FRÉDÉRIC ROUDAUT

Le protocole IPv6

Degré de difficulté

Le nombre d'adresses attribuées actuellement est proche des limites du protocole IPv4 utilisé pour la communication Internet. Comme le montrent diverses études : on assistera à une pénurie d'adresses à l'horizon 2011. Le protocole IPv6 est la solution standardisée pour palier à ce manque d'adresses.

Depuis les années 80, l'Internet connaît un succès incroyable. La majeure partie des entreprises y est maintenant directement connectée, le nombre de particuliers détenteur d'un abonnement Internet auprès d'un FAI (*Fournisseur d'Accès Internet*) est en constante croissance. La demande est telle que le nombre d'adresses attribuées actuellement est proche des limites du protocole IPv4 utilisé pour la communication Internet. Les études réalisées par les autorités responsables de l'allocation d'adresses ainsi que la commission européenne convergent : on assistera à une pénurie d'adresses à l'horizon 2011. Il était donc nécessaire d'étendre le plan d'adressage. Le protocole IPv6 est la solution standardisée pour palier à ce manque d'adresses.

La définition de ce nouveau protocole IPv6 a été également une opportunité de corriger certains problèmes inhérents au protocole IPv4. Ces problèmes avaient été mis en exergue par la communauté Réseau au cours des dernières années. De nouveaux besoins tels que la sécurité, la mobilité, une facilitation des mécanismes de configuration sont également apparus et ont pu être pris en compte lors de la standardisation d'IPv6.

Ces différentes modifications font d'IPv6 un protocole à part entière et non une simple extension d'IPv4. Il s'agissait donc d'adapter ces corrections sur l'ensemble des protocoles du modèle en couche TCP/IP expliquant ainsi le travail relativement complexe et colossal effectué par l'organisme de standardisation de l'Internet, IETF (*Internet Engineering Task Force*) ces dix dernières années principalement (même si les spécifications initiales d'IPv6 datent de 1988).

À défaut d'espace, toutes les Figures parues dans l'article sont disponibles sur le CD joint au magazine (dossier : articleIPv6).

Adressage IPv6 et nommage

L'évolution la plus visible d'IPv6 concerne l'extension de son espace d'adressage pour palier à la pénurie d'adresse du protocole actuel IPv4. Ce paragraphe vous expliquera le format de ces nouvelles adresses ainsi que le nouveau découpage en classes usité par IPv6.

Format des adresses

Les adresses IPv6 sont constituées de 16 octets (128 bits). On dispose ainsi d'environ 3,4 × 1038 adresses, soit plus de 667 millions de milliards d'adresses par millimètre carré de surface terrestre. Elles sont découpées en 8 mots de 16 bits (4 chiffres hexadécimaux) séparés par des :. En comparaison les adresses IPv4 sont constituées de 4 octets, chaque octet étant noté par sa forme décimale; les différents octets étant séparés par des ..

Exemple : *fe80:0000:0000:0000:0240:96ff:fea7:00d3* est une adresse IPv6

CET ARTICLE EXPLIQUE...

Le nouveau mode d'adressage.

Les mécanismes de communication sous-jacents.

La configuration automatique.

Les différences entre 2 protocoles : Ipv4 et Ipv6.

CE QU'IL FAUT SAVOIR...

Afin d'appréhender au mieux cet article, il est préférable d'avoir des connaissances relativement solides d'IPv4 et en particulier du modèle en couche TCP/IP.

Cette notation pouvant être fastidieuse, les méthodes de simplification suivantes ont été définies :

- La notation :: permet de représenter plusieurs 0 consécutifs au sein de plusieurs mots de 16 bits. Le nombre de 0 peut être retrouvé en examinant le nombre de mots présents dans l'adresse. Cet élément ne peut être présent qu'une fois au sein de l'adresse,
- Au sein d'un mot de 16 bits les chiffres hexadécimaux de poids fort positionnés à 0 peuvent être omis.

La méthode de simplification 1 sur l'exemple précédent nous donne *fe80::0240:96ff:fea7:00d3*.

En appliquant la méthode 2 on obtient l'adresse *fe80::240:96ff:fea7::d3*, qui est beaucoup plus lisible.

En IPv6 on abandonne le format classique de masque usité en IPv4 pour décrire un réseau ou un sous-réseau par le nombre de bits pertinents après le symbole */*.

Exemple :

- *2a01:e35:2EC0:B6A0::/64* décrit un réseau IPv6 composé des 64 premiers bits,
- *FE80::/64* décrit également en format simplifié un réseau IPv6 de 64 bits. Il s'agit en fait de *FE80:0000::/64*.

Généralement, les 64 premiers bits de l'adresse IPv6 servent à l'adresse de sous-réseau, tandis que les 64 bits suivants identifient l'hôte à l'intérieur du sous-réseau.

Les différents types d'adresses

IPv6 définit 3 types d'adresses les adresse Unicast, Multicast et Anycast. Voici leur description :

- les adresses *Unicast* sont destinées : destinées à la communication avec une interface unique,
- les adresses Multicast sont destinées à la communication avec un groupe d'interfaces. La notion de *broadcast* utilisée en IPv4 pour joindre l'ensemble des interfaces d'un lien n'existe plus en IPv6 mais est remplacée par ce type d'adresse beaucoup plus fin,
- les adresses Anycast sont destinées à la communication avec une seule interface d'un groupe donné.

Ces notions seront explicitées par la suite.

Adressage Unicast

Les adresses Unicast sont destinées à la communication avec une interface unique. Ces adresses sont de deux types selon leur portée.

- Les adresses *Lien-local* : destinées à la communication au sein d'un lien,
- Les adresses *globales* : ayant une portée mondiale et destinées aux échanges de l'Internet IPv6.

Similairement à IPv4, on retrouve en plus une adresse de loopback ainsi qu'une adresse indéterminée.

La notion d'adressage public/privé utilisé en IPv4 est revisitée en IPv6. Chaque interface possède une adresse Lien-local ainsi que potentiellement une ou plusieurs adresses globales. L'adresse utilisée sera l'une ou l'autre selon la portée de la communication. Le concept principal d'IPv6 est la communication de bout en bout. Le *NAT/PAT* (*Network Address Translation/Protocol Address Translation*) n'a plus sa place en IPv6. En effet, le NAT bien que considéré comme un moyen de protection et masquage des réseaux posait de sérieux problèmes sur cette communication de bout en bout. Un certain nombre d'artifices sont ainsi généralement utilisés pour pallier à la modification des adresses IPv4 et des ports TCP/UDP (et identifiants ICMP) au niveau des routeurs de bordures. Il est ainsi parfois nécessaire de disposer en sus d'ALGs (*Application Level Gateway*) ou passerelles applicatives pour modifier le contenu des charges utiles des paquets lorsque celles-ci contiennent des informations relatives aux adresses privées. C'est le cas du protocole FTP (*File Transfert Protocol*) par exemple, dans lequel les échanges initiaux de contrôle indiquent au niveau de la charge utile l'adresse ainsi que le port de la session de donnée.

Tableau 1. *Code Fabricant (OUI) sur 3 octets*

Code Fabricant (OUI) sur 3 octets en hexadécimal	Vendeur/Fabricant
00 – 00 – 0C	Cisco
00 – 03 – 93	Apple
02 – 80 – 8C	3COM
08 – 00 – 20	SUN
08 – 00 – 5A	IBM

Tableau 2. *Portée des adresses Multicast*

Valeur	Portée
1	Interface-local
2	Lien-local
4	Admin-local
5	Site-local (actuellement déprécié)
0, 3, F	Réservé
6, 7, A, B, C, D	Non assigné

Tableau 3. *Quelques Groupes Multicast prédéfinis*

Préfixe	Groupe
FF01::1	Tous les nœuds de l'interface
FF02::1	Tous les nœuds du lien
FF01::1	Tous les routeurs de l'interface
FF02::2	Tous les routeurs du lien
FF02::9	Tous les routeurs RIPng

La limitation majeure concerne la sécurisation des échanges de bout en bout. Un chiffrement ou une authentification utilisant l'adresse de l'émetteur devient ainsi invalide dès lors que l'adresse de celui-ci est modifiée par le routeur de bordure. Contrairement aux idées reçues le NAT/PAT n'est pas une sécurité fiable et peut donc facilement être abandonné au profit de mécanisme de sécurisation de bout en bout. LE NAT/PAT est simplement un artifice pour palier à la pénurie d'adresses IPv4.

Identifiant EUI-64

Les adresses *MAC* (*Media Access Control*) sont des identifiants physiques stockés dans les cartes ou les interfaces réseaux afin de permettre un adressage mondial pratiquement unique.

