfixes du lien.



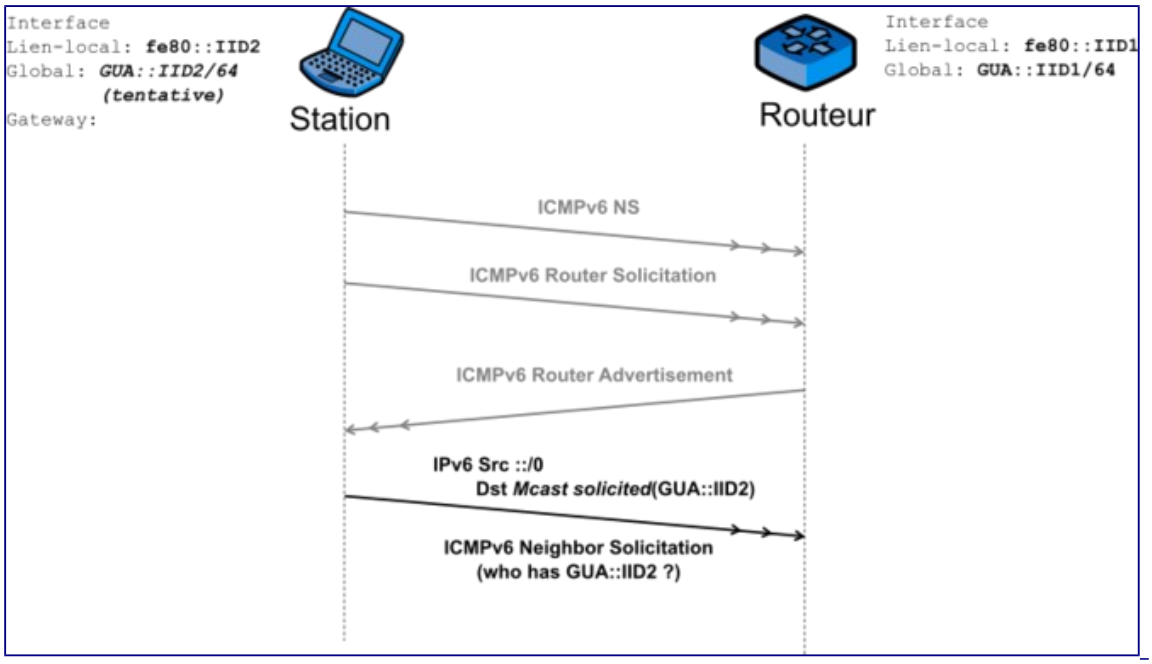

```
Ethernet Src: 1a:0:20:c:7a:34
 Dst: 33:33:0:0:0:1
 Type: 0x86dd
IPv6
 Version: 6 Priorité: 0xf0 Label: 000000
 Longueur: 56 octets (0x0038) Protocole: 58 (0x3a, ICMPv6)
 Nombre de sauts: 255 (0x0ff)
 Source: fe80::1800:20ff:fe0c:7a34 (ganesha, lien-local)
 Desti.: ff02::1 (multicast, tous les noeuds 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, -1 (infinie)
  Préfixe: 2001:db8:12:3::/64

0000: 6f 00 00 00 00 38 3a ff fe 80 00 00 00 00 00 00
0010: 18 00 20 ff fe 0c 7a 34 ff 02 00 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
```

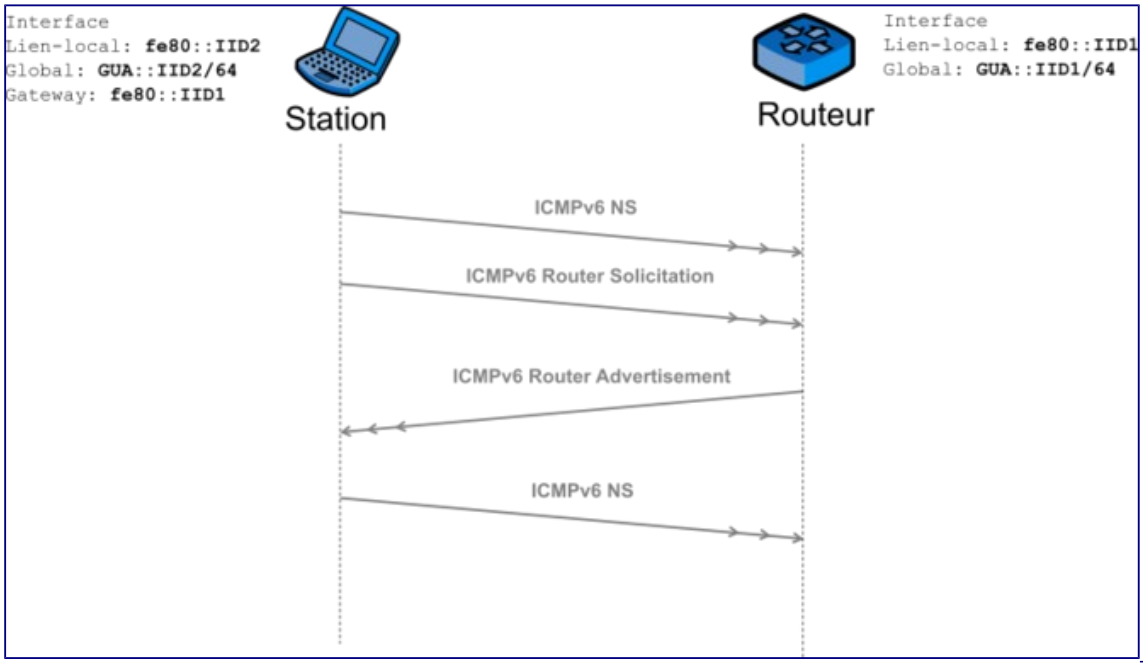
Le message annonce de routeurs est émis vers le groupe de tous les noeuds IPv6 du lien. Le drapeau A étant positionné, le préfixe annoncé peut alors servir à la construction de l'adresse unicast globale. La durée de validité de cette adresse n'est pas limitée. La station se construit donc l'adresse `2001:db8:12:3:a00:20ff:fe0a:aa6d` à partir du préfixe et de l'identifiant d'interface issu de l'adresse MAC. D'autres adresses peuvent être construites à partir des identifiants d'interface aléatoires.

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.



Une fois l'unicité de cette adresse vérifiée, la station configure dans sa table de routage l'adresse lien-local du routeur comme passerelle par défaut. Elle est désormais capable de communiquer avec des équipements situés au delà de ce routeur. D'autres informations comme notamment le DNS qui peuvent être communiquées dans le message d'annonce de routeur sont elles aussi utilisées pour la configuration de la station.



Conclusion

La configuration automatique des paramètres réseau IPv6 permet une connectivité fonctionnelle de l'interface réseau dès son branchement. Ce mécanisme ne nécessite aucune intervention humaine et facilite donc grandement la vie des administrateurs réseau. La configuration est centralisée sur le routeur du réseau qui devient l'équipement crucial à son bon fonctionnement. Les stations sont ensuite autonomes pour récupérer ces paramètres et décider de leur adresse IPv6 afin de se configurer.

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

Activité 33: Contrôler la configuration réseau par DHCPv6

Introduction

L'auto-configuration à états utilise un serveur pour allouer des adresses IPv6 et/ou des paramètres de configuration à des machines IPv6. Elle réduit les efforts de configuration des machines IPv6, tout comme l'auto-configuration sans état. Elle offre, à la différence de l'auto-configuration 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 et sans états, ne sont pas exclusives et peuvent coexister dans un même environnement.

Une machine peut, par exemple, obtenir son adresse unicast globale par auto-configuration sans état et obtenir les informations relatives au service de noms (DNS) par l'auto-configuration à états.

Tout le mécanisme d'auto-configuration à états 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 3315 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 standard a cependant fait l'objet de nombreux débats. DHCP est un élément important du fonctionnement d'un réseau. En conséquence, la parution tardive d'un standard finalisé a entraîné une adoption lente, de son support et de son déploiement.

Présentation générale du protocole DHCPv6

Le protocole DHCPv6 est un protocole de niveau application. Il fonctionne conformément au modèle client-serveur.

Ces échanges utilisent une pile de communication UDP/IPv6/réseau physique. Ils font intervenir quatre types d'entités: les clients, les serveurs, les relais et les interrogateurs (Requestor).

Les clients sollicitent les serveurs pour obtenir des adresses IPv6 et/ou des paramètres de configuration du réseau. Ils communiquent directement avec les serveurs DHCPv6 lorsqu'ils se trouvent sur le même lien.

Les relais relaient les messages des clients et les acheminent vers les serveurs lorsque clients et serveurs ne se trouvent pas sur les mêmes liens. Ils relaient également dans ce cas les messages des serveurs destinés aux clients.

Les administrateurs utilisent les interrogateurs pour obtenir des informations relatives aux paramètres de configuration des clients de leurs serveurs DHCPv6.

Les échanges DHCPv6 se composent de requêtes et de réponses. Il existe deux types de messages: ceux échangés entre client et serveurs et ceux échangés, soit entre relais, soit entre relais et serveurs.

Pile de communication utilisée par DHCPv6

DHCPv6 utilise le protocole de transport UDP. Il utilise deux ports: 546 pour le client et 547 pour les serveurs ou les relais.

Un client DHCPv6 envoie les requêtes depuis le port 546 et les envoie vers le port 547.

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 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.

Les messages UDP sont encapsulés dans des datagrammes IPv6.

En fonction des indications du serveur DHCPv6, les communications peuvent, au niveau IPv6, se faire en point à point ou en diffusion sélective (multicast) pour la découverte des serveurs DHCPv6 d'un site.

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.

Présentation des entités du protocole DHCPv6

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 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 être sur le même lien), soit avec un relais DHCPv6, soit avec le serveur DHCPv6. Il ignore l'existence des relais DHCPv6. Il émet des messages DHCPv6 qu'il envoie au serveur DHCPv6. Il a l'impression de communiquer directement avec le serveur DHCPv6.

Le serveur DHCPv6 centralise les paramètres de configuration des équipements du réseau. Il peut ou non se trouver sur le même lien qu'un client DHCPv6. Il gère la configuration des clients

situés sur le même lien ou sur des liens différents. Lorsqu'il ne se trouve pas sur le même lien que son client, les échanges DHCP transitent par un ou plusieurs relais DHCPv6.

Les relais DHCPv6 sont des équipements éventuellement 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. Leur action est transparente pour les clients. Ils transmettent éventuellement des informations locales aux serveurs. 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 les interrogateurs pour demander des informations relatives aux clients à un serveur DHCPv6.

Un administrateur peut, ainsi, obtenir des informations relatives aux baux d'un client ou à la machine qui utilise une adresse à un instant donné ou encore pour connaître 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 (auto-configuration sans état). Il a alors besoin pour communiquer d'informations telles que l'adresse IPv6 du serveur DNS (le service de nommage convertit, en particulier, les noms en adresses IP).

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 états (*stateful*), alloue une ou plusieurs adresses IPv6 au client. Ces adresses font l'objet d'un contrat de location temporaire, un bail de location. 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 états.

De plus, un serveur qui redémarre, relit son fichier des baux. Il peut alors exactement reprendre son activité là où il l'avait laissée.

Fonctions des Messages du protocole DHCPv6

Cette partie présente 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 transaction et des associations d'identités.

Les serveurs DHCPv6 utilisent les identificateurs de transaction pour associer leurs réponses aux demandes correspondantes des clients.

Les identificateurs de transaction changent pour chaque transaction et sont globalement uniques.

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

SOLICIT / ADVERTISE

Un client utilise le message SOLICIT (champ type 1) pour localiser les serveurs configurés pour allouer des adresses et/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). Messages de demande d'information de configuration

REQUEST / REPLY

Un client utilise le message REQUEST (champ type 3) pour demander des adresses et/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, REQUEST, RENEW, REBIND reçu d'un client DCHPv6.

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 que ces paramètres sont adaptés au lien auquel le client 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 SOLICIT,

REQUEST, 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 donné.

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
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 et/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 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-FORW	12	Relai	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-REPL	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 de message relayé de ce message pour le lui remettre.

Pour en savoir plus: 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

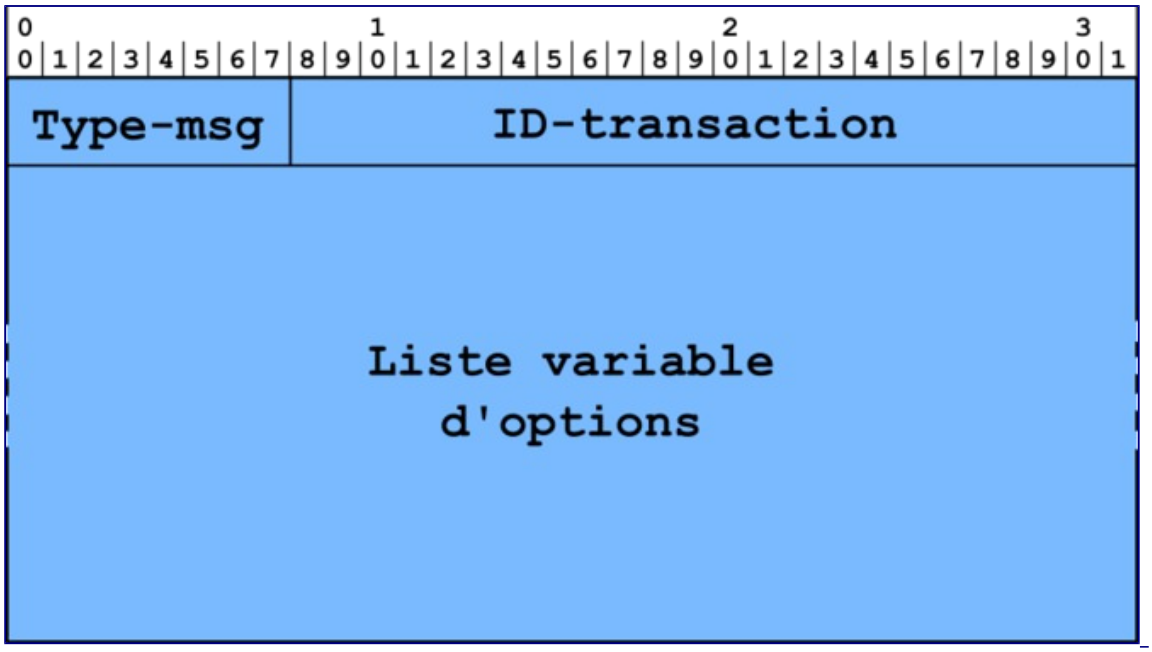
Le document RFC 3315 décrit l'ensemble des éléments du protocole DHCPv6. A l'instar de nombreux protocoles de l'Internet, le protocole d'échange d'informations est découplé de l'information elle-même. La nature des informations échangées peut donc changer et évoluer rapidement, sans impacter les mécanismes 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. 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.



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

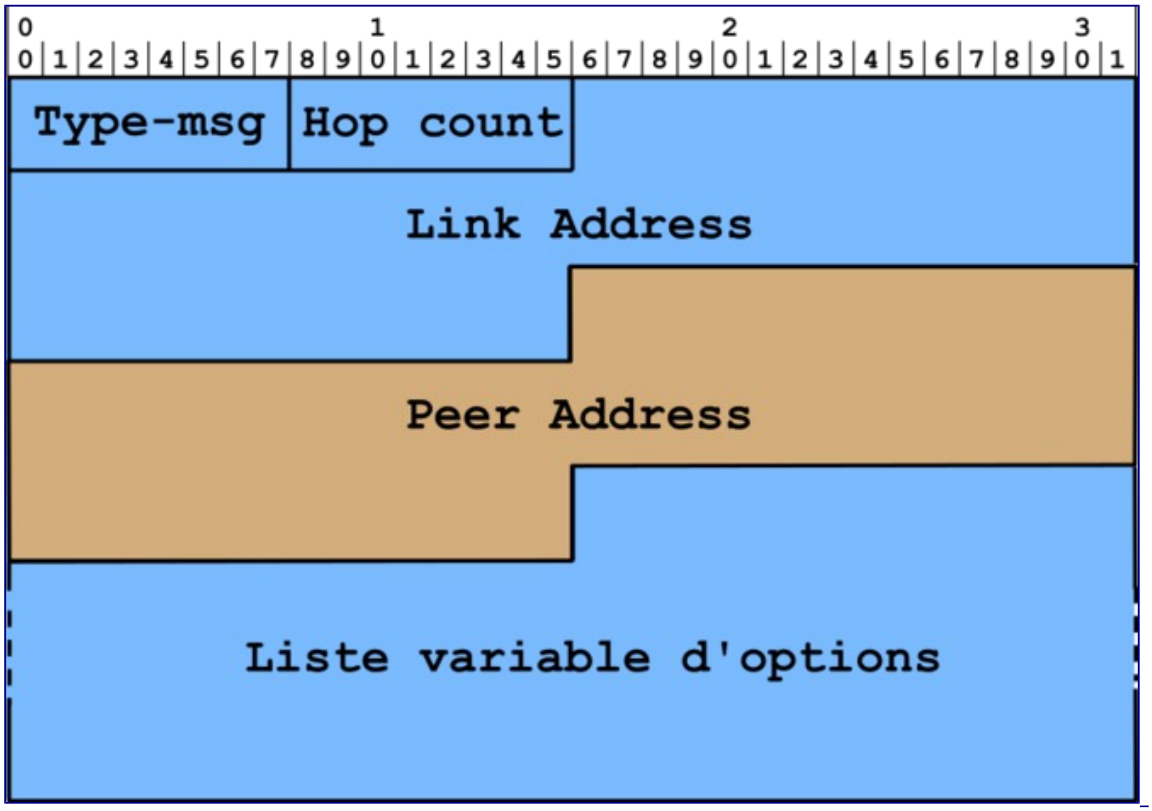
Id-transaction . L'identificateur de transaction identifie un échange (question-réponse). Il change pour chaque échange et est globalement unique. Il est codé sur 3 octets.

Option list . La liste des options du message est de taille variable. Elle correspond à une succession d'options rangées séquentiellement, les unes derrière les autres, 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. *Annexe1. Options du protocole DHCPv6* .

Structure des messages échangés entre relais et serveur DHCPv6

La structure des messages échangés entre relais et serveur est la suivante:



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 le 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 locale au lien désigne l'interface du relais émettrice du message (RELAY-FORWARD) ou destinataire du message (RELAY-REPLY). *Peer-address* . L'adresse du pair est une adresse globale ou locale au site. Elle identifie, pour chaque relais, l'interface du relais, côté client. Pour le dernier relais, dans le cas du transit d'un message du serveur vers le client, cette adresse identifie l'interface du relais derrière laquelle se trouve le client.

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 de relais contenu dans l'option de message de relais du message RELAY-REPLY reçu. 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 saut indique le nombre de relais traversés par ce message pour atteindre le serveur.

Link-address . L'adresse locale au lien d'un message RELAY-FORWARD est une adresse globale ou une adresse locale au site que le serveur utilise pour identifier le lien où se trouve le client (adresse du relais côté client). C'est l'adresse du relais, du côté du client.

Peer-address . L'adresse du pair est l'adresse IPv6 de l'interface depuis laquelle le relais a envoyé le message au serveur. C'est l'adresse du relais du côté du serveur.

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 saut 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

Le RFC 3315 définit 3types d'identificateurs unique DHCPv6 (DUID).

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 dépendant du serveur, celle-ci n'est pas utilisable par le serveur DHCP pour identifier un client. Le serveur identifie 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 son identifiant, même en cas de remplacement ultérieur de cette carte réseau.

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

Les DUID peuvent être générés selon 3 méthodes: combinaison d'une adresse physique et d'une horodate, dérivée d'un numéro de constructeur ou d'un numéro unique affecté par un constructeur, ou enfin, dérivée de l'adresse MAC d'une interface de réseau.

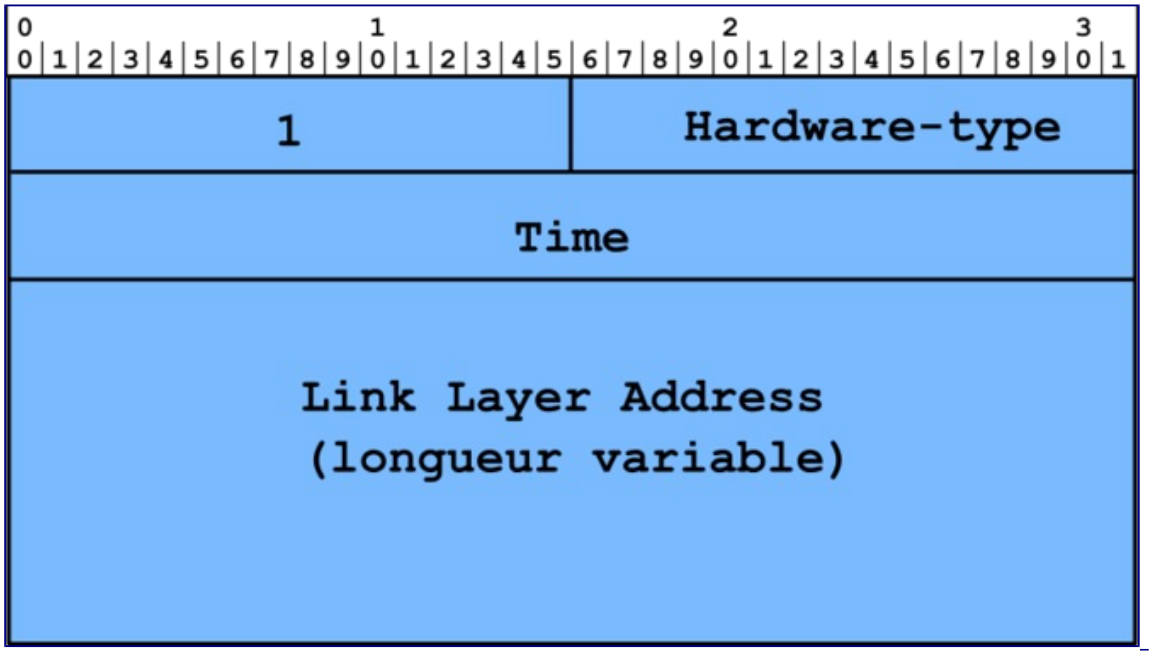
Les valeurs des champs type de DUID sont les suivantes.

- 1: Link-layer address plus time (DUID-LLT)
- 2: Vendor-assigned unique ID based on Enterprise Number (DUID-EN)
- 3: Link-layer address (DUID-LL)

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 donnée qui selon le mode de construction, contient des types de valeurs différents.

DUID construit à partir de l'adresse physique + horodate (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.

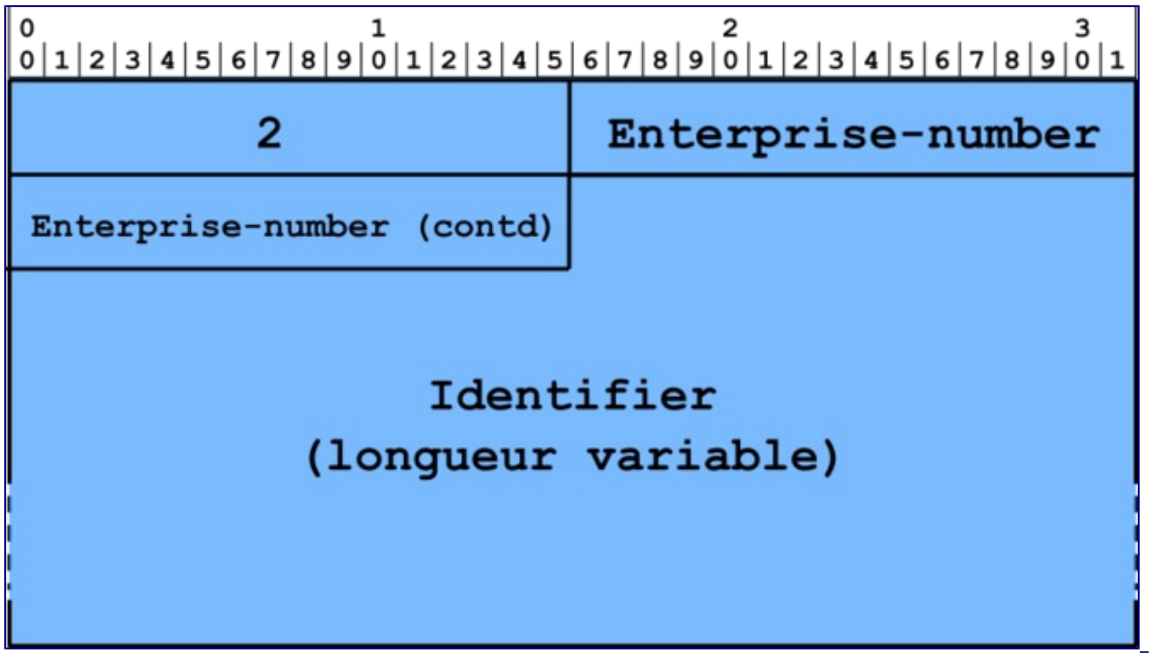
Le choix de l'interface dont on utilise l'adresse physique est indifférent tant que l'identification est unique. Le DUID doit être enregistré dans une mémoire non volatile et doit continuer à être utilisé, même en cas de remplacement ultérieur de l'interface qui a servi à le générer.

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 structure de l'option est la suivante:

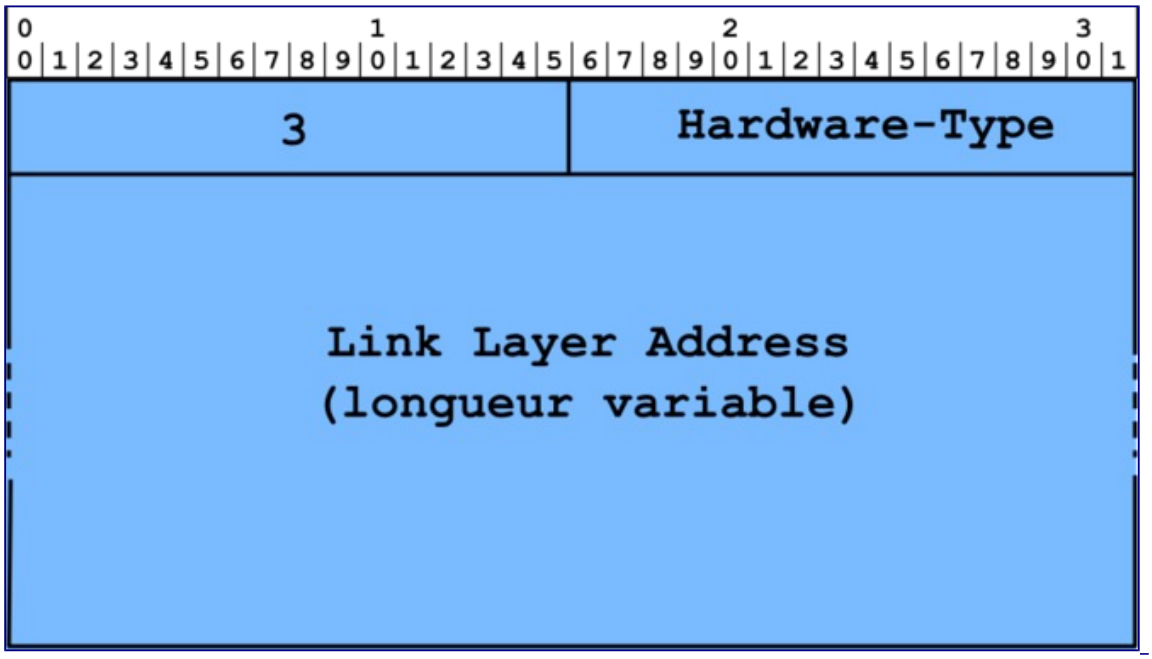


Le constructeur affecte généralement cet identificateur unique à l'équipement lors de sa construction. Il l'enregistre généralement dans une mémoire non volatile de l'équipement.

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 structure de l'option est la suivante:



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).

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 (voir le principe de la renumérotation).

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 et 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 domaine.

Options du protcole DHCPv6

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 structurée en plusieurs champs (voir Annexe 1. Options du protocole DHCPv6).

Chaque option est codée en format TLV: type, longueur, valeur, à savoir, le type de l'option, la longueur, en octet, du champ valeur du paramètre qui suit et le champ valeur du paramètre de configuration.

Le champ type de l'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.

La longueur totale en octet d'une option est donc souvent 4 + longueur de la valeur du champ longueur de l'option. Elle peut être plus importante si l'en-tête de l'option inclut des informations de taille fixe.

Le tableau qui suit présente les options du protocole DHCPv6, leur code et leur définition. L'annexe 1 *Annexe1. Options du protocole DHCPv6* présente leur structure.

Options de DHCPv6

Désignation	Code	Définition
OPTION_CLIENTID	1	Identification du client
OPTION_SERVERID	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 relay-forw ou relay-repl
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 aavec lui.
OPTION_STATUS_CODE	13	Indique le statut du message DHCPv6 qui transporte cette option.
OPTION_RAPID_COMMIT	14	Permet, dans un message solicit, à un client 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 (Solicit reply).
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 les client et 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 reconfiguration, si le client doit répondre par un message renew ou information-request.
OPTION_RECONF_ACCEPT	20	Indique à un serveur si le client accepte ou refuse les messages reconfigure ou annonce à un client qu'il peut ou non accepter les messages reconfigure.

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

Un client relié au même lien que le serveur 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 a priori ce client ignore l'adresse IPv6 du serveur, le client DHCPv6 envoie toujours (c'est le mode de fonctionnement par défaut) ce message à l'adresse

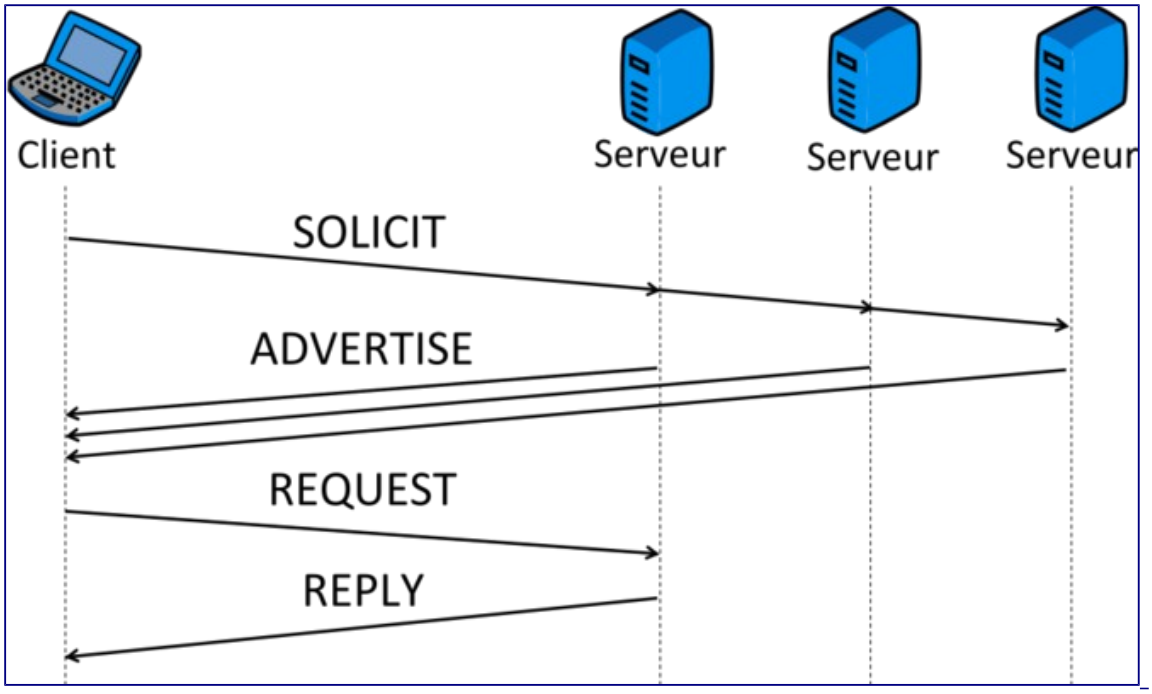
de diffusion sélective IPv6 *ALL_DHCP_And_ Relay_Agent* .

Les serveurs capables d'allouer des adresses au client répondent avec un message DHCPv6 ADVERTISE. Ils font une offre au client DHCPv6.

Le client, si plusieurs serveurs DHCPV6 sont disponibles, 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 sélectionné. Il envoie ce message à l'adresse de diffusion sélective *ALL_DHCP_And_ Relay_Agent* .

Tous les serveurs qui ont répondu à la demande du client savent ainsi si leur offre a ou non été retenue. Le serveur dont l'offre à été retenue, et lui seul, renvoie un message REPLY au client.

La figure ci-dessous résume les messages DHCPv6 échangés dans ce cas.



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 disponible.

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 diffusion sélective (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 transmettre 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. Ceci permet d'utiliser, au niveau 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.

Le client DHCPv6 envoie donc la trame Ethernet sur le lien, vers le serveur DHCPv6.

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'avions 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 au serveur. Le relais DHCPv6 envoie ensuite le message RELAY-FORWARD 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.

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.

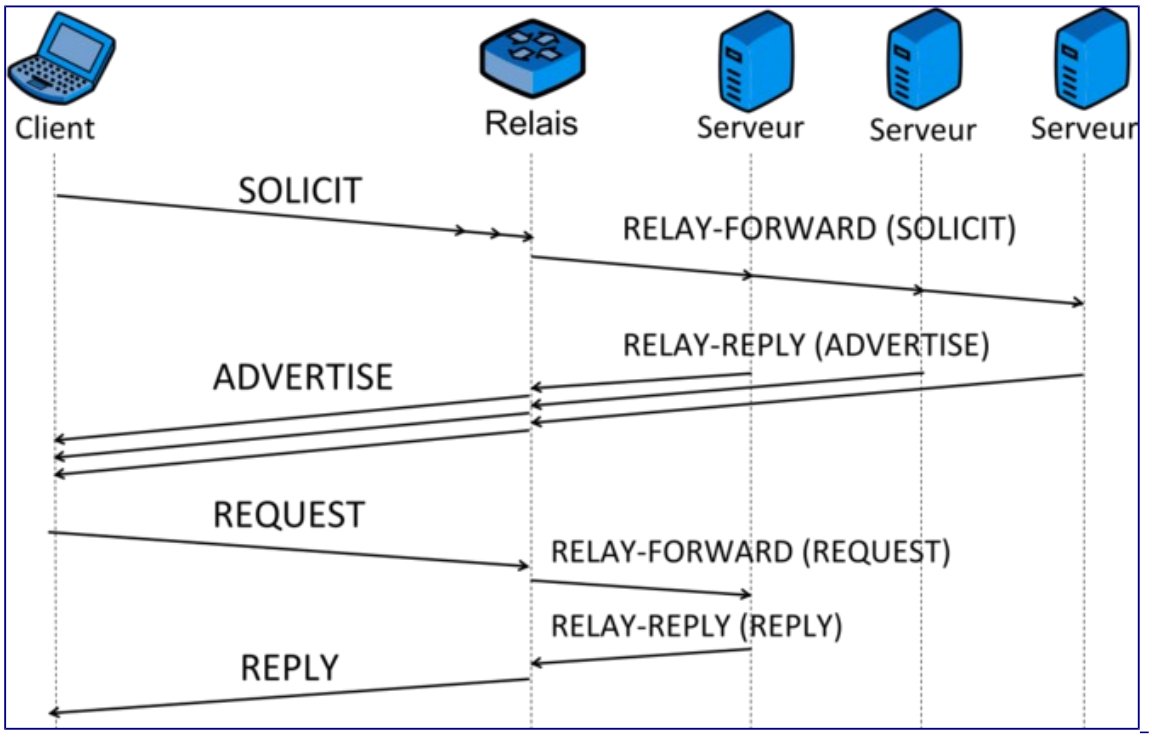
Notez que le message du client est recopié dans l'option message relayé message RELAY-FORWARD du premier relais DHCPv6 traversé.

Si le message traverse plusieurs relais, l'option message relayé du relais courant contient le message RELAY-FORWARD du relais précédent.

Lorsque 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 calculer. Le protocole DHCPv6 peut donc ainsi faire parvenir sa réponse du serveur au client.



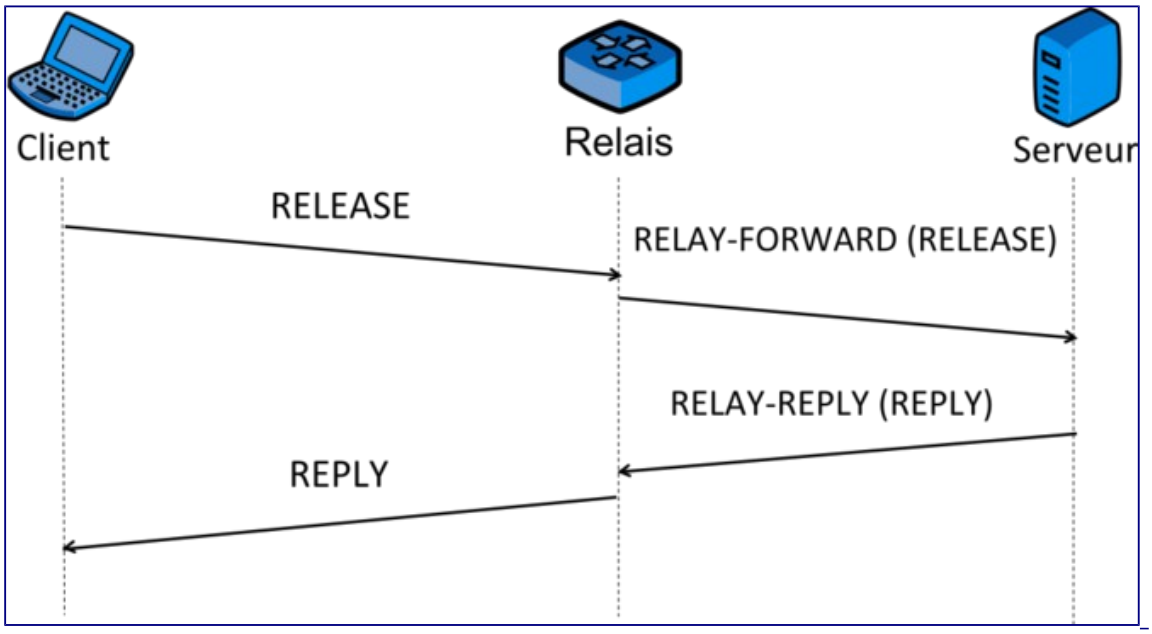
Après la phase d'acquisition de l'adresse IPv6, le client DHCPv6 vérifie (DAD) que l'adresse ipv6 allouée n'est pas déjà en service (DAD: détection d'adresse dupliquée). Il configure alors ses interfaces de réseau.

L'utilisateur qui travaille sur le client DHCPv6 peut alors accéder au réseau.

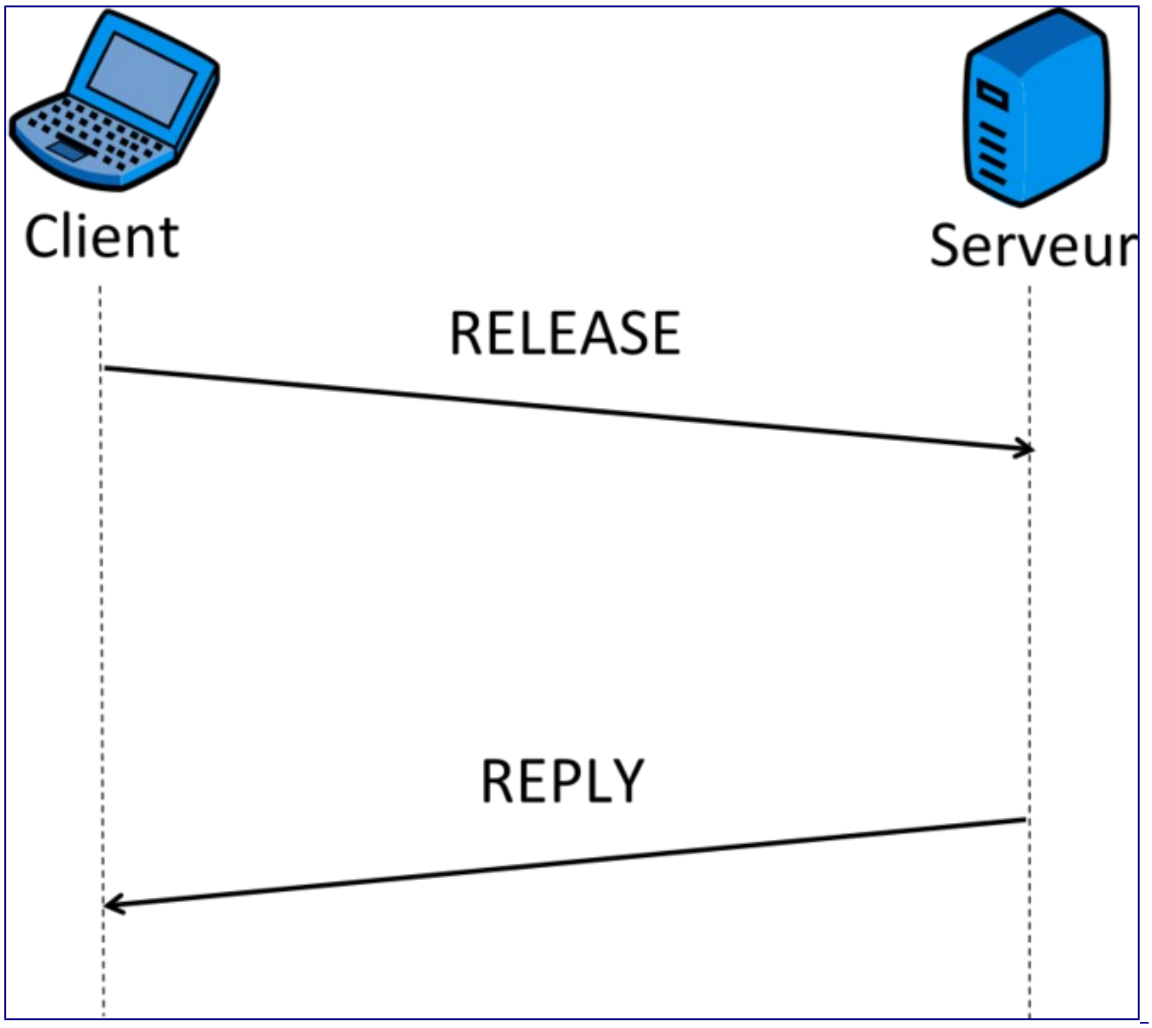
Le processus DHCPv6 client devient alors inactif jusqu'à ce que l'utilisateur qui travaille sur le client DHCPv6 ferme sa session et arrête le client. Il se réactive alors pour libérer (release) l'adresse IPv6 allouée.

Libération de l'adresse IPv6 par le client DHCPv6 avec présence d'un relais

Le processus d'arrêt normal du client DHCPv6 inclut la libération de l'adresse IPv6 allouée par le serveur. La figure ci-dessous présente la libération de l'adresse IPv6 en présence d'un relais.



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



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).

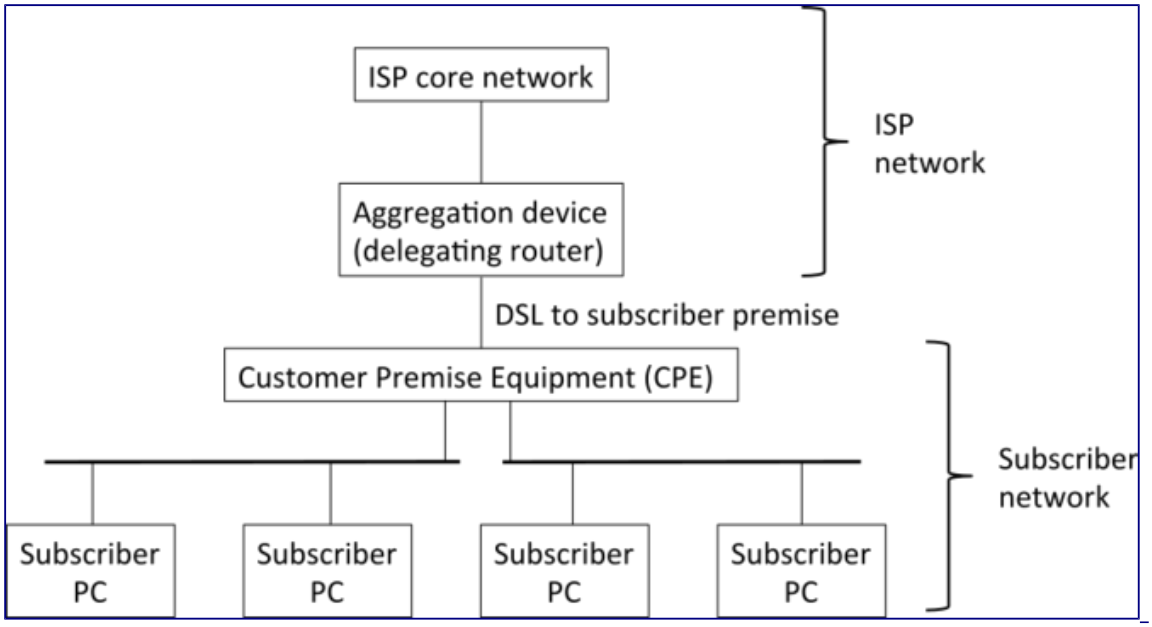
Notez que la délégation de préfixe à états est indépendante de l'allocation des adresses IPv6.

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 Internet) (ISP: Internet service Provider) qui alloue un préfixe au routeur d'accès d'un client (CPE: Customer Premise Equipment) reliant un réseau interne au réseau du FAI.

La figure ci-dessous présente un exemple où la délégation de préfixe à états est possible.



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 renummérotation 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.

Sur chaque machine, toutes les communications utilisant l'ancienne adresse sont préservées aussi longtemps que nécessaire (RENEW).

Toutes les nouvelles communications sont établies à l'aide de la nouvelle adresse.

Lorsque la dernière machine du réseau cesse d'utiliser son ancienne adresse, la renumérotation est terminée.

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 adresse en fonction de la date d'échéance cible.

Lorsque la date d'échéance arrive, aucune utilisation d'ancienne adresse n'est plus possible. TOutes les communications utilisant les anciennes adresses sont coupées.

Elles sont, en cas de besoin, rétablies en utilisant les nouvelle adresses.

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

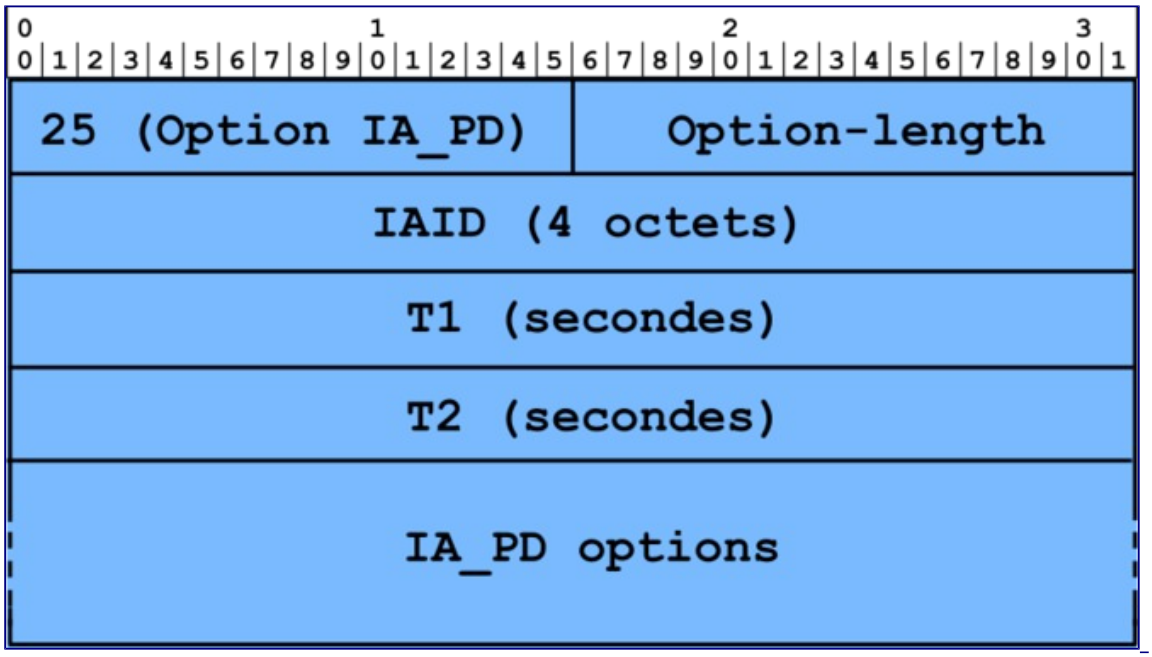
Notez que l'utilisation du préfixe alloué sur le routeur demandeur est impossible sur le lien donnant accès au routeur délégataire. Ceci empêche par conséquent l'agrégation des routes d'acès au routeur demandeur et d'accès au réseau qu'il dessert.

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 que l'interface qui le relie au routeur délégataire. il utilise alors des deux options du [RFC 6603]

Structure de l'option d'association d'identités pour la délégation de préfixes (RFC 3633 , RFC 7550)

La structure de cette option est la suivante:



OPTION_IA_PD . Le champ type de cette option a pour valeur 25.

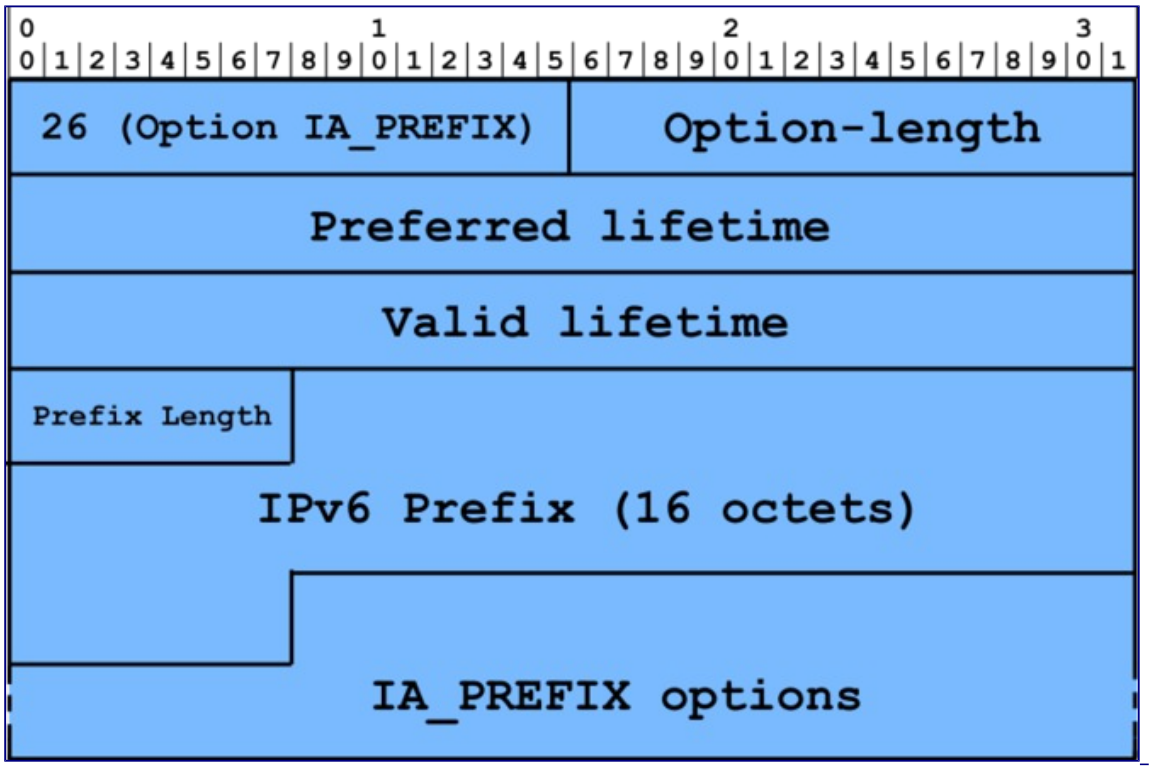
Option-length . La longueur de l'option est la longueur, en octet, de la valeur des options IA_PD options.

IAID . L'IAID 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 structure de cette option est la suivante:



`Msg-type` Le champ type de cette option vaut 26.

`Option-length` Le champ longueur du champ option est la longueur en octet du champ d'option de cette option.

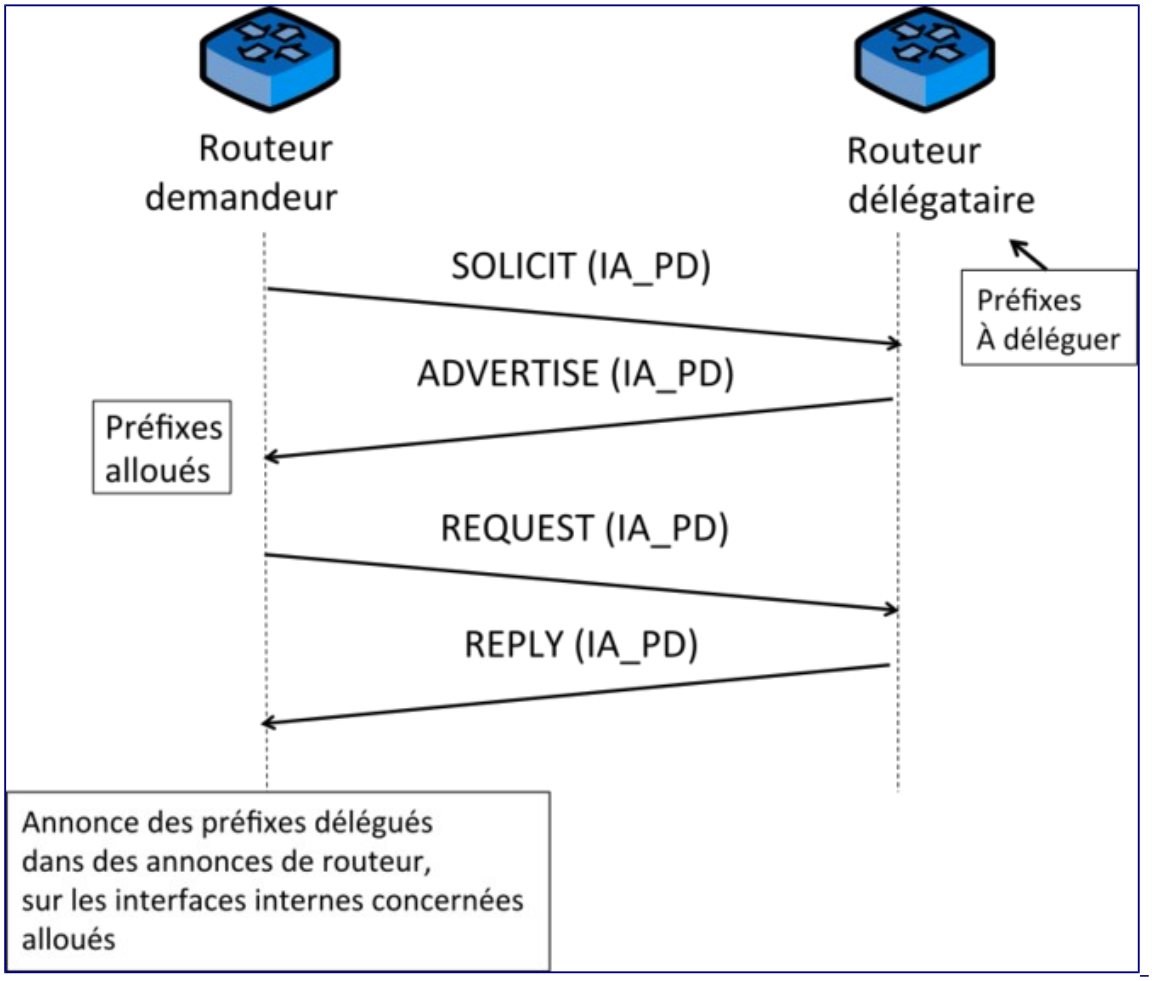
`Preferred- lifetime, valid lifetime` Les durées de vie préférée et totale sont celles du préfixe.

`Prefix-length` Le 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.

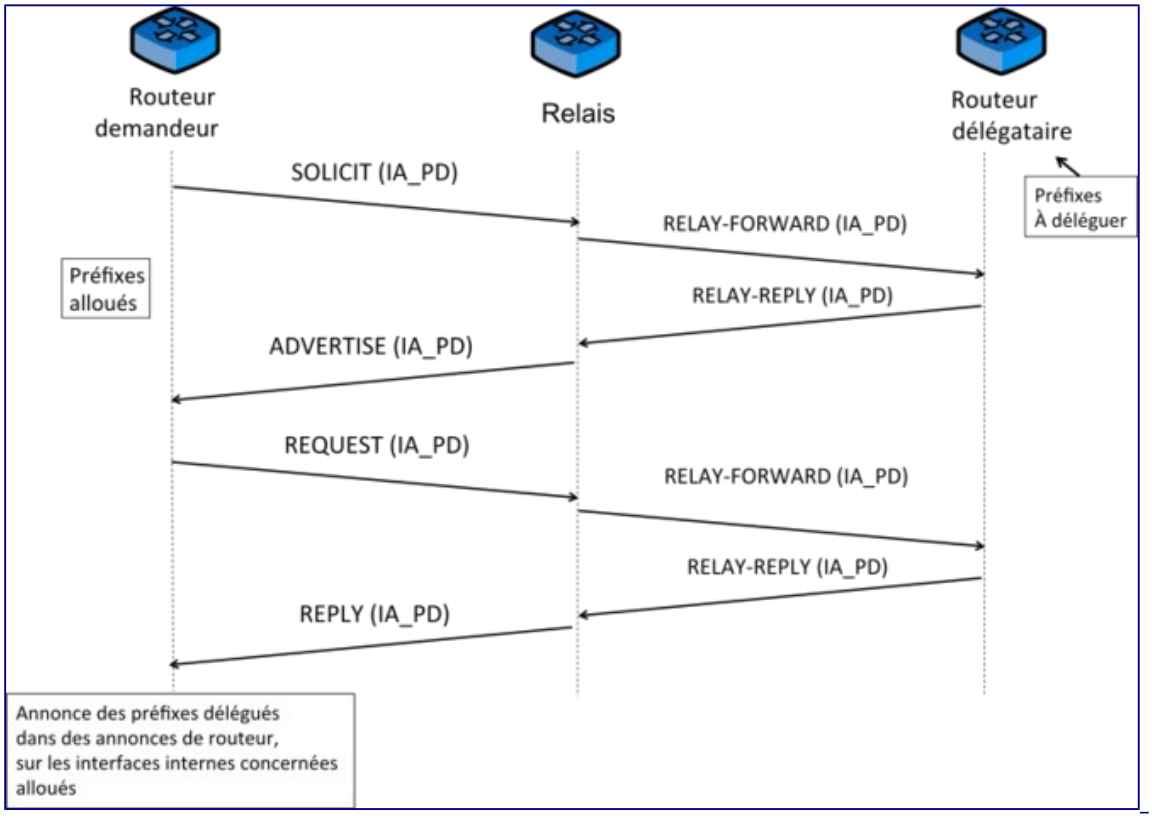
Principe de l'allocation de préfixe à états sans relais



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.

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 l'en-tête du message RELAY-REPLY.

Conclusion

DHCPv6 est un protocole de niveau application. Il utilise le service de transport d'UDP. Il fonctionne en mode client-serveur. Les messages échangés transportent l'identité de l'émetteur (DUID), celui 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 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.

Initialement, ni la délégation de préfixe, ni l'exclusion de préfixe n'existaient. Il a suffit de définir deux options et leur gestion en émission et en réception pour ajouter cette nouvelle fonctionnalité dans DHCPv6.

Ceci a impliqué RFC 7550 des modifications pour clarifier ou préciser la spécification de DHCPv6 RFC 3315 et entraînera prochainement la publication d'une nouvelle version de la spécification du protocole DHCPv6.

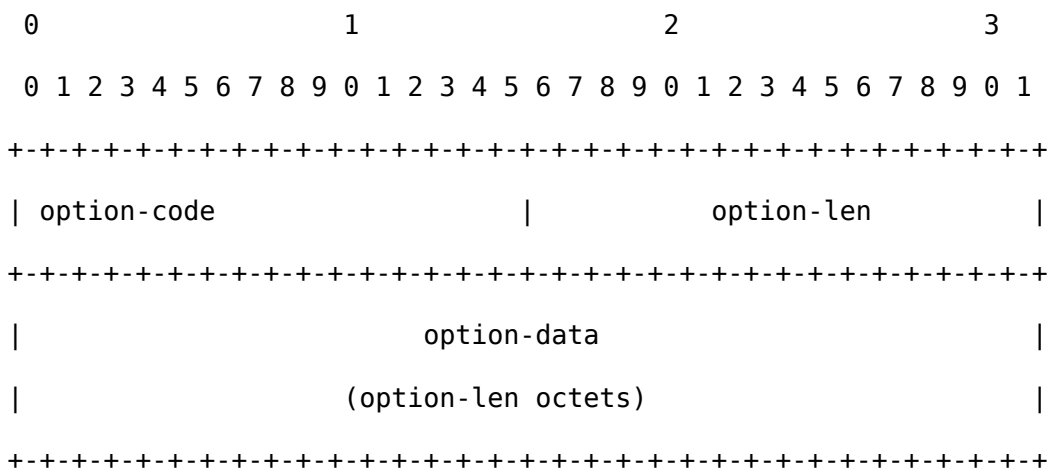
Annexe 1. Structure des options du protocole DHCPv6

La structure générale des options est la suivante. 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 octet du champs de données de l'option. Le champs type de l'option en est exclus. 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'un association d'identités, d'autres encore sont spécifiques d'une adresse, dans une association d'identités.

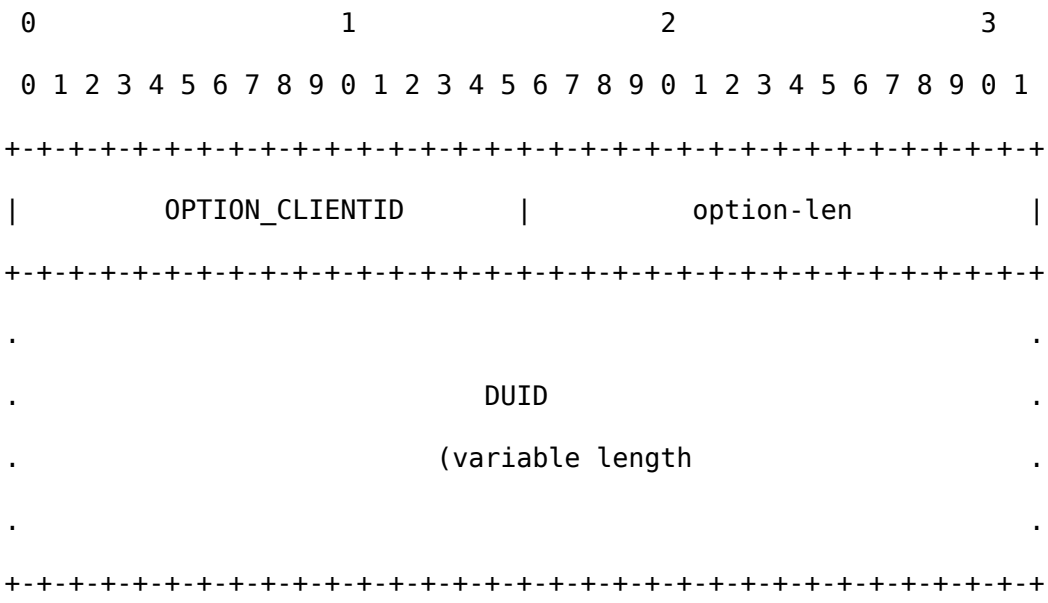
La structure générale d'une option est la suivante:



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.

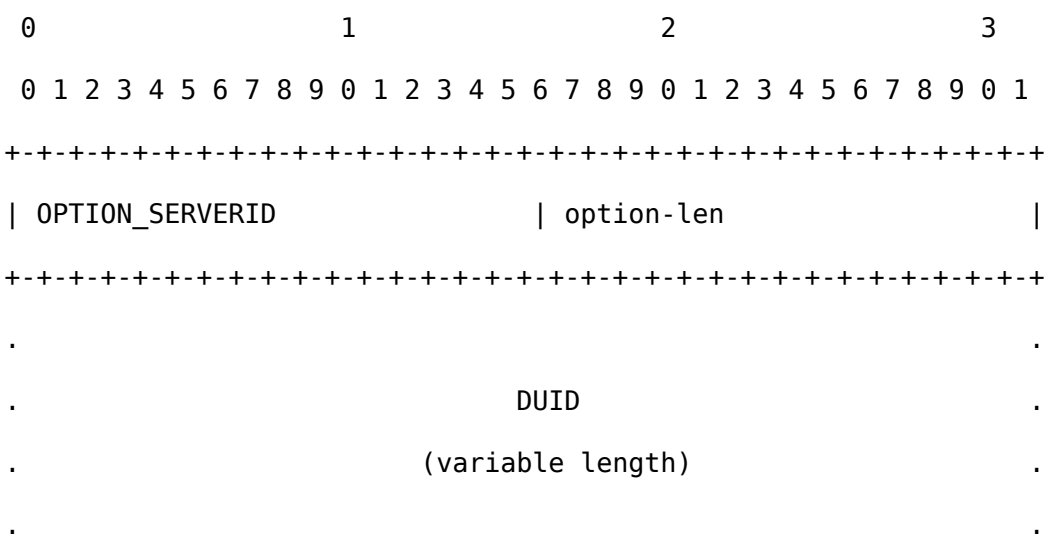
La structure de cette option est la suivante:



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:

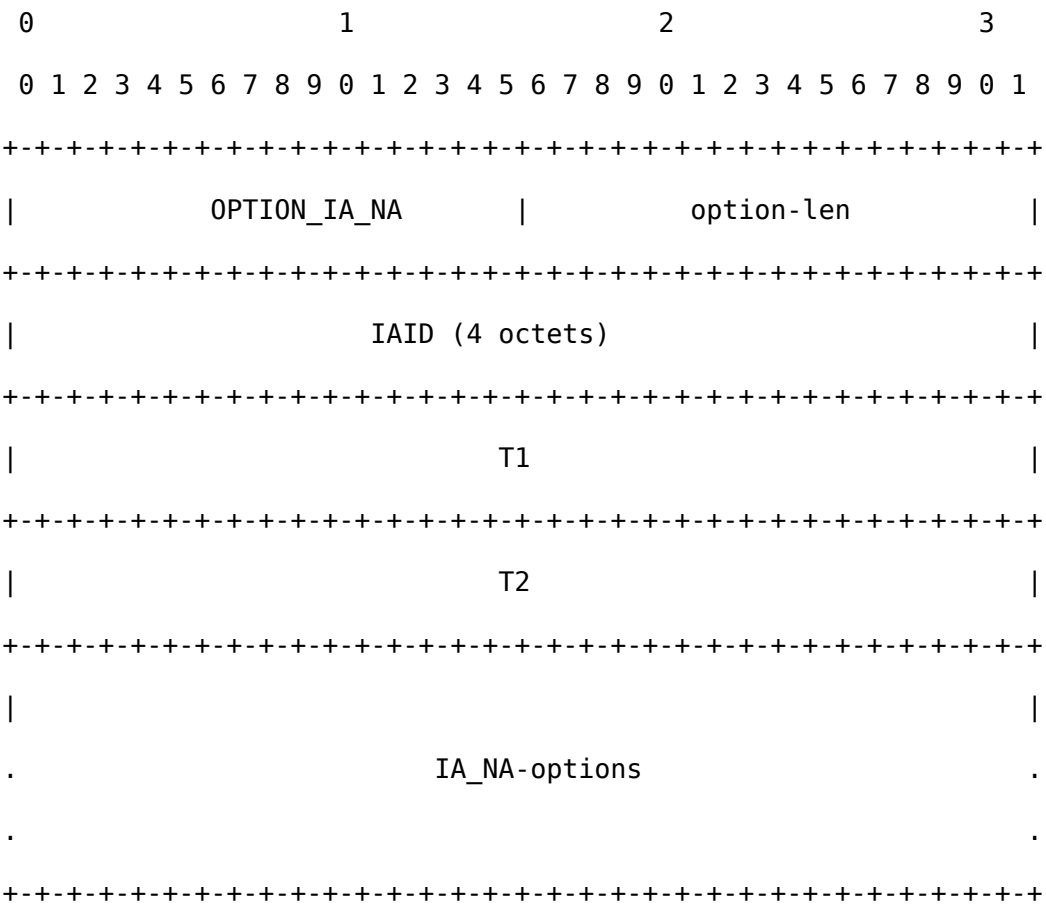


Option association d’identité pour les adresses non temporaires

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:

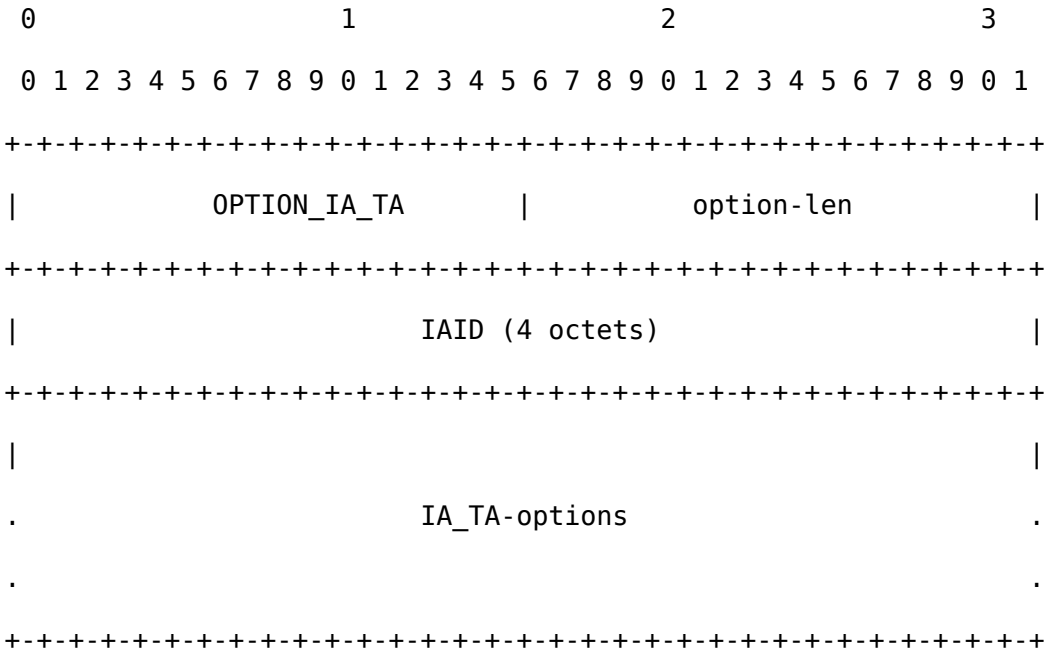


Option d'association d’identité pour les adresses temporaires

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 adresses temporaires associées 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.

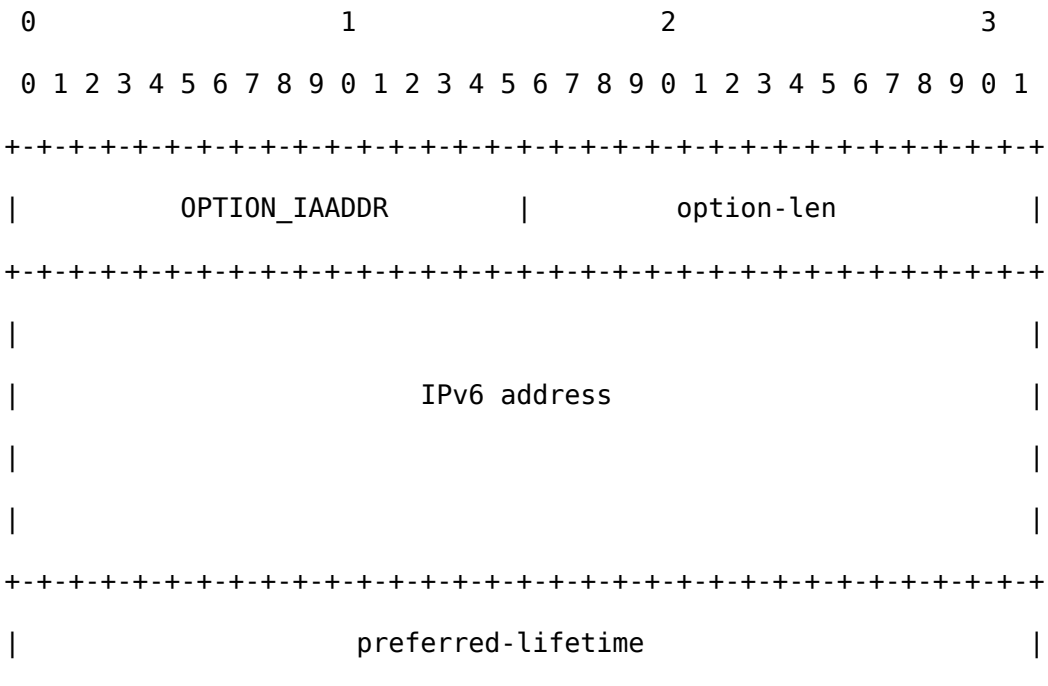
La structure de cette option est la suivante:

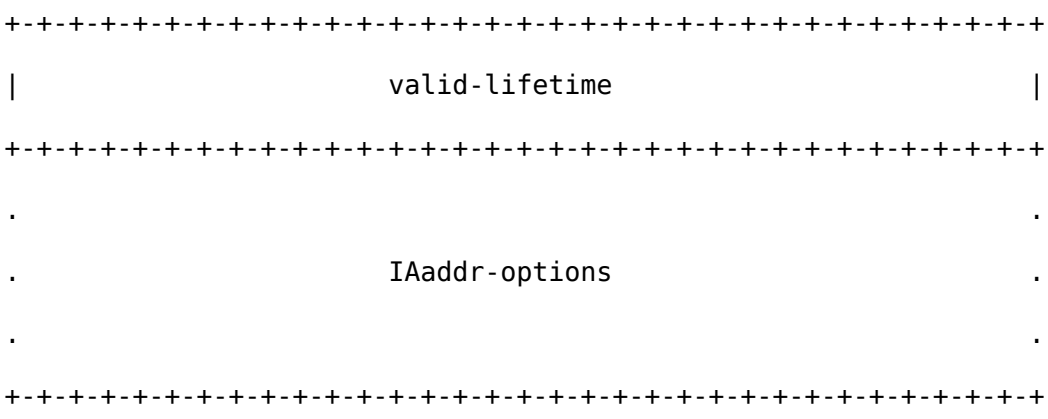


Option d’adresse d'association d'identités

L'option d'adresse'association d'identités (IA Address Option) spécifie une adresse IPv6 associée à une association d'identités IA_NA ou IA_TA. Elle apparaît dans le champ d'option d'une association d'identités pour adresse non temporaire ou temporaire. Une option statut indique l'état de toute opération impliquant cette adresse.

La structure de cette option est la suivante:

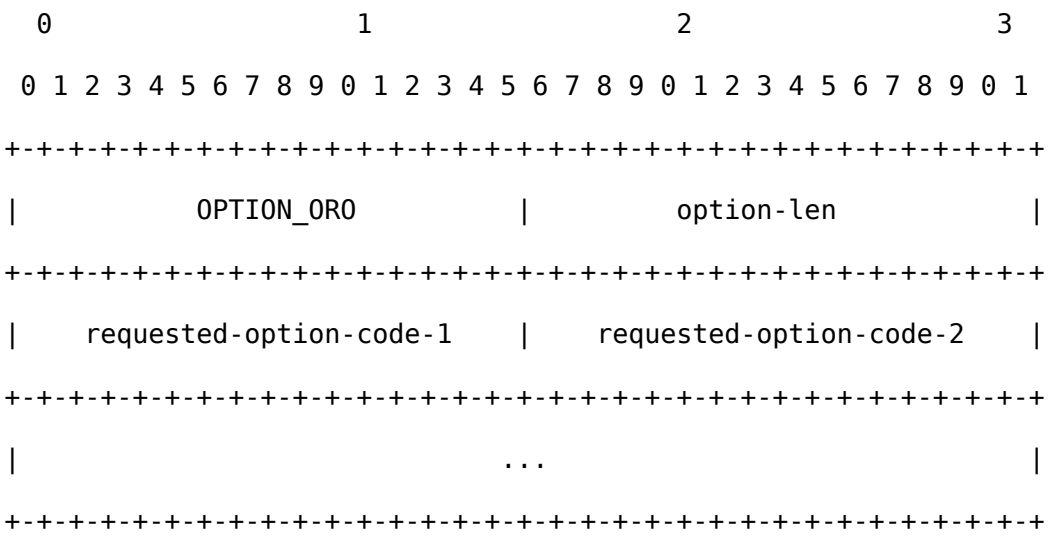




Option de demande d'options

L'option de demande d'option (Options Request Option) identifie la liste des options demandées par le client ou fournies ou cconcernées pour le serveur.

La structure de cette option est la suivante:

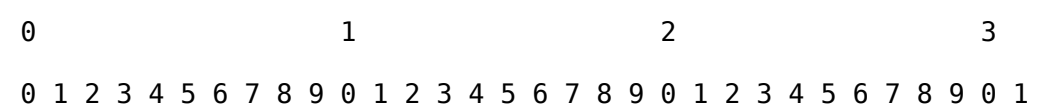


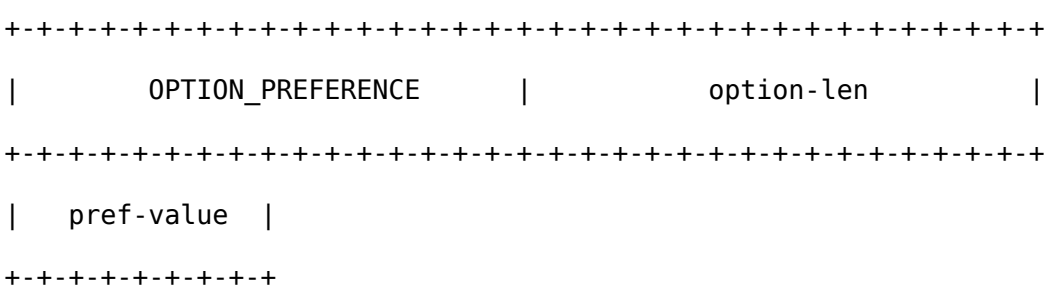
Option de priorité (du serveur)

L'option de priorité (Preference Option) indique la priorité du serveur au client.

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:



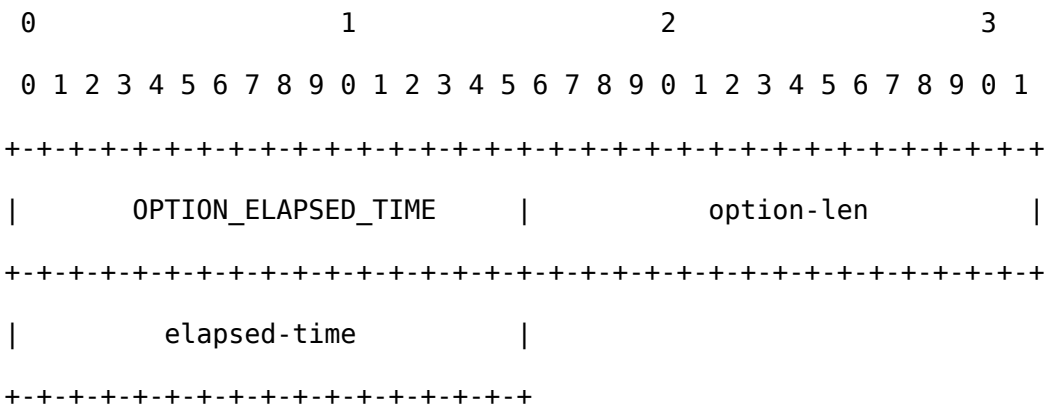


Option temps écoulé (depuis le début d'un échange)

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

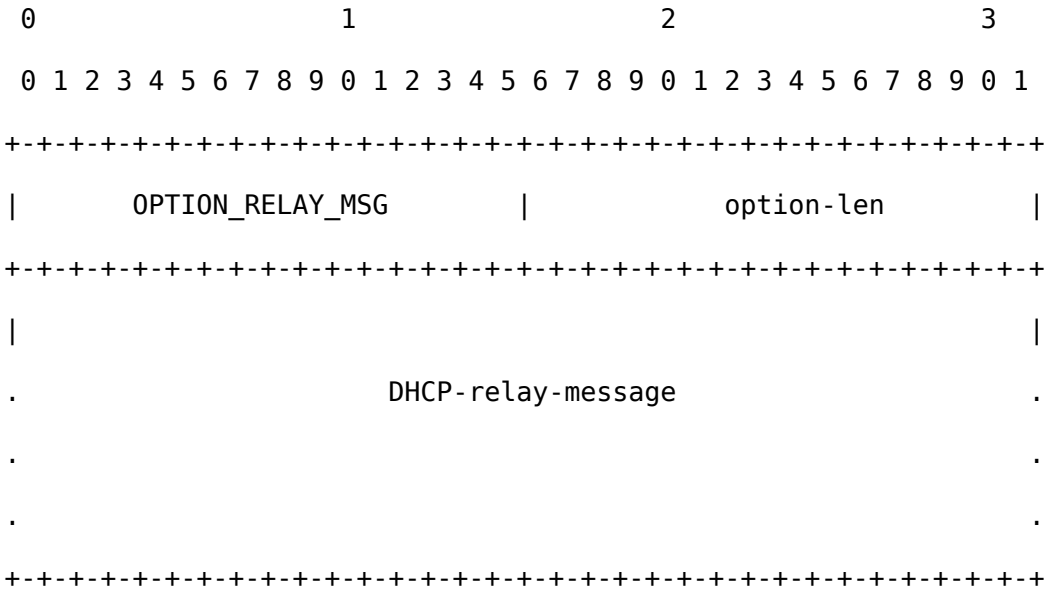
L'option message relayé (RELAY Message Option) contient le message DHCPv6 relayé dans un message RELAY-FORWARD ou RELAY-REPLY.

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 d'authentification

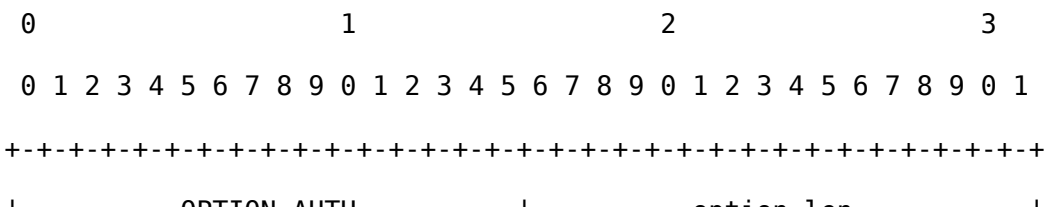
L'option d'authentification (Authetication Option) transporte une information d'authentification. Cette information authentifie l'identité de l'émetteur et l'intégrité du message DHCPv6. Cette option fournit un environnement qui prend en compte différents protocoles d'authentification, ce qui permettra d'en prendre en compte de nouveaux.

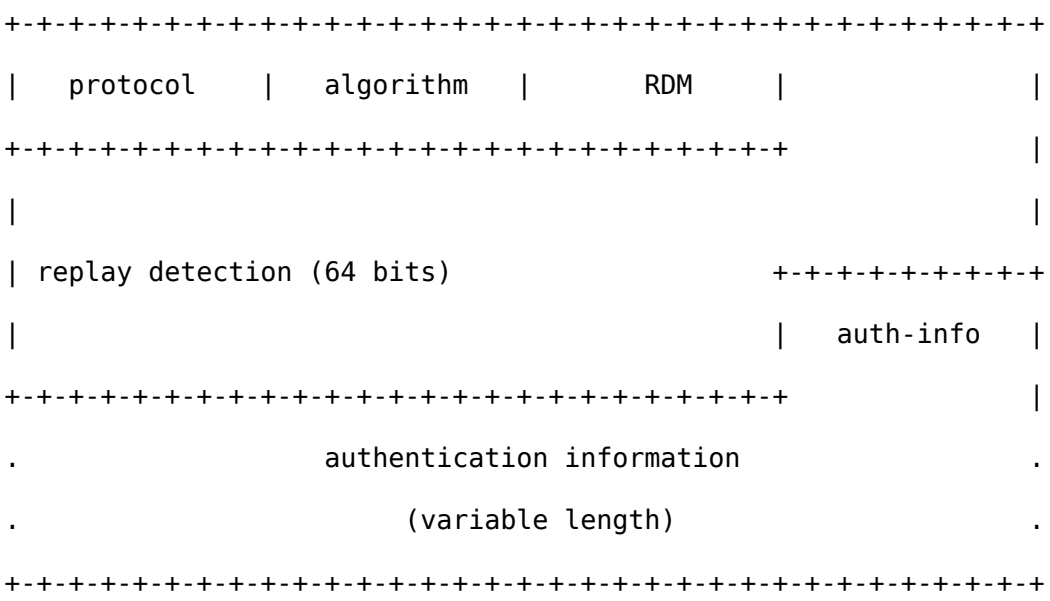
Cette option décrit donc le protocole d'authentification utilisé, la méthode de protection contre le rejeu, l'algorithme de génération du condensé (MAC: Message Authentification Code) qui authentifie le message, et bien entendu la valeur du condensé (128 bits, pa exemple).

Rappel: le principe de l'authentification consiste à calculer un condensé de taille fixe qui ne dépend que de l'information prise en compte (le message DHCPv6, par exemple) en utilisant un algorithme tel que deux informations différentes produisent très probablement des condensés différents. La comparaison des condensés reçu et calculé par le récepteur permet de décider si les données reçues sont ou ne sont pas acceptables. Si ces condensés sont ientiques, l'information est acceptable. Elle ne l'est pas sinon.

La sécurisation des échanges DHCPv6 entre serveurs et relais adjacents utilise IPsec.

La structure de cette option est la suivante:

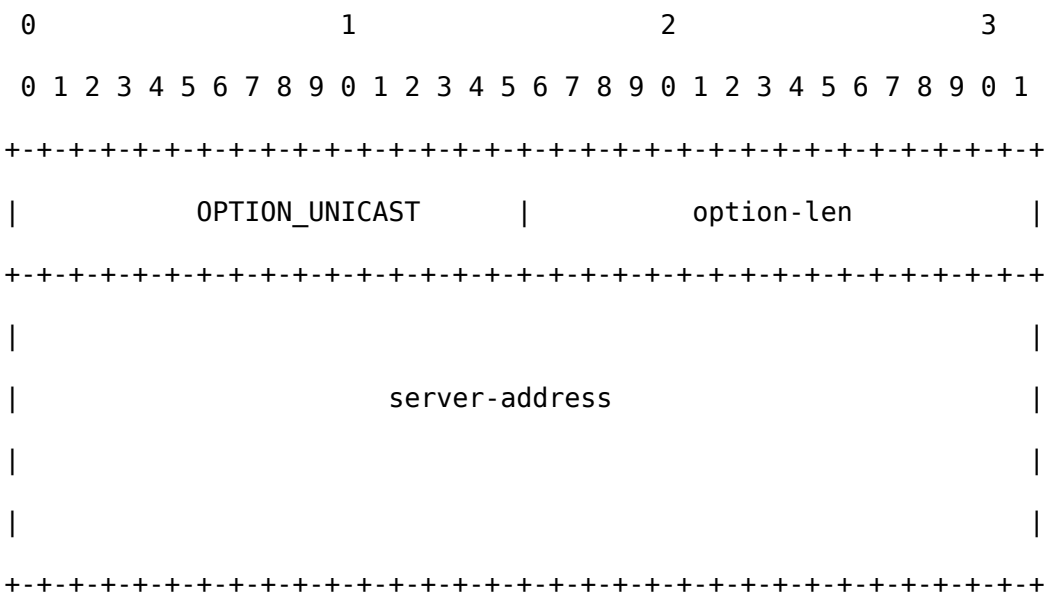




Option d'utilisation de l'adresse individuelle du serveur

L'option d'utilisation de l'adresse individuelle du serveur (Server Unicast Option), qu'émet un serveur, autorise le client DHCPv6 qui reçoit cette option à échanger avec le serveur en utilisant son adresse individuelle au lieu de l'adresse de diffusion sélective All_DHCP_Relay_Agents_and_Servers address.

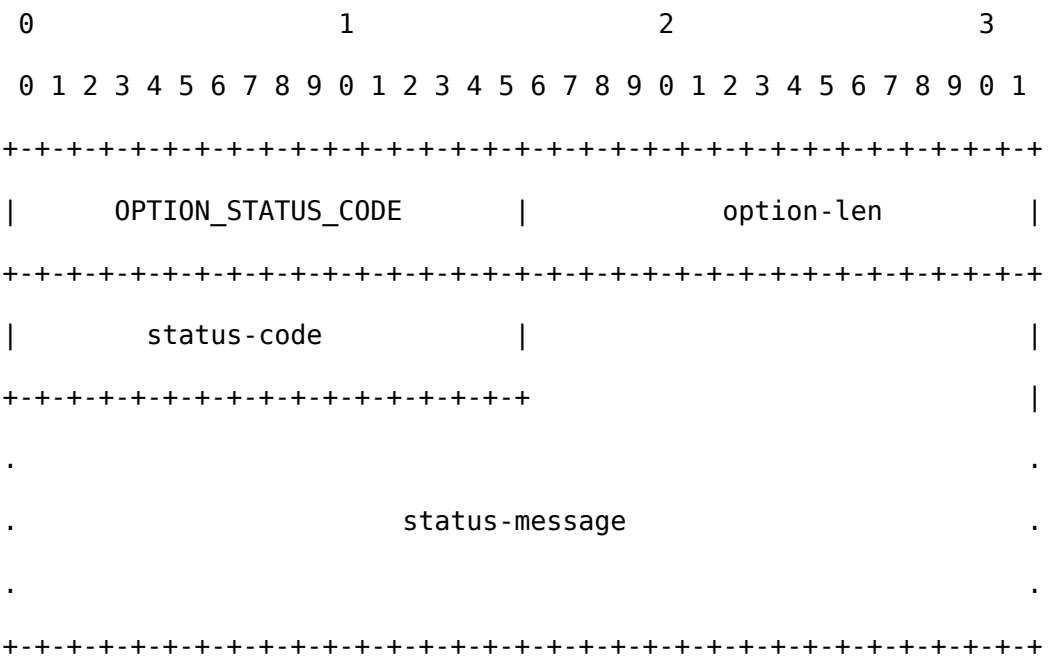
La structure de cette option est la suivante:



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 (success).

La structure de cette option est la suivante:



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

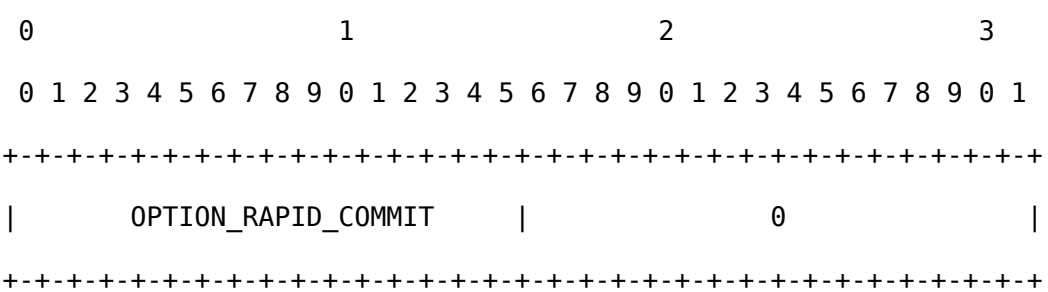
Option de Validation rapide

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:

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

Un serveur doit inclure cete option le message REPLY qui répond au SOLICIT du client transportant l'option de validation rapide.

La structure de cette option est la suivante:

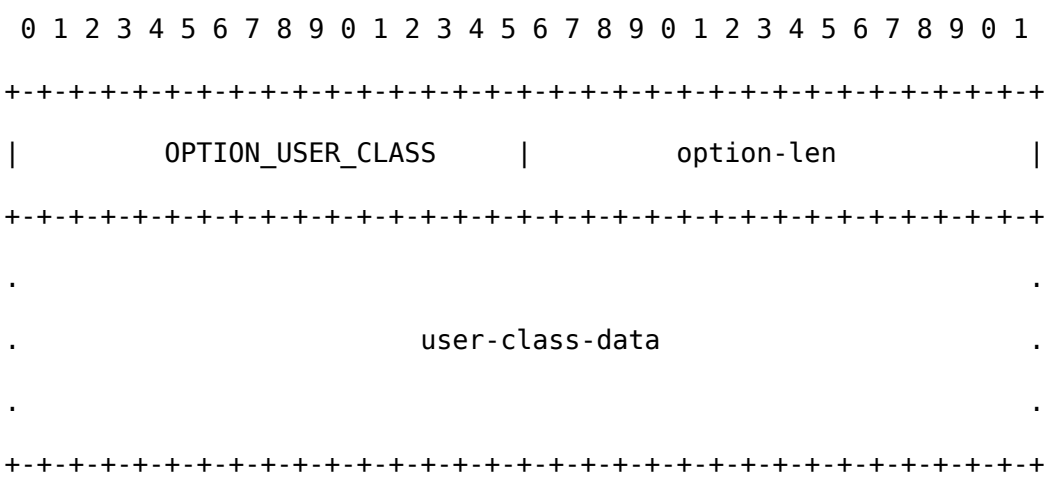


Option classe d'utilisateur

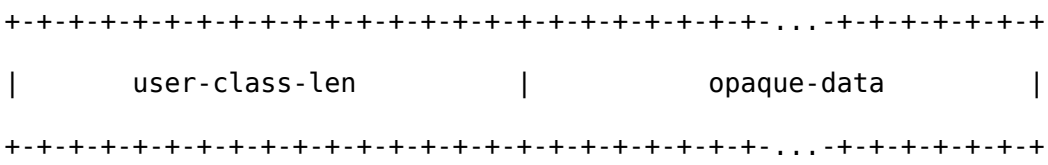
L'option classe d'utilisateur (User Class Option)identifie un type ou une classe d'utilisateurs ou d'applications qu'ils représentent. La partie données de cette option contient plusieurs champs non interprétés (Opaque) par DHCPV6. Ces champs représentent la classe d'utilisateur à laquelle appartient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

Un serveur choisit les informations de configuration du client en fonction de la classe identifiée par l'option.

La structure de cette option est la suivante:



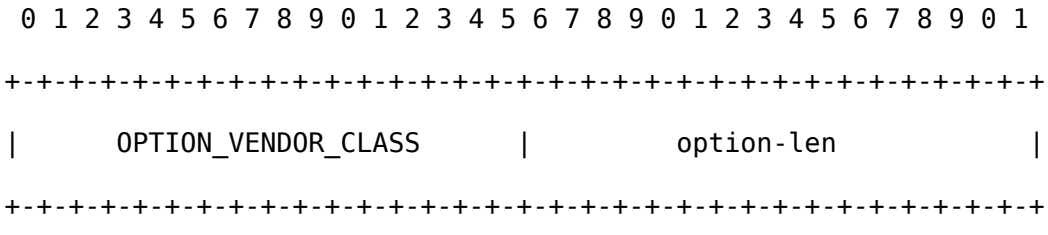
La structure de la partie données de cette option peut apparaître plusieurs fois. Elle est la suivante:

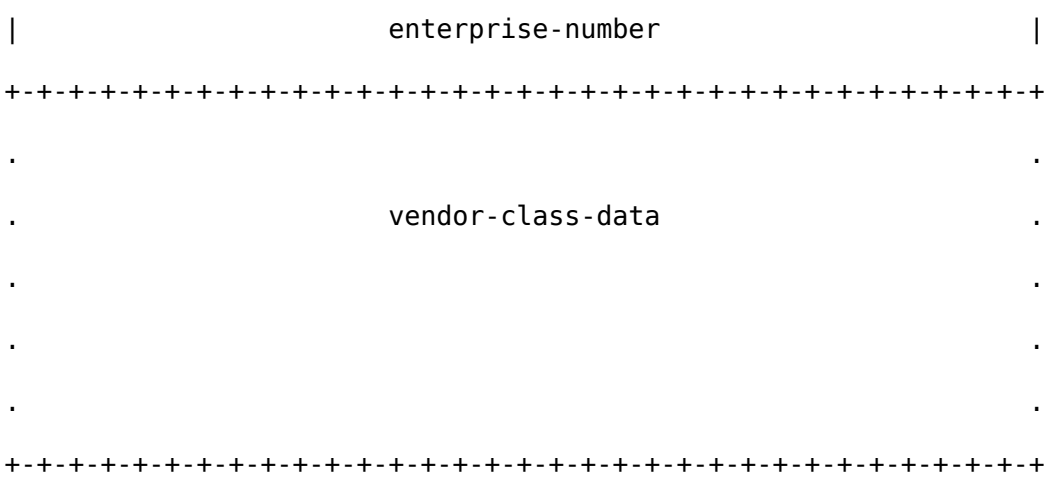


Option de classe de constructeur

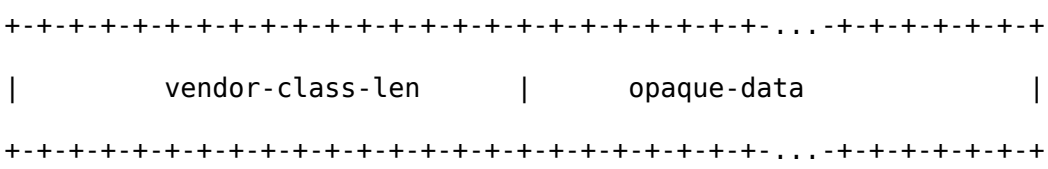
L'option de classe de constructeur (Vendor Class Option)identifie le constructeur du matériel qui supporte le client DHCPv6. Le numéro d'entreprise identifie le constructeur.

La structure de cette option est la suivante:





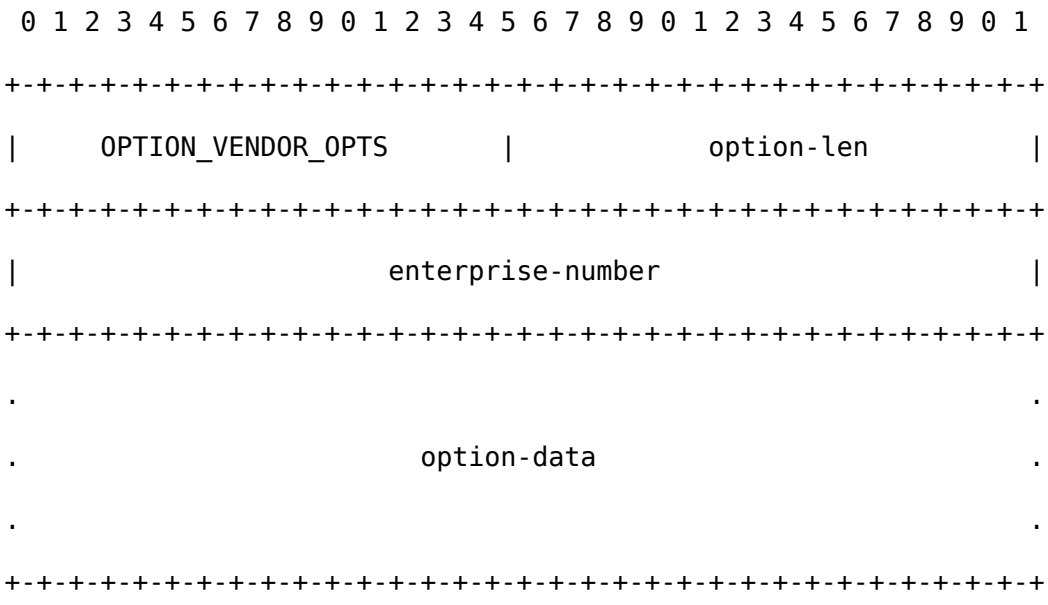
Les paramètres définissant la classe du constructeur se suivent les uns les autres dans le champ de données de classe de constructeur. Chaque paramètre est codé en format LV. DHCPv6 n'interprète pas la valeur (opaque) de ces paramètres.



Option d'information spécifique d'un constructeur

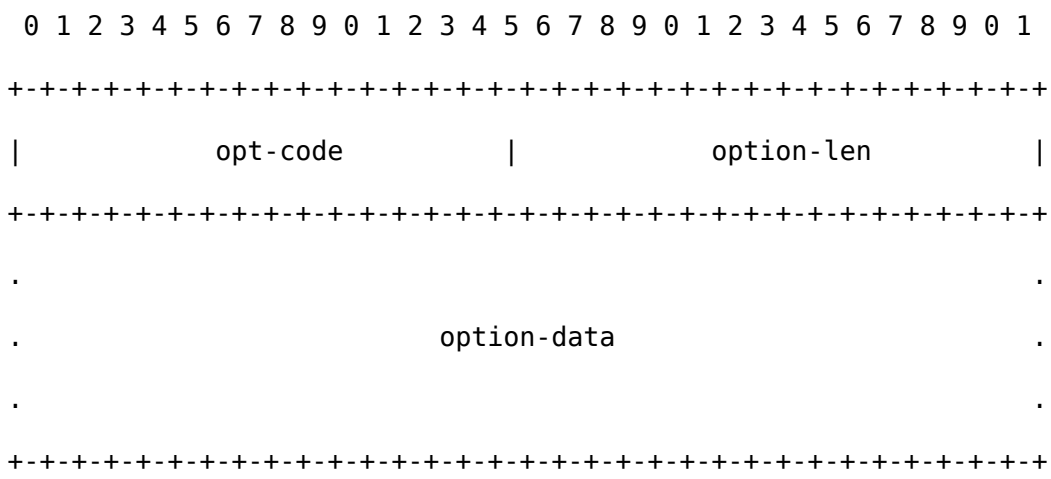
L'option d'information spécifique d'un constructeur (Vendor-specific Information Option) permet que les clients et serveurs DHCPv6 échangent des informations spécifiques d'un constructeur. Le numéro d'entreprise identifie le constructeur.

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'optiond'information spécifique d'un constructeur.

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



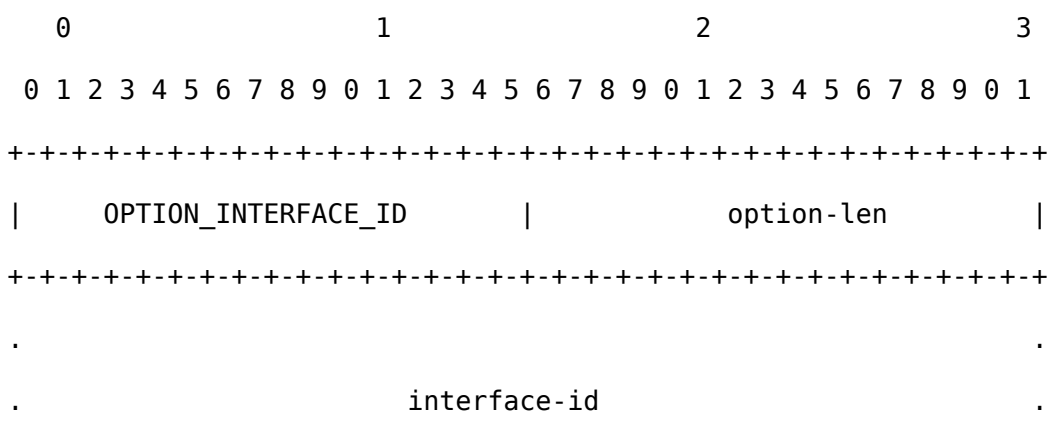
Option d'identification d'interface

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 es spécifique des messages RELAY-FORWARD et RELAY-REPLY. Ils peuvent également utiliser cete information pour appliquer une politique d'allocation basée sur la correspondance exacte de la valeur de cette option.

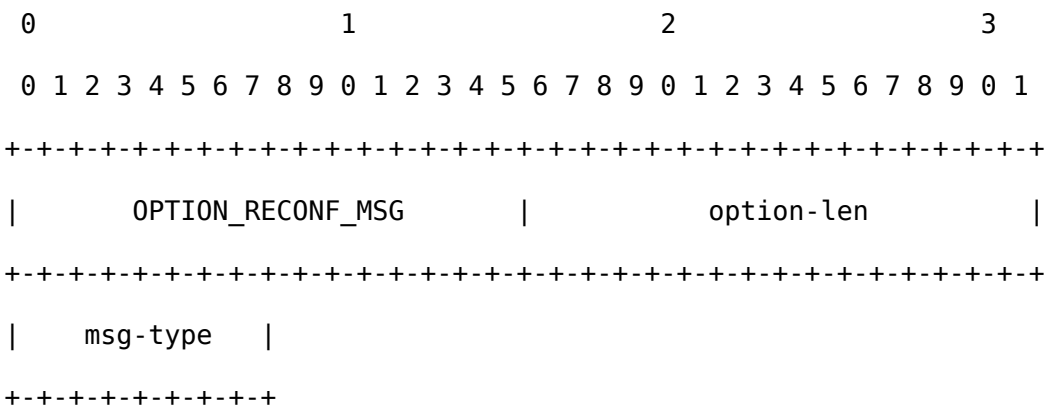
La structure de cette option est la suivante:



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:

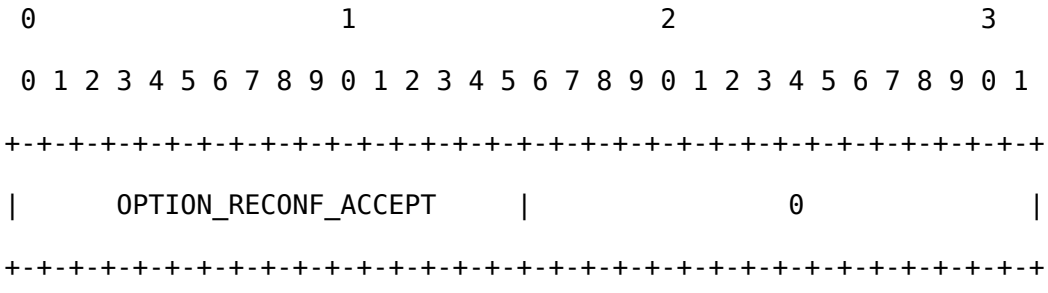


Option d'acceptation de reconfiguration

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 refux d'accepter des messages de reconfiguration. La présence de cette option indique au client s'il doit ou non accepter les messsages 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.

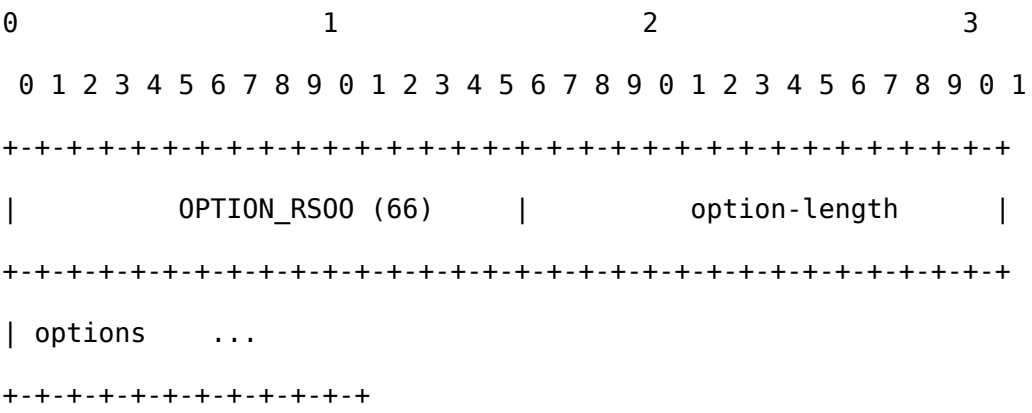
L'option d'options spécifiques de relais (RELAY-SUPPLIED OPTIONS OPTION) dans les messages RELAY-FORW adressés au serveur DHCPv6 contient alors toutes les options correspondant à des paramètres que le relais souhaite porter à la connaissance du client. Cette possibilité n'est effective que pour des paramètres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

Le serveur DHCPv6 qui reçoit un message RELAY-FORW contenant une option RSSO enregistre les option 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 transmette au client.

Notez que le relais transmet ces paramètres spécifiques de relais au serveur. Le serveur décide ensuite de transmettre tout ou partie de ces informations au client, éventuellement en fonction de la politique définie par l'administrateur du réseau.

Un relais DHCPv6 n'a pas le droit de modifier le contenu d'une réponse (REPLY) destinée à un client.

La structure de cette option est la suivante:



Pour en savoir plus lisez le RFC 6422 .

Annexe 2. Codes d'état du protocole DHCPv6

Cette annexe présente les codes d'état du protocole DHCPv6. Ils sont extraits du RFC 3315 .

Name	Code	Description

```
Success          0 Success.

UnspecFail       1 Failure, reason unspecified; this
                   status code is sent by either a client
                   or a server to indicate a failure
                   not explicitly specified in this
                   document.

NoAddrsAvail     2 Server has no addresses available to assign to
                   the IA(s).

NoBinding        3 Client record (binding) unavailable.

NotOnLink        4 The prefix for the address is not appropriate for
                   the link to which the client is attached.

UseMulticast     5 Sent by a server to a client to force the
                   client to send messages to the server.
                   using the All_DHCPV6_Relay_Agents_and_Servers
                   address.
```

Activité 34: Faire correspondre adresse et nom de domaine

Introduction

Cette activité introduit le système de nommage communément appelé le 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.

Concepts de base du DNS

Le DNS est un système de base de données hiérarchique et distribuée. 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 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.

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

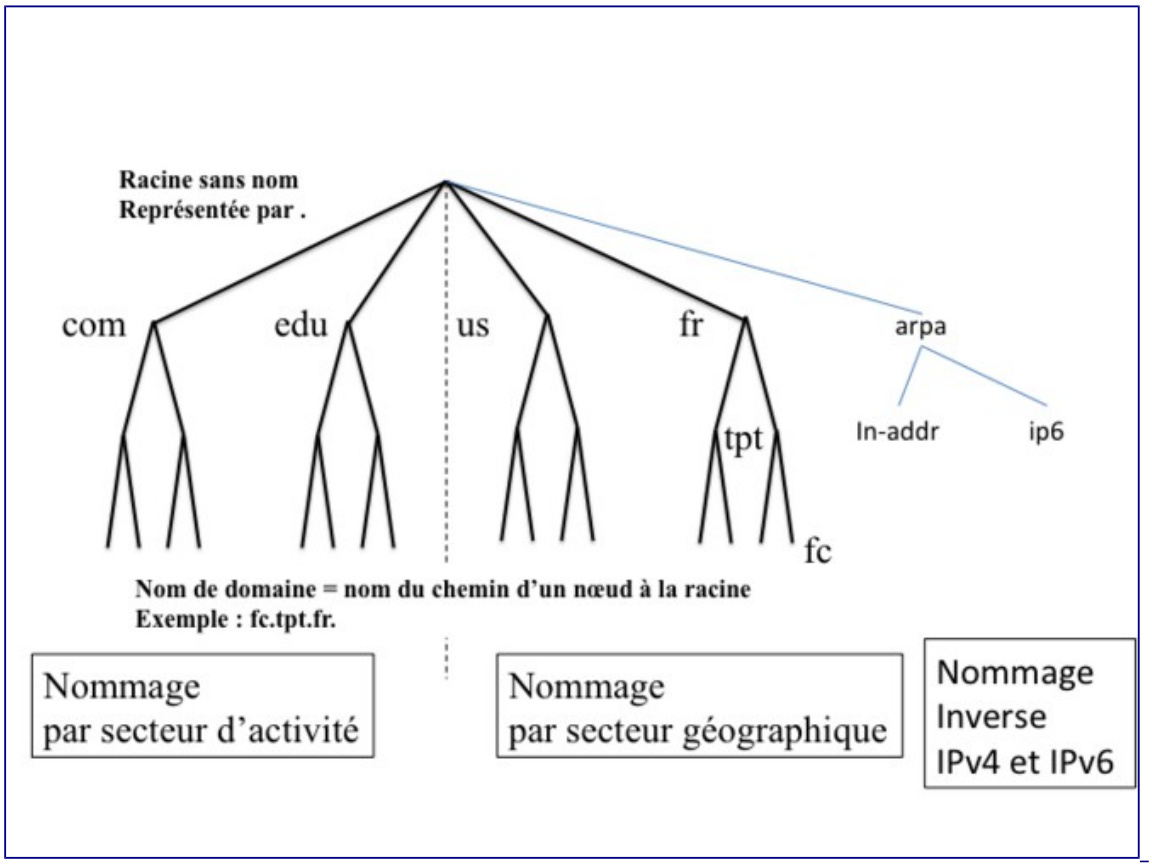
systèmes Unix). Un équipement disposant localement d'une version à jour du fichier de nommage peut ainsi communiquer avec toutes les machines connues dans ce fichier. Dès le début des années 80, la croissance exponentielle du nombre de noms et d'adresses IP utilisées et le besoin de plus en plus fréquent 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'impose 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, utilise un système de nommage hiérarchique pour garantir l'unicité des noms. Le système de nommage hiérarchique utilise une structure d'arbre. 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.



Arbre de nommage

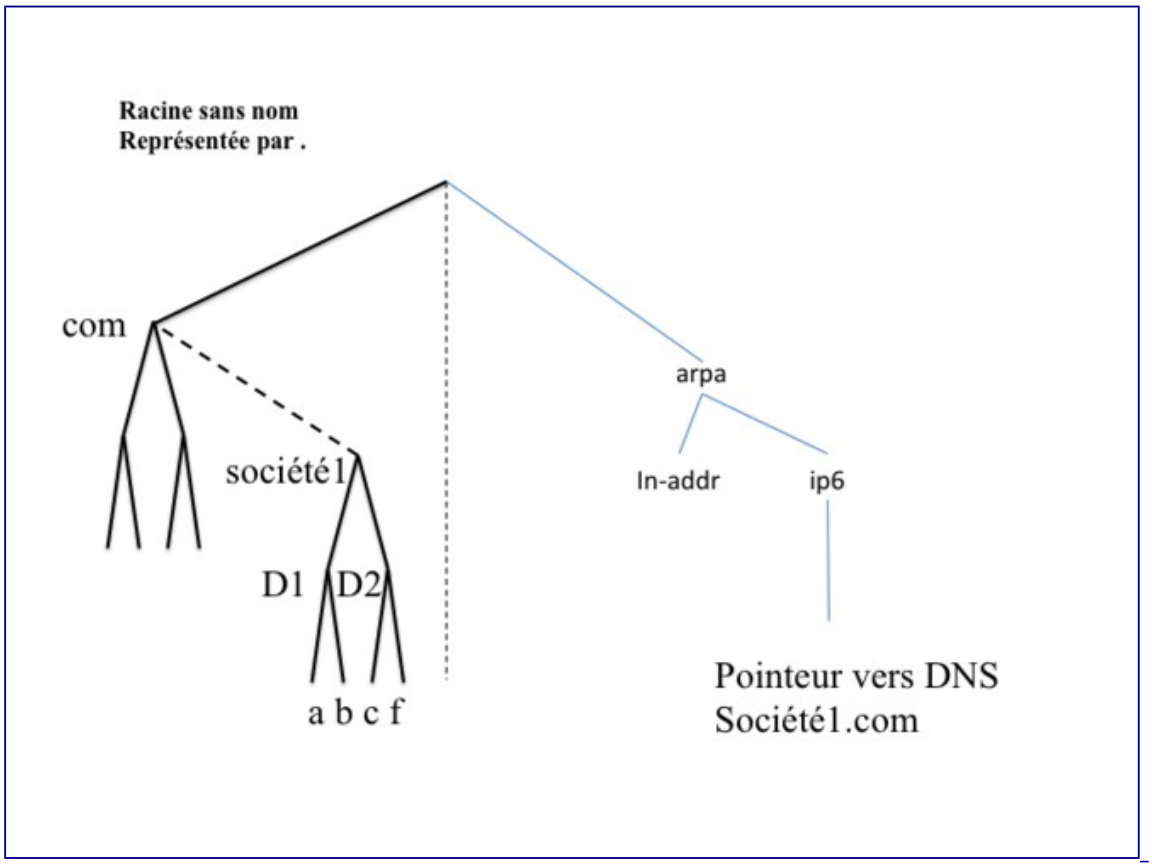
Figure 1. Arbre de nommage. Le nommage se fait, soit en fonction du secteur d'activité, soit en fonction du code pays (ISO). Deux sous-arbres sont dédiées à la résolution inverse: *in-addr* pour IPv4 et *ip6* pour IPv6.

- **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.

- *Equilibrage 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). 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.

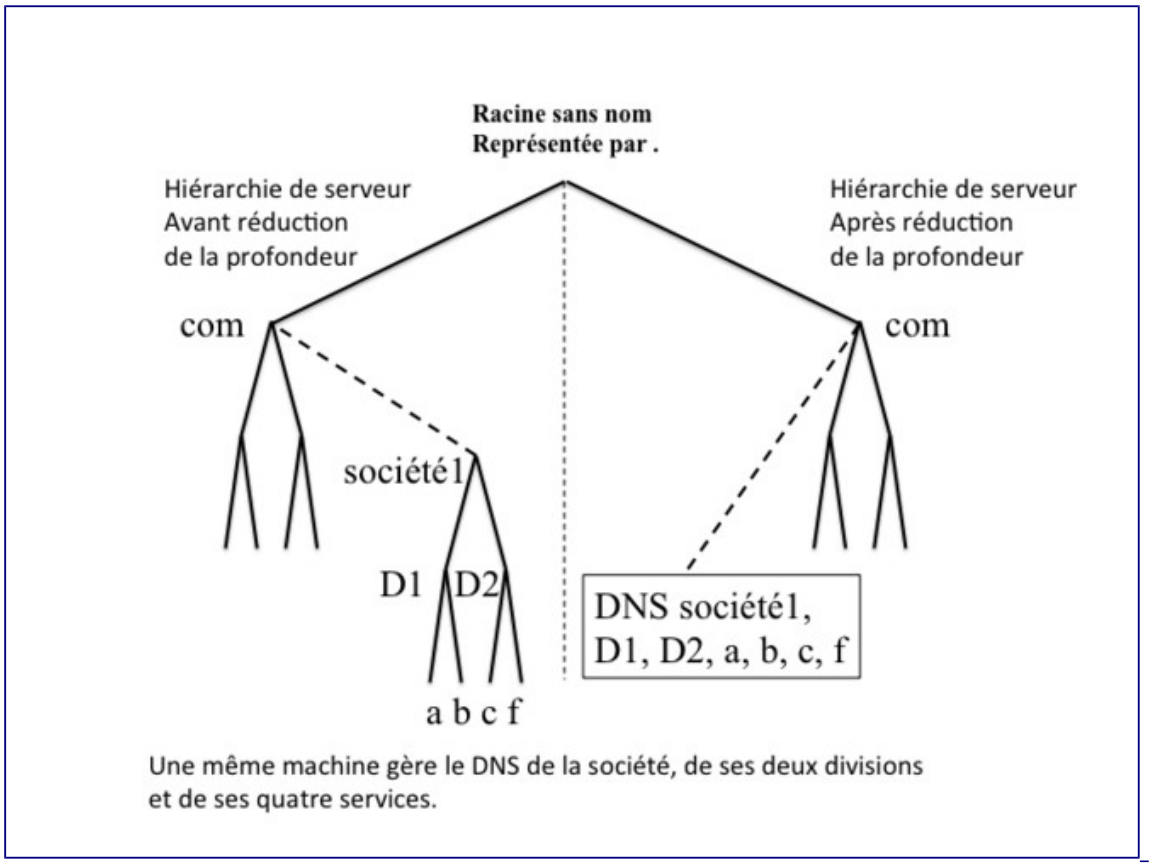


Extension de l'arbre de nommage

L'idée, simple, mais géniale, a été de concevoir un système client-serveur pour cela. 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.

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. 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.



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

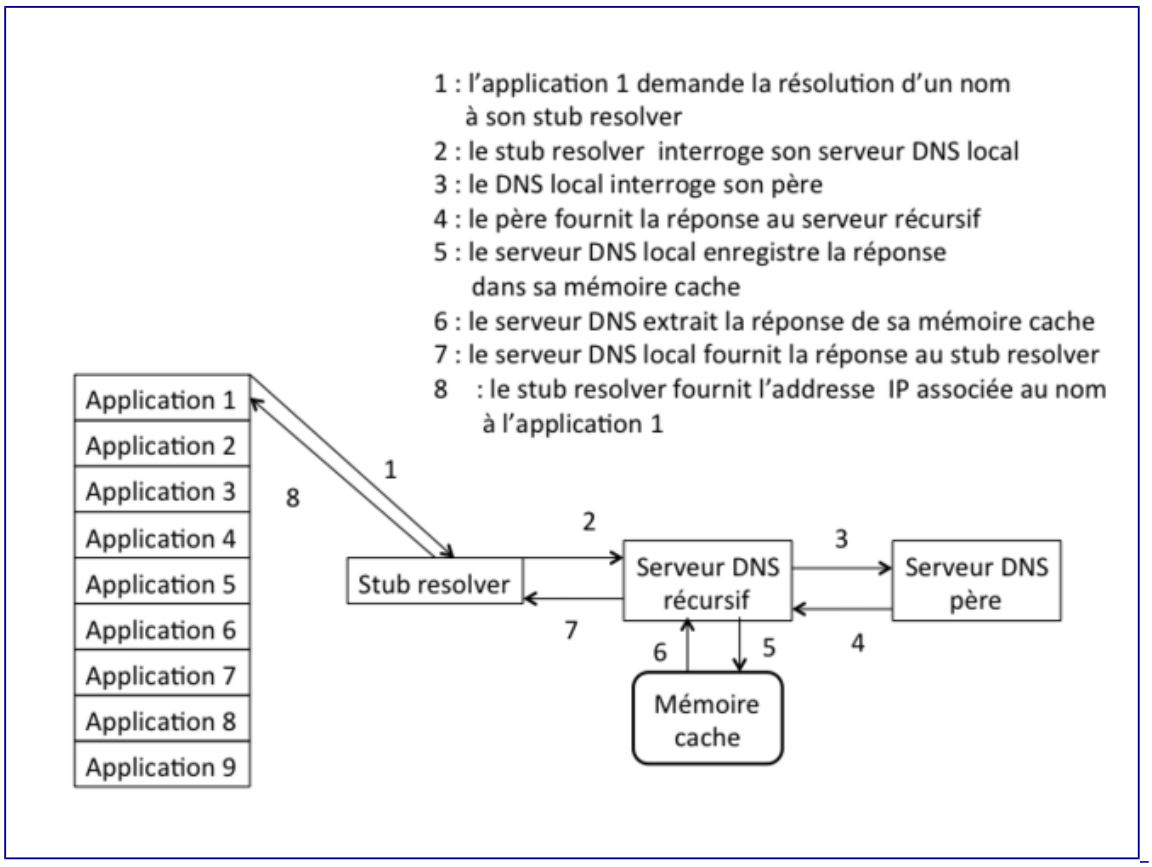
Chacune des applications d'une machine s'adresse au résolveur unique de cette machine (stub resolver) et lui demande des informations associées à des noms de domaines, comme des adresses IP, des relais de messagerie (enregistrement de type MX) ou des serveurs de noms (enregistrement de type NS).

Initialement, le résolveur de la machine locale interroge successivement chacun des serveurs

(résolution itérative), jusqu'à ce qu'il s'adresse au serveur officiel du domaine concerné. Le résolveur de chaque machine gère donc localement un cache des informations de nommage. Cela accélère le traitement des requêtes ultérieures, mais s'accompagne d'une réplication potentielle des mêmes informations au niveau de chaque machine.

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. Pour l'utiliser, les programmes d'application invoquent les procédures de la bibliothèque.

Aujourd'hui, pour optimiser 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.



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. La résolution itérative des requêtes des clients est alors déportée au niveau du serveur DNS local.

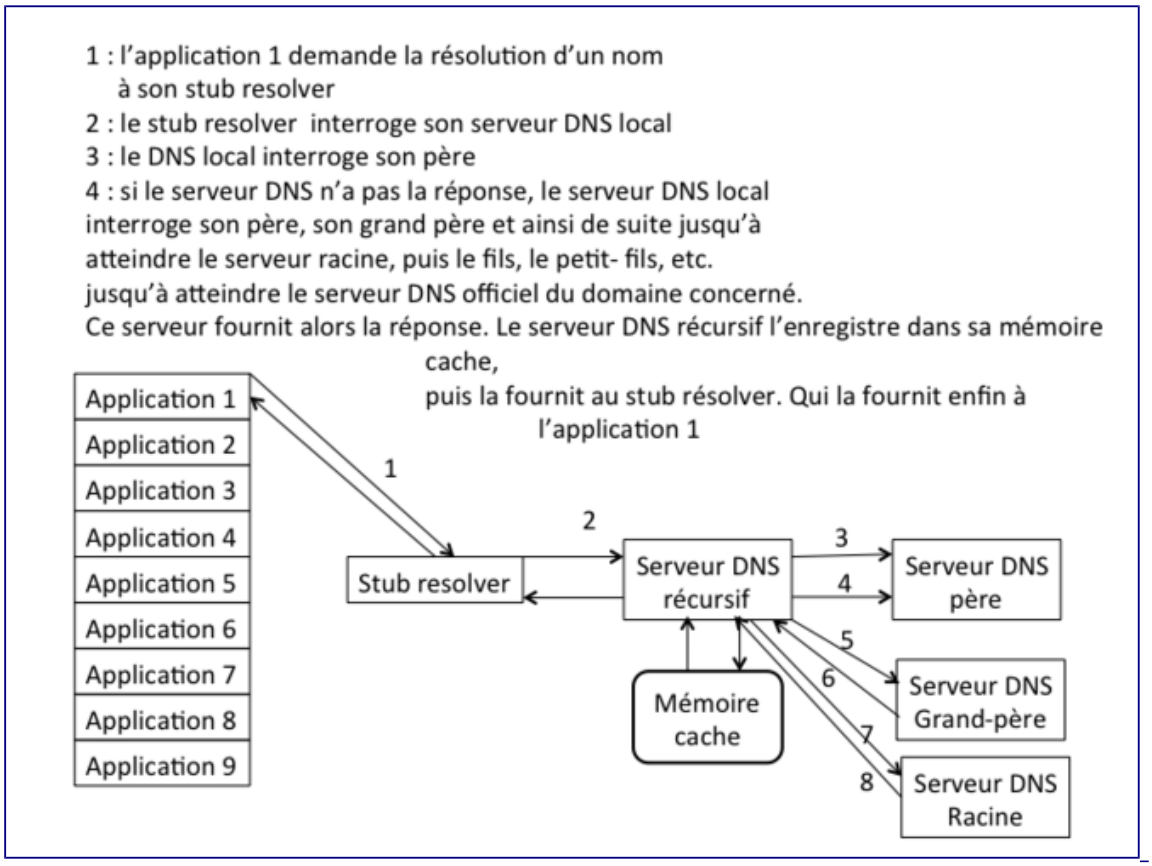
Le serveur DNS local résout un nom de façon très simple: il commence par s'adresser à son serveur DNS père et lui demande «Pourrais-tu me fournir l'adresse (ou les adresses) associées à ce nom?».

Le serveur DNS père peut, soit disposer de la réponse et il la fournit alors au serveur DNS local, soit ignorer la réponse, et dans ce cas, il demande au serveur DNS local de s'adresser au

serveur DNS grand-père. Il fournit alors au serveur DNS local, son client, le nom et l'adresse IP du grand-père.

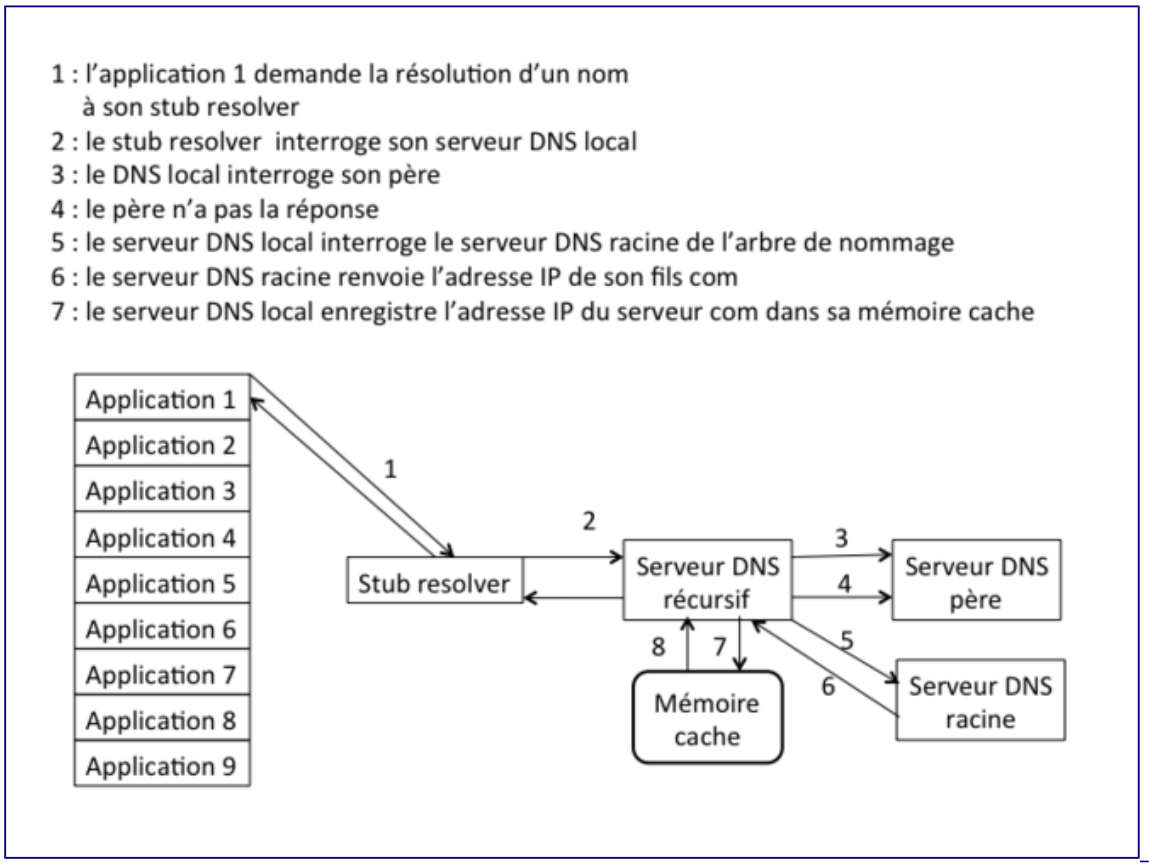
Le serveur DNS local enregistre alors ces informations dans sa mémoire cache. Il interroge alors le serveur grand-père à qui il repose la même question.

Ce processus se répète jusqu'à ce que le client pose la question au serveur DNS racine de l'arbre.



Résolution itérative non optimisée depuis un serveur local

Notez qu'une optimisation du système consiste à ce que le serveur DNS local interroge directement la racine de l'arbre lorsque son serveur DNS père ne dispose pas des informations de nommage demandées.



Résolution itérative optimisée depuis un serveur local

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. Il ne dispose localement d'aucune information de nommage. Il n'enregistre également pas 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, puis au serveur DNS petit-fils du serveur DNS racine. Il finit par adresser sa demande au serveur DNS officiel qui gère les informations de nommage recherchées.

Le serveur DNS officiel concerné 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.

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 lorsquque cette information est valide et s'y trouve.

Si la question concerne le serveur DNS officiel, déjà connu, d'un domaine, le serveur DNS local contacte directement le serveur DNS concerné.

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.

Notez que dans tous ces cas, le traitement des demandes d'information de nommage est accéléré.

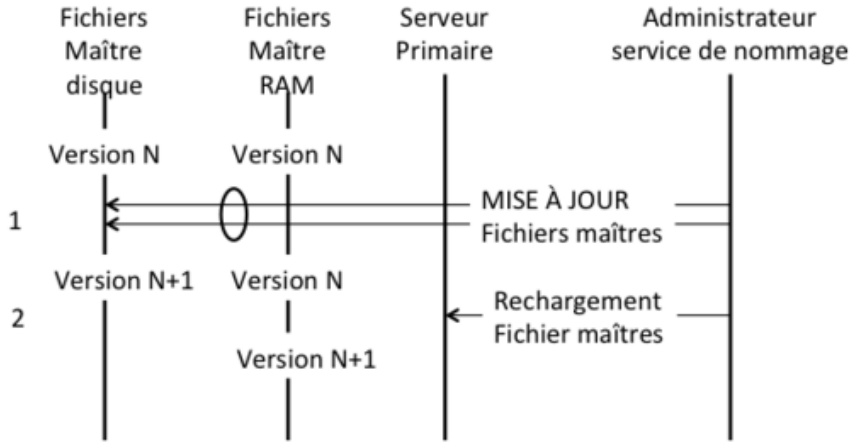
Serveurs de noms primaires et secondaires

Le système DNS distingue, pour une zone donnée, deux types de serveurs de noms: primaire et secondaires.

Notez tout d'abord que les serveurs de noms primaire et secondaires 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.



1 : L'administrateur du service de nommage met à jour les fichiers maîtres sur le disque. Lorsqu'il termine ses modifications, il incrémente le numéro de version de chaque fichier. Le serveur primaire continue d'utiliser les fichiers en version N.

2 : L'administrateur demande au serveur primaire de relire ses fichiers maîtres. Le serveur charge alors les fichiers maître en version N+1 en RAM. S'il est configuré pour cela, il annonce son changement de version aux serveurs secondaires concernés. Sinon, il répond aux requêtes en utilisant la version N+1 des fichiers maîtres.

Mise à jour d'un serveur DNS primaire par l'administrateur du réseau

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.

Un serveur DNS secondaire, par exemple de premier niveau, peut jouer le rôle de serveur DNS primaire pour la synchronisation d'un serveur DNS secondaire, de second niveau.

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. Il déclenche la prise en compte des modifications en redémarrant le serveur DNS primaire ou en le réinitialisant.

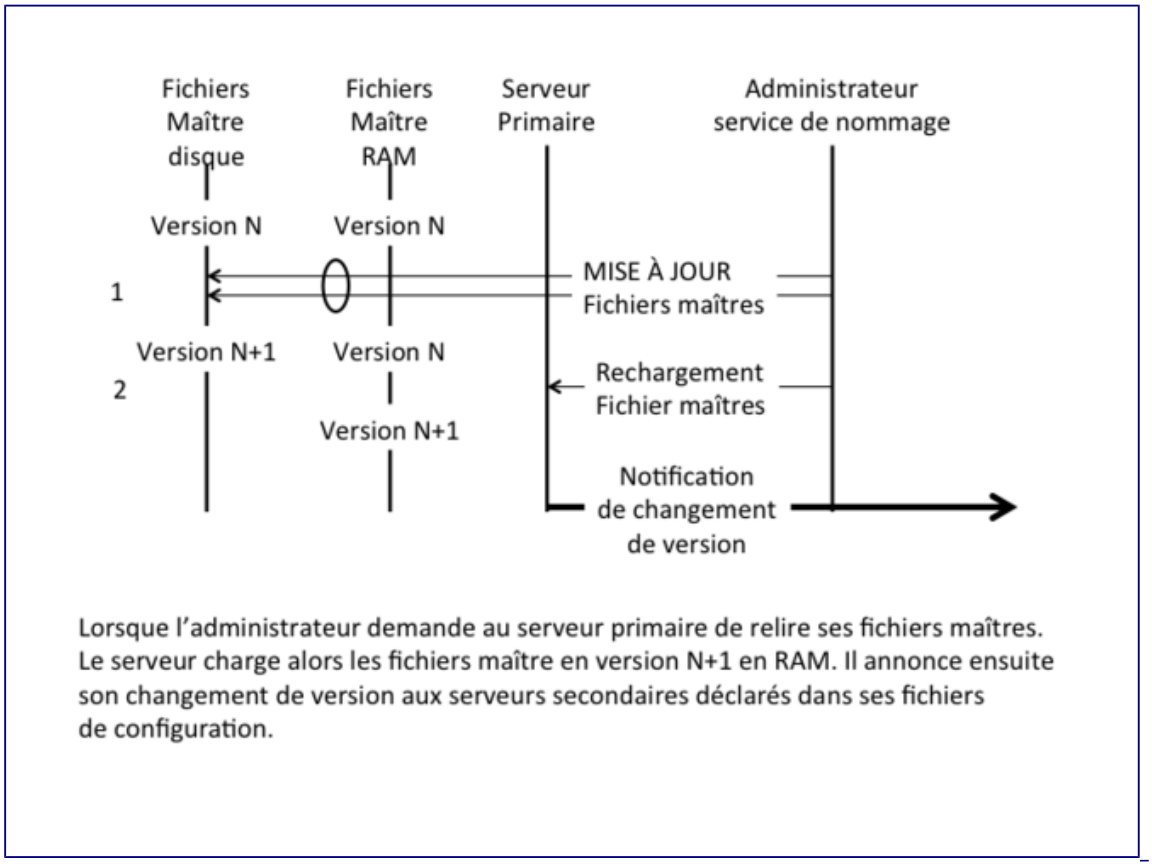
Redémarage . Le serveur DNS primaire relit son fichier de configuration et ses fichiers de zone et les charge en mémoire RAM: Random Access Memory. Il n'utilise ensuite que les informations disponibles en RAM.

Réinitialisation . Le serveur DNS primaire ne relit que ses fichiers de zone et les charge en mémoire RAM: Random Access Memory. 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).

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.



Notification d'un changement de version de base de nommage par le serveur DNS primaire

Synchronisation à partir du seul serveur DNS primaire . Dix serveurs DNS secondaires, au plus par défaut, peuvent immédiatement établir une session de synchronisation avec le serveur DNS primaire.

Les autres serveurs DNS secondaires attendent pendant un temps supérieur ou égal au délai de synchronisation des serveurs DNS secondaires.

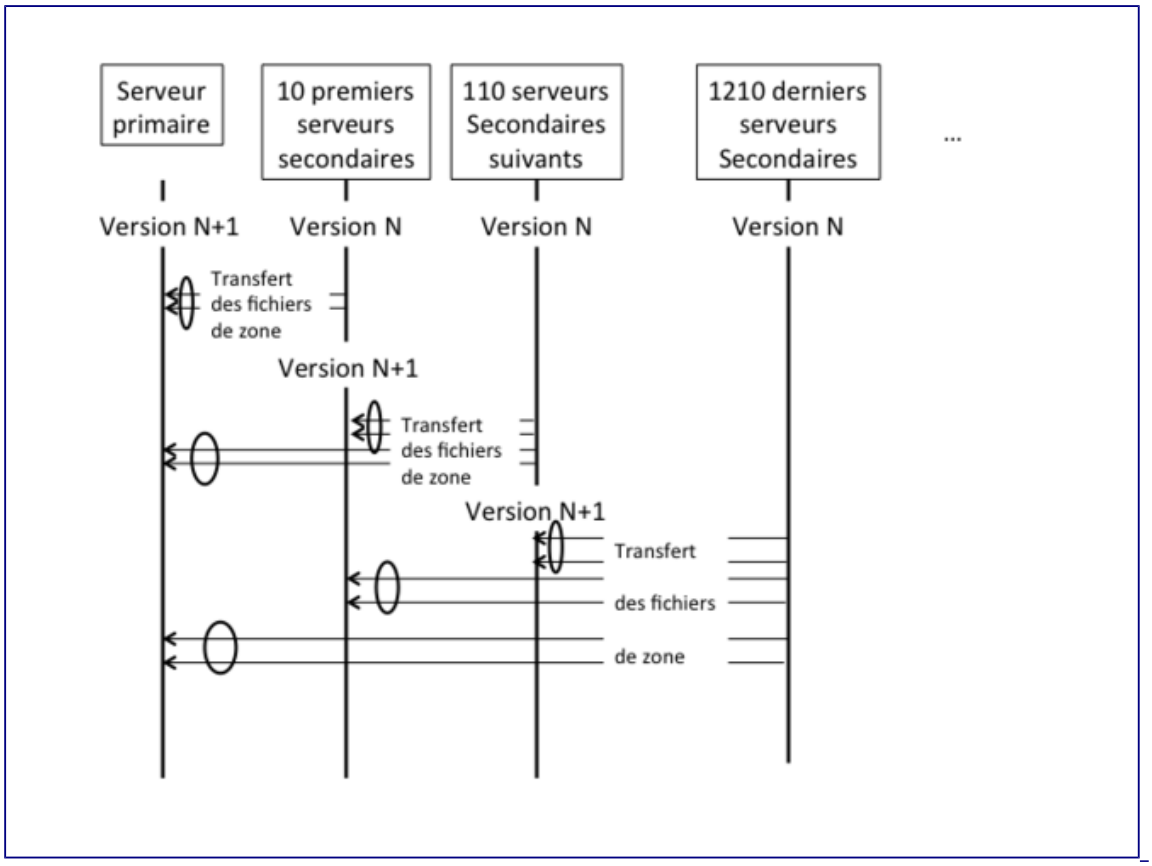
Lorsque les dix premiers serveurs DNS secondaires sont synchronisés, une deuxième vague de 10 serveurs DNS secondaires peut éventuellement démarrer sa synchronisation à partir du serveur DNS primaire.

Les serveurs DNS secondaires qui n'ont pas pu établir de session de synchronisation avec le serveur DNS primaire attendent.

Ce processus continue jusqu'à ce que tous les serveurs DNS secondaires soient synchronisés.

Dans ce processus, les serveurs DNS secondaires se synchronisent 10 par 10, ce qui peut

durer longtemps lorsqu'ils sont nombreux.

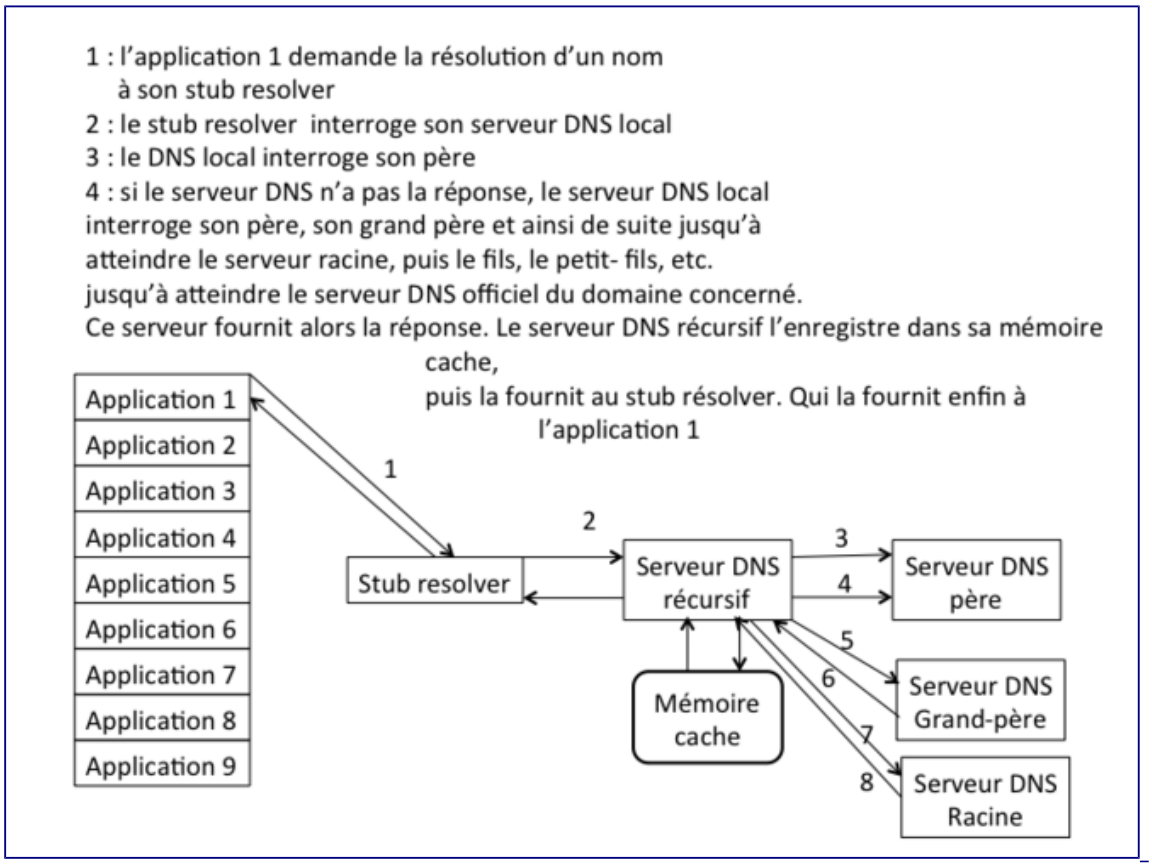


Mise à jour des serveurs DNS secondaires depuis le serveur DNS primaire

Synchronisation à partir du serveur DNS primaire et des serveurs DNS secondaires déjà synchronisés . Dans ce cas, l'administrateur du réseau peut avoir configuré chacun des serveurs DNS secondaires synchronisés pour qu'ils notifient leur changement de numéro de version de fichiers de zone à 10 serveurs DNS secondaires de deuxième niveau.

Ainsi dans les domaines de très grande taille où un grand nombre de serveurs DNS secondaires existe, tous ne peuvent alors pas, en pratique, se synchroniser à partir du seul serveur DNS primaire. La synchronisation 10 par 10 de ces serveurs DNS secondaires prendrait trop de temps.

Pour accélérer la synchronisation. Une fois les dix premiers serveurs DNS secondaires synchronisés, le serveur DNS primaire et chacun de ces dix serveurs DNS secondaires synchronisés peuvent à leur tour prendre en charge 10 serveurs DNS secondaires, soit 110 serveurs DNS secondaires. Et si cela ne suffit pas, les serveurs DNS secondaires synchronisés lors des première et seconde vagues de synchronisation permettent la synchronisation d'une troisième vague de 1210 serveurs DNS secondaires ((1+10+110)*10=1210).



Mise à jour des serveurs DNS secondaires depuis le serveur DNS primaire et les serveurs secondaires déjà synchronisés

Notez que de cette façon, la synchronisation d'un grand nombre de serveurs DNS secondaires est possible rapidement.

Cependant, dans la mesure où un grand nombre de serveurs DNS secondaires se synchronisent simultanément. Une quantité éventuellement importante de bande passante du réseau risque d'être indisponible pendant la phase de synchronisation des serveurs DNS secondaires. Ceci explique la limitation à 10 par défaut du nombre de synchronisations simultanées autorisées au niveau d'un serveur DNS.

L'administrateur du réseau peut modifier cette valeur pour l'adapter à son environnement.

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 ne numéro de version de la base de nommage du serveur DNS primaire n'a pas changé, le serveur DNS secondaire attend un certain temps 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 (au minimum la durée de la synchronisation), à l'expiration duquel il tente à nouveau de se synchroniser.

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.

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.

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.

Il est alors prudent, s'il ne reste plus alors de secondaire, de configurer un ou plusieurs autres serveurs DNS secondaires conservant une copie des fichiers de zone, localement, dans une mémoire non volatile…

Notez également que 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. 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 serveur 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. Et ces serveurs DNS interrogent alors les serveurs DNS de l'Internet pour le compte des serveurs DNS internes.

Serveurs DNS à rôles multiples

Un serveur DNS BIND peut simultanément se comporter comme un serveur primaire pour certaines zones, comme serveur DNS secondaire pour certaines autres zones, et comme serveur DNS récursif pour un certain nombre de clients.

Cependant comme les fonctions des serveurs DNS officiels et récursifs sont logiquement séparées, il est souvent bénéfique de les activer 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.

Spécifications du service de nommage

Rappelons au passage que pour ces applications, une phase lors de laquelle le client DNS local, appelé «stub resolver», interroge son serveur DNS récursif (ou cache) précède la communication. Le serveur DNS récursif effectue les requêtes itératives nécessaires, en partant, s'il le faut, de la racine de l'arbre de nommage et renvoie les ressources demandées.

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.

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.

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.

Les adresses IPv6 sont quatre fois plus grandes que les adresses IPv4 (16 octets). Elles peuvent être attribuées automatiquement ou auto-configurées. Elles sont représentées en notation hexadécimale (double) pointée, par exemple, *2001:db8:330f::beef:cafe:deca:102* .

Tous ces facteurs ont considérablement réduit les chances qu'un humain mémorise ces adresses IPv6. Afin de supporter le nouveau schéma d'adressage d'IPv6, deux extensions DNS ont été définies (RFC 3596): l'enregistrement de ressource AAAA et un nouveau sous-domaine dédié à la résolution inverse (adresse-nom) en IPv6, *ip6.arpa* .

L'enregistrement de ressource AAAA (prononcé «quad A») enregistre les correspondances nom - adresse IPv6. Le code de ce nouveau type d'enregistrement de ressource 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). Un quartet correspond à un chiffre hexadécimal.

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. Nous verrons quelques restrictions dans le chapitre *Deux impossibilités d'accéder au service de nommage* .

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 enregistrement de 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 multidomiciliée ou routeur, par exemple).

Une requête DNS de type AAAA concernant une machine particulière renvoie dans ce cas tous les enregistrements AAAA publiés dans le DNS et correspondant à cette machine.

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:

```
nom [ttl] IN AAAA adresse
```

L'adresse est écrite suivant la représentation classique des adresses IPv6 (RFC 4291) (représentation hexadécimale pointée). 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 et/ou IPv6 correspondant à un équipement 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 l'équipement 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 resolveur 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.

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 inverse au suffixe *ip6.arpa* . Par exemple, l'adresse *2001:660:3006:1::1:1* (adresse de *ns3.nic.fr* donne le nom de domaine suivant:

```
1.0.0.0.1.0.0.0.0.0.0.0.0.0.0.0.1.0.0.0.6.0.0.3.8.0.6.6.0.1.0.0.2.ip6.arpa.
```

L'administrateur de la zone concernée publie alors dans le DNS inverse l'enregistrement PTR correspondant au nom de domaine inverse ci-dessus. Dans cet exemple, l'enregistrement PTR vaut *ns3.nic.fr* .

En pratique, on procède par délégation de zone inverse afin de répartir les enregistrements PTR sur un système hiérarchique de serveurs DNS.

Les données de résolution inverse se trouvent ainsi distribuées sur les différents sites. Ceci facilite la gestion des données de résolution inverse.

Ainsi, pour une zone donnée, l'administrateur de la zone gère localement la base de

correspondance nom-adresse et les bases de données de résolution inverse, à raison d'une par lien dans la zone.

Le fichier de correspondance directe, nom-adresse, et les fichiers de correspondance inverse, adresse-nom, contiennent chacun un numéro de version.

Le numéro de version d'un fichier change chaque fois que l'administrateur en modifie le contenu ou, dans le cas des mises à jour dynamiques, lorsqu'un certain nombre de modifications ont été effectuées ou qu'il s'est écoulé un certain temps.

L'ensemble des fichiers de correspondance directe et inverse constituent, la base de nommage d'un serveur DNS. Le numéro de la base de nommage change dès que le numéro de version d'un de ces fichiers change, c'est-à-dire, dès qu'il a été modifié.

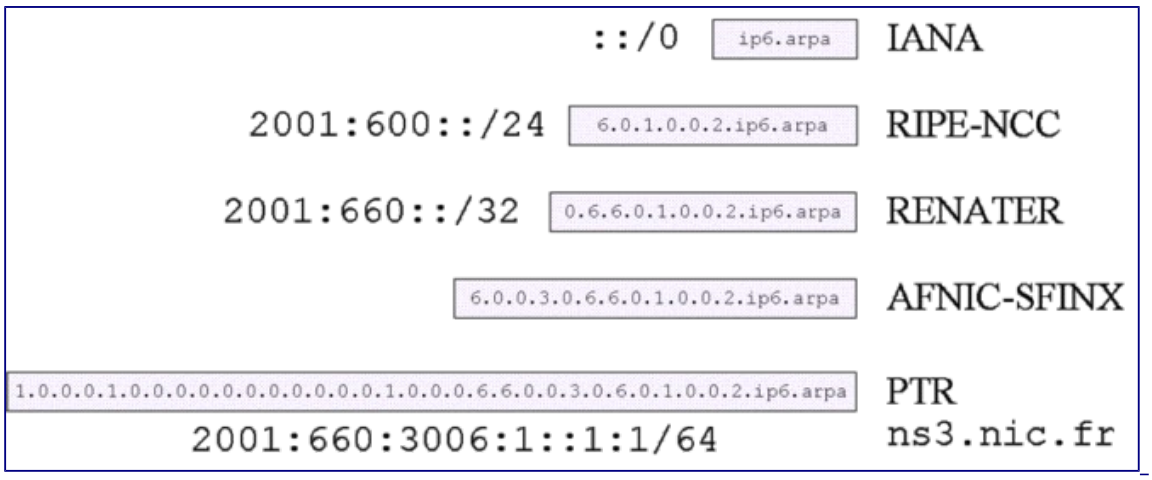
Notez que pour optimiser le processus de synchronisation des serveurs DNS secondaires, il suffit de ne transmettre que les fichiers modifiés.

La délégation DNS inverse suit le schéma classique d'attribution des adresses IP (identique pour IPv4 et IPv6).

1) L'IANA délègue (en termes de provision) de grands blocs d'adresses IPv6 aux registres Internet régionaux (RIR: Regional Internet Registry), typiquement des préfixes de longueur 12 selon la politique actuelle.

2) Les RIR provisionnent des blocs d'adresses IPv6 plus petits pour les registres Internet locaux (LIR: Local Internet Registry), c'est-à-dire aux fournisseurs d'accès Internet locaux, typiquement des préfixes de longueur 32 bits ou plus courts selon le besoin. Notez que dans les régions APNIC et [LACNIC]] des registres nationaux intermédiaires (NIR) existent entre le RIR et les LIR présents dans ces pays.

3) Les LIR attribuent des préfixes IPv6 aux clients finaux. Ces préfixes ont typiquement des une longueur variable entre 48 et 64 bits. La longueur du préfixe varie selon le besoin du client et selon la politique du LIR en vigueur).



Délégation du nommage inverse

La figure montre qu'une liste de serveurs DNS est associée à chaque nœud présent dans le

sous-arbre de nommage DNS inverse.

Cette liste inclut généralement un serveur DNS primaire et un ou plusieurs serveurs DNS secondaires, tous considérés des serveurs DNS officiels pour cette zone DNS inverse.

L'administrateur d'un site responsable du nommage publie (ou non, en fonction de la politique locale) les enregistrements PTR correspondant aux adresses IPv6 qu'il utilise) dans ses zones DNS inverse.

Par exemple, Renater a reçu le préfixe *2001:660::/32* et la délégation de la zone DNS inverse *0.6.6.0.1.0.0.2.ip6.arpa* de la part du RIPE-NCC. Renater a affecté le préfixe *2001:660:3006::/48* à l'AFNIC et lui a délégué la zone DNS inverse correspondante:

```
6.0.0.3.0.6.6.0.1.0.0.2.ip6.arpa. IN NS ns1.nic.fr.
6.0.0.3.0.6.6.0.1.0.0.2.ip6.arpa. IN NS ns2.nic.fr.
6.0.0.3.0.6.6.0.1.0.0.2.ip6.arpa. IN NS ns3.nic.fr.
```

L'AFNIC publie alors dans sa zone DNS inverse les enregistrements PTR correspondant aux adresses IPv6 utilisées. Voici un extrait du fichier de zone DNS:

```
$ORIGIN 6.0.0.3.0.6.6.0.1.0.0.2.ip6.arpa.
1.0.0.0.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.0.0 IN PTR ns3.nic.fr.
```

Découverte de la liste de serveurs DNS récursifs

Pour renforcer le déploiement d'IPv6, la communauté IPv6 a mis en œuvre un mécanisme de découverte automatique des serveurs DNS récursifs avec ou sans DHCPv6.

Trois propositions ont ainsi vu le jour dans le cadre des travaux des groupes «ipv6», «dhc» et «dnsop» de l'IETF.

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.

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 6106 définit deux options d'annonce de routeur: une option qui fournit une liste de serveurs DNS récursifs (RDNSS) et une option pour définir la liste des noms de domaines

recherchés (DNSSL).

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* .

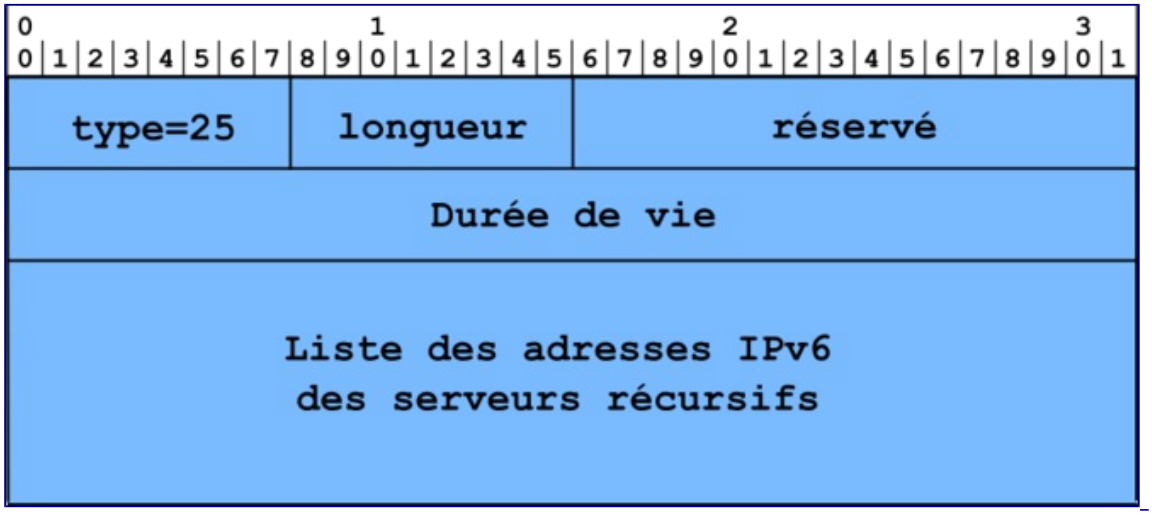
L'autoconfiguration, avec configuration complète du service DNS, sert dans les réseaux dépourvus de serveur DHCPv6 ou pour des machines IPv6 dépourvues de client DHCPv6.

Elle fonctionne sur tout réseau supportant la découverte des voisins. Les configurations du réseau et du service DNS sont alors simultanées.

L'administrateur du réseau configure manuellement les annonces des routeurs pour cette cette autoconfiguration.

Option de liste de serveurs DNS récursifs (RDNSS)

Cette option d'annonce de routeur contient l'adresse IPv6 d'un ou plusieurs serveurs DNS récursifs.



Type Le champ type a pour valeur 25.

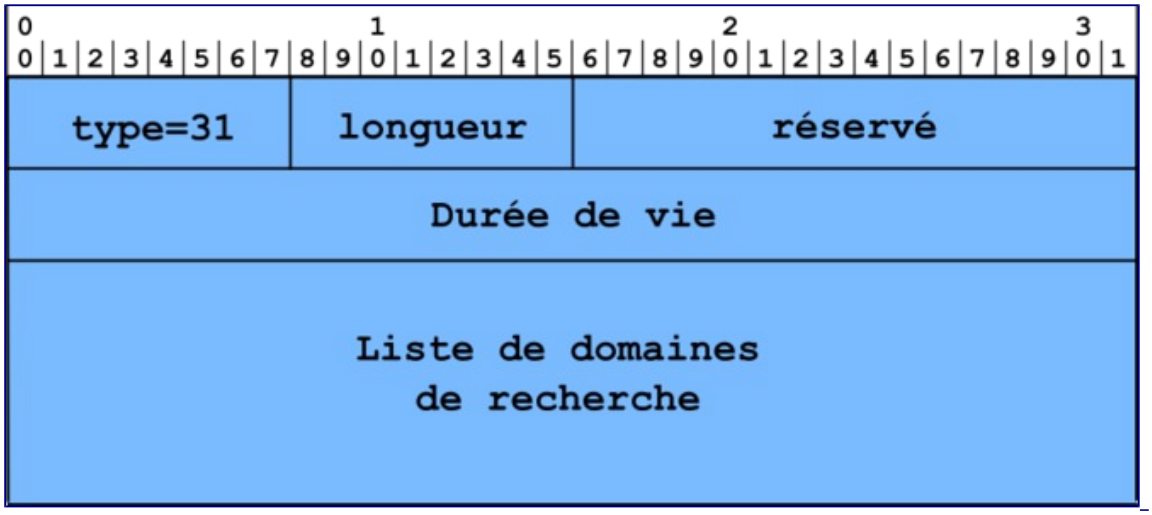
Longueur Ce champ indique la longueur totale de l'option. Les champs types et longueur sont inclus (en multiples de 8 octets). Ce champ permet que l'utilisateur calcule facilement le nombre d'adresses de serveurs DNS récursifs.

Durée de vie . Ce champ indique la durée de vie durée de vie maximum (en seconde) 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.

Addresses Ces champs contiennent les adresses IPv6 des serveurs DNS récursifs, codées sur 128 bits.

Option de liste de domaine recherchés (DNSSL)

L'option DNSSL contient un ou plusieurs suffixes de noms de domaines. 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.



`Type` Le champ type a pour valeur 31.

`Length` Ce champ 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.

`Lifetime` Ce champ 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.

`Noms de domaines` Ce champ contient la liste des noms de domaines à utiliser pour effectuer les résolutions directes.

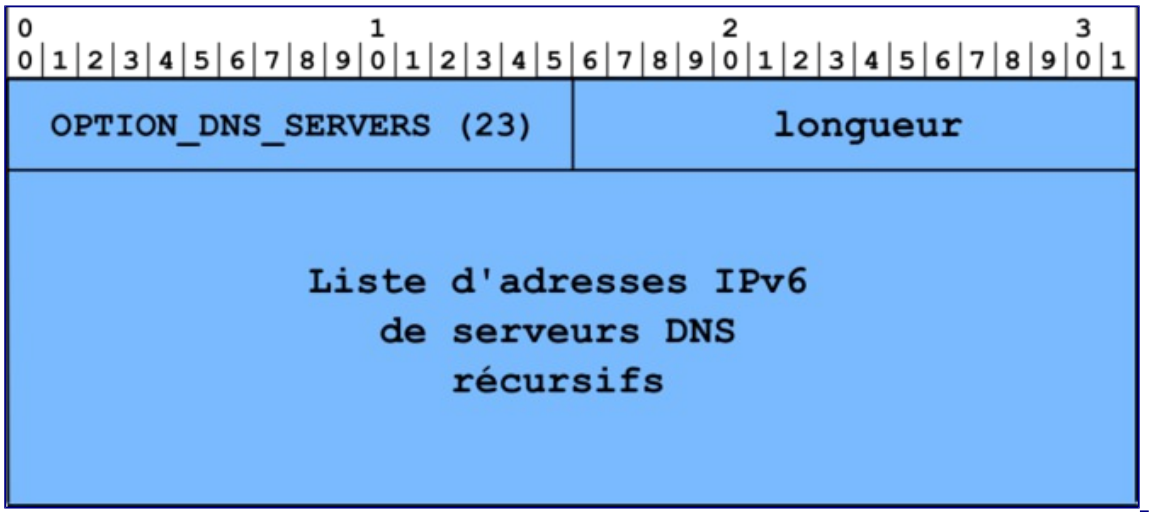
Pour simplifier les choses, les noms de domaines ne sont pas compressés. Les bits excédentaires sont mis à 0.

Extension de la configuration à états, DHCPv6

Le RFC 3315 spécifie le protocole d'autoconfiguration à états, 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.

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.

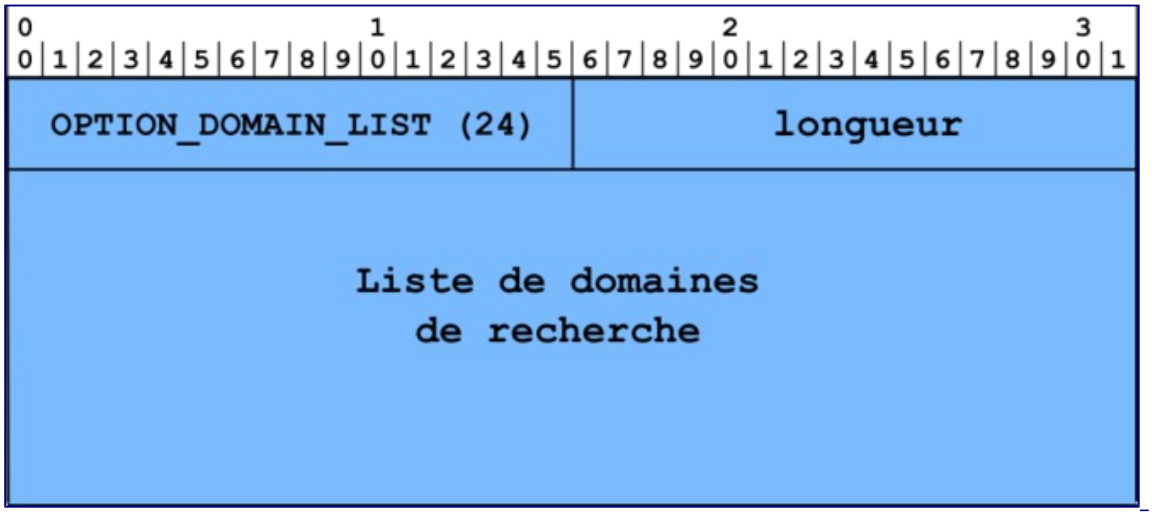


OPTION_DNS_SERVERS Le code option vaut 23.

Longueur La longueur de l'option est exprimée en multiple de 16 octets. La valeur du champ indique le nombre d'adresses de serveurs DNS récursifs contenu dans l'option.

DNS-recursive-name-server . Ce champ contient l'adresse IPv6 d'un serveur DNS récursif. Il peut apparaître plusieurs fois.

Option liste de suffixes de nom de domaine



OPTION_DOMAIN_LIST Le code de cette option vaut 24.

Longueur Ce champ donne la longueur de l'option en octets.

Searchlist Ce champ donne la liste de suffixes de nom de domaine. 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.

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 états et avec états, notamment lorsque plusieurs serveurs sont susceptibles de répondre.

Voici maintenant un résumé des trois propositions: RA, DHCPv6, anycast.

RA . Le mécanisme à base d'annonce de routeur (RA) est spécifié dans le RFC 6106 . 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.

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 propose une utilisation dans le DHCPv6 à états, la seconde propose une utilisation dans le DHCPv6 sans état.

DHCPv6 à états . Un DHCPv6 à états (RFC 3315) 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).

DHCPv6 sans état . Un serveur DHCPv6-lite ou DHCPv6 sans état (RFC 3736) 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, un équipement est en double pile, et s'il 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.

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.

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. 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 PT) et au niveau du transport IPv6 des messages DNS.

Néanmoins, certains ne supportent encore IPv6 qu'au niveau de la base de nommage.

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.

Par exemple, l'ISC: Internet Systems Consortium développe la distribution BIND9. Cette distribution représente la référence de fait dans le domaine. En effet, il s'agit d'un logiciel complet: 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 génétique.

Performances . Les performances sont reconnues: des tests de performances comparant, d'un côté, NSD et BIND, et de l'autre, Unbound et BIND montrent la supériorité respective des premiers sur les seconds).

Diversité génétique . La diversité générique concerne la diversité des plates-formes logicielles

supportant ces serveurs DNS.

Présentation du 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 un serveur DNS secondaire et préparer le fichier de configuration des clients du service de nommage.

Configuration du serveur DNS primaire . 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.

Configuration du serveur DNS secondaire . 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.

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.

Il faut également déclarer, au niveau du serveur DNS primaire, les serveurs DNS secondaires autorisés à se synchroniser.

L'outil *named-checkconf* vérifie les fichiers de configuration du serveurs DNS secondaire.

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 pas 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 (resource record).

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 tous les fichiers de zone.

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.

Notez que les fichiers de zones sont plus faciles à lire s'ils sont documentés. L'ordre des enregistrements n'a aucune importance. Les enregistrements de ressource doivent commencer dans la première colonne d'une ligne.

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.

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* .

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.

Types d'enregistrement de ressource DNS

Les principaux enregistrements de ressource 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.

SOA: Start Of Authority . Cet enregistrement de ressource 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).

NS: Name Server . Cet enregistrement de ressource 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.

MX: Mail eXchanger . Cet enregistrement de ressource désigne un agent de transfert ou un serveur de courrier officiel pour une zone donnée.

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

A . Cet enregistrement de ressource définit une correspondance nom-adresse IPv4.

AAAA . Cet enregistrement de ressource définit une correspondance nom-adresse IPv6.

PTR . Cet enregistrement de ressource définit une correspondance inverse, adresse-nom. Les pointeurs ne désignent que le nom canonique d'une machine.

CNAME . Cet enregistrement de ressource définit un nom canonique ou un 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 l'implémentation de référence en matière de DNS.

Réseau virtualisé utilisé pour générer ces exemples

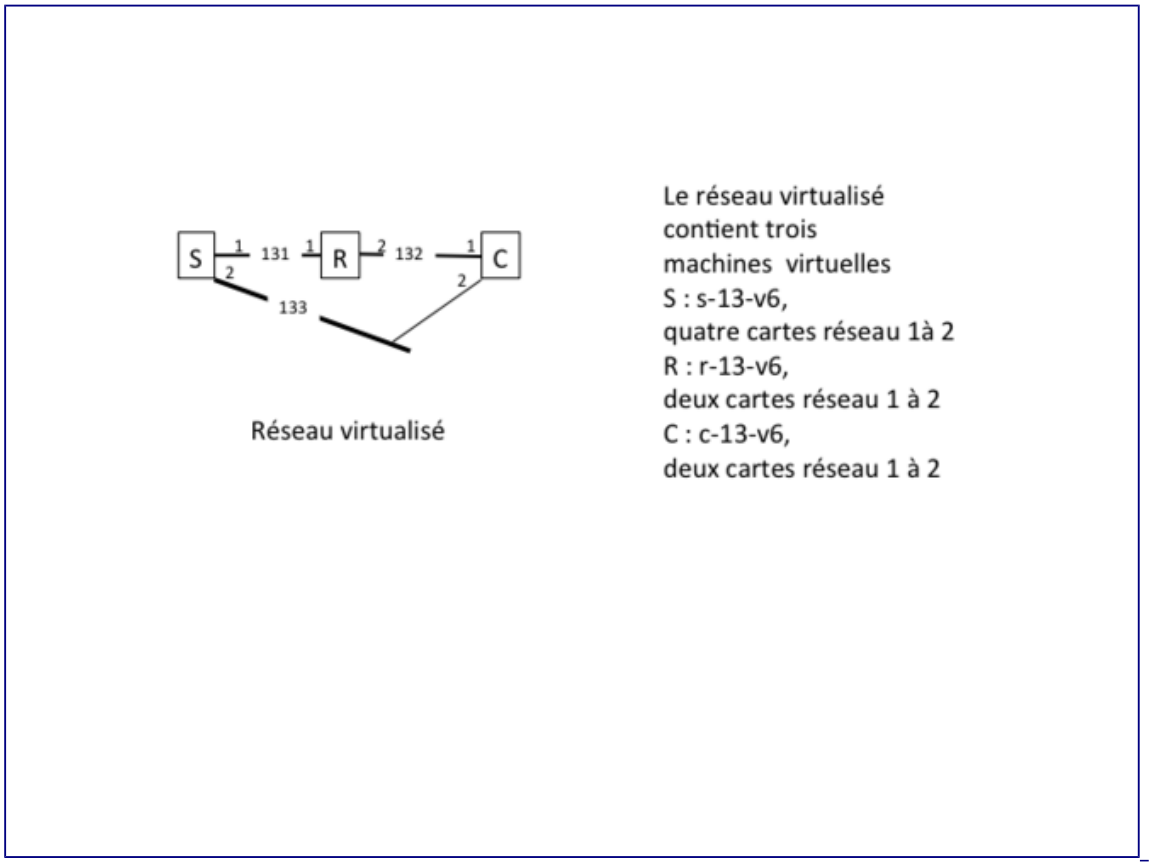
Les exemples de fichiers qui suivent ont été configurés dans un environnement réseau incluant trois machines supportant respectivement un serveur, un relais et un client DNS.

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 états 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 états en présence d'un relais DHCPv6.

Le client, *c-13-v6* , doté de deux interfaces de réseau. La première est connectée d'une part, 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 états.



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.

Il y a, en IPv6, un fichier de résolution inverse par lien dans la zone.

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 comme valeurs possibles:

any . Dans ce cas, le serveur écoute sur toutes les adresses IPv4 opérationnelles.

liste . Une liste explicite comprenant une ou plusieurs adresses IPv4 données indique que, pour

le transport IPv4, le serveur n'écoute les requêtes et réponses DNS que sur les interfaces configurées avec ces adresses IPv4.

none . Dans ce cas, 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* .

L'option *listen-on-v6* peut avoir comme valeurs possibles:

any . Dans ce cas, le serveur DNS écoute sur toutes ses adresses IPv6 opérationnelles.

liste Une liste explicite comprenant une ou plusieurs adresses IPv6 données indique que le serveur DNS le serveur n'écoute les requêtes et réponses DNS que sur les interfaces configurées avec ces adresses IPv6.

none . Dans ce cas, 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;
                };
        };
```

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 6 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
                                        AAAA    2001:db8:330f:a0d4::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         ; expiration (5 semaines 6 jours 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.

3.5.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0 IN      PTR     s-13-v6.tpt.example.com.
7.1.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0 IN      PTR     s-13-v6.tpt.example.com.
7.9.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0 IN      PTR     r-13-v6.tpt.example.com.
```

Fichier db.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         ; expiration (5 semaines 6 jours 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.

7.9.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0 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. root.s-13-v6.tpt.example.com. (
                2               ; Numéro de série
                3600            ; rafraîchissement (1 heure)
                900             ; Nouvelle tentative (15 minutes)
                3600000         ; expiration (5 semaines 6 jours 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 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 nameserver 2001:db8:330f:a0d1::53
nameserver 2001:db8:330f:a0d1::217

search tpt.example.com
```

Outils de vérification de la configurations 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+rl005.12-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 -x 2001: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:
;7.1.2.0.0.0.0.0.0.0.0.0.0.0.0.0.1.d.0.a.f.0.3.3.8.b.d.0.1.0.0.2.ip6.arpa. IN  PTR

;; ANSWER SECTION:
7.1.2.0.0.0.0.0.0.0.0.0.0.0.0.0.1.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
3.5.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.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
7.9.1.0.0.0.0.0.0.0.0.0.0.0.0.0.1.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
7.9.1.0.0.0.0.0.0.0.0.0.0.0.0.0.2.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
7.1.2.0.0.0.0.0.0.0.0.0.0.0.0.0.3.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.

Le DNS est l'application TCP/IP client-serveur qui gère la base de données distribuée à la plus grande échelle qui soit.

Le DNS 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: la fragmentation de l'espace de nommage. Il convient donc, soit de les éviter, soit de trouver les solutions adéquate 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.

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 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 forwarders) 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.

Recommandation. 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.

Recommandation. 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 {liste des adresses des serveurs 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 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 (referal) vers des serveurs DNS racine (TLD), les serveurs DNS qui gèrent les domaines racines (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 se convaincre que la publication des adresses IPv6 des serveurs DNS racines 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 racines 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 à 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 faible (adresses 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 domaine 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: tous les prérequis doivent être satisfaits pour que la mise à jour 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 serveurs 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 de sécurité, 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 au délai de mise à jour des données de zone du serveurs 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 ressource 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 est unique. Il contient les correspondances nom-adresse IPv4 et IPv4 pour toutes les machines de la zone.

Le fichier de 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* . c'es 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 réseau 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 la 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.

Conclusion

Cette séquence a présenté un ensemble de mécanismes permettant de déployer et opérer efficacement un réseau IPv6:

- **ICMPv6** fournit premièrement un protocole complet pour contrôler le bon fonctionnement du réseau et renvoyer si besoin des rapport d'erreur à l'émetteur d'un message problématique.
- La **découverte des voisins** et **l'auto-configuration** avec ou sans état permettent au réseau de se configurer automatiquement sans que l'utilisateur et l'administrateur n'ait besoin d'intervenir trop souvent.
- Enfin le **système de nommage DNS** offre une gestion de correspondance entre nom et IP permettant de mieux identifier les services de l'internet.

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

- l'utilisation d'ICMPv6 pour les rapports d'erreur demande de bonnes pratiques de filtrage
- les annonces de routeurs, garants du bon fonctionnement du réseau local, ne doivent pas être perturbés (voir cet article de S.Bortzmeyer discutant du problème des RAcailles)
- le choix entre auto-configuration avec ou sans état doit être réfléchi
- l'auto-configuration sans-état des postes client 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.

Aujourd'hui les réseaux IPv6 seuls ou déployés conjointement avec IPv4 (double-pile) deviennent de plus en plus courant. Les bonnes pratiques de déploiement et d'administration émergent progressivement. Il est donc important de se tenir informer, de partager et d'adapter ses propres pratiques en fonction des expériences de chacun.