MOOC
Objectif IPv6 !
vers l'internet nouvelle génération

Document Compagnon[1]

Séquence 4

L'intégration d'IPv6 dans l'Internet

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

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

Les auteurs

Bruno Stévant

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

Jacques Landru

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

Jean-Pierre Rioual

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

Pascal Anelli

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

Joël Grouffaud

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

Pierre Ugo TOURNOUX

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

Remerciements à :

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

Table des activités

Les auteurs **5**

Introduction **9**
Références bibliographiques 9

Activité 41: Déployer IPv6 maintenant **11**
Où en est IPv4? 12
Motivations à IPv6 15
Où est est IPv6? 16
Quel scénario pour le déploiement? 19
Principes des mécanismes d'intégration 19
Double pile 20
Tunnel 21
Traduction 21
Conclusion 23
Références bibliographiques 24
Pour aller plus loin 24

Activité 42: Déployer IPv6 dans un réseau **27**
Introduction 27
Technique de la double pile 27
Plan de migration originel 28
Étude et préparation du déploiement d'IPv6 30
Méthode 30
Vérification de la disponibilité d’IPv6 32
Obtenir un préfixe IPv6 34
Préfixe ULA 34
Préfixe GUA 36
Définition du plan d’adressage de sous-réseau avec IPv6 36
Déploiement des équipements en double pile 38
Configuration d'adresses 38
Résolution d’adresses 39
Administration du réseau 39
Déploiement d'IPv6 pour les services 40
Les adresses IPv4 imbriquées dans une adresse IPv6 40
Au niveau des applications 41
Problèmes liés à la double pile 42
Conclusion 45
Références bibliographiques 46
Pour aller plus loin 46

Activité 43: Établir la connectivité IPv6 ..... **49**
Problématique 49
Principe du tunnel IPv6 sur IPv4 49
Tunnel configuré 51
Tunnel automatique 51
Connectivité d'un site isolé: Tunnel Broker 54
Connectivité sur une infrastructure IPv4: 6rd 56
Conclusion 59
Références bibliographiques 60

Pour aller plus loin 60

Activité 44: Interopérer les applications par traduction 61
Contexte d'utilisation de la traduction 61
Principe de la traduction entre protocoles IP 62
Transposition protocolaire des champs de l'en-tête (RFC 7915) 63
Les adresses pour les traducteurs d'adresse NAT64, NAT46 (RFC 6052) 64
Traduction des adresses 65
Mécanismes complémentaires 67
Traduction des paquets ICMP 67
Relais-traducteur DNS auxiliaire (RFC 6147) 67
Mécanisme de transition NAT64/DNS64 68
NAT64: traduction "sans état" RFC 7915 69
NAT64: traduction "avec état" RFC 6146 69
Conclusion 72
Références bibliographiques 73
Pour aller plus loin 73

Activité 45: Interopérer des applications par passerelles applicatives 75
Contexte d'utilisation des passerelles applicatives 75
Principe des passerelles applicatives 75
Cas du service Web 76
ALG placée du coté du client 77
ALG placée du coté du service 77
Déploiement d'un relais inverse 78
Utilisation d'un service d'hébergement ou de distribution des contenus 79
Conclusion 80
Références bibliographiques 81
Pour aller plus loin 81

Conclusion 83
Pour en savoir plus 84
Remerciements 84

Introduction

Cette séquence d'activités traite du thème du déploiement d'IPv6. Elle part des limitations d'IPv4 introduites par la pénurie d'adresses. Cette pénurie a profondément changé la nature de l'Internet. Elle a introduit une complexité et un coût de connectivité grandissant. Il devient de plus en plus évident, avec l'apparition des nouveaux réseaux, qu'IPv4 devient inadapté pour répondre aux besoins d'interconnexion. Après près de 50 ans d'existence, IPv4 a atteint la fin de ses possibilités et devient problématique dans le développement de l'Internet. Au sein de l'IETF, il y a des voix qui s'expriment pour rendre IPv4 obsolète. Cette volonté se concrétise début 2016 par la publication d'un document de travail qui prône de rendre IPv4 historique [1] . Ce document illustre bien qu'IPv4 est limité et qu'il est temps de passer à IPv6. Car c'est bien dans les limitations d'IPv4 que la motivation au passage d'IPv6 est à trouver. Nous expliquerons en quoi IPv6 est indispensable pour le développement des services innovants et en quoi IPv6 permet de retrouver les principes qui ont fait le succès de l'Internet comme, notamment, une connectivité simplifiée.

Au cours de cette séquence, nous rappellerons le plan de migration vers IPv6 initialement planifiée. Nous exposerons aussi la méthode du passage à IPv6. Enfin, pour chacun des problèmes soulevés par la migration à IPv6, nous détaillerons les propositions adaptées. Nous conclurons cette séquence par un exercice de mise en oeuvre pratique.

Références bibliographiques

1. ↑ Pépin G. (2016) Article en ligne sur Next Inpact. Un brouillon de RFC propose de déclarer l'IPv4 obsolète.

Activité 41: Déployer IPv6 maintenant

Généralement, l'identification d'une *killer application* est recherchée pour justifier un passage rapide vers IPv6. Ce fut le cas avec IPv4 quand le Web est apparu. Les sites sont massivement passés de protocoles propriétaires (IPX, NetBEUI) vers IPv4 pour accéder aux informations par un navigateur; ce qui a conduit au concept d'intranet. On ne connaît pas actuellement d'application particulière pouvant forcer massivement le passage vers IPv6. Les fonctionnalités avec IPv4 sont les mêmes, puisqu'il ne s'agit que d'une nouvelle version du protocole IP. La qualité de service est souvent évoquée, mais il s'agit d'un leurre, car les mécanismes de réservation ou de différenciation sont pris en charge par les deux versions du protocole. Il n'y a pas une fonctionnalité qu'aurait IPv6 qui ne soit pas dans IPv4. Il peut y avoir des simplifications apportées, comme dans la configuration d'un réseau. Mais ce genre d'avantage ne justifie pas le coût de la migration d'IPv4 vers IPv6. Les raisons poussant au passage à IPv6 ne sont pas à chercher du coté de la demande mais trouvent leurs origines dans les limitations d'IPv4.

Il n'y a plus de préfixe réseau public disponible ni, a fortiori, d'adresse publique. Or, l'adresse est un élément indispensable à la connectivité au réseau Internet. Sans adresse, un noeud est invisible. Il ne peut rien recevoir ni envoyer, et rend toute communication impossible. La demande de connectivité à Internet, autrement dit d'adresses, loin de diminuer, va au contraire s'accélérer dans les prochaines années avec les nouvelles applications telles que la domotique et la route intelligente. Ces dernières impliquent une masse importante d'objets numériques connectés. Ces applications se développent en IPv6, car IPV4 n'a pas les capacités pour les supporter. Il n'est pas adapté pour interconnecter la multitude des composants numériques: son plan d'adressage à 2^32, soit environ 4.3 milliards, adresses, est trop restreint. Il n'aurait même pas assez d'adresses pour chaque être humain sur la planète, même si l'allocation d'adresses était parfaite.

Cette taille insuffisante du plan d'adressage n'est pas due à une erreur des concepteurs d'IPv4 mais provient du progrès technologique. Le paradigme de l'ordinateur a beaucoup évolué depuis les années 60. Au début, il y avait un ordinateur par organisation. Puis il y a eu un ordinateur par département. Ensuite, l'arrivée de la micro-informatique a amené un ordinateur par personne. Enfin, avec la généralisation du numérique dans divers objets du quotidien, on en arrive à plusieurs ordinateurs (machines ou objets connectés) par personne. L'espace d'adressage IPv4 de l'Internet est devenu insuffisant et n'est plus capable de répondre au besoin d'interconnexion des ordinateurs. IPv6 vise justement à répondre à ce changement. De plus, IPv6 possède quelque chose que IPv4 n'a plus: c'est le principe fondamental de bout en bout. Ce principe a été perdu avec le changement de l'architecture de l'Internet, entraîné par le manque d'adresses, comme nous allons le voir. Pour les applications et l'extensibilité du réseau, ce principe peut tout changer. La véritable motivation du passage à IPv6 se situe à ce niveau: avoir un Internet adapté à l'informatique d'aujourd'hui.

Où en est IPv4?

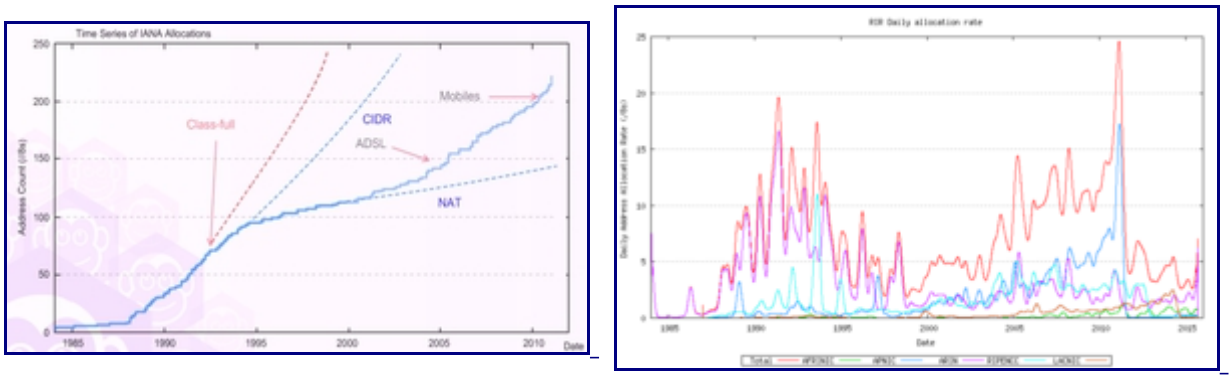
L'Internet vit depuis des années en situation de pénurie d'adresses. Cette pénurie d'adresses a été prédite dès le milieu des années 1990, peu après la naissance du web. Des mesures palliatives ont été prises pour ralentir la consommation des adresses et ralentir l'apparition de la pénurie complète des adresses IPv4. La première mesure a été de retenir une méthode plus efficace d'attribution des adresses IPv4 en s'appuyant sur des longueurs de préfixe réseau de taille variable. Ce changement connu sous le nom de CIDR (*Classless Inter-Domain Routing*) n'était pas suffisant. Il fallait toujours une adresse IP par noeud se connectant à l'Internet. La seconde mesure a été de restreindre l'attribution des adresses aux noeuds par une allocation temporaire et non plus permanente. Ceci revient plus exactement à partager, dans le temps, une adresse IP entre plusieurs noeuds. Ce partage des adresses a validé le constat qu'il y a bien une pénurie d'adresses dans l'Internet. En pratique, le partage des adresses IPv4 a été possible avec l'introduction d'un nouveau dispositif: le NAT (*Network Address Translation*) [RFC 2663] et le recours à l'adressage privé [RFC 1918], comme le préfixe `192.168.0.0/16` largement utilisé dans les accès des particuliers.

Plan d'adressage privé IPv4 RFC1918

Le plan d'adressage privé [RFC 1918] réserve des préfixes pour des réseaux de différentes tailles qui sont dans l'ordre décroissant: 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16. Ces préfixes sont non routables sur l'Internet public, mais les réseaux issus de ces préfixes peuvent être routés sur des topologies privatives (réseaux de campus, réseaux d'entreprise, réseaux domestiques...).

Un ensemble de noeuds derrière le NAT et identifié par l'adressage privé (routable sur une topologie privative) se partage une ou plusieurs adresses IP globales (aussi appelés adresses publiques, routables sur l'Internet public). Le NAT est une fonction de la "box" (routeur résidentiel) que chacun utilise à domicile pour accéder à Internet. Le NAT remplace dynamiquement les adresses privées par des adresses globales dans un sens et inversement dans l'autre sens. Lorsque qu'il n'y a qu'une simple adresse IP globale de disponible, à partager entre plusieurs machines d'adresse privée, la mise en correspondance avec cette adresse globale nécessite d'utiliser le numéro de port. Dans ce cas, en plus de traduire l'adresse, le NAT change aussi le numéro de port, on parle alors de NAPT (*Network Address and Port Translation*).

La figure 1 représente le cumul des adresses IPv4 consommées et l'effet des mesures de réduction de consommation des adresses. [1] . Les adresses IPv4 sont exprimées par le préfixe de longueur 8 bits. Cette figure montre bien une diminution du taux de consommation des adresses IPv4. Du temps était ainsi gagné pour promouvoir une solution définitive. Mais le développement de l'Internet dans la téléphonie mobile et la banalisation des accès ADSL ont accéléré la pénurie. Le graphique (b) de la figure 1 montre que, depuis 2011, la pénurie est aigüe par cette chute du taux de consommation des adresses.



(a) **(b)**

Figure 1: Cumul de consommation des adresses IPv4 et taux de consommation.

Notation "/8"

Dans les diagrammes montrant l'usage des adresses IPv4, celles-ci sont agrégées par "/8". Comme l'espace d'adressage IPv4 est un champ de 32 bits, il y a 4 294 967 296 valeurs uniques représentées dans ce contexte par une séquence de 256 "/8" bits où chaque "/8" correspond à 16 777 216 adresses uniques.

Cependant, la solution du NAT rend la connectivité Internet coûteuse et complexe. Le code réseau des applications devient de plus en plus complexe et donc coûteux à développer du fait des techniques de contournement à mettre en œuvre pour que les applications retrouvent une connectivité globale (à savoir, pouvoir être appelées ou appelantes). De plus, le NAT introduit un état dans le réseau qui fragilise la robustesse du système de communication. Il convient ici de ne pas oublier qu'un principe fondateur de l'Internet est de rendre le fonctionnement de l'infrastructure de communication indépendante du fonctionnement des producteurs et consommateurs de données. Ce principe connu sous le nom de "bout en bout" a conduit à définir le service réseau en mode "non connecté". Aucune marque ou état, issu d'une communication, ne se matérialise dans le réseau: tout est indiqué dans le paquet. On parle d'unité de transfert auto-descriptive. L'en-tête du paquet comporte toutes les informations pour aller de la source à la destination. Le NAT est en complète contradiction avec ce principe. Le paquet n'est plus auto-descriptif de la source à la destination car chaque passerelle NAT traversée modifie les informations de l'acheminement du paquet. On peut considérer que chaque NAT traversé conduit à constituer un tronçon du chemin pour atteindre la destination. C'est cette succession de tronçons qui devient le chemin de la source à la destination. On peut voir que, d'une infrastructure de communication de bout en bout, l'Internet a évolué vers une infrastructure de communication devant gérer des changements de tronçons. Or, ces changements de tronçons demandent des états complexes à gérer en mode "non connecté", ce qui rend le système fragile. En effet, une panne d'un NAT suffit à interrompre toutes les communications le traversant, ce qui n'est pas le cas quand cela arrive à un routeur. Certes, des solutions existent, à base de redondances de NAT, pour maintenir la disponibilité de ce dispositif. Ces solutions sont coûteuses et complexes à mettre en oeuvre et ne constituent pas le cas courant.