Cette assertion n'est cependant pas garantie, la plupart des drivers permettent maintenant de la modifier manuellement.

Les espaces de nommage suivants, gérés par *l'IEEE* (*Institute of Electrical and Electronics Engineers*) sont couramment utilisés afin d'adresser ces différentes interfaces :

- EUI-64 sur 64 bits,
- MAC-48 sur 48 bits.

Les adresses unicast IPv6 utilisent intensivement une version modifiée de l'EUI-64 (Extended Unique Identifier sur 64 bits) afin de permettre l'identification d'une interface sur un lien. Il est donc requis que ces identifiants d'interface soient uniques au sein d'un préfixe réseau.

Celles-ci sont formées depuis un identifiant EUI-64 (cf. Figure 1) par inversion d'un bit particulier noté u (*universal/local bit*).

Les cartes Ethernet, elles, possèdent un identifiant de la forme MAC-48 (cf. Figure 2) et sont exprimées sous la forme de 12 chiffres hexadécimaux :

- les 3 premiers octets notés *OUI* (*Organizationally Unique Identifiers*) sont administrés par l'IEEE et identifient le constructeur,
- les 3 suivants sont à la charge du constructeur et forment le numéro de série de la carte.

Le tableau 1 vous présente ainsi quelques numéros OUI attribués par l'IANA. Chaque carte réseau vendue par Cisco commence donc par le préfixe 00-00-0C.

Un identifiant EUI-64 modifié est formé depuis cette adresse MAC par inversion du bit u (universal/local bit) et insertion de la valeur hexadécimale sur deux octets FFFE entre le numéro constructeur et le numéro d'interface (cf. Figure 3).

Exemple : L'adresse Mac *00-40-96-A7-C5-D3* donne ainsi l'identifiant d'interface *02-40-96-FF-FE-A7-C5-D3*

Adresses Lien-local

Au niveau d'un lien, les adresses IPv6 sont formées par concaténation du préfixe *FE80::/64* à l'identifiant d'interface au format EUI-64 modifié. La Figure 4 présente un tel type d'adresse.L'unicité au niveau lien de l'identifiant d'interface assure ainsi l'unicité de l'adresse IPv6 Lien-local. Ce type d'adresse ne traverse jamais les routeurs.

Adresses Globales

Les adresses Globales sont formées de manière similaire aux adresses Lien-local par concaténation du préfixe réseau à l'identifiant d'interface au format EUI-64 Modifié. L'unicité au niveau du préfixe de l'identifiant d'interface assurera également l'unicité mondiale de l'adresse IPv6, les préfixes réseaux étant délivrés par des fournisseurs de service ou directement des autorités de régulation. La nomenclature d'une telle adresse a été clairement définie afin de permettre des attributions hiérarchiques de préfixes jusqu'aux sites finaux.

Ce type d'adresse est utilisé pour une communication généralement en dehors d'un réseau local ou LAN (*Local Area Network*) voir à l'échelle de l'Internet Ipv6. L'adresse de loopback (*127.0.0.1* en IPv4) est utilisée pour représenter le nœud lui-même. Elle ne transite jamais sur le réseau. Elle est notée ::1. L'adresse indéteminée est utilisée par exemple lorsque l'interface n'a pas encore connaissance de son adresse (*0.0.0.0* en IPv4). Elle est notée ::.

Adressage Multicast

En IPv4, on dispose de la notion de broadcast afin de permettre la diffusion

Tableau 4. *Champs de l'entête IPv6*

Champs	Taille	Rôle
Version	4 bits	Décrit la version du protocole. Vaut 6 pour IPv6.
Traffic Class	8 bits	Destiné pour faire de la QoS par priorisation, shaping de trafic … ayant pour but d'offrir des fonctions de qualité de service comme Diffserv.
Flow Label	20 bits	Incomplètement spécifié actuellement. Numéro unique choisi par la source, ayant pour but d'offrir des fonctions de qualité de service comme RSVP.
Payload Length	16 bits	Longueur en octet de la charge utile du paquet. En présence d'extension d'entête, ceux-ci sont comptabilisés par ce champ. En IPv4, un champ similaire Total Length comptabilise en plus l'entête ce qui finalement est inutile et limite la taille totale de la charge utile du paquet.
Next Header	8 bits	Décrit l'entête de la couche immédiatement supérieure ou la prochaine extension d'entête. Similaire au champ Protocol en IPv4.
Hop Limit	8 bits	Décrémenté par chaque routeur présent le long du chemin. Le paquet est jeté si ce champ devient nul permettant ainsi d'éviter que le paquet boucle indéfiniment dans le réseau. Similaire au champ TTL en IPv4.
Source Address	128 bits	Contient une adresse unicast de l'émetteur du paquet.
Destination Address	128 bits	Contient l'adresse du ou des destinataires du paquet.

de paquets à l'ensemble des interfaces présentes sur un lien. IPv6 raffine cette notion et lui oppose le concept de multicast pour représenter un groupe d'interfaces potentiellement de portée mondiale. Les adresses multicast (cf. Figure 5) utilisent le préfixe *FF00::/8*.

Le champ *Flag* est composé de 4 bits. Les 3 premiers sont réservés et généralement positionnés à 0, le dernier représente la durée de validité de l'adresse et est positionné à 1 si l'adresse est permanente, le cas échéant il est positionné à 0.

Le champ *Scope* indique la portée de l'adresse selon le Tableau 2.

Les adresses multicast se dérivent encore en 2 groupes :

- Adresses multicast prédéfinies : leur champ Flag vaut 0 et elles sont généralement définies par une autorité telle que l'IANA (Internet Assigned Numbers Authority) chargée de la gestion de l'espace d'adressage IP ainsi que d'autres ressources partagées de numérotation requises soit par les protocoles de communication, soit pour l'interconnexion de réseaux à l'Internet. Le Tableau 3 donne pour exemple certains groupes importants régis par l'IANA.
- Adresses Multicast sollicitées : ce type d'adresse est une fonction des adresses anycast/unicast. Pour chaque adresse unicast ou anycast configurée, une interface écoute sur une adresse multicast de ce type. Ce type d'adresse est formé par combinaison du préfixe *FF02 :0 :0 :0 : 0 :1 :FF00/104* avec les 24 derniers bits de l'adresse unicast/anycast. Nous verrons par la suite que ce type d'adresse permet de limiter la diffusion pour le protocole de *Neighbor Discovery* (découverte des voisins) et en particulier le DAD (*Détection D'adresse Dupliquée*) contrairement au protocole ARP (*Address Resolution Protocol*) d'IPv4. Pour exemple l'adresse IPv6 *4037::01:800:200E:8C6C* donne l'adresse sollicitée *FF02::1:FF0E:8C6C*.

Adressage Anycast

Une adresse anycast permet de représenter un service plutôt qu'une interface donnée. On veut ainsi pouvoir joindre une machine fournissant certains services sans se soucier de la machine contactée. Elle est très peu utilisée pour l'instant et pose naturellement des problèmes de sécurité.

DNSv6

On rappelle que le DNS (*Domain Name System*) permet l'obtention d'un nom plus lisible à partir d'une adresse et inversement à partir du moment où ce nom a été enregistré dans une hiérarchie de serveur DNS. Le format des adresses IPv6 étant naturellement plus long, la question d'une mise à jour du DNS était évidente.

En IPv4, la transformation nom DNS vers adresse IP est définie par un enregistrement nommé *A*. En IPv6, la taille des adresses étant 4 fois plus importante le quadruplé *AAAA* est utilisé.

La capture présentée en Figure 6 montre ainsi une requête DNSv6 sous Windows XP en utilisant la commande `nslookup`. On remarquera en particulier que le serveur DNS est adressé en IPv4 bien que l'on fasse une requête IPv6. Il en va de même pour un serveur IPv6, celui-ci peut très bien répondre à des requêtes pour des adresses IPv4. Si l'on avait ici effectué une requête pour obtenir toutes les adresses de *www.kame.net*, on aurait obtenu l'ensemble de ses adresses IPv4 et IPv6.

La transformation inverse depuis l'adresse vers le nom est similaire en IPv4 et IPv6 et utilise un enregistrement PTR (Seul l'arbre DNS inverse est différent : *in.addr.arpa* pour *ipv4* et *ip6.arpa* pour IPv6).

Il a précédemment été indiqué qu'une adresse IPv6 était formée par combinaison d'un préfixe et d'un identifiant d'interface dépendant de son adresse de niveau liaison de donnée. On comprend donc qu'un changement d'interface impacte potentiellement l'adresse IPv6 de l'interface et de là même les enregistrements DNSv6. Ce souci a amené des réflexions sur la mise à jour sécurisée de ces différents enregistrements au travers d'un protocole baptisé DNSsec (*Domain Name System Security Extensions*). Ce protocole ne sécurise pas uniquement les échanges mais protège également les données ainsi que les enregistrements DNS de bout en bout par une hiérarchie de clés. Chaque domaine signant son sous-domaine … Malheureusement à l'heure actuelle ce protocole est très peu usité. Seuls quelques domaines sont ainsi sécurisés. Le RIPE-NCC (responsable du domaine *in-addr.arpa*) par exemple signe ainsi ses enregistrements avec DNSsec.

Tableau 5. *Valeurs principales du champ Next Header*

Protocoles	Valeur
TCP	6
UDP	17
IPv6	41
ICMPv6	58
No Next Header	59
Encapsulation IP	94

Tableau 6. *Valeurs principales du champ Next Header pour les extensions d'entête*

Protocoles	Valeur
Hop-by-Hop Options Header	0
Fragment Header	44
Destination Options Header	60
Authentication Header	51
Encapsulating Security Payload	50

Entête IPv6

L'entête IPv6 a été soigneusement pensé afin de supprimer les incohérences et problèmes rencontrés en IPv4.

Tableau 7. *Champ de l'extension d'entête Hop-By-Hop Option Header*

Champs	Taille	Rôle
Next Header	8 bits	Décrit l'entête de la couche immédiatement supérieure ou la prochaine extension d'entête.
Hdr Ext Len	8 bits	Longueur de l'entête en mot de 8 bits sans prendre en compte les 8 premiers bits.
Options	Variable	Contient une ou plusieurs sous-options adéquates au protocole usité. La taille de ce champ est telle que l'extension d'entête soit un multiple de 8 octets.

Le *checksum* ou somme de contrôle calculé sur l'entête n'est plus présent. On considère les réseaux suffisamment fiables pour ne pas avoir besoin de vérifier ce champ d'autant plus que dans le cas d'IPv4 celui-ci est vérifié et recalculé par chaque routeur présent le long du chemin. En effet ce checksum inclus le champ TTL (*Time To Live*) décrémenté par chaque routeur lors du transfert du paquet. Ce champ TTL est destiné à éviter qu'un paquet boucle indéfiniment dans le réseau. Ainsi les routeurs rencontrant un paquet avec un champ TTL à 0 se doivent de le jeter. Les protocoles de niveau supérieur (niveau transport) auront la charge de vérifier l'intégrité des paquets,

Les champs relatifs à la fragmentation ont été supprimés de l'entête. En IPv6, es paquets ne sont plus fragmentés le long du chemin mais sont fragmentés par la source. La source se doit donc de connaître le Path MTU (*Maximum Transmission Unit*) ou taille maximum des informations pouvant transiter le long du chemin. Un protocole dédié baptisé *Path MTU Discovery* a donc été soigneusement défini dans cette optique. Afin de faciliter ce protocole un MTU minimum de 1280 octets est exigé sur les différents liens utilisant IPv6,

Les options et en particulier leur alignement étaient assez mal gérés en IPv4 rendant ainsi difficile une gestion hardware de celles-ci. Ces problèmes ont été corrigés en IPv6. Elles sont maintenant baptisées extensions header et sont chaînées entre elles de manière plus cohérente.

Bien entendu les adresses IPv6 étant 4 fois plus grandes, l'entête IPv6 est également plus grand. Il fait 40 octets pour l'entête minimal alors que l'entête IPv4 n'en fait que 20. On constatera néanmoins que les 2 adresses IPv6 de l'entête représentent déjà 32 octets alors que les 2 adresses IPv4 n'en font que 8 (i.e. 24 octets de plus).

La Figure 7 vous présente l'entête IPv6 minimal ainsi que l'entête IPv4 pour comparaison. La signification des différents champs de l'entête IPv6 est précisé dans le Tableau 4.

Chaînage des entêtes et extensions d'entête

Les différents entêtes de niveau supérieur ainsi que les options IPv6 sont chaînés entre eux par l'utilisation d'un champ *Next Header*.

Le Tableau 5 présente quelques-unes des valeurs de ce champ *Next Header* définies par *l'IANA*.

Le nombre d'options minimales et obligatoires implémentées sur les piles IPv6 est moins important qu'en IPv4. On distingue les extensions d'entêtes suivants :

- *Hop-by-Hop Options Header*,
- *Fragment Header*,
- *Destination Options Header*,
- *Authentication Header* (AH),
- *Encapsulating Security Payload* (ESP).

Historiquement IPv6 incluait également un entête baptisé *Routing Header de type 0*, similaire à l'option Loose Source Routing en IPv4. Cette option permet de traverser des routeurs prédéfinis lors de l'acheminement du paquet. Cet entête contenait ainsi une liste d'adresses à traverser et lorsque la première cible indiquée dans l'adresse de destination du paquet était atteinte, celle-ci échangeait son adresse avec la première adresse de la liste et ainsi de suite jusqu'au dernier élément de la liste. Le nombre d'adresses présentes pouvant être relativement important au sein de cette liste, cette option pouvait être plus facilement utilisée en IPv6 pour effectuer des attaques par déni de service en créant des boucles dans ces listes. Cette extension d'entête a donc été dépréciée.

Le Tableau 6 présente les valeurs utilisées dans les champs Next Header précédent afin de permettre le chaînage de ces différentes extensions d'entête.

Ces extensions d'entête ont parfois des contraintes d'alignement. Dans ces cas-là des sous-options de bourrage sont utilisées. La Figure 8 présente les extensions d'entête qui seront explicitées par la suite.

Hop-By-Hop Option Header

Cette extension d'entête est analysée par l'ensemble des routeurs présent le long du chemin.

Le rôle des différents champs est précisé dans le Tableau 7. On pourra se reporter à la Figure 8 pour le format de l'entête.

Fragment Header

Cette extension d'entête est utilisée pour transférer un fragment de paquet, le

Tableau 8. *Champ de l'extension d'entête Fragment Header*

Champs	Taille	Rôle
Next Header	8 bits	Décrit l'entête de la couche immédiatement supérieure ou la prochaine extension d'entête.
Reserved	8 bits	Réservé pour une utilisation future.
Fragment Offset	13 bits	Offset en unité de 8 octets relativement au début de la partie fragmentable du paquet originel.
Res	2 bits	Réservé.
M	1 bit	Positionné à 1 s'il y a d'autres fragments ultérieurs, à 0 sinon.
Identification	32 bits	Identificateur commun à l'ensemble des fragments.

Tableau 9. *Champ de l'extension d'entête Destination Option Header*

Champs	Taille	Rôle
Next Header	8 bits	Décrit l'entête de la couche immédiatement supérieure ou la prochaine extension d'entête.
Hdr Ext Len	8 bits	Longueur de l'entête en mot de 8 bits sans prendre en compte les 8 premiers bits.
Options	Variable	Contient une ou plusieurs sous-options adéquates au protocole usité. La taille de ce champ est telle que l'extension d'entête soit un multiple de 8 octets.

paquet originel étant supérieur au *Path MTU* (MTU minimum entre la source et la destination). En IPv6 la fragmentation est effectuée de bout en bout et non plus dans le cœur de réseau comme en IPv4. Ceci sera reprécisé par la suite avec la notion de *Path MTU Discovery*.

Grossièrement l'émetteur fragmente le paquet originel en petits fragments de taille inférieure ou égale au *Path MTU*. Ces fragments seront réassemblés par la destination avant transmission à la couche de niveau transport.

Le rôle des différents champs est précisé dans le Tableau 8. On pourra se reporter à la Figure 8 pour le format de l'entête.

Destination Option Header

Cette extension d'entête est utilisée pour transférer des informations complémentaires uniquement analysées par la destination.

Le rôle des différents champs est précisé dans le Tableau 9. On pourra se reporter à la Figure 8 pour le format de l'entête.

Authentication Header & Encapsulating Security Payload

Ces extensions d'entête sont utilisées par le protocole IPsec, chargé de la sécurité du paquet. Ces extensions ainsi que le protocole IPsec seront décrits ultérieurement.

Protocoles de niveau Transport pour IPv6 : TCP & UDP

Les protocoles de niveau transport sont légèrement impactés par cette nouvelle version du protocole. Leurs comportements ainsi que leur entêtes restent à l'identique mais étant donné que l'entête IPv6 n'inclut pas de checksum et que les adresses sont plus grandes, quelques modifications sont à prendre en considération.

- En IPv4, UDP n'a pas pour obligation de remplir son champ checksum. Avec IPv6, ce calcul devient obligatoire pour UDP étant donné que l'entête IPv6 n'inclut pas de champ checksum,
- Les calculs de checksum, aussi bien en IPv4 qu'en IPv6 pour UDP et TCP s'effectuent sur l'entête UDP ou TCP suivi de la charge utile du protocole de niveau transport et précédé d'un pseudo-header incluant entre autre les adresses source et destination IP. Ce pseudo-Header étant légèrement différent entre les deux versions, les adresses étant de tailles différentes, le calcul du checksum est lui aussi légèrement modifié. Sans rentrer dans les détails le calcul est le complément à 1 sur 16 bits de la somme des compléments à 1 de tous les mots de 16 bits présents dans cette concaténation d'entêtes.

La Figure 9 présente le pseudo-Header utilisé en IPv4 ainsi qu'en IPv6.

Tableau 10. *Champ de l'entête ICMPv6*

Champs	Taille	Rôle
Type	8 bits	Indique le type de message ICMPv6.
Code	8 bits	Positionné à 0.
Checksum	16 bits	Somme de contrôle afin de détecter des erreurs éventuelles de transmission.
Content	Variable	Spécifique au type de message ICMPv6 usité.

Tableau 11. *Champ Type des messages d'erreur ICMPv6*

Messages d'erreurs	Valeur du champ Type
Destination Unreachable	1
Packet too Big	2
Time Exceeded	3
Parameter Problem	4

Tableau 12. *Champs de l'entête ICMPv6 Destination Unreachable*

Champs	Taille	Rôle
Type	8 bits	Vaut 1.
Code	8 bits	Précise la cause de rejet du paquet : 0 : Pas de route pour la destination, 1 : Interdiction administrative, 2 : Portée de la destination en inadéquation avec la source, 3 : Adresse non joignable, 4 : Port non joignable, 5 : Rejet suite à la politique ingress/egress de l'adresse source, 6 : Rejet suite à une politique de route vers la destination.
Checksum	16 bits	Somme de contrôle afin de détecter des erreurs éventuelles de transmission.
Unused	32 bits	Positionné à 0.
Content	Variable	Contient une partie du paquet responsable de telle manière que la taille totale du paquet ICMPv6 ne dépasse pas le MTU minimum IPv6.

ICMPv6

ICMPv6 (*Internet Control Message Protocol*) est un protocole de niveau 3 sur le modèle en couche TCP/IP, qui permet le contrôle des erreurs de transmission. En effet, comme le protocole IPv6 ne gère que le transport des paquets et ne permet pas l'envoi de messages d'erreur, c'est grâce à ce protocole qu'une machine émettrice peut savoir qu'il y a eu un incident de réseau.

ICMPv6 est plus que le pendant d'ICMP pour IPv4, dans la mesure où il reprend ses spécificités et y ajoute d'autres autrefois subdivisées dans divers protocoles indépendants. On distingue en particulier :

- la résolution d'adresse, la détection d'adresse double … intégrés auparavant dans ARP (*Address Resolution Protocol*) pour IPv4. Ces nouveautés seront par la suite décrites au sein du protocole baptisé *Neighbor Discovery*,
- la gestion de groupes multicast définie auparavant dans IGMP (*Internet Group Management Protocol*) pour IPv4. Ce mécanisme est à présent nommé MLD (*Multicast Listener Discovery*),

La découverte du *Path MTU*, par le mécanisme *Path MTU Discovery*.

Ces différents messages se classifient en 2 catégories :

- Messages d'erreur : notés de 0 à 127 inclus,
- Messages informationnels : notés de 128 à 255 inclus.

Entête ICMPv6

L'entête commun à l'ensemble des messages ICMPv6 est présenté en Figure 10. Le rôle des champs génériques principaux est indiqué dans le Tableau 10.

Messages d'erreurs

Les messages d'erreurs ICMPv6 sont similaires à ceux utilisés en ICMPv4. Ceux-ci sont indiqués dans le Tableau 11.

Les entêtes de ces différents messages d'erreurs sont relativement proches et sont décrites par la Figure 11.

Destination Unreachable

Un routeur ne pouvant transférer un paquet pour une quelconque raison telle que par exemple par manque de connaissance sur la route à emprunter, par cause d'outre-passement de la politique de sécurité devrait générer un tel message à l'entité émettrice avant de rejeter le paquet. La charge utile de ce message contient une partie du paquet responsable de telle manière que la taille totale du paquet ICMPv6 ne dépasse pas le MTU minimum IPv6 (i.e. 1280 octets). En cas de rejet par cause de congestion un routeur ne doit jamais générer un tel paquet, il ne ferait qu'accentuer la congestion.

De la même manière une entité destinatrice peut générer un tel paquet si le protocole de niveau transport ne dispose pas par exemple de serveur en écoute sur le port que l'émetteur cherche à joindre.

Le rôle des différents champs est précisé dans le tableau 12. On pourra se reporter à la Figure 11 pour le format de l'entête.

Packet Too Big

Ce type de message est particulièrement intéressant pour la détection du MTU minimum présent le long du chemin (Cf. Path MTU Discovery). Un routeur devant transférer un message sur un lien ne pouvant le contenir utilise ce type de message en précisant la taille du MTU limitatif pour en informer l'émetteur. La charge utile de ce message contient une partie du paquet responsable de telle manière que la taille totale du paquet ICMPv6 ne dépasse pas le MTU minimum IPv6 (i.e. 1280 octets).

Le rôle des différents champs est précisé dans le Tableau 13. On pourra se reporter à la Figure 11 pour le format de l'entête.

Time Exceeded

Ce type de message est généré par un routeur lorsque le champ Hop Limit du paquet IPv6 à transmettre atteint ou est égal à 0. Les paquets IPv6 présentant une telle spécificité sont jetés.

C'est en particulier ce type de message qui permet de connaître la route utilisée entre 2 points de communication par la commande classique traceroute6. Dans un cadre d'utilisation classique de cette commande, l'émetteur transmet des paquets IPv6 vers la destination en incrémentant le champ Hop Limit à partir de la valeur 1. Le premier routeur recevra donc un tel paquet avec un champ Hop Limit à 1, décrémentera ce champ à 0 et générera un paquet Time Exceeded vers la source ; le second routeur recevra également un paquet avec un champ Hop Limit à 1, décrémentera ce champ à 0 et générera un paquet *Time Exceeded* vers la source … et ainsi de suite jusqu'à la destination finale.

Le rôle des différents champs est précisé dans le tableau 14. On pourra se reporter à la Figure 11 pour le format de l'entête.

Parameter Problem

Ce type de message est généré par un routeur ne pouvant parser un paquet IPv6 suite à une erreur rencontrée dans l'entête ou dans les entêtes d'extension.

Le rôle des différents champs est précisé dans le Tableau 15. On pourra se reporter à la Figure 11 pour le format de l'entête.

Messages informationnels

Les messages d'information ICMPv6 principaux sont indiqués dans le Tableau 16.

Dans ce paragraphe, seuls seront précisés les messages à caractère informatif. Ceux relatifs au Neighbor Discovery seront explicités dans le paragraphe associé.

Echo Request / Echo Reply

Ces paquets sont utilisés comme sonde pour détecter si une machine est joignable ou non.

Lorsque un paquet ICMPv6 Echo Request est transmis à une interface celle-ci doit répondre à la machine émettrice par un paquet *ICMPv6 Echo Reply* en utilisant un adressage source de même portée. Ce type de message est principalement utilisé par la commande classique `ping6`.

Tableau 13. *Champs de l'entête ICMPv6 Packet Too Big*

Champs	Taille	Rôle
Type	8 bits	Vaut 2.
Code	8 bits	Positionné à 0.
Checksum	16 bits	Somme de contrôle afin de détecter des erreurs éventuelles de transmission.
MTU	32 bits	Indique le MTU limitatif.
Content	Variable	Contient une partie du paquet responsable de telle manière que la taille totale du paquet ICMPv6 ne dépasse pas le MTU minimum IPv6

Tableau 14. *Champs de l'entête ICMPv6 Time Exceeded*

Champs	Taille	Rôle
Type	8 bits	Vaut 3.
Code	8 bits	Positionné à 0.
Checksum	16 bits	Somme de contrôle afin de détecter des erreurs éventuelles de transmission.
Unused	32 bits	Positionné à 0.
Content	Variable	contient une partie du paquet responsable de telle manière que la taille totale du paquet ICMPv6 ne dépasse pas le MTU minimum IPv6

Le rôle des différents champs est précisé dans le Tableau 17, le format de l'entête est indiqué dans la Figure 12.

Fragmentation & Path MTU Discovery

On rappelle auparavant que le MTU (*Maximum Transmission Unit*) est la quantité d'information maximum pouvant traverser un lien. Le *Path MTU* est ainsi le MTU minimum du chemin entre la source et la destination.

Il a auparavant été précisé qu'en IPv6 le concept de fragmentation est complètement différent étant donné que le cœur de réseau n'a plus cette tâche. Si besoin est de transmettre des paquets de taille supérieure au Path MTU, la fragmentation est réalisée par l'initiateur des paquets. Afin de limiter les problèmes de transmission de paquets, IPv6 impose un MTU minimum de 1280 octets.

Toutefois cette définition du minimum n'impose en aucune façon que les paquets transmis fassent au plus 1280 octets. La découverte du Path MTU peut s'effectuer à l'aide d'un protocole très simple baptisé Path MTU Discovery. Celui-ci repose principalement sur les paquets ICMPv6 Packet Too Big. Un routeur devant transmettre un paquet d'une taille supérieure au lien doit rejeter ce paquet et envoyer un paquet ICMPv6 Packet Too Big à l'émetteur en lui indiquant le MTU du lien concerné. L'émetteur aura alors à charge de fragmenter le paquet en utilisant les extensions d'entête Fragment Header et de réexpédier ces fragments qui pourront par la suite éventuellement poser problème pour un autre routeur.

Exemple : dans l'exemple de la Figure 13, H1 souhaite transférer 1460 octets de données vers R2. Avec les 40 octets d'entête IPv6, H1 génère un paquet de 1500 octets et le transmet à R1. Or le MTU entre R1 et R2 est de 1280 octets (MTU minimum pour IPv6) ; R1 transmet donc un paquet ICMPv6 Packet Too Big à H1 en lui indiquant ce MTU de 1280 octets ; charge sera alors à H1 de fragmenter ce paquet et d'émettre à nouveau ces fragments. La composition des fragments peut donc être de 1232 octets de données (+40 octets d'entête IPv6 +8 octets d'entête de fragmentation) pour le 1er fragment et 128 octets de données (+40 octets d'entête IPv6 +8 octets d'entête de fragmentation) pour le 2ème fragment.

Au cours du temps ce Path MTU peut bien entendu augmenter à nouveau, parce que la route est modifiée, un tunnel est supprimé, une interface changée … Dans un souci d'un meilleur remplissage des paquets, la source enverra de temps en temps des paquets d'une taille supérieure au Path MTU détecté afin de tester une éventuelle augmentation de celui-ci.

Neighbor Discovery Protocol (Découverte des voisins)

Le protocole de Neighbor Discovery est un protocole indissociable d'IPv6. Il a été conçu pour faire d'IPv6 un protocole plug-and-play. L'idée sous-jacente du Neighbor Discovery est de supprimer toute configuration réseau manuelle des interfaces. Il suffit de connecter l'interface sur un réseau, pour qu'automatiquement, les adresses, la route par défaut, le MTU … soient initialisés. Bien évidemment, il ne s'agit ici que des machines n'ayant pas le rôle de routeur.

Tableau 15. *Champs de l'entête ICMPv6 Parameter Problem*

Champs	Taille	Rôle
Type	8 bits	Vaut 4.
Code	8 bits	Précise la cause du problème rencontré : 0 : Erreur dans un champ de l'entête, 1 : Erreur dans le champ Next Header, 2 : Erreur dans une extension d'entête.
Checksum	16 bits	Somme de contrôle afin de détecter des erreurs éventuelles de transmission.
Pointer	32 bits	Pointeur sur l'octet responsable de l'erreur.
Content	Variable	Contient une partie du paquet responsable de telle manière que la taille totale du paquet ICMPv6 ne dépasse pas le MTU minimum Ipv6.

Tableau 16. *Champ Type des messages informationnels ICMPv6*

Messages Informationnels	Valeur du champ Type	Caractère du message
Echo Request	128	Informatif
Echo Reply	129	
Group Membership Query	130	Gestion des groupes Multicast (MLD)
Group Membership Report	131	
Group Membership Reduction	132	
Router Solicitation	133	Neighbor Discovery
Router Advertisement	134	
Neighbor Solicitation	135	
Neighbor Advertisement	136	
Redirect	137	

Ce protocole regroupe les fonctionnalités suivantes :

- découverte des routeurs présents sur le lien,
- découverte des préfixes du lien,
- découvertes de certains paramètres du lien : MTU …,
- configuration automatique sans état des adresses Lien-local et globale,
- résolution d'adresse,
- découverte des routes par défaut ainsi que des prochains routeurs pour une destination donnée,
- *Neighbor Unreachability Detection* (NUD) : permet de déterminer qu'une entité du lien n'est plus joignable,
- *Duplicate Address Detection* (DAD) : détection d'adresse dupliquée,
- mécanisme de redirection.

Ce protocole n'est pas réellement sécurisé (même si les spécifications initiales laissent supposer une utilisation potentielle conjointe d'IPsec) dans sa version usuelle étant donné qu'il prend place sur un réseau local principalement. Des extensions telles que SEND (*SEcure Neighbor Discovery*) utilisant des signatures RSA sont possibles afin d'améliorer la sécurité de celui-ci.

Ce protocole a cependant l'inconvénient de nécessiter un sous-réseau à diffusion. Les réseaux NBMA (*Non Broadcast Multiple Access*) tel qu'ATM ou X25 nécessitent l'utilisation d'un protocole spécifique comprenant par exemple un routeur disposant d'une connexion point à point avec toutes les interfaces présentes.

Entêtes de messages du Protocole Neighbor Discovery

Le *Neighbor Discovery* s'appuie essentiellement sur la couche ICMPv6 et définit pour cette couche 5 messages : 2 pour la communication entre une interface et un routeur, 2 pour le dialogue entre voisins et 1 pour la redirection (celle-ci souvent non autorisée sera laissée de côté).

Sollicitation et annonces des routeurs

Ces types de messages sont utilisés en particulier pour obtenir :

- l'adresse des routeurs disponibles,
- les préfixes réseaux à utiliser,
- le routeur par défaut,
- le mécanisme de configuration des adresses : avec Etat (DHCPv6), sans Etat,
- la valeur des champs génériques à utiliser dans les paquets IPv6 générés : Hop Limit, MTU du lien,
- les valeurs de certains timers spécifiques : durée de vie d'un routeur, temps de conservation des adresses des voisins…

Dans cet optique 2 messages ont été définis :

- *Router Solicitation* : une machine placée sur un lien envoie spontanément à l'adresse *FFO2 ::1* (tous les routeurs sur le lien) une telle requête afin de disposer des informations nécessaires à sa configuration. Lorsque l'équipement ne dispose pas encore de son adresse, ce type de requête est émis en utilisant l'adresse indéterminée (::) en tant que source,
- *Router Advertisement* : spontanément un routeur positionné sur un lien envoie à intervalles réguliers ce type d'annonce afin de permettre aux machines présentes sur le lien de s'autoconfigurer. Ce type de message est également transmis en réponse à une requête *Router Solicitation*. Dans tous les cas l'adresse source utilisée est l'adresse lien-local du routeur. Selon les cas l'adresse destination est l'adresse de tous les nœuds ou l'adresse de la machine ayant effectué la requête.

Comme plusieurs routeurs peuvent émettre ce type d'annonce, les machines présentent sur le lien pourront ainsi disposer de plusieurs routeurs en cas de panne. Ceci permet également de faire du Multihoming si le site concerné a établit des accords avec plusieurs ISP (*Internet Service Provider*) ou FAI (*Fournisseur d'Accès à Internet*) en français.

L'entête *Router Solicitation* est très proche de celle usitée pour les messages

Tableau 17. *Champs de l'entête ICMPv6 Echo Request/Echo Reply*

Champs	Taille	Rôle
Type	8 bits	Vaut 128 pour Echo Request, 129 pour Echo Reply.
Code	8 bits	Vaut 0.
Checksum	16 bits	Somme de contrôle afin de détecter des erreurs éventuelles de transmission.
Identifier	16 bits	Permet d'identifier le couple Echo Request/Echo Reply.
Sequence Number	16 bits	Permet d'identifier le couple Echo Request/Echo Reply.
Data	Variable	Données éventuelles à l'identique dans l'Echo Request et l'Echo Reply.

Tableau 18. *Champs de l'entête ICMPv6 Router Solicitation*

Champs	Taille	Rôle
Type	8 bits	Vaut 133.
Code	8 bits	Positionné à 0.
Checksum	16 bits	Somme de contrôle afin de détecter des erreurs éventuelles de transmission.
Reserved	32 bits	Positionné à 0.
Options	Variable	Peut contenir par exemple une sous-option Source Link Layer Address indiquant l'adresse MAC de l'expéditeur.

d'erreurs ICMPv6 décrite dans la Figure 11. Le tableau 18 présente les différents champs de l'entête *Router Solicitation*.

Le Tableau 19 présente les différents champs de l'entête Router Advertisement, l'entête quant-à lui, est précisé dans la Figure 14.

Sollicitation et annonces des voisins

Ces types de message sont principalement utilisés pour :

- obtenir l'adresse MAC d'une machine à partir de son IP ou inversement,
- vérifier l'unicité d'une adresse IPv6 avant son utilisation,
- forcer une mise à jour des caches associant adresses MAC et adresses IP (caches NDP).

En IPv4, ces fonctionnalités sont effectuées par le protocole ARP (*Address Resolution Protocol*) ou RARP (*Reverse Address Resolution Protocol*). Une machine voulant envoyer un paquet à une autre (elles peuvent être toutes deux des routeurs) doit obtenir dans un premier temps son adresse MAC. Elle effectue alors une requête ARP. La réponse associée permet de remplir un cache ARP associant adresses MAC et adresses IP. Chaque entrée à une durée de vie.

En IPv6, un tel cache existe également, le cache NDP (*Neighbor Discovery Protocol*). Il contient des adresses Lien-local ou globale mais présentes sur le lien. Deux messages permettent ces différentes fonctionnalités :

- *Neighbor Solicitation* : une machine voulant effectuer une résolution d'adresse envoie sur le lien ce type de requête en unicast. De la même manière, une machine s'initialisant vérifie l'unicité d'une adresse au niveau du lien avant de pouvoir l'utiliser en envoyant ce type de requête à l'adresse multicast sollicitée associée,

- *Neighbor Advertisement* : une interface recevant un *Neighbor Sollicitation* doit répondre avec un *Neighbor Advertisement* soit pour renseigner son adresse MAC, soit pour invalider l'adresse IP qu'une autre interface tenterait d'utiliser. Les *Neighbor Advertisements* peuvent également être émis spontanément pour mettre à jour les entrées des caches NDP.

Le Tableau 20 présente les différents champs de l'entête *Neighbor Solicitation*, l'entête quant-à lui, est précisé dans la Figure 15.

Le Tableau 21 présente les différents champs de l'entête Neighbor Advertisement, l'entête quant-à lui, est aussi précisé dans la Figure 15.

Détection d'Adresse Dupliquée (DAD) & Autoconfiguration sans état

Par simple combinaison des messages du *Neighbor Discovery*, une interface peut s'autoconfigurer automatiquement.

- Une machine s'initialisant sur un lien, construit dans un premier temps son identifiant d'interface EUI-64 modifié à l'aide de l'adresse de sa couche de niveau liaison de données. Puis elle construit son adresse Lien-local temporaire par concaténation du préfixe *FE80::/64* avec cet identifiant,
- Elle a ensuite pour charge de vérifier l'unicité de cette adresse lien-local ainsi que de son identifiant d'interface. Pour cela elle utilise un algorithme de détection d'adresse dupliquée (DAD : *Duplicate Address Detection*). Elle envoie à l'adresse solicited multicast un paquet Neighbor Solicitation avec pour champ adresse de la cible l'adresse provisoire. Ne disposant pas encore d'une adresse validée, elle utilise comme adresse source

Tableau 19. *Champs de l'entête ICMPv6 Router Advertisement*

Champs	Taille	Rôle
Type	8 bits	Vaut 134.
Code	8 bits	Positionné à 0.
Checksum	16 bits	Somme de contrôle afin de détecter des erreurs éventuelles de transmission.
Cur Hop Limit	8 bits	Valeur par défaut que les Machines doivent utiliser pour le champ Hop Limit des paquets générés. La valeur 0 indique que ce champ n'est pas spécifié par le routeur.
M	1 bit	Positionné à 1 pour indiquer que la configuration d'adresse se fait par DHCPv6.
O	1 bit	Positionné à 1 pour indiquer que certains paramètres de configuration supplémentaires sont disponibles via DHCPv6. Ce champ est redondant lorsque le champ M est positionné à 1.
Reserved	6 bits	Positionné à 0.
Router Lifetime	16 bits	Indique la durée de vie (en secondes) de ce routeur en tant que routeur par défaut. Lorsque ce champ est positionné à 0, le routeur ne doit pas être considéré comme le routeur par défaut.
Reachable Time	32 bits	Durée (en millisecondes) pendant laquelle une machine doit considérer qu'une machine est toujours joignable depuis sa dernière détection.
Retrans Timer	32 bits	Temps (en millisecondes) entre 2 retransmissions de messages Neighbor Solicitation.
Options	Variable	Peut contenir par exemple une sous-option : Source Link Layer Address indiquant l'adresse MAC de l'expéditeur, MTU indiquant l'adresse la taille de MTU usitée, Prefix Information indiquant les préfixes réseaux à utiliser.

l'adresse indéterminée. Si elle ne reçoit pas de réponse en retour, cette adresse est considérée comme étant unique, et est donc associée à l'interface. Dans le cas où une réponse, lui parvient, elle ne pourra donc pas utiliser cette adresse, une intervention humaine sera alors indispensable. Il est clair que dans le cas d'une panne de lien, au moment de la réparation de ce dernier un conflit d'adresse pourra être détecté.

- La machine disposant à présent d'une adresse Lien-local, il s'agit donc pour elle d'obtenir une adresse globale routable sur l'internet IPv6. Pour cela, elle dispose de deux possibilités :
- Soit par l'intermédiaire d'un serveur DHCPv6 (autoconfiguration avec état), procédure standard en IPv4,
- Soit par autoconfiguration sans état éventuellement complétée par un serveur DHCPv6.
- Dans le cas de l'autoconfiguration sans état, l'interface cherche à acquérir un Router Advertisement (spontanément ou par un *Router Solicitation*). Ce Routeur Advertisement lui donnera les préfixes réseaux, les routes par défaut, éventuellement le MTU du lien, les durées de validités de certains timers … Elle peut donc ainsi construire à l'aide de son identifiant d'interface, déterminé comme unique, ses différentes adresses globales,
- Dans le cas où le bit O du *Router Advertisement* est positionné à 1, l'interface cherchera à obtenir des informations complémentaires par DHCPv6 (tel que le DNS par exemple).

Le diagramme d'états de la Figure 16 résume ces différentes étapes.

Résolution d'adresse

La résolution d'adresse en IPv6 est comme précisée auparavant identique à celle d'IPv4. Une machine désireuse d'envoyer un paquet à une autre (routeur ou non) doit auparavant résoudre son adresse de niveau liaison de donnée.

Elle effectue ainsi les étapes suivantes :

- Vérification de la présence de l'adresse IP dans le cache NDP. Si celle-ci est déjà présente et qu'elle n'a pas expirée, le processus est achevé, elle peut transmettre le paquet à l'interface en question. Dans le cas contraire, elle émet un Neighbor Solicitation pour l'interface concernée,
- L'interface ainsi atteinte répond par un Neighbor Advertisement afin de renseigner son adresse de niveau liaison de données,
- L'émetteur remplit alors son cache NDP avec le couple (adresse IPv6, adresse MAC) en y ajoutant une durée de validité. L'émetteur peut alors transmettre le paquet en utilisant l'adresse MAC correspondante.

Ces étapes sont graphiquement présentées dans la Figure 17.

Mécanismes de transition & Intéropérabilité IPv4/IPv6 ?

L'ensemble des protocoles de niveau réseau ayant été modifié avec IPv6, vient naturellement la question de l'intéropérabilité avec IPv4; d'autant plus que même les protocoles de niveau transport (UDP, TCP) et applicatif (DNS, FTP …) se sont adaptés pour cette prise en compte de l'augmentation de l'espace d'adressage.

Par défaut les mondes sont ainsi totalement distincts, l'Internet IPv6 étant disjoint de l'Internet IPv4. Cette séparation ayant été évaluée comme l'un des principaux freins au déploiement d'IPv6, l'IETF a originellement mis l'accent sur un certain nombre de mécanismes de transition pour assurer des passerelles entre ces deux mondes. Néanmoins devant le surnombre de propositions, le Working Group de l'IETF ngstrans chargé de la standardisation des mécanismes de transition a finalement considéré qu'il fallait conserver uniquement quelques mécanismes et promouvoir plutôt une migration progressive mais totale des sites en IPv6. Le risque évalué étant que les différents sites restent en IPv4 et utilisent l'un ou l'autre des mécanismes de transition pour accéder aux backbones IPv6.

Dans ce paragraphe nous présenterons succinctement certains des mécanismes de transition les plus usités.

Double Pile IPv4/IPv6

Ce mécanisme est le plus usité actuellement. La majeure partie des systèmes d'exploitation propose maintenant une pile IPv6 en plus de la pile IPv4. Ce mécanisme permet ainsi, selon les besoins, de se connecter à l'Internet IPv6 ou l'Internet IPv4, à la condition bien entendu d'être connecté au monde IPv6. Auparavant seules les entreprises publiques ou les universités avaient accès à ces réseaux. On constate que certains FAI proposent actuellement des accès IPv6. Peu à peu on assiste ainsi à la création de bulles IPv6/IPv4 dans les universités et chez les particuliers. Le problème restant étant de faire migrer également les entreprises pour l'instant réticentes, peut-être par manque de confiance, de compétence, ou tout simplement ne souhaitant pas investir, considérant l'inutilité de faire évoluer leur système bancal mais qui marche encore …

Les topologies ainsi crées avec les doubles piles n'ont pas besoin d'être superposées, les plans d'adressage peuvent être totalement disjoints, les équipements hardwares intégrant également de plus en plus ce mécanisme de double pile, il suffirait dans un premier temps de définir un plan d'adressage et une topologie IPv6 cohabitant avec l'IPv4 et qui évoluerait avec les changements *hardware* et *software*.

Avec l'augmentation de la taille des adresses, bien évidemment les interfaces de communication réseaux ont été également adaptées. Des points d'accès par sockets aux services de la couche transport ont été spécialement définis pour IPv6. Dans un contexte de double pile on pourrait donc imaginer qu'il faille un client/serveur spécifique IPv6 et de même un client/serveur spécifique IPv4 (i.e. un serveur telnet utilisant les sockets IPv4 et un serveur utilisant les sockets IPv6). Ce contexte de double-pile offre cependant une particularité intéressante avec la définition d'un type d'adresse particulier : les adresses IPv4 Mappées. Ces adresses ont le format IPv6 mais incluent l'adresse IPv4. Seules les sockets IPv6 sont ainsi nécessaires, selon l'adressage utilisé le paquet est redirigé ou non vers la pile IPv4.

Passerelle Applicative (ALG : Application Level Gateway)

Le principe des ALGs est assez similaire pour l'ensemble des passerelles applicatives : un proxy équipé d'une double pile permet la cohabitation entre les 2 mondes. Si l'on prend l'exemple d'HTTP (présenté en Figure 18), on peut considérer une machine M sur un site entièrement équipé en IPv6 réalisant une requête en IPv6 pour une URL en IPv4. Le proxy HTTP est lui connecté en IPv6 sur le site et en IPv4 sur l'Internet. Il réalise alors la requête en IPv4 pour l'URL IPv4, récupère la page et la transmet en IPv6 à M. La symétrie est identique, dans le cas où le site est entièrement en IPv4 et la requête pour une URL IPv6.

Ce type de mécanisme permet ainsi de faire cohabiter les 2 mondes et fonctionne dès lors que le protocole concerné de niveau applicatif permet l'utilisation d'un proxy équipé d'une double pile ou nécessite la connexion distante sur un serveur équipé d'une double pile. C'est par exemple le cas des relais DNS, des serveurs POP3, IMAP4, SMTP…

Traduction d'entête : SIIT/NAT-PT

Les mécanismes de traduction d'entête permettent tout simplement comme leur nom l'indique la traduction d'un entête IPv4 vers un entête IPv6 ou inversement (avec éventuellement les entêtes de niveau transport). Le mécanisme le plus complet est SIIT/NAT-PT (*Stateless IP, ICMP Translation Algorithm/Network Address Translation – Protocol Translation*). Même si en théorie SIIT peut fonctionner seul, son utilisation sans état et la clarté de sa spécification en font un protocole difficilement utilisable seul. Conjointement avec NAT-PT (ou l'extension NAPT-PT : *Network Address Port Translation-Protocol Translation*), ce protocole permet donc de faire cohabiter les 2 mondes. Il fonctionne à la manière du NAT, garde des informations d'états, alloue des adresses IPv4 au besoin depuis un pool, fait éventuellement de la translation de port et connaît les mêmes difficultés liées à cette allocation dynamique.

Si l'on prend pour exemple la Figure 19, une machine M dans un réseau IPv6 désireuse de contacter une machine N dans un réseau IPv4, celle-ci transmet un paquet IPv6 vers un boîtier NAT-PT avec pour destination une adresse spéciale IPv6 composée d'un préfixe NAT-PT suivi de l'adresse IPv4 transformée en hexadécimal. Ce boîtier a alors la charge de traduire l'entête en IPv4, d'allouer une adresse IPv4 pour l'émetteur, de remplir une table d'association entre adresse IPv4 allouée et adresse IPv6 de l'expéditeur, de récupérer l'adresse IPv4 du destinataire, pour finalement transmettre ce nouveau paquet sur le réseau IPv4 vers la destination. La réponse potentielle suivra le chemin inverse. La table d'association permettra de retrouver le destinataire effectif de cette réponse.

On comprend donc qu'il est très difficile de maintenir des serveurs sur le réseau IPv6. Cette table d'association doit en effet être statique pour les serveurs potentiels. De plus les informations liées aux adresses et contenues dans les charges utiles devront être également modifiées par ce type de traducteur. D'autres protocoles doivent par conséquent se greffer en plus pour le FTP, le DNS … Ce type de protocole a également du mal à conserver la sémantique des paquets

Tableau 20. *Champs de l'entête ICMPv6 Neighbor Solicitation*

Champs	Taille	Rôle
Type	8 bits	Vaut 135.
Code	8 bits	Positionné à 0.
Checksum	16 bits	Somme de contrôle afin de détecter des erreurs éventuelles de transmission.
Reserved	32 bits	Positionné à 0.
Target Address	Variable	Adresse de la cible sollicitée. Ne doit pas être une adresse Multicast.
Options	Variable	Peut contenir par exemple une sous-option Source Link Layer Address indiquant l'adresse MAC de l'expéditeur.

Tableau 21. *Champs de l'entête ICMPv6 Neighbor Advertisement*

Champs	Taille	Rôle
Type	8 bits	Vaut 136.
Code	8 bits	Positionné à 0.
Checksum	16 bits	Somme de contrôle afin de détecter des erreurs éventuelles de transmission.
R	1 bit	Positionné à 1 pour indiquer que l'expéditeur est un routeur.
S	1 bit	Positionné à 1 pour indiquer que la réponse l'est suite à une requête d'un message Neighbor Solicitation.
O	1 bit	Positionné à 1 pour indiquer que cette réponse doit mettre à jour l'entrée du cache NDP.
Reserved	29 bits	Positionné à 0.
Target Address	32 bits	En réponse à des *Neighbor Solicitation*, contient l'adresse de l'entité ayant effectuée cette requête, sinon contient l'adresse dont l'identifiant d'interface a changé.
Options	Variable	Peut contenir par exemple une sous-option *Target Link Layer Address* indiquant l'adresse MAC de l'expéditeur de ce message.

lors de la traduction, se marie très mal avec les mécanismes de sécurité IPsec, la mobilité de machines (MIPv6), la mobilité de réseaux (NEMO), le multicast … Devant les nombreux problèmes rencontrés, l'IETF a donc décidé de ne pas encourager la mise en œuvre de ce protocole et l'a déprécié.

Tunneling

Ce type de mécanisme repose principalement sur des besoins de communication de sites ou d'îlots isolés IPv6 (respectivement IPv4) sur une infrastructure IPv4 (respectivement IPv6). Le nombre de mécanismes imaginés dans ce contexte foisonne, aucun n'étant réellement satisfaisant. 2 techniques différentes s'opposent ici :

- les tunnels configurés manuellement ou par un fournisseur public (*Tunnel Broker*),
- les tunnels automatiques : 6to4, Teredo, Isatap …

L'idée n'est pas ici de les présenter tous en détail, on pourra se reporter aux spécifications au besoin.

Tunnel brokers

Plusieurs fournisseurs proposent ainsi des interfaces WEB permettant après inscription la configuration d'un tunnel IPv6 sur IPv4 vers le site de l'intéressé. Il suffit donc de configurer manuellement ou par des scripts fournis l'autre partie du tunnel jusqu'au routeur du fournisseur pour accéder à l'Internet IPv6. C'est finalement la méthode la plus appropriée pour joindre l'Internet IPv6 pour un particulier dont le FAI ne propose pas d'IPv6.

6to4

Ce type de mécanisme est principalement usité pour permettre la communication entre îlots IPv6 sur une infrastructure IPv4 tout en permettant un accès à l'Internet IPv6. Le principe est relativement simple : chaque îlot isolé, qualifié de réseau 6to4 dispose en bordure d'une passerelle 6to4 équipée d'une double pile. Chaque îlot utilise un préfixe particulier qualifié de préfixe 6to4 formé de la manière suivante : *2002:: Adresse IPv4 du Relais ::/48*

Dans la Figure 20, une machine M désireuse de communiquer avec une machine N d'un autre réseau 6to4, transmettra donc les paquets IPv6 à sa passerelle par routage. Le champ destination indiquera une adresse formée d'un préfixe 6to4 incluant l'adresse IPv4 de la passerelle de destination. Ce paquet sera ainsi automatiquement encapsulé dans un paquet IPv4 avec comme adresse source l'adresse de la passerelle émettrice et comme adresse destination celle de réception. La passerelle réceptrice désencapsulera ce paquet avant de le transmettre sur son îlot. Chaque passerelle pointe également vers un relais par défaut connecté à l'Internet IPv6.

Ce mécanisme simple à mettre en œuvre a de nombreuses failles de sécurité : en particulier il est sensible au déni de service et n'offre pas de contrôle sur le trafic reçu. Une passerelle 6to4 a en effet la possibilité de vérifier la cohérence d'adressage entre l'entête IPv6 encapsulée et l'entête IPv4 dans le cas où l'émetteur est une passerelle 6to4 mais cette vérification est presque impossible si la source est une adresse native. Dans ce cas-là le paquet sera transmis sur le réseau IPv6 protégé par la passerelle, sans certitude qu'il ne s'agit pas d'un paquet forgé. Le risque est d'autant plus important que l'adresse IPv4 de l'émetteur sera probablement perdue après le transfert.

Quelques autres protocoles …

Teredo (*Tunneling IPv6 over UDP through NAT*) est assez similaire à 6to4. Il définit une méthode permettant d'accéder à l'Internet IPv6 derrière un équipement réalisant du NAT en encapsulant les paquets IPv6 dans de l'UDP sur IPv4 entre le client et le relais Teredo à l'aide d'un serveur Teredo.

ISATAP (*Intra-Site Automatic Tunnel Addressing Protocol*) permet la connectivité IPv6 au dessus d'une infrastructure IPv4.Il supporte un réseau NBMA (*Non Broadcast Multicast Access*), génère des adresses Lien-local depuis les adresses IPv4 et permet l'utilisation du protocole *Neighbor Discovery* au-dessus de cette infrastructure IPv4.

Conclusion

De nombreux mécanismes de transition ont été définis pour permettre :

- la cohabitation des 2 mondes,
- la communication entre les 2 mondes,
- la communication entre îlots isolés IPv6 (respectivement IPv4) dans une infrastructure IPv4 (respectivement IPv6),
- la communication entre un îlot isolé IPv6 (respectivement IPv4) et l'Internet IPv6 (respectivement IPv4).

Aucun de ces mécanismes n'est pleinement satisfaisant, mais ils n'ont pas pour vocation d'exister durablement. Ils devraient décroître dans le temps en fonction du nombre d'équipements IPv6 présents sur le réseau.

Afin d'anticiper le passage à IPv6, les applications réseaux futures devraient déjà prendre en compte ce nouveau mode d'adressage et en particulier utiliser les sockets IPv6 qui dans tous les cas permettent la communication avec les 2 mondes. Les anciennes applications doivent également être adaptées dans cet esprit. Bien entendu, il sera difficile de faire migrer celles dont on ne dispose plus des sources.

Vous trouverez dans le Tableau 23 les références aux normes décrites au sein de cet article.

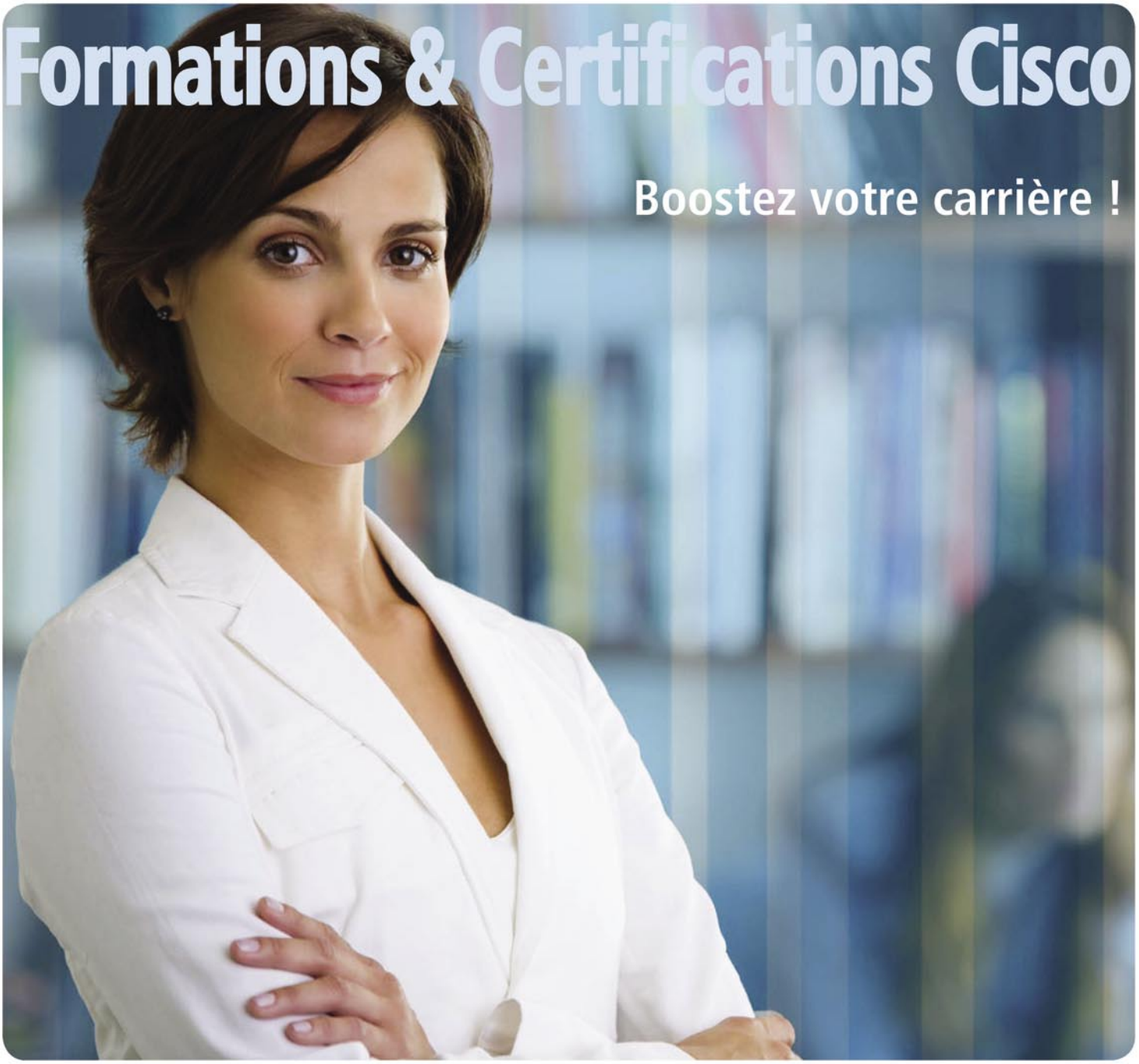
Références

Voici quelques références glanées sur Internet :

- *http://standards.ieee.org/regauth/oui/tutorials/EUI64.html* – Guidelines For 64-Bit Identifier (EUI-64) Registration Authority,
- *http://standards.ieee.org/regauth/oui/oui.txt* – OUI définis par l'IEEE,
- *http://www.iana.org/assignments/ipv6-multicast-addresses* – IPv6 Multicast Adresses,
- *http://www.iana.org/assignments/protocol-numbers/* – Protocol Numbers,
- *http://livre.point6.net/index.php* – IPv6 Théorie et Pratique – Gisèle Cizault.

À propos de l'auteur

Frédéric Roudaut travaille actuellement chez Orange Labs (anciennement France Telecom R&D) à Sophia Antipolis pour le compte d'Orange Business Services IT&Labs depuis 1 an et demi.
Pour contacter l'auteur : *frederic.roudaut@free.fr*



Global Knowledge est le centre de formation agréé Cisco le plus important, aussi bien au niveau national, européen que mondial. Nous vous proposons le catalogue de formations Cisco le plus complet du marché, représentant 95% des technologies Cisco dans les domaines suivants:

- **Routage & Commutation**
- **Management de Réseaux**
- **Sécurité**
- **Communications & Services**
- **Téléphonie & VoiP**
- **Stockage & Data Centers**
- **Wireless**
- **Préparation CCIE**

Pour plus d'information sur notre offre de formation et de certification Cisco, rendez-vous sur notre site internet : **www.globalknowledge.fr/Cisco** ou contactez un conseiller formation au **+33 (0)1 78 15 34 00**

Global Knowledge™