L'introduction du mécanisme NAT a changé l'architecture de l'Internet: il n'y a plus de bout en bout [RFC 2993]. La conséquence est que déployer des nouveaux services ou des nouveaux

protocoles de transport est devenu quasi impossible. Car, non seulement NAT change l'adresse IP, mais il modifie souvent aussi le numéro de port situé au niveau de la couche de transport, ce qui a pour conséquence de figer les protocoles de transport actuels. L'ajout d'un nouveau protocole de transport nécessite de mettre à jour le code de tous les NAT en activité, ce qui représente une opération quasi impossible du fait de la diversité des NAT et de leur nombre. Cette idée de rigidification de l'Internet est nommée par le terme d'"ossification". Devant cet état de fait, des réflexions sont menées dans les instances de la gouvernance Internet pour essayer de sortir de cette impasse [RFC 7663].

Le modèle d'interaction se trouve aussi, d'une certaine manière, rigidifié. Dans le modèle d'interaction client-serveur, les clients qui sont derrière le NAT peuvent s'accommoder de partager une simple adresse IP. Il en est tout autrement pour les serveurs qui ont besoin d'une adresse IP qui leur soit propre afin d'être contactés. Ainsi, ce changement architectural de l'Internet l'a transformé petit à petit en un système minimaliste à l'image des services télématiques utilisés à l'époque du minitel. Il est composé de clients et de serveurs. Les possédants d'un adressage public ont ainsi un avantage pour promouvoir leur service. Une certaine forme de contrôle des services est ainsi donnée aux hébergeurs et opérateurs. La conséquence de cette évolution est qu'il est très difficile pour un utilisateur derrière un NAT d'offrir un service. Il en est de même pour les applications de type "pair à pair" (comme la téléphonie sur IP, les jeux répartis...) qui sont devenues terriblement complexes pour contourner les difficultés introduites par le NAT pour les connexions entrantes [RFC 5128]. De fait, l'innovation dans ce type d'application est d'une certaine manière réduite. Le NAT est le composant qui participe à limiter l'apparition de nouveaux acteurs et à maintenir une certaine forme de rente pour les acteurs en place.

Enfin, certains ont vu dans le NAT un élément de sécurité d'un réseau local, dans la mesure où le NAT agit comme un filtre en bloquant les paquets entrants non sollicités. Les attaques sont de nos jours dans le contenu, au niveau de l'application, comme les chevaux de Troie ou les codes malveillants (*malware*) dans les pages Web. Le NAT n'améliore donc pas la sécurité car il n'apporte aucune protection contre ces attaques [2] . Le RFC 4864 montre comment avoir le même niveau de sécurité qu'un NAT en IPv6 sans en reprendre les inconvénients.

La pénurie d'adresses ne faisant que s'aggraver avec le temps, on en arrive à la situation que les adresses publiques ne sont plus suffisantes pour être attribuées aux opérateurs eux-mêmes. C'est ce que montre la figure 2 [3] . Cette figure représente, sous forme d'un histogramme, l'état des allocations et donc la situation de l'adressage dans l'Internet IPv4. L'histogramme est composé de 256 barres indiquées par la valeur du premier octet de l'adresse d'IPv4 (notée ici "/8"). Pour la même valeur du premier octet, est alors indiqué l'état de l'usage des 3 autres octets. Cette figure montre qu'il ne reste quasiment plus rien à allouer (en vert). Les RIR (*Regional Internet Registries*) sont sur leur réserve. Ils allouent maintenant les dernières adresses publiques sous des conditions draconiennes et donc, le plus souvent, n'allouent plus d'adresses publiques.

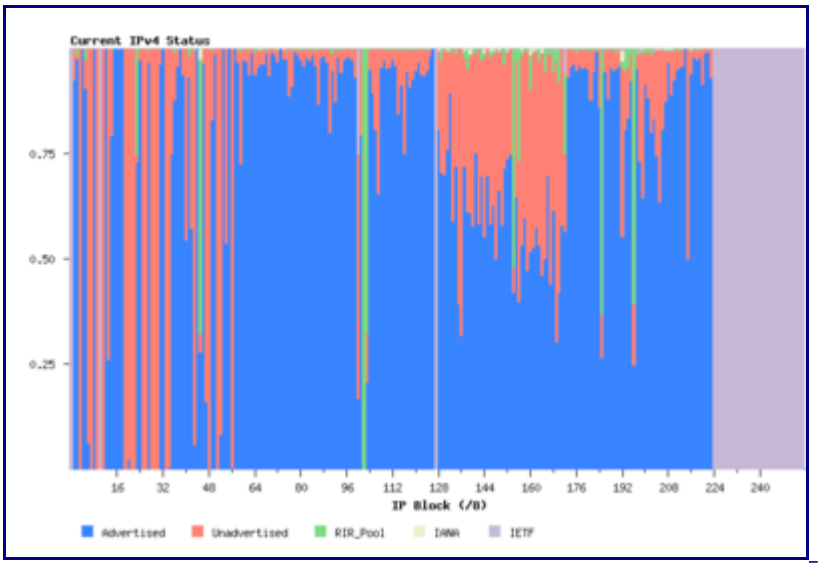


Figure 2: État du plan d'adressage IPv4 en 2015.

Aussi, certains opérateurs, par manque d'adresses publiques, ont recours au NAT444, encore appelée technique du "double NAT" ou CGN (Carrier Grade Nat) RFC 6888 . Le réseau de l'opérateur est, lui-même, en adressage privé. Ainsi, le client de l'opérateur n'a même plus une adresse publique. Le NAT du client final se retrouve à faire un passage d'un adressage privé à un autre adressage privé. D'un point de vue de la terminologie, le NAT du client est dorénavant qualifié de NAT44 pour un changement d'adressage de derrière (le coté client) à devant (le coté opérateur) cet équipement.

Un NAT ou des NAT?

La traduction, qui se veut une solution provisoire, s'est intégrée dans l'architecture de l'Internet comme une technique classique. À tel point qu'elle se décline en différents usages. Stéphane Bortmeyer parle du "zoo des sytèmes de traduction d'adresse IP" [4] lorsqu'il en recense les différentes évolutions.

Le déploiement des super NAT, ou NAT444, pose de nombreux problèmes. Par exemple, il était complexe pour un client d'un opérateur d'héberger un serveur derrière un NAT44, mais ceci devient maintenant impossible derrière un NAT444. Les RFC 5684 et RFC 7021 dressent d'ailleurs une liste des ennuis apparus par l'introduction des NAT444. La seule solution a toutes ces complexités réside au passage d'IPv6 pour sortir enfin de la pénurie.

Motivations à IPv6

C'est en partant du constat des limitations et des problèmes induits par l'utilisation d'IPv4 que les motivations à l'adoption d'IPv6 apparaissent. Il faut aujourd'hui un grand espace d'adressage. Les nouveaux usages de l'Internet avec les nouveaux objets connectés demandent énormément d'adresses. Dépasser la pénurie d'adresse, c'est ouvrir la voie à de nouveaux services, c'est laisser la porte ouverte à de nouveaux acteurs innovants, c'est pouvoir créer de nouveaux marchés pour de nouveaux besoins. Le passage à IPv6 devient une nécessité car, en attribuant une adresse à chaque noeud du réseau, la connectivité en IPv6 retrouve les principes qui ont fait le succès du fonctionnement de l'Internet, et notamment celui du "bout en bout". La technologie de l'infrastructure de communication retrouve sa simplicité

originelle. Il n'est pas soutenable que la croissance du réseau s'effectue avec une complexité croissante comme avec IPv4. Tout ceci est bien connu et cette évolution est qualifiée "par non passage au facteur d'échelle" (*no scalable*). Ainsi, avec cette simplicité retrouvée, de nouveaux champs d'application s'ouvrent à l'Internet en IPv6. Le RFC 7368 en donne une illustration avec la domotique.

En plus de la simplicité retrouvée, IPv6 en apporte de nouvelles facilités, comme la configuration automatique d'un réseau. Avec IPv6, le réseau peut se gérer uniquement au niveau des routeurs, les stations construisant leurs adresses automatiquement, alors qu'avec IPv4, chaque équipement doit se voir attribuer une adresse et obtenir sa configuration depuis un serveur qui reste à gérer. Pour les réseaux avec un grand parc de machines, c'est d'autant plus intéressant.

Geof Huston dans l'article [5] ajoute un autre argument lié à la sécurité dans l'Internet des objets. Comme un balayage de l'espace d'adressage IPv4 prend 5 minutes, un objet peut être victime d'une action "pirate". En IPv6, l'espace d'adressage est si grand qu'il est impossible de balayer tout un réseau pour trouver les adresses utilisées, ce qui rend les noeuds quasiment indétectables. En effet, il faut 41000 ans en IPv6 pour balayer exhaustivement un préfixe /64. Cette caractéristique sur la taille rend IPv6 indispensable pour l'Internet des objets car elle rend les objets indétectables par un simple sondage, tout en les laissant accessibles. En pratique, le RFC 7707 montre que cette affirmation n'est pas si vraie. Les adresses IPv6 peuvent être attribuées selon des conventions d'adressage comme "utiliser l'identifiant 1 pour le routeur". Des stratégies de balayage "malin" peuvent débusquer les noeuds dans un réseau. La connaissance à priori du constructeur des interfaces réseaux, donc de son identifiant OUI (Organisationnally Unique Identifier) réduira l'espace des identifiants d'interface (IID) de 64 à 24 bits, par exemple. Dissimuler les adresses IP des noeuds est de la sécurité par l'obscurité: cela peut ralentir l'attaquant, mais cela ne doit certainement pas être utilisé comme unique moyen de défense car, tôt ou tard, l'attaquant trouvera ces adresses. Il n'en reste pas moins que le balayage est bien plus facile et rapide en IPv4 qu'en IPv6.

Où est est IPv6?

Depuis le premier RFC sur IPv6 publié en décembre 1995, la version IPv6 a quitté les laboratoires. L'étape de standardisation des protocoles de base de IPv6 (*core specs*) est achevée depuis le début des années 2000.

Une adresse IP s'utilise dans l'infrastructure de communication mais également dans les applications des systèmes d'extrémités. Dans la communication, l'adresse IP a un double rôle: localisation et identification. Comme la pénurie d'adresses IPv4 est prévue depuis longtemps, et puisque IPv6 a été conçu très tôt dans cette phase de pénurie, les fabricants et développeurs ont eu le temps de fournir des matériels compatibles IPv6. Si bien, qu'aujourd'hui, tous les équipements informatiques comportent IPv6. Les systèmes d'exploitation de tous les équipements terminaux (PC, stations de travail, imprimantes, etc.), comme ceux des périphériques intermédiaires (commutateurs, routeurs, etc.) disposent d'une pile IPv6 aisément configurable. Ceux-ci s'intègrent à un Internet v6 comme les équipements IPv4 ont pu s'intégrer

à leur époque dans l'Internet v4. Il n'y a pas de réelle difficulté à faire fonctionner des équipements en IPv6 comme le note le RFC 6586 . Tant qu'il s'agit d'acheminer des paquets en utilisant les adresses IPv6, il a été démontré que les traitements de niveau réseau fonctionnent sans problème. Les difficultés commencent à apparaître quand d'autres fonctions, comme la sécurité, ou dans la couche applicative, utilisent des adresses IP comme un identificateur codé sur 32 bits. Les fabricants n'ont pas toujours appliqué des procédures de test complètes, ni pu valider les équipements en IPv6. Cela est dû à un marché encore de taille modeste bien qu'en croissance. Cela devient plus compliqué pour les logiciels propriétaires ou pour les logiciels anciens dont le code source n'est pas disponible. Tout ceci n'est pas un gros problème en soi, mais c'est le risque de multiplication qui va rendre la tâche de migration délicate. C'est actuellement un facteur bloquant pour le déploiement massif d'IPv6.

En 2015, l'usage d'IPv6 vu par les serveurs de Google est proche de 7%. La figure 3 montre l'évolution des usages [6] . Cette courbe montre un doublement de l'adoption d'IPv6 tous les ans depuis 2010. Les utilisateurs de Google peuvent émettre des requêtes en IPv6 s'ils ont un accès IPv6 offert par leur fournisseur d'accès à Internet. En aout 2016, aux USA, IPv6 représente plus de la moitié du trafic mobile vers Facebook [7] .

Figure 3: Évolution du pourcentage de requêtes reçues en IPv6 par Google.

Le figure 4 [8] montre le pourcentage des organisations annonçant un préfixe IPv6. L'Europe, de manière générale, est active dans le déploiement d'IPv6 et la Belgique en particulier [9] . Pour suivre l'évolution de l'adoption d'IPv6, la page web de *world ipv6 launch* référence les mesures faites par différents opérateurs [10] .

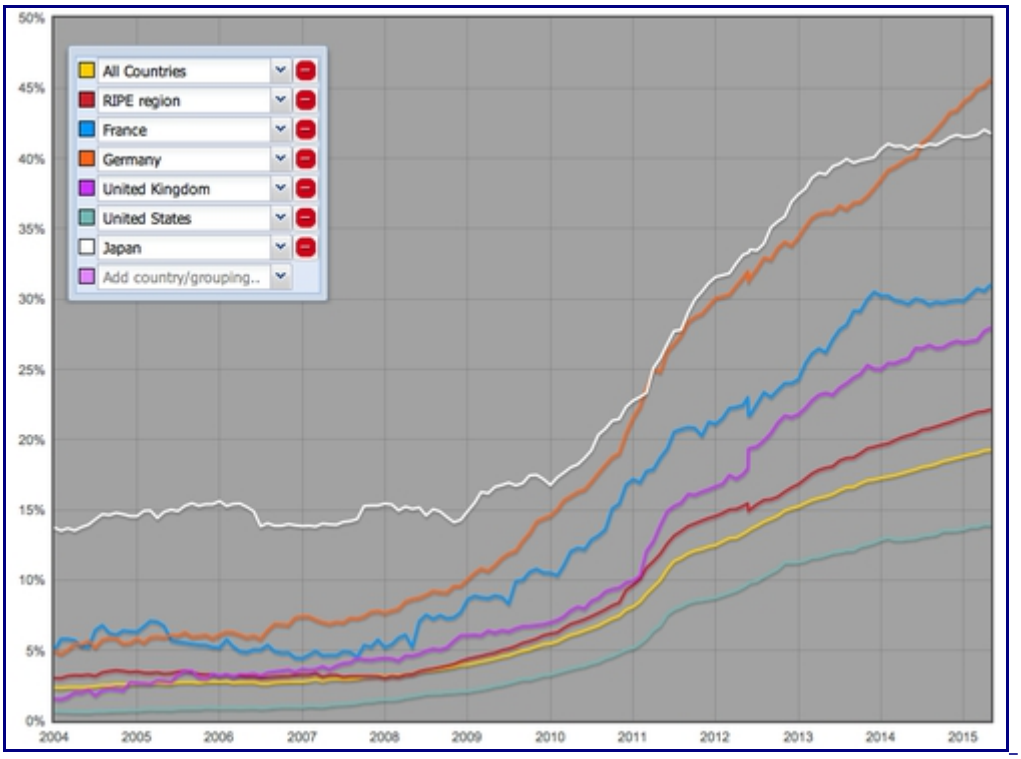


Figure 4: Évolution du pourcentage d'organisations annonçant au moins un préfixe IPv6 par région.

L'adoption d'IPv6 est aussi une question de formation. Le protocole IPv6 n'est plus au stade expérimental; il est indispensable pour un fonctionnement normal de l'Internet. Nous entendons par "normal", un fonctionnement respectant les principes fondateurs de l'Internet, dont celui du "bout en bout". Si les principes de ces deux versions d'IP sont très similaires, IPv4, nous venons de le voir, adopte de plus en plus des principes non conventionnels pour continuer de fonctionner. L'apprentissage du fonctionnement de l'Internet doit se faire de nos jours principalement avec IPv6, et accessoirement avec IPv4. Il faut rendre banale la nouvelle version du protocole IP. Dans un article [11] , Geof Huston dresse une liste de fausses assertions et de rumeurs pour justifier de ne pas commencer le travail de migration vers IPv6. Si ces fausses assertions circulent, elles démontrent à quel point le besoin de formation et d'information sur la situation de l'Internet est nécessaire. Nous espérons que ce cours contribuera à combler ce manque.

Bien qu'IPv6 soit une technologie mature, le déploiement de l'Internet v6 reste encore limité. Néanmoins, son usage devient de plus en plus pressant. Non seulement l'Internet continue de grandir, mais de nouveaux usages et de nouveaux équipements apparaissent, ne faisant qu'accélérer sa croissance. Cela a été écrit: IPv4 n'est pas capable de répondre à ce défi. Et pourtant, le principal frein au passage à IPv6 est de se satisfaire de la situation présente. Le coût du passage à IPv6 constitue un investissement qui, comme tout investissement, s'amortit dans le temps et fournit un retour. Maintenir IPv4 ne produit qu'une dépense, sans aucun espoir de retour. Pire, continuer à déployer des systèmes IPv4 rend la nécessaire migration vers IPv6 plus lente et plus coûteuse. Comment procéder pour réaliser cet investissement? C'est ce que

nous allons étudier par la suite.

Quel scénario pour le déploiement?

Nous avons vu, dans les séquences précédentes, les détails de la technologie de communication liée à IPv6. Nous avons pu constater que le format des paquets et des adresses sont différents de ceux d'IPv4, et ces différences font que ces deux versions d'IP ne peuvent interopérer. L'internet actuel fonctionne en IPv4 mais il a besoin d'IPv6 pour continuer sa croissance. Quelle que soit la version d'IP utilisée, l'objectif est de maintenir une connectivité globale. Se pose alors le problème de la coexistence des deux versions d'IP au sein d'un seul Internet. Plus exactement, le monde IPv6 doit intégrer des mécanismes afin qu'il puisse interopérer avec l'Internet version 4, c'est-à-dire la partie de l'Internet qui utilise encore IPv4. Comme il n'y aura pas de jour du grand basculement d'IPv4 à IPv6, l'introduction d'IPv6 dans l'Internet s'effectuera de façon progressive et en s'étalant dans le temps. Elle doit même se faire sans que l'utilisateur puisse s'en apercevoir. La phase de transition doit être simple ou, au minimum, moins compliquée qu'une utilisation prolongée d'IPv4. Cette introduction d'IPv6 progressive et sans rupture dans l'Internet démontre qu'IPv6 est une évolution d'IPv4. La migration doit se focaliser sur les nouveaux réseaux tout en laissant les anciens fonctionner sous IPv4. L'apparition d'IPv6 ne signifie pas que IPv4 cesse d'exister. En effet, la base d'équipements et de logiciels installés est tellement importante que cela assure au protocole IPv4 une durée de vie quasi "illimitée" à l'échelle humaine. Ceci rend l'idée de la migration sans fin. En fait, c'est notamment au travers des extensions du réseau actuel qu'IPv6 viendra suppléer IPv4. Cet objectif de déployer IPv6 tout en laissant fonctionner IPv4 est rappelé dans le RFC 7381 , qui décrit la démarche pour le déploiement d'IPv6 dans un réseau administré.

Cette idée d'un protocole visant à soulager IPv4 est marquée par le terme d' *intégration* . Le terme de *transition* , lorsqu'il est utilisé, porte l'idée du remplacement d'IPv4 par IPv6. Cette idée est plus anxiogène car elle annonce une migration d'un système de communication qui fonctionne pour aller vers un système plus inconnu. Le but du maintien d'IPv4 en activité est aussi d'éliminer la peur de détruire quelque chose qui fonctionne. De plus, dans le contexte actuel d'un Internet en IPv4, déployer IPv6 ne signifie pas que le réseau ne doit utiliser qu'IPv6. Au contraire, le déploiement d'IPv6 doit s'intégrer dans le réseau actuel et être vu comme une extension du réseau présent. La suite de ce document va présenter les types de mécanismes d'intégration, leurs principes et leurs limites.

Principes des mécanismes d'intégration

Ainsi, IPv6 doit se déployer sans remettre en cause l'existant, qui est opérationnel. Mais que faut-il faire pour passer son réseau en IPv6? En fait, il n'y a pas une solution unique, mais plusieurs réponses qui dépendent de la place occupée par IPv6 dans le système de communication. Il faut distinguer la bordure (les hôtes) et l'infrastructure de communication. L'infrastructure de communication traite du transport des données. Les hôtes sont les consommateurs et producteurs de données ou, de manière classique, les clients et les serveurs. La distinction entre hôte et réseau conduit à identifier six cas [12] :

1. Un hôte IPv4 qui communique avec un hôte IPv4 via un réseau IPv4;
2. un hôte IPv6 qui communique avec un hôte IPv6 via un réseau IPv6;
3. un hôte IPv6 qui communique avec un hôte IPv6 via un réseau IPv4;
4. un hôte IPv4 qui communique avec un hôte IPv4 via un réseau IPv6;
5. un client IPv4 qui communique avec un serveur IPv6;
6. un client IPv6 qui communique avec un serveur IPv4.

Chaque cas pose un problème particulier qui demande un mécanisme dédié. En contrepartie, chaque mécanisme de transition introduit une charge administrative supplémentaire dans le réseau. Ces mécanismes dits d'intégration n'ont pas pour vocation à exister durablement. Ils devraient décroître dans le temps en fonction du nombre d'équipements IPv6 présents sur le réseau. Ils servent à rendre le coût du déploiement supportable en partant des composants existants. Les nouvelles applications, comme par exemple la domotique, pourraient directement démarrer en IPv6 natif et se passer des mécanismes.

Double pile

Le premier cas exprime le point de départ de la migration; le second cas en représente le point d'arrivée. La première idée, pour passer de IPv4 à IPv6, est d'avoir des noeuds qui soient bilingues en quelque sorte, c'est-à-dire capable de parler en IPv6 ou en IPv4 en fonction des capacités de leur correspondant. Pour cela, IPv4 et IPv6 coexistent dans les mêmes noeuds et les mêmes réseaux. Ainsi, les noeuds IPv6 restent compatibles avec les noeuds IPv4. Lorsqu'une nouvelle machine est déployée, elle possède donc une adresse IPv4 et une adresse IPv6. Avec cette idée, la croissance de la taille de l'Internet de ces dernières années aurait été aussi celle d'IPv6. La figure 5 schématise le principe de la communication en double pile. Le déploiement d'IPv6 en double pile était le plan originel de migration. Après la période de spécification que furent les années 90, les années 2000 devaient servir au déploiement des solutions d'intégration. Ainsi, quand le plan d'adressage IPv4 viendrait à épuisement dans la première moitié des années 2010, IPv6 aurait été déployé. Hélas, cette idée n'a pas abouti car elle avait un coût immédiat dû à la double configuration pour un gain futur (à la fin du plan d'adressage IPv4). L'attentisme a régné au niveau du marché et des acteurs comme les fournisseurs d'accès. Ceux-ci n'ont pas montré un réel empressement à déployer une infrastructure en IPv6 pour fournir des préfixes IPv6 afin que leurs clients fonctionnent en double pile. Le déploiement de noeuds double pile a été au final très limité. Nous nous retrouvons maintenant avec deux problèmes à gérer simultanément: l'intégration d'IPv6 et l'épuisement des adresse IPv4 disponibles. Il est à noter que les mécanismes qui suivent (tunnel et traduction) reposent sur des machines à double pile. Elles sont capables de communiquer dans les deux protocoles.

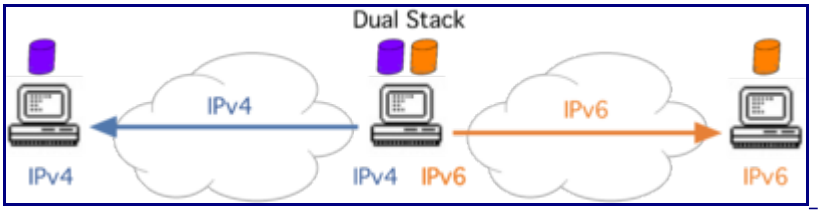


Figure 5: Double pile.

Tunnel

Les cas 3 et 4 se résolvent à l'aide de tunnels. Le paquet de la source est placé dans une enveloppe qui est en fait un paquet dans la version IP du réseau. Dans le troisième cas, une connectivité IPv6 est offerte au travers d'une infrastructure IPv4 existante comme le représente la figure 6. On parle de câbles virtuels (*softwire*): un câble virtuel est un tunnel dans lequel une extrémité du tunnel encapsule les paquets IPv6 dans des paquets IPv4. Les paquets IPv4 transitent dans l'infrastructure IPv4 pour rejoindre l'extrémité du tunnel qui va désencapsuler le paquet IPv6. Le câble virtuel forme une liaison point à point entre 2 noeuds IPv6. IPv4 est alors vu comme un système de transmission, comme peut l'être Ethernet ou une liaison Wifi. Le masquage de la topologie du réseau IPv4 à IPv6 peut conduire à faire un routage des paquets IPv6 susceptible d'être "sous-optimal". Par conséquent, la solution des tunnels doit se faire en essayant de suivre la topologie du réseau et ces tunnels doivent être les plus courts possibles en terme de routeurs IPv4 traversés. Comme les systèmes d'extrémités sont compatibles, la solution à base de tunnels introduit certes une complexité, mais ce n'est pas la plus forte.

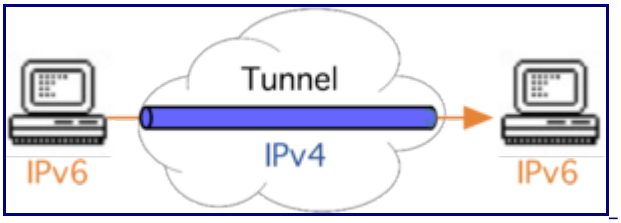


Figure 6: Tunnel.

Traduction

Les deux derniers cas traitent la situation où les extrémités sont incompatibles. Pour certaines catégories d'applications, comme le mail ou le web, le succès d'IPv6 est fortement lié à l'interopérabilité avec IPv4 puisque, jusqu'à présent, la majorité des informations et des utilisateurs ne sont accessibles qu'avec cette version du protocole. Pour des applications distribuées, la technique de traduction (*translation*) consiste à rendre possible la communication entre un système IPv6 et un système IPv4, comme indiqué par la figure 7. C'est l'idée du NAT d'IPv4 appliquée à IPv6. Dans le cas du NAT IPv4, le format du paquet reste le même, mais avec IPv6, le format du paquet change en même temps que les adresses. Ainsi, un coté du NAT est en IPv4 et l'autre coté repose sur IPv6.

Cette traduction peut se faire à différents niveaux de l'architecture réseau:

- Au niveau applicatif, par des passerelles ou ALG (*Application Level Gateway*). Le proxy est un exemple d'ALG qui comporte, en plus des fonctions de traduction, un cache. Le principe d'une traduction par une ALG consiste à ce que le client envoie sa requête en IPv6 à la passerelle applicative. Celle-ci la renvoie vers le serveur en IPv4.

Dans l'exemple du DNS, ceci se conçoit très facilement. Le *resolver* du client envoie la requête au serveur local en IPv6. Ce dernier envoie la requête au serveur suivant en IPv4. De même, certains protocoles applicatifs, tel le protocole de transfert de courrier SMTP, fonctionnent nativement en mode relais. Le message passe de relais en relais pour atteindre le serveur de courrier de destination. Le relayage s'effectuant au niveau applicatif, chaque saut peut

indifféremment s'effectuer en v6 ou en v4. Pour ces applications largement diffusées, comme le web, la messagerie, le DNS, ou encore les serveurs d'impression, la traduction est donc relativement simple à faire. On peut également souligner que le web et la messagerie constituent une part significative des flux Internet actuels. Cette méthode de migration devrait permettre de traiter la majorité des flux. Mais sa mise en oeuvre est complexe car l'ALG est très liée à l'application et la multiplication des applications empêche d'avoir une proposition universelle.

- Au niveau réseau, par des NAT qui agissent au niveau de l'en-tête IP. Le paquet IPv4 est construit à partir d'informations déjà contenues dans l'en-tête IPv6, en particulier différents formats d'adressage permettent de véhiculer une adresse IPv4 dans une adresse IPv6 (le RFC 6052 formalise les différentes variantes d'embarquement d'une adresse IPv4 dans une adresse IPv6). La difficulté d'assurer la compatibilité entre les deux mondes n'est, cependant, pas symétrique. Il est beaucoup plus facile d'initier une session partant du monde IPv6 pour aller vers le monde IPv4. Autrement dit, il est plus facile d'avoir le client du coté IPv6 et le serveur du coté IPv4. En effet, un client IPv6 peut gérer une adresse IPv4 (une adresse sur 128 bits peut contenir une adresse sur 32 bits). Dans le sens inverse, c'est plus complexe: le client IPv4 se retrouve à gérer une adresse en 128 bits et, de plus, il est impossible de modifier l'existant en IPv4.
- Au niveau transport, au moyen de relais SOCKS [RFC 1928] ou de relais TRT (*Transport Relay Translator*) [RFC 3142]. Les relais transport peuvent être perçus comme des "proxys génériques" pour relayer de manière contrôlée les protocoles TCP ou UDP. L'équipement relais accepte les flux ou connexions entrantes issus du client, auprès de qui il se fait donc passer pour le serveur, et les relaie vers le serveur authentique en se faisant passer pour le client. Ce type de solution n'est pas totalement satisfaisante d'un point de vue sécurité car le relais a un comportement de type « *Man in the Middle* » qui intercepte et éventuellement manipule les flux, y compris les flux sécurisés tels que TLS ou SSH. Ce relais peut en effet négocier une clé intermédiaire lors de l'initialisation de la session sécurisée comme SSH (*Secure Shell*) et déchiffrer le flux SSH reçu avant de le réémettre chiffré avec sa propre clé sur la connexion de sortie. Le relayeur aurait alors tout loisir d'observer le flux en clair. C'est une des limitations importante des passerelles de niveau transport. Quel niveau de confiance peut-on accorder à la passerelle transport? On notera également que, compte tenu de son niveau (transport), le relais bloque les flux de contrôle de niveau réseau (ICMP, ICMPv6). Pour ces raisons, l'usage de relais transport est donc aujourd'hui déconsidéré en faveur des deux autres mécanismes précédents.

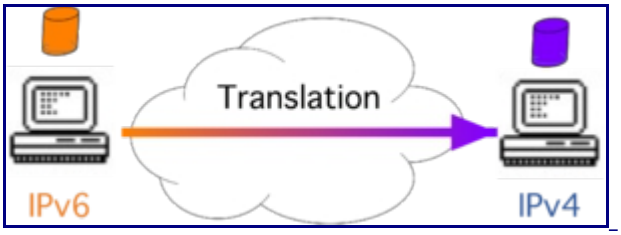


Figure 7: Traduction IPv6-IPv4.

Conclusion

L'adoption d'IPv6 dépend des besoins de chacun mais aussi de la hausse du coût généré par la pénurie d'adresses IPv4. Quand ce coût dépasse une valeur admise propre à chaque acteur, la décision du passage à IPv6 s'impose. IPv6 peut s'utiliser dans le réseau de son site, que son réseau de communication soit à construire ou qu'il existe déjà, que la connectivité de son opérateur soit ou non en IPv6. Notons qu'il est envisageable de déployer un intranet en IPv6 tout en laissant les communications avec l'Internet en IPv4. Quoi qu'il en soit, tant qu'il y aura de l'Internet version 4, il faut maintenir cette connectivité depuis le monde IPv6. Donc, en plus du déploiement d'IPv6, il faut installer des éléments pour réaliser cette connectivité.

Pour chaque situation, l'IETF a développé des mécanismes de coexistence. Chaque mécanisme répond à une problématique précise du déploiement d'IPv6 dans un monde IPv4. La migration vers IPv6 ne soulève pas tous les problèmes possibles. Par conséquent, il faut choisir les mécanismes qui s'appliquent à sa situation. Le fait qu'il y ait un choix à faire dans la multitude des mécanismes est même devenu un argument pour ne pas passer à IPv6. Cette multitude renvoie une image de complexité. Il faut comprendre que chaque technique répond à un problème bien précis, et qu'il n'est pas nécessaire de maîtriser toutes les techniques. C'est à partir de l'étude de ses propres besoins qu'il faut identifier lesquelles des techniques sont à appliquer. La démarche consiste, à partir de l'inventaire du réseau IPv4, à se demander ce qui n'est pas compatible IPv6. Dans la situation d'un nouveau réseau IPv6, ce sont les services accessibles uniquement en IPv4 qui vont guider le choix. La question à élucider quelle que soit la situation est la suivante: quels sont les problèmes qui vont apparaître en utilisant IPv6? C'est à partir de ce constat que les techniques de transition vont être retenues. Alors, ce sont ces techniques-là qu'il convient d'apprendre et de maîtriser. Par exemple, après une étude de son réseau de communication, l'utilisation d'IPv6 montre un problème sur la connectivité avec l'Internet version 6 car son fournisseur d'accès Internet est resté en IPv4. Un tunnel statique peut être la solution (voir la technique TSP de l'activité 43).

Il convient de garder à l'esprit que la finalité n'est pas d'installer des mécanismes d'intégration. Ces mécanismes sont vus comme temporaires, mais sur une période temporaire qui peut durer. L'objectif final est d'avoir l'Internet en IPv6 partout comme le rappelle le RFC 6180 . Le but des mécanismes de coexistence est de faciliter le déploiement progressif et indépendant du protocole IPv6 dans tous les segments du réseau constituant l'Internet. Lorsque cela sera fait, ces mécanismes deviendront obsolètes et leur disparition rendra l'usage d'IPv6 beaucoup plus simple, à l'image d'IPv4 avant l'apparition de son problème de pénurie d'adresses.

La démarche du déploiement d'IPv6 dans un réseau administré d'une organisation est décrite dans le RFC 7381 . Ce document suggère 3 phases:

1. Une phase de préparation et d'analyse au cours de laquelle l'inventaire de l'existant est effectué afin de déterminer quels sont les matériels et les logiciels fonctionnant en IPv6. Le choix de la phase suivante est aussi décidé en fonction des priorités de l'organisation.
2. Une phase interne consistant à déployer IPv6 pour les communications internes.
3. Une phase externe dans laquelle il s'agit de traiter la connectivité de son Intranet avec l'Internet.

Les auteurs [13] montrent aussi que selon l'usage du réseau (mobile, fixe ou de voix sur IP), la stratégie de migration n'est pas la même et doit prendre en compte leurs spécificités. Plusieurs mécanismes de la migration vers IPv6 sont présentés dans la suite de ce chapitre: le déploiement d'IPv6 dans le réseau local en premier lieu, le maintien de la connectivité entre les îlots IPv6 ensuite et, pour finir, l'interopérabilité avec les services en IPv4.

Références bibliographiques

1. ↑ Huston, G (2013). APNIC Labs. A Primer on IPv4, IPv6 and Transition
2. ↑ Bortzmeyer, S. (2012) La traduction d'adresses (NAT) apporte-t-elle vraiment de la sécurité?
3. ↑ Huston, G. IPv4 Address Report
4. ↑ Bortzmeyer, S. (2010), "Le zoo des systèmes de traduction d'adresse IP"
5. ↑ Huston, G. (2015) The ISP Column. The Internet of Stupid Things
6. ↑ Google. Statistics. IPv6 Adoption
7. ↑ Col P. (2016) ZDNet. IPv6 représente plus de la moitié du trafic mobile vers Facebook aux USA
8. ↑ RIPE NCC. IPv6 Enabled Networks
9. ↑ Cole, P. (2016). ZDnet. La Belgique championne du monde d'IPv6, bien loin devant la France!
10. ↑ World IPv6 Launch IPv6 Measurements
11. ↑ Huston, G. (2011). Cisco Internet Protocol Journal, Vol. 14, No. 1, pp. 14-21, March. Transitional Myths
12. ↑ Soussi, M. (2011). AFNIC's Issue Papers. IPv6, A Passport For The Future Internet
13. ↑ Boucadair, M.; Binet, D. et Jacquenet, C. (2011). Techniques de l'ingénieur. Transition IPv6 - Outils et stratégies de migration

Pour aller plus loin

Pénurie d'adresses IPv4

- Bortzmeyer, S (2014), article de blog: Épuisement des adresses IPv4
- Huston, G. (2014) The Internet in Transition: The state of the transition to IPv6 in Today's Internet and of measures to support the continued use of IPv4 .
- Huston, G (2015). Addressing 2014 - And then there were 2!
- Van Beijnum, I. (2014). With the Americas running out of IPv4, it's official: The Internet is full

Statistiques sur IPv6

- APNIC IPv6 Deployment Report
- APNIC Lab List of statistics
- APNIC IPv6 deployment support site . (*Useful and up to date information on IPv6')*
- RIPE IPv6 statistics
- RIPE Lab List of statistics

- Internet Society Liste de pointeurs
- IPv6 Users by Country
- IPv6 CIDR report
- IPv6 host count by IPv6 matrix

Techniques de transition

- Wikipedia IPv6 transition mechanism

RFC et leur analyse par S. Bortzmeyer:

- RFC 1918 Address Allocation for Private Internets Analyse
- RFC 1928 SOCKS Protocol Version 5
- RFC 2663 IP Network Address Translator (NAT) Terminology and Considerations Analyse
- RFC 2993 Architectural Implications of NAT Analyse
- RFC 3142 An IPv6-to-IPv4 Transport Relay Translator
- RFC 4864 Local Network Protection for IPv6
- RFC 5128 State of Peer-to-Peer (P2P) Communication across Network Address Translators (NATs) Analyse
- RFC 5157 IPv6 Implications for Network Scanning Analyse
- RFC 5684 Unintended Consequence of NAT deployments with Overlapping Address Space Analyse
- RFC 6052 : IPv6 Addressing of IPv4/IPv6 Translators Analyse
- RFC 6180 Guidelines for Using IPv6 Transition Mechanisms during IPv6 Deployment Analyse
- RFC 6269 Issues with IP Address Sharing Analyse
- RFC 6319 : Issues Associated with Designating Additional Private IPv4 Address Space Analyse
- RFC 6586 Experiences from an IPv6-Only Network Analyse
- RFC 6888 : Common requirements for Carrier Grade NATs (CGNs) Analyse
- RFC 7021 Assessing the Impact of Carrier-Grade NAT on Network Applications Analyse
- RFC 7368 IPv6 Home Networking Architecture Principles Analyse
- RFC 7381 Enterprise IPv6 Deployment Guidelines Analyse
- RFC 7663 IAB Workshop on Stack Evolution in a Middlebox Internet (SEMI) Report Analyse
- RFC 7707 : Network Reconnaissance in IPv6 Networks Analyse

Activité 42: Déployer IPv6 dans un réseau

Introduction

Une organisation, qui a une infrastructure de communication reposant sur le protocole IPv4, rencontre des difficultés pour faire croître son réseau de manière simple. Elle décide de passer à IPv6 avec, comme cahier des charges:

- déployer IPv6 sans casser ou perturber ce qui fonctionne en IPv4,
- rendre le déploiement complètement transparent à l'utilisateur,
- viser des améliorations en terme de simplicité de gestion et de performance du réseau ou, au pire, que cette dernière soit équivalente à celle obtenue en IPv4,
- maintenir la connectivité avec l'Internet IPv4.

Afin d'avoir un déploiement progressif d'IPv6, elle s'oriente vers un déploiement en double pile qui est un des premiers mécanismes de coexistence, et le plus recommandé. En effet, il évite les problèmes liés à l'utilisation des tunnels. C'est la technique de transition originellement envisagée comme nous le rappellerons. La suite de ce document décrit les principaux éléments relatifs à l'activation d'une double pile. Dans un premier temps, l'adressage et la configuration à mettre en place sont étudiés. Ensuite, les points propres à chacune des principales applications réseaux (DHCP, DNS, pare-feu, supervision) à prendre en compte lors du passage en IPv6 sont soulevés. Enfin, les problèmes induits par l'utilisation de la double pile, ainsi que leurs solutions, sont précisés.

Technique de la double pile

Le mécanisme de double pile IP (*Dual Stack*), spécifié par le RFC 4213 , consiste à doter un équipement du réseau de la pile protocolaire IPv6, en plus de celle d'IPv4, et d'affecter une adresse IPv4 et IPv6 à chaque interface réseau. La configuration des équipements réseaux en double pile exige clairement un double travail de configuration à la fois en IPv4 et en IPv6. L'utilisation parallèle des piles IPv4 et IPv6 vise l'intégration de IPv6 tout en assurant aux noeuds en double pile une compatibilité parfaite avec le réseau IPv4 existant. Ainsi, les noeuds en double pile sont capables de communiquer dans les deux versions du protocole IP. La figure 1 illustre ce principe.

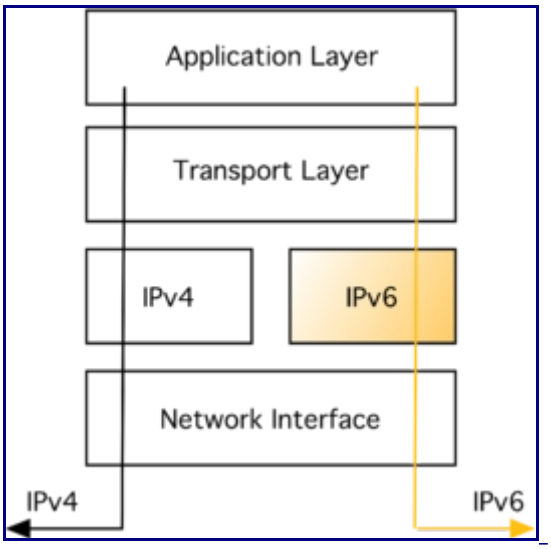


Figure 1: Architecture d'un noeud en double pile.

Dans le cas d'un routeur, il y a une table de routage pour chaque version du protocole. Le routeur est ainsi capable de relayer à la fois les paquets IPv6 et IPv4. De cette façon, IPv4 et IPv6 coexistent sur la même infrastructure. Autrement dit, IPv6 n'a pas besoin d'une infrastructure dédiée.

La technique de la double pile résout le problème d'interopérabilité lié à l'introduction de la pile IPv6. Quand cela est possible, la communication se fait en utilisant la nouvelle version du protocole. Dès qu'un des éléments n'est pas compatible (réseau, système d'exploitation, application), le protocole IPv4 est utilisé. Mais, pour que cette technique soit pleinement utilisable, cela implique que les routeurs entre les 2 correspondants soient aussi configurés pour router les deux types de paquets et que des applications soient capables de traiter des communications avec des adresses IPv6. Avec cette technique, il est possible d'écrire des applications en IPv6 qui restent compatibles avec les applications IPv4 existantes.

Plan de migration originel

La double pile a été proposée dès le début d'IPv6. Le plan originel de migration de l'Internet reposait d'ailleurs sur ce mécanisme, comme le rappelle G. Huston [1] par la figure 2.

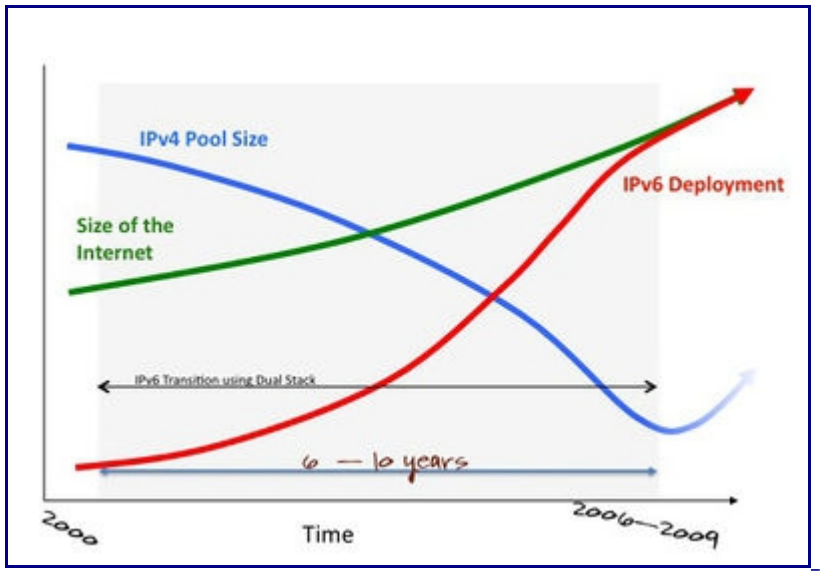


Figure 2: Plan de migration vers IPv6.

Cependant, le problème de la pénurie d'adresses IPv4 n'est pas résolu avec ce mécanisme, puisque l'interface réseau d'un équipement en double pile possède une adresse de chaque version IP. La croissance de l'internet continue de consommer des adresses IPv4. Mais cela offre la possibilité de déployer des noeuds IPv6 afin de vérifier, dans un premier temps, la compatibilité de son réseau avec ce nouveau protocole. Les problèmes inhérents à l'utilisation d'IPv6 peuvent donc être identifiés très tôt. Ensuite, dans un second temps, cela augmente la base des noeuds IPv6 installés. Au fur à mesure du déploiement de ces noeuds, les communications pourront se faire de plus en plus souvent en IPv6. En effet, le client en double pile utilisera en priorité IPv6 pour joindre un serveur lui-même en double pile. Le protocole IPv4 reste cantonné au cas où la tentative échoue en IPv6, ou si le serveur est resté sur l'ancienne version d'IP. Enfin, dans un dernier temps, quand la majorité des services sera accessible en IPv6, la croissance de l'Internet pourra se poursuivre en IPv6 uniquement. Il deviendra envisageable de se passer d'IPv4 et de ses NAT (*Network Address Translation*). Un cercle vertueux est enclenché. L'effort d'interopérabilité aura changé de camp, rendant IPv4 encore plus complexe à utiliser, et par conséquent, accélérant encore le passage à IPv6.

Malgré la disponibilité des équipements supportant la double pile, les acteurs de l'Internet tels que les FAI (Fournisseurs d'accès à Internet), les hébergeurs et les administrateurs de sites n'ont pas perçu l'urgence d'intégrer IPv6 dans leurs activités. Les doubles piles déployées sur les noeuds de l'Internet restent largement inutilisées par rapport au plan initial, comme le montre la figure 3. La croissance de l'Internet s'est poursuivie en IPv4, et celle-ci a donc été affectée par plusieurs effets néfastes comme nous l'avons vu dans l'activité précédente. L'échec du plan initial est largement dû à la dérégulation appliquée dans le secteur des télécommunications qui a conduit les acteurs à privilégier le court terme, et les rend incapables de prendre en compte les besoins à plus long terme dans leurs activités [1] . Dans l'incapacité de réaliser un déploiement coordonné d'IPv6 qui profiterait à tous, chaque acteur a des actions individuelles qui sont raisonnables pour lui, mais coûtent cher à tous. Comme le note S. Bortzmeyer:"déployer IPv6 coûte à celui qui le déploie, ne pas le déployer coûte équitablement à tout le monde" [2] .

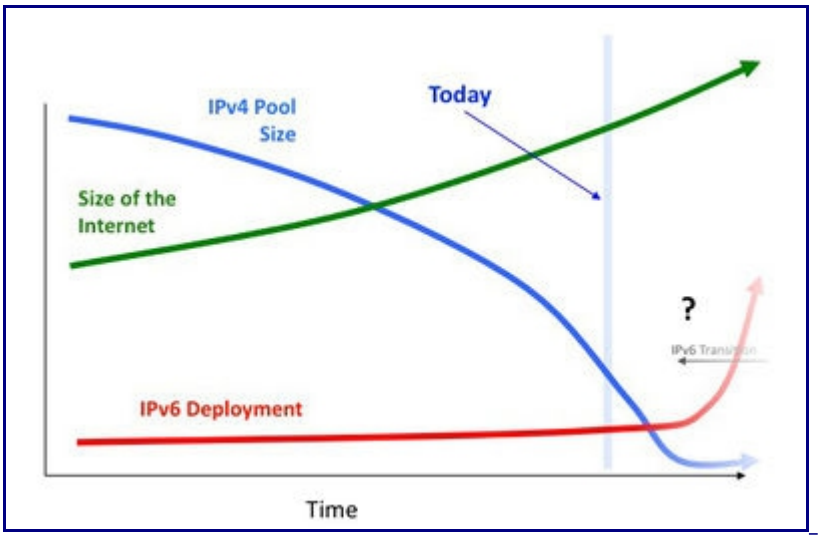


Figure 3: État du plan de migration initial.

Avec l'intégration d'IPv6 dans les principaux systèmes d'exploitation [3] et malgré l'attentisme d'une grande majorité des acteurs de l'Internet, de plus en plus d'infrastructures de communication et d'hébergeurs proposent leurs services en IPv6. Certains FAI donnent maintenant une connectivité IPv6 à leurs clients et ceux qui n'ont pas cette chance peuvent se rabattre sur un accès IPv6 via des tunnels. Ces derniers sont souvent gratuits [4] . Les performances en IPv6 ont été fortement améliorées avec la multiplication des points de présence des FAI en IPv6. Un point de présence est un lieu géographique du FAI contenant un noeud de son réseau fédérateur; autrement dit, un point de connectivité pour le réseau de distribution de ses utilisateurs. De nos jours, comme un grand nombre d'applications (mail, supervision, firewall...) intègre désormais IPv6, il est beaucoup plus aisé de déployer IPv6 dans son réseau qu'il y a une dizaine d'années. Mais il faut faire ce passage le plus tôt possible de manière à traiter progressivement et sereinement les inévitables bugs logiciels et erreurs de configuration qui surviendront.

Étude et préparation du déploiement d'IPv6

En fonction du contexte de déploiement, les enjeux et contraintes ne seront pas les mêmes. Il faut distinguer le réseau résidentiel de l'utilisateur domestique, qui se caractérise par l'absence d'administration, du réseau d'entreprise qui est administré.

- Au sein d'un réseau résidentiel, les problématiques sont liées à la configuration des équipements terminaux, au déploiement des services de résolution de noms et configuration d'adresses, ainsi qu'aux performances perçues par l'utilisateur.
- Dans le cas d'un réseau d'entreprise, il faudra ajouter les problématiques d'obtention du préfixe IPv6, la définition du plan d'adressage, et la configuration du routage IPv6, en plus de celui d'IPv4. Comme les réseaux d'entreprises hébergent de nombreux services tels que le DNS ou le web, il faut aussi prendre en compte la mise à niveau de ces services.

Méthode

L'intégration d'IPv6 dans un réseau d'entreprise demande de la méthode, comme le montre le RFC 7381 . Une phase d'étude et d'analyse est un préalable indispensable pour identifier les points bloquant à l'intégration d'IPv6 dans le contexte professionnel.

L'intégration d'IPv6 commence par la désignation d'une personne en charge de suivre et coordonner les actions liées à l'intégration d'IPv6. Sa première tâche consistera à dresser un inventaire des équipements du réseau afin de déterminer ceux qui supportent IPv6. Cet inventaire va être un élément clef pour orienter le choix des techniques de transition. Par exemple, si de nombreux segments du réseau ne sont pas "IPv6 compatible", il n'est pas question de tout jeter et de racheter, mais il faudra retenir la technique de transition adaptée à son réseau. En plus du matériel, il faut également faire l'inventaire des logiciels utilisés pour déterminer lesquels supportent IPv6 et lesquels nécessitent une mise à jour.

Les applications "métiers", développées en interne, doivent être modifiées le plus tôt possible afin de les rendre capables de manipuler des adresses sur 128 bits. Le RFC 4038 propose des méthodes pour développer du code indépendant des versions d'IP. Dans une note [5] S. Bortzmeyer propose d'utiliser des bibliothèques de langage de plus haut niveau d'abstraction. Ainsi, les détails de la communication ne remontent pas jusqu'au développeur d'application. Le RFC 6724 indique comment sélectionner les adresses sources. Le RFC 8305 liste et solutionne les problèmes liés à la baisse de performance parfois observée dans les déploiements "double pile". Ce dernier point est développé dans la section "Déploiement au niveau des services" de cette activité.

Un point, dans cette phase d'étude, à ne pas négliger concerne la sécurité. L'essentiel des failles de sécurité d'un réseau IPv6 est commune avec celles d'un réseau IPv4. Celles qui sont spécifiques à IPv6 peuvent être dues au manque de support d'IPv6 par les fournisseurs d'équipement de sécurité tels que les NIDS(*Network Based Intrusion Detection System*), pare-feu, outils de monitoring... Ces dispositifs doivent supporter IPv6 aussi bien qu'IPv4 mais ce n'est pas toujours le cas. La faible maturité du code source est également une faille relevée par le RFC 7381 . Les problèmes de sécurité spécifiques à IPv6 peuvent aussi être dus à la configuration. Les pare-feu et ACL (*Access Control List*) des logiciels peuvent avoir des règles strictes pour IPv4 mais beaucoup moins pour IPv6. Étant donné que leur réseau est beaucoup moins sollicité en IPv6, des administrateurs sont tentés de ne pas fournir autant d'efforts que pour la sécurisation d'IPv4. L'utilisation d'adresses protégeant la vie privée des utilisateurs [RFC 4941] complique également la tâche des administrateurs. Elles sont un frein pour une identification rapide des noeuds. Les mécanismes de transition reposant sur des tunnels encapsulant IPv6 sur les réseaux IPv4 apportent également des failles inhérentes à l'utilisation des tunnels [RFC 7123]. S'ils sont mal déployés, ils peuvent créer des *back doors* qui offrent un moyen de passer outre les sécurités de l'entreprise (en particulier avec Teredo et 6to4 [RFC 6180]).

Même si IPv6 n'est pas déployé dans un réseau, il faut malgré tout prendre en compte IPv6 pour la sécurisation. En effet, la plupart des hôtes sont désormais en double pile. Ils ont une adresse IPv6 "lien-local" qui peut être utilisée pour la communication entre les équipements d'un même lien. Ce trafic peut être filtré sur les équipements de niveau 2 s'ils le permettent. La double pile rend le noeud sensible aux attaques par fausses annonces de routeurs (*rogues RA*

) [RFC 6104]. Ces annonces configurent chez les hôtes une fausse connectivité IPv6. Les hôtes enverront le trafic au routeur par défaut, lequel pourra fournir une connectivité IPv6 aux utilisateurs via des tunnels et mettre en œuvre des attaques de type MitM (*Man in the Middle*). Le RFC 7113 propose une méthode d'analyse de l'en-tête IPv6 appelée 'RA-Guard *(* IPv6 Router Advertisement Guard *) à mettre en oeuvre au niveau des commutateurs.* En dépit du fait que les annonces de routeurs illégitimes soient la plupart du temps le fait d'erreurs de configuration de machines hôtes qui émettent des RA, il ne faut pas néanmoins les négliger car une connectivité IPv6 non fonctionnelle ou de mauvaise qualité va affecter la qualité de service perçue par l'utilisateur (voir le paragraphe "problèmes liés à la double pile" de cette activité). Notons que la sécurisation des mécanismes d'auto-configuration n'est pas un problème spécifique à IPv6. En IPv4, des serveurs DHCP mal intentionnés (idem pour DHCPv6) peuvent également envoyer des informations erronées suite à une requête DHCP. Aussi bien les RA que le DCHP peuvent être sécurisés via l'authentification des messages, mais ces solutions sont très peu déployées en pratique.

L'impact des différences entre les deux versions d'IP est souvent mal évalué. Par exemple, l'utilisation d'un préfixe IPv6 GUA pour les hôtes et l'absence de NAT, notamment dans les routeurs SOHO (*Small Office / Home Office*) est perçue comme une faille de sécurité. En plus des règles de filtrage nécessaires à la sécurisation, les RFC 6092 et RFC 7084 imposent que les routeurs SOHO filtrent par défaut les connexions venant de l'extérieur au réseau. De cette manière, l'absence de NAT dans le cadre d'IPv6 n'ouvrira pas plus de faille de sécurité que sur les routeurs SOHO en IPv4. La sécurité de IPv6 peut aussi être surévaluée, comme dans les attaques par balayage de l'espace d'adressage. Malgré la taille gigantesque de l'espace d'adressage en IPv6, le RFC 7707 montre que IPv6 est malgré tout sensible aux attaques par balayage, et qu'il faut s'en protéger. A cet effet, le RFC 6018 propose l'utilisation de *greynets* pour IPv4 et IPv6.

Ensuite, vient la problématique du routage interne. Les principaux protocoles de routage intègrent depuis longtemps IPv6. OSPFv3 supporte IPv4 et IPv6 mais diffère de OSPFv2 sur certains points. Notons qu'il est possible d'utiliser des protocoles de routage différents pour les trafics IPV4 et IPV6. Le document [6] (en cours d'étude au moment de la rédaction de ce document) détaille les choix de conception spécifiques au routage IPv6.

La phase de préparation inclut également le plan d'adressage et l'allocation des adresses. Ces points sont abordés en détail dans la suite de ce document.

Il apparaît donc clairement que l'intégration d'IPv6 nécessite d'impliquer de nombreux corps de métiers. Les formations adéquates doivent donc être proposées au personnel de l'entreprise. Cela inclut aussi bien les administrateurs système et réseau, ceux en charge du routage, de l'infrastructure, les développeurs, que le personnel des centres d'appel du support technique. À titre d'exemple, citons l'article [7] qui rapporte l'expérience de la migration en IPv6 d'un industriel.

Vérification de la disponibilité d'IPv6

Le protocole IPv6 et ses protocoles associés sont pris en charge par les systèmes d'exploitation

depuis plus de 10 ans. Il en découle qu'une grande majorité des noeuds de l'Internet comporte IPv6. Ainsi, au démarrage d'un noeud, même en l'absence d'un routeur IPv6 sur le lien de ce noeud, l'interface se configure automatiquement avec une adresse IPv6. Les exemples ci-dessous montrent que c'est le cas pour les OS les plus courants. Pour chacun des OS, une adresse "lien-local" (*link-local address*) a été allouée [voir la séquence 1]. Elle est utilisée pour les communications locales uniquement, comme par exemple le mécanisme de découverte de voisins [RFC 4861]. Elle n'est pas routable ni, par conséquent, utilisable pour une communication indirecte (passant par un routeur).

Pour que cette vérification soit une formalité, il est nécessaire, bien en amont de l'intégration d'IPv6, d'exiger, dans les achats de matériels et logiciels, la disponibilité d'IPv6 ou la compatibilité [8] . Par exemple, c'est ce qu'a fait le département nord-américain de la défense [9] .

MacOSX 10.9.2

```
ifconfig en0
en0: flags=8863UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST mtu 1500
    ether 14:10:9f:f0:60:46
    inet6 fe80::1610:9fff:fef0:6046%en0 prefixlen 64 scopeid 0x4
    inet 192.168.1.143 netmask 0xffffff00 broadcast 192.168.1.255
    nd6 options=1PERFORMNUD
    media: autoselect
    status: active
```

Linux 2.6.32:

```
ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 60:eb:69:9b:87:97
          inet addr:195.154.87.139  Bcast:195.154.87.255
Mask:255.255.255.0
          inet6 addr: fe80::62eb:69ff:fe9b:8797/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:75115704 errors:5 dropped:0 overruns:0 frame:5
          TX packets:17934141 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:6583563265 (6.5 GB)  TX bytes:5944865545 (5.9 GB)
          Memory:feae0000-feb00000
```

Windows:

```
c:\ ipconfig
Windows IP Configuration
Ethernet adapter Local Area Connection:
Connection-specific DNS Suffix .:
IPv6 Address. . . . . . . . . . . .: 2001:db8:21da:7:713e:a426:d167:37ab
Temporary IPv6 Address. . . . . .: 2001:db8:21da:7:5099:ba54:9881:2e54
Link-local IPv6 Address . . . . .: fe80::713e:a426:d167:37ab%6
IPv4 Address. . . . . . . . . . . .: 157.60.14.11
Subnet Mask . . . . . . . . . . . .: 255.255.255.0
Default Gateway . . . . . . . . . .: fe80::20a:42ff:feb0:5400%6
IPv4 Default Gateway  . . . . . .: 157.60.14.1
Tunnel adapter Local Area Connection* 6:
Connection-specific DNS Suffix .:
IPv6 Address. . . . . . . . . . . .: 2001:db8:908c:f70f:0:5efe:157.60.14.11
Link-local IPv6 Address . . . . .: fe80::5efe:157.60.14.11%9
Site-local IPv6 Address . . . . .: fec0::6ab4:0:5efe:157.60.14.11%1
```

```
Default Gateway . . . . . .: fe80::5efe:131.107.25.1%9
fe80::5efe:131.107.25.2%9
Tunnel adapter Local Area Connection* 7:
Media State . . . . . .: Media disconnected
Connection-specific DNS Suffix .:
```

Obtenir un préfixe IPv6

Pour une communication indirecte, il faut compléter la configuration avec une adresse IPv6 unicast qui soit routable. Il existe deux types d'adresses qui répondent à ce critère: les adresses "unicast locales" ULA (*Unique Local Address*) [RFC 4193] et les adresses "unicast globales" GUA (*Global Unicast Address*) [RFC 3587]. Pour rappel, les différences majeures entre ces deux types d'adresses sont les suivantes:

- **Portée** : les adresses GUA sont des adresses publiques tandis que les adresses ULA sont des adresses privées. Les adresses privées ne peuvent être utilisées que pour des communications dans un intranet.
- **Routage** : Les adresses GUA peuvent être routées dans l'Internet. Les adresses ULA, routables uniquement sur une topologie privative, doivent être filtrées par les routeurs en bordure de site. Les prefixes ULA ne doivent pas être annoncés ni acceptés par les routeurs inter-AS.
- **Obtention** : Un préfixe ULA est généré de manière aléatoire par l'administrateur d'un site. Le GUA est obtenu auprès d'un opérateur tiers qui gère un registre d'allocation.

Mais quelle type d'adresse routable utiliser dans un site? Quelles sont les cas d'utilisation des adresses ULA? Les éléments de réponses à ces questions sont abordés dans le RFC 5375 , qui développe les considérations à prendre en compte pour la mise en place de l'adressage unicast d'IPv6. Ainsi, il recommande un préfixe de lien de /64 pour, notamment, le bon fonctionnement de la procédure d'auto-configuration d'adresses. Le RFC 6177 discute du préfixe à allouer à un site d'extrémité. Ce préfixe peut varier de /48 à /64. Il est recommandé de donner des possibilités de sous-réseaux à l'intérieur du site, ce que ne permet pas une allocation de préfixe à /64.

Il faut tout d'abord noter que le préfixe alloué à un site est souvent très confortable au niveau du plan d'adressage. Il n'y a rien de commun avec ce qui est connu en IPv4. Lorsqu'un site obtient un préfixe /48, il peut avoir 2^16 sous-réseaux différents et 2^64 noeuds dans chacun de ces sous-réseaux. Même l'allocation d'un préfixe /64, qui reste problématique pour déployer des sous-réseaux, donne un nombre d'adresses disponibles qui dépasse de plusieurs ordres de grandeur le nombre de noeuds qu'il peut y avoir dans un réseau.

Préfixe ULA

Le préfixe ULA [RFC 4193] est l'équivalent, dans son usage, aux préfixes privés d'IPv4 [RFC 1918], mais quasi unique et sans registre central. Ce dernier point rend le préfixe ULA non agrégeable, et donc les adresses ULA non routables sur l'Internet. La caractéristique d'unicité du préfixe ULA supprime le risque de conflit entre les 2 plans d'adressage lorsque 2 sites privés fusionnent, ce qui est loin d'être le cas en IPv4.

Ce RFC propose, dans un espace réservé `fc00::/7` , de constituer, selon un algorithme, des adresses quasi uniques. Le format des adresses de type ULA est présenté dans l'activité 13. Il est rappelé que le format d'adresse ULA se compose d'un préfixe de 48 bits dont 40 bits (*Global ID* , GID) pour identifier le site. Les 40 bits du GID sont générés en utilisant une fonction de hachage (*i.e.* SHA-1) de l'heure et de l'adresse MAC de la machine, exécutant l'algorithme détaillé dans le RFC. Outre le script, sous licence libre GPL et développé par Hartmut Goebel, indiqué dans l'activité 13, il existe des sites pour générer automatiquement un préfixe ULA comme http://unique-local-ipv6.com/ ou http://www.kame.net/~suz/gen-ula.html , ou bien encore celui du SIXXS qui, en plus de fournir un préfixe ULA, l'enregistre dans un registre.

Notons que les raisons conduisant à l'utilisation des adresses privées d'IPv4 ne s'appliquent plus dans le cas d'IPv6. Citons:

- **Manque d'adresses IP publiques** . Dans l'internet IPv4, la motivation principale pour l'utilisation des adresses privées est que l'espace d'adressage publique n'est pas suffisant pour l'ensemble des machines. Dans le cas d'IPv6, cette motivation n'a clairement plus lieu d'être.
- **Accroitre le niveau de sécurité** . L'utilisation des adresses privées dans IPv4 induit que les machines situées derrière un NAT sont plus difficilement accessibles de l'extérieur par un unique effet de bord. Cela rend les machines derrière le NAT moins vulnérables aux attaques extérieures. Certains estiment donc que les adresses GUA exposent les machines directement aux attaquants de l'Internet et trouvent là une justification à l'utilisation d'adresses privées. On notera que cet argument est fallacieux car, avec un adressage privé, il faut malgré tout utiliser un pare-feu pour prévenir les attaques, ce qui montre que la sécurisation n'est pas une question de type d'adresse publique ou privée. Donc, une simple règle sur un pare-feu pour interdire l'ouverture de connexion depuis l'extérieur peut fournir le même niveau de sécurité qu'un NAT.
- **Facilité de déploiement** . L'accès Internet, pour un site avec un adressage ULA, nécessite un NAT66 dénommé aussi NPTv6 (*Network Prefix Translation*) [RFC 6296] pour le changement d'adresses ULA en GUA. En plus de l'achat et de la maintenance de cet équipement, ce sont certaines tares du NAT qui reviennent dans le réseau IPv6 [RFC 5902]. L'usage d'ULA dans le cas d'un accès Internet n'économisera pas l'obtention d'un préfixe GUA (pour l'extérieur du NAT). Au final, un réseau basé sur les adresses ULA introduit un travail plus complexe et plus important qu'un équivalent GUA.

Aussi, les seuls cas où l'utilisation des adresses ULA est réellement motivée sont les réseaux de tests (enseignement, bancs d'essais, déploiement de prototype) et les réseaux nécessitant un niveau de sécurité très élevé par un isolement complet, comme les réseaux tactiques ou d'hôpitaux. Le RFC 6296 propose une autre utilisation d'un plan d'adressage construit sur un préfixe ULA. Pour des sites de taille petite ou moyenne, un préfixe ULA couplé à un NAT66, offre une solution simple pour changer d'opérateur ou pour gérer la multi-domiciliation sans nécessiter un préfixe PI (*Provider Independent*). Ainsi, en cas de changement de fournisseur d'accès, la renumérotation n'impactera que le NAT. Les adresses ULA forment ainsi une sorte de substitut aux adresses PI. Cette idée peut avoir un sens tant que des mécanismes simples de renumérotation du réseau ne seront pas effectifs [RFC 7010]. Cette question de la

renumérotation n'est pas une question simple { RFC 5887]. Dans tous les autres cas, les adresses GUA sont plus faciles à déployer et à administrer. C'est aussi le conseil donné par l'auteur de cette note [10] .

Préfixe GUA

Pour rappel, les préfixes GUA sont sous l'autorité de l'IANA [11] qui délègue aux RIR (*Regional Internet Registry*) l'allocation. Les RIR délèguent eux-mêmes aux NIR (*National Internet Registery*) puis aux LIR (*Local Internet Registery*) et/ou finalement aux FAI. En Europe, le RIR est le RIPE-NCC . Il délègue directement aux FAI/LIR sans passer par des NIR. Les LIR et certains FAI se voient déléguer des préfixes /32. Ils ont obligation d'allouer les blocs IPv6 à des utilisateurs finaux tels que des organismes ou des FAI. Le RIPE-NCC ne prévoit pas de recommandation sur la taille des préfixes alloués par les LIR aux FAI.

Le préfixe GUA peut être alloué par un FAI, par un LIR ou par un RIR. Le choix s'effectue selon le type de préfixe à détenir. Si le préfixe est destiné à un site, on parlera d'un préfixe PA (*Provider Assigned* ou *Provider Aggregatable*); si le site est multi-domicilié, il faut un préfixe dit PI (*Provider Independent*).

Le préfixe de type PA est attribué par le FAI/LIR. Il n'y a pas de formalités particulières à remplir. Le préfixe est alloué en même temps que la connectivité. Le préfixe est donc spécifique à un site et associe ce site à un opérateur. Ce dernier assure les services suivants:

- allocation du préfixe à l'organisme,
- transport du trafic de l'utilisateur,
- annonce d'un préfixe BGP dans lequel est inclus celui du site.

La taille du préfixe alloué varie selon les opérateurs. Certains donneront un /52, voire un /60. Le préfixe alloué est au maximum /64. Si un site doit avoir un préfixe de moins de 48 bits, la demande doit être motivée. Si le FAI change, il faut rendre le préfixe et renuméroter le réseau du site, et cette action est pénible [RFC 5887]. Pour éviter ce désagrément, il est possible de demander un préfixe PI auprès d'un RIR. Ce type de préfixe est une nécessité pour les sites multi-domiciliés ou pour les sites qui doivent changer de FAI sans changer d'adresses. La demande de préfixe doit être faite directement à RIPE-NCC qui attribue un préfixe /48 ou un préfixe de longueur inférieure si la demande est motivée. Il faut que l'organisation qui en fait la demande soit membre de RIPE ou que la demande soit parrainée par un FAI/LIR membre de RIPE. Il est ensuite nécessaire que les FAI annoncent et routent le préfixe PI.

A noter que si un FAI ne propose pas IPv6, il est possible d'utiliser un service de tunnels. Certains d'entre eux (*e.g.* Hurricane Electric) attribuent gratuitement un préfixe /48 lors de l'établissement d'un tunnel.

Définition du plan d'adressage de sous-réseau avec IPv6

Les préfixes alloués dans la majorité des cas laissent de nombreux bits pour gérer les liens à l'intérieur d'un site. Lorsque le préfixe alloué au site est un /48, le SID (*Subnet Identifier*) est codé sur 16 bits. Il est évident que la structuration du plan d'adressage est radicalement

différente selon que l'on soit en IPv4 ou en IPv6. En IPv4, l'essentiel du travail sur l'adressage a pour but d'économiser les quelques adresses disponibles, pour pouvoir fonctionner malgré la pénurie. En IPv6, ce problème disparaît et la définition du plan d'adressage vise la facilité de son administration tout en rendant l'agrégation de routes efficace. La mise en oeuvre des politiques de sécurités doit aussi être prise dans la définition du plan d'adressage interne. Dans l'article [12] , l'auteur montre comment ces critères doivent servir à guider la définition d'un plan d'adressage pour un site. Comme nous l'avons vu dans l'activité 16, il est possible de structurer le routage interne de plusieurs manières:

- Reproduire le schéma IPv4 déjà déployé. Ainsi, par exemple, le préfixe privé (RFC 1918) `10.0.0.0/8` offre 24 bits d'identification locale à l'administrateur pour la structuration des sous-réseaux. En pratique, sur cet exemple, les plus petits sous-réseaux ont rarement des préfixes supérieurs à /24, ce qui laisse 16 bits (24 - 8) pour la structuration. Dans ce cas, il est donc possible de reproduire le plan d'adressage privé IPv4 à l'aide des 16 bits du SID.
- Numéroter de manière incrémentale les sous-réseaux (e.g. 0001,0002,0003…). Simple à mettre en oeuvre, cette technique peut cependant conduire à un adressage plat et difficile à mémoriser. Elle peut également complexifier l'écriture des règles de filtrage ainsi que l'agrégation.
- Utiliser le numéro de VLAN, ce qui est tout à fait possible puisque le VLAN ID n'occupe que 12 bits. Cette méthode permet d'éviter de mémoriser plusieurs niveaux de numérotation.
- Séparer les types de réseaux et utiliser les chiffres de gauche pour les désigner. D'autres niveaux de structuration peuvent être définis sur les bits restant. Cette technique permet de faciliter les règles de filtrage, tout en utilisant des règles appropriées à la gestion de ces sous-réseaux pour la partie de droite. À titre d'exemple, le tableau 1 contient le plan de numérotation d'une université localisée sur plusieurs sites prenant en compte les différentes communautés d'utilisateurs. Ainsi, le préfixe:
  - `2001:DB8:1234::/52` servira pour la création de l'infrastructure, donc en particulier les adresses des interfaces des routeurs prises dans cet espace;
  - `2001:DB8:1234:8000::/52` servira pour le réseau Wi-Fi des invités; la manière dont sont gérés les 12 bits restants du SID n'est pas spécifié;
  - `2001:DB8:1234:E000::/52` servira pour le réseau des étudiants. L'entité représente la localisation géographique du campus. Dans chacun de ces campus, il sera possible d'avoir jusqu'à 16 sous-réseaux différents pour cette communauté.

Communauté	4 bits	8 bits	4 bits
Infrastructure	0	valeurs spécifiques	
Tests	1	valeurs spécifiques	
Tunnels	6	allocation de /60 aux utilisateurs	

Invités Wi-Fi	8	valeurs spécifiques	
Personnels	A	Entité	sous-réseaux
Étudiants	E	Entité	sous-réseaux
Autre	F	valeurs spécifiques	

Tableau 1: Exemple de découpage du SID

Déploiement des équipements en double pile

Les services indispensables au fonctionnement d'un réseau doivent être déployés et ceux existants doivent intégrer IPv6; par exemple, la configuration d'adresse (DHCP / SLAAC), le nommage (DNS) et l'administration de l'infrastructure (supervision, sécurité et métrologie). Cette section traite des problématiques liées à leur configuration.

Un hôte en double pile présente une interface réseau de la manière suivante dans un environnement Unix:

```
eth0: flags=8843UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST mtu 1500
inet 192.108.119.134 netmask 0xffffff00 broadcast 192.108.119.255
inet6 2001:db8:1002:1:2b0:d0ff:fe5c:4aee/64
inet6 fe80::2b0:d0ff:fe5c:4aee/64
ether 00:b0:d0:5c:4a:ee
media: 10baseT/UTP half-duplex
supported media: autoselect 100baseTX
```

Notons qu'un réseau peut être entièrement en double pile ou partiellement, à condition que les segments IPv4 soient masqués par des tunnels dans lesquels IPv6 est encapsulé dans IPv4. Tous les équipementiers de coeur de réseau supportent ces mécanismes, ce qui permet rapidement d'acheminer du trafic IPv6 dans une infrastructure IPv4 existante. Lorsque le déploiement est partiel, une attention particulière doit être portée au protocole de routage utilisé, l'activation de fonctions permettant de gérer plusieurs topologies (v4 et v6) pouvant s'avérer nécessaire.

Configuration d'adresses

La configuration des interfaces réseaux en IPv6 peut s'effectuer selon plusieurs méthodes.

Avec la méthode SLAAC (*StateLess Address Auto Configuration*) [RFC 4862], l'interface génère elle-même ses adresses à partir des informations émises par le routeur local. Si SLAAC est sans doute plus simple et plus rapide à déployer, elle peut présenter des inconvénients:

- Absence du DNS. SLAAC n'intègre pas de champ pour transmettre le serveur DNS local. Ce n'est pas un problème si l'adresse d'un serveur DNS est obtenue via le DHCP de l'interface IPv4, mais cela rend donc indispensable l'existence d'une telle interface. Toutefois, le RFC 6106 rend désormais possible l'ajout d'une option DNS dans les messages RA (*Router Advertisment*).
- Absence de contrôle sur les adresses. Il n'y a pas de moyen fiable d'enregistrer

l'association "adresse MAC - adresse IP". Le logiciel NDPMON (*Neighbor Discovery Protocol Monitor*) permet cependant d'écouter le réseau en permanence et de mémoriser les correspondances entre les adresses IP et MAC.

Avec DHCPv6 [RFC 3315], le client obtient son adresse et ses informations auprès du serveur DHCP. Ce dernier peut donc contrôler les informations indiquées à chaque machine, contrôler les adresses attribuées et mémoriser ces dernières. Le serveur DHCP est aussi l'endroit logique où faire des mises à jour dynamiques du DNS pour refléter les changements d'adresses IP. Comme DHCP offre davantage de contrôle que SLAAC, DHCP est en général apprécié dans les réseaux d'organisations.

Lorsque DHCP est utilisé dans sa version sans état, comme le permet le RFC 3736 , il sert à distribuer uniquement des paramètres statiques, comme les adresses des serveurs de noms. Dans cette situation, la méthode SLAAC est utilisée pour allouer les adresses et le noeud doit récupérer les informations manquantes à sa configuration par le serveur DHCP sans état.

Lors du déploiement de DHCPv6 en double pile, l'inconvénient majeur va être la gestion des informations recueillies via des sources différentes. Ce problème bien connu est notamment décrit dans le RFC 4477 . En effet, des informations pouvant être reçues à la fois du DHCPv4 et du DHCPv6, il peut y avoir inconsistance. Par exemple, des informations relatives à la pile IPv6 renseignées manuellement dans la configuration de l'OS (e.g. /etc/resolv.conf) peuvent être effacées par le client DHCPv4. Le client doit savoir s'il doit utiliser les informations les plus récentes ou fusionner ces informations selon des critères bien précis. Ce problème est encore plus prononcé si les réseaux IPv6 et IPv4 n'ont pas les mêmes administrateurs.

Résolution d'adresses

Les points importants relatifs au DNS (*Domain Naming System*) dans le déploiement d'IPv6 sont présentés dans le RFC 4472 . Pour IPv6, le DNS est d'autant plus indispensable que les adresses sur 128 bits ne sont pas simples à lire ni à mémoriser. Le DNS est utilisé pour associer les noms avec les adresses IP. Un nouvel enregistrement (*resource record*) appelé AAAA a été défini pour les adresses IPv6 [RFC 3596]. Les "résolveurs" DNS (clients du DNS) doivent être capables d'interpréter les enregistrements A pour IPv4 et les enregistrements AAAA pour IPv6. Lorsque les deux types sont retournés par le serveur DNS, le "résolveur" doit trier l'ordre des enregistrements retournés de manière à favoriser IPv6. Par ailleurs, le client (de la couche application) doit pouvoir spécifier au "résolveur" s'il souhaite obtenir les entrées de type A ou AAAA.

Administration du réseau

Il est indispensable que IPv6 et IPv4 soient iso-fonctionnels. Pour ce faire, il faut maîtriser les outils d'administration réseau IPv6 et en particulier s'assurer du bon fonctionnement des services et équipements IPv6.

L'administration d'un réseau peut se décomposer en trois tâches: la supervision, la métrologie et la sécurité. Les pare-feux sont depuis longtemps capables d'appliquer leurs règles de filtrage au trafic IPv6. Il est à noter que les mécanismes de chiffrement et les certificats n'ont pas été

impactés par IPv6. Les outils de métrologie sont généralement assez faciles à adapter à IPv6 puisqu'il y a peu de dépendance entre les logiciels.

La difficulté principale réside dans les outils de supervision. Le protocole de supervision SNMP sert à collecter dans des bases de données appelées MIB (*Management Information Base*) diverses informations qui sont stockées sur les équipements réseaux. Net-SNMP intègre IPv6 depuis 2002. Cette intégration était nécessaire pour interroger les noeuds uniquement IPv6. Cette intégration d'IPv6 n'était pas indispensable dans le cas d'un réseau double pile puisqu'il est possible d'interroger un équipement via SNMP depuis son interface IPv4. L'évolution des MIB a été beaucoup plus délicate mais elle est achevée et le RFC 2851 prévoit que l'adresse IP soit de longueur variable et constituée de deux champs, un pour identifier le type d'adresse et un pour l'adresse elle-même.

Les principales solutions de supervision (e.g. Nagios) et équipementiers supportent désormais largement IPv6. Il faut malgré tout s'assurer que l'ensemble des outils utilisés dans le cadre de SNMP supportent la version unifiée et modifiée de la MIB.

Déploiement d'IPv6 pour les services

Les adresses IPv4 imbriquées dans une adresse IPv6

Les premières adresses IPv4 imbriquées dans une adresse IPv6 ont été décrites dès les premières spécifications des mécanismes d'interopérabilité, dont certains ont depuis été officiellement dépréciés. Parmi ces adresses historiques nous trouvons:

- **adresse IPv4 compatible** (*IPv4-Compatible IPv6 address* RFC 2893 , RFC 3513) **`::a.b.c.d/96`** ou **`::xxxx:xxxx/96`** . Ces adresse ont été dépréciées par le RFC 4291 .
- **«IPv4 mappées»** (*IPv4-mapped IPv6 address* RFC 4291) **`::ffff:a.b.c.d/96`** ou **`::ffff:xxxx:xxxx/96`** . Ces adresses font référence à un noeud supportant uniquement IPv4.
- **«IPv4 translatées»** (*IPv4-translated IPv6 address* RFC 2765) **`::ffff:0:a.b.c.d/96`** ou **`::ffff:0::xxxx:xxxx/96`** . Ces adresses référençaient dans l'espace v4 un noeud uniquement v6, dans le cadre du protocole, aujourd'hui obsolète, SIIT (RFC 2765). Elles se distinguent des «IPv4 mappées» par un décalage à gauche de 16 bits du mot `ffff` .

Les préfixes de ces adresses sont composés de mots nuls ou tout à 1 (`:ffff:`), ce qui les rend neutres vis à vis du calcul du checksum intégrant le pseudo entête *(cf sequence 3)* .

Les longs préfixe nuls de ces adresses les rendent difficilement routables sur le réseau. Ces adresses sont cependant adaptées pour les interfaces logiques internes aux machines double pile (*dual-stack*). Les adresses «IPv4 mappées» sont par exemple utiliséees pour aiguiller les flux vers la pile IPv4, dans le cadre d'applicatifs conformes IPv6 hébergés sur des machines double pile.

Au niveau des applications

La version de protocole IP utilisée doit être transparente au niveau de l'application et cela ne doit rien changer. Il faut cependant que l'application puisse exprimer l'adresse de son correspondant, que ce soit en IPv4 ou en IPv6. Pour cela, les adresses doivent être codées sur 128 bits. Un type d'adresse IPv6 a été défini à cet usage, à savoir comporter l'adresse IPv4 d'une communication IPv4 (*IPv4 mapped IPv6 address* , «IPv4 mappées»). L'adresse IPv4 est imbriquée dans une adresse IPv6 comme le montre la figure 4. Le format des adresses IPv4 imbriquées est `::ffff:ipv4 address` , comme par exemple `::ffff:192.0.2.1` (affichée `::ffff:c000:201`). Avec ce type d'adresse, l'espace d'adressage IPv4 est vu comme une partie de l'espace d'adressage IPv6.

tolérance de notation (rappel)

Lorsque l'adresse IPv4 occupe la partie basse de l'adresse IPv6, les 32 bits de poids faible (bits 97 à 128), la notation décimale pointée traditionnelle d'IPv4 est tolérée. Ainsi l'adresse `2001:db8:900d:cafe::` **`c0a8:a05`** peut être notée `2001:db8:900d:cafe::` **`192.168.10.5`** lors d'une saisie (configuration manuelle d'interface ou passage de paramètre en ligne de commande, ...). Cependant elle sera affichée sous sa forme canonique (RFC 5952) `2001:db8:900d:cafe::c0a8:a05` dans le journal de bord (log système) de la machine. Dans ce cas si la saisie peut nous sembler familière, la correspondance entre l'adresse IPv6 et l'adresse IPv4 embarquée est moins évidente à l'affichage.

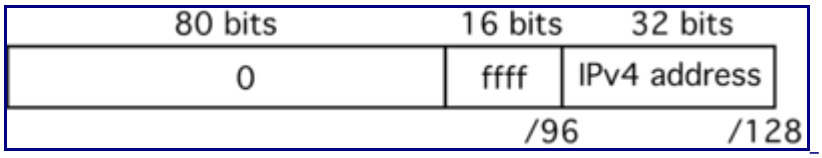


Figure 4: Adresse IPv4 imbriquée dans une adresse IPv6.

Quand la pile IPv4 d'un équipement reçoit un paquet et qu'une application utilise le format d'adresse d'IPv6, les adresses IPv4 imbriquées "source" et "destination" sont construites à partir des informations contenues dans l'en-tête du paquet. Réciproquement, quand une application émet des paquets avec des adresses IPv4 imbriquées, ceux-ci sont aiguillés vers la pile IPv4.

L'exemple suivant illustre ce fonctionnement. Le client Telnet compilé en IPv6 et fonctionnant sur une machine double pile peut contacter les équipements IPv4 en utilisant leur adresse IPv4 mais, bien sûr, les équipements IPv6 avec leur adresse IPv6.

```
telnet rhadamanthe
Trying 2001:db8:1002:1:2b0:d0ff:fe5c:4aee...
Connected to rhadamanthe.ipv6.rennes.enst-bretagne.fr.
Escape character is '^]'.

FreeBSD/i386 (rhadamanthe.ipv6.rennes.enst-br) (ttyp3)

login:^D

telnet bloodmoney
Trying::ffff:193.52.74.211...
Connected to bloodmoney.rennes.enst-bretagne.fr.
Escape character is '^]'.


SunOS UNIX (bloodmoney)
```

```
login:
```

Nous venons de le voir: une application compatible IPv6 peut dialoguer indifféremment en IPv4 et en IPv6, alors qu'une application utilisant un format d'adresse IPv4 restera limitée à ce protocole. Ceci ramène au problème du développement du code lié à la communication des applications. Plus généralement, le développement d'applications *IPv6 compatible* demande de nouvelles méthodes et pratiques au niveau de la programmation du fait du changement de la longueur de l'adresse IP, de la suppression de la diffusion d'IPv4 [13] . Pour rendre une application "IPv6 compatible", il faut qu'elle soit compilée ou recompilée avec l'interface de programmation (API) IPv6 ou, pour les applications écrites avec un langage de haut niveau d'abstraction, que la bibliothèque intègre IPv6. Ceci n'est bien sûr possible que sur les équipements pourvus d'un système ayant une pile IPv6, ce qui est aujourd'hui vrai dans la quasi-totalité des cas. Reste le problème des applications non recompilables (code source non disponible): ce genre de situation est traité par la suite dans l'activité de traduction.

Devant le coût des développements, la problématique de la compatibilité des applications à IPv6 doit être traitée dès le début, dans la stratégie de migration vers IPv6.

Problèmes liés à la double pile

L'intégration d'IPv6 devrait être indolore: l'utilisateur ne devrait pas voir de différence lorsqu'il accède à un service en IPv6. Cependant, en l'absence d'un minimum de précaution, ce souhait peut ne pas être satisfait, et le déploiement d'IPv6 en double pile peut dégrader le fonctionnement des services. Nous allons voir quels sont les problèmes engendrés au niveau du service perçu et comment les prévenir.

Le premier problème porte sur la phase d'établissement de la connexion comme expliqué par cet article [14] . Pour l'illustrer, prenons un service “monservice.org” accessible aux adresses IPv4 et IPv6 comme représenté sur la figure 5. L'application du client demande au "résolveur" DNS la liste des adresses IP pour joindre “monservice.org”, et ce dernier retourne une adresse IPv6 et une adresse IPv4. Conformément aux préconisations du RFC 6724 , la connexion commence avec l'adresse IPv6. Si la connexion IPv6 échoue, une autre adresse, potentiellement en IPv4, est essayée. Si le service est accessible sur une des adresses retournées par le DNS, le client finira par établir une connexion au service. L'inconvénient de cette méthode est que les tentatives de connexion sont bloquantes et donc effectuées de manière séquentielle. Le délai d'attente pour considérer qu'une connexion a échoué est de l'ordre de plusieurs dizaines de secondes.

Dans l'état actuel du déploiement d'IPv6, bien des sites ont une connexion IPv6 totalement ou partiellement inopérante. Si un serveur fonctionne en IPv4 et en IPv6, et que son client n'a qu'IPv4, il n'y aura pas de problème. Mais si le client a IPv6, tente de l'utiliser, mais que sa connectivité IPv6 est plus ou moins défaillante, il aura des temps de réponse très importants. Les utilisateurs percevront le service comme très dégradé. C'est la raison pour laquelle, encore aujourd'hui, il y a si peu de sites Internet accessibles en IPv6.

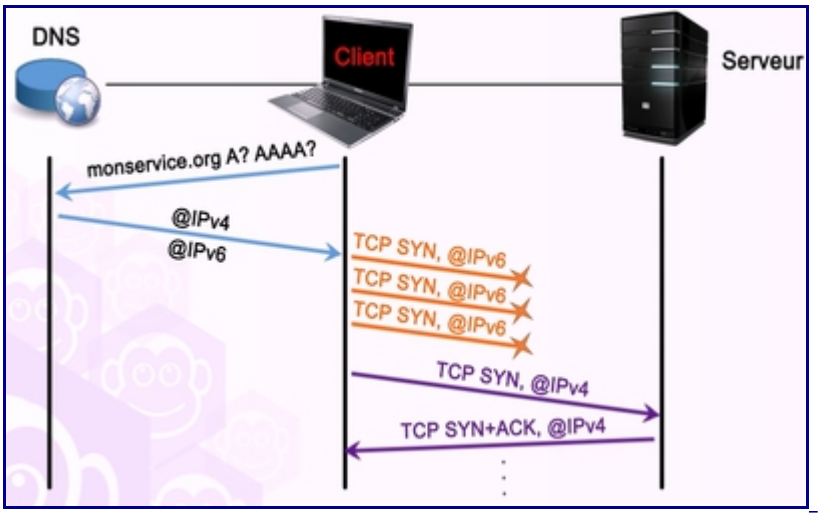


Figure 5: Établissement de connexion d'un client en double pile.

Le second problème est relatif à la taille des paquets IPv6, comme montré dans cet article [15] . Une fois la connexion établie en IPv6, l'utilisateur peut rencontrer des problèmes pour les échanges avec le serveur. En effet, en raison de l'utilisation de tunnels, IPv6 présente un problème de MTU bien plus souvent que IPv4. Le lien «standard» sur Internet a une MTU de 1500 octets, héritée d'Ethernet. Si, de bout en bout, tous les liens ont cette MTU, la machine émettrice peut fabriquer des paquets de 1500 octets et ils arriveront intacts. Mais, s'il y a sur le trajet un tunnel qui réduit la MTU, le problème de MTU peut se produire, comme la figure 6 le représente. Le problème de MTU se manifeste par le fait que les paquets de petite taille, tels ceux utilisés lors de l'établissement de la connexion, passent, mais les gros paquets, comme les transferts de fichiers avec HTTP, bloquent mystérieusement. Les paquets dépassant la MTU du chemin ne sont jamais remis à la destination. Si les messages ICMP avertissant de ce problème sont bloqués par un routeur sur le chemin, la source n'apprendra pas le problème et ne pourra donc pas s'adapter. La connexion va finalement se fermer pour cause d'inactivité (aucune réception n'est faite). Ce problème est assez sérieux dans l'Internet et a fait l'objet du RFC 4459 . Dans l'article [15] , le problème de MTU est détaillé et illustré par des captures de traces.

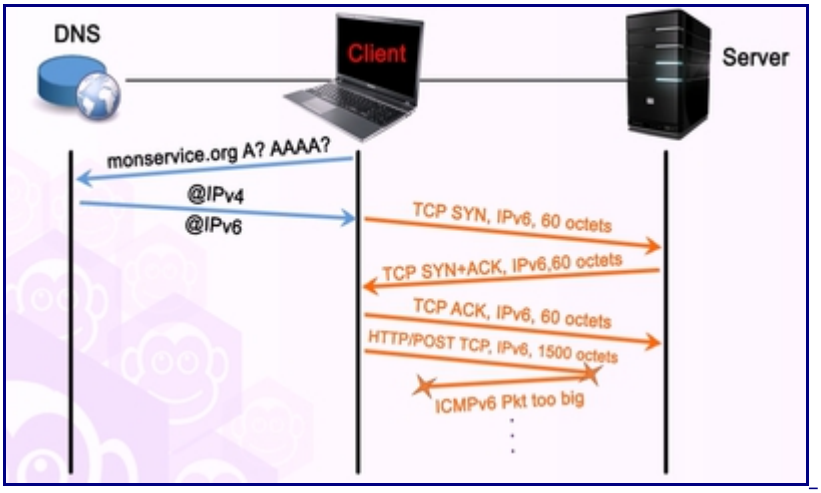


Figure 6: Le problème de MTU.

Le troisième problème porte sur la performance perçue pour un service reposant sur la

connectivité IPv6. Celle-ci sera évaluée comme dégradée à l'image de l'interactivité. La connectivité IPv6 est souvent constituée de tunnels. Si les sorties des tunnels sont trop éloignées du point d'entrée, le temps de réponse peut significativement augmenter et dépasser les valeurs souhaitables pour les applications interactives (ToIP, vidéoconférence, jeux en ligne...) et même pour le Web. L'utilisateur verra alors sa qualité de service chuter par rapport au réseau simple pile IPv4 et ce, même si la connectivité IPv6 est parfaitement fonctionnelle. Ce problème de délai important en IPv6 est illustré par la figure 7 dans laquelle le temps de réponse (noté RTT *Round Trip Time*) est plus long en IPv6 du fait d'un chemin plus long en terme de noeuds de commutation et en distance.

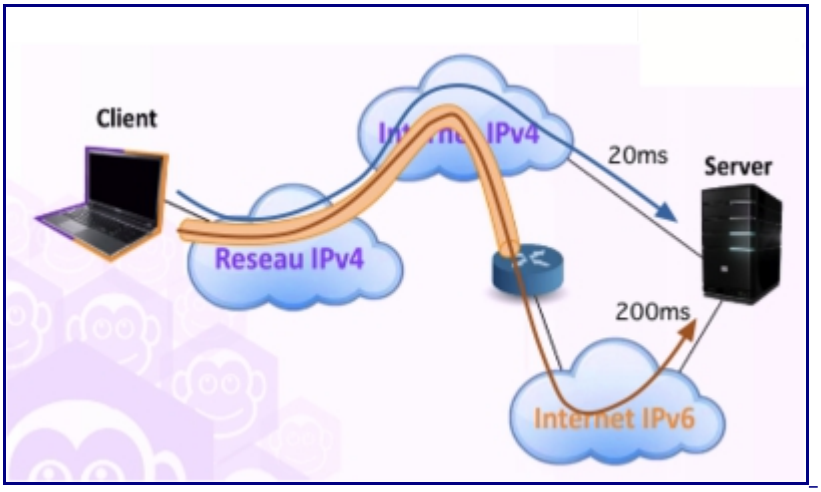


Figure 7: Illustration des délais importants en IPv6.

Des solutions ont été proposées pour éviter que les utilisateurs désactivent IPv6 en réponse à la baisse de performance qu'ils observent. Il est ici intéressant de noter que les problèmes que nous venons de décrire trouvent leur origine dans l'utilisation d'IPv4 dans la connectivité IPv6. La bonne solution serait de généraliser IPv6 pour un usage sans IPv4. En attendant, les solutions proposées sont détaillées par la suite afin qu'IPv6 fonctionne aussi bien qu'IPv4.

Les problèmes qui apparaissent lors de la phase d'établissement de la connexion sont dus au fait que le client tente de se connecter séquentiellement aux différentes adresses du service. IPv6 étant testé en premier lieu, il faut attendre que la tentative de connexion échoue, ce qui peut prendre plusieurs secondes. Le RFC 8305 propose d'essayer d'établir une connexion TCP à la fois en IPv4 et en IPv6 et de conserver la première connexion établie. Le RFC précise que les demandes de connexion doivent être émises de sorte que ce soit celle portée par IPv6 qui puisse être conservée. Les navigateurs Internet ont pris en compte ces recommandations mais les mises en oeuvre divergent comme le rapporte l'article [16] :

- Le navigateur Safari conserve, dans une table, le délai moyen pour atteindre chaque adresse du serveur. L'adresse ayant le délai le plus court est utilisée en priorité, mais si elle ne répond pas avant le délai attendu, l'adresse suivante est essayée. La demande de connexion est émise en décalé sur les différentes adresses du serveur. La première connexion établie sera utilisée pour la suite des échanges. Cette solution peut cependant induire un délai non négligeable si le serveur comporte de nombreuses adresses et que seule la dernière (celle de plus long délai moyen) est accessible.

- Le navigateur Chrome mesure les délais pour l'obtention des adresses IPv4 et IPv6 via le DNS. Il tente d'établir une connexion avec le protocole dont l'adresse a été obtenue en premier. Notons que pour maximiser les chances de réussite, il envoie deux segments SYN en parallèle avec des ports "source" différents. Si aucun segment SYN + ACK n'est reçu après 250 ms, un dernier segment SYN est émis depuis un troisième port. Si aucun segment SYN + ACK n'est reçu après un total de 300 ms, le protocole suivant sera essayé. Dans le cas où un problème apparaît avec un seul des protocoles, le délai est donc au maximum allongé de 300 ms. Si un problème apparaît avec les deux protocoles, c'est la méthode par défaut de l'OS qui sera utilisée. Notons que si le RTT est supérieur à 300 ms, les deux protocoles seront systématiquement utilisés.
- Le navigateur Firefox implémente strictement les recommandations du RFC 8305 et essaye les deux protocoles en parallèle.

Des mises en oeuvre similaires à celles des navigateurs sont à développer pour les clients des différentes applications (e.g. mail, VoIP, chat...). Pour ne pas avoir les inconvénients des accès séquentiels, il faudrait ne pas attendre l'expiration des temporisateurs de l'OS, mais choisir des temps de garde plus agressifs et ayant moins d'impact pour les utilisateurs. Par exemple, si IPv6 ne répond pas avant un délai de 300 ms ou deux RTT, alors IPv4 est essayé.

Notons cependant que le parallélisme a un effet pervers pour les opérateurs. En effet, l'utilisation des CGN pour la connectivité IPv4 leur est coûteuse et le maintien des états relatifs à l'ouverture de chaque connexion consomme des ressources. En suggérant l'ouverture de plusieurs connexions en parallèle, le RFC 8305 va à l'encontre des intérêts des opérateurs et potentiellement des utilisateurs si les CGN sont saturés. C'est pourquoi, il suggère d'essayer en priorité le protocole qui ne générera pas d'état dans le réseau, à savoir IPv6.

Pour les problèmes de MTU, la solution réside dans le fait de forcer les utilisateurs à choisir une faible MTU, par exemple 1400 octets, dans l'espoir qu'il n'y ait pas un lien sur la route dont la MTU soit inférieure à cette valeur. Cela peut être fait lors de la configuration de l'interface réseau ou lors de l'établissement d'une connexion TCP en réduisant la taille maximum des segments autorisée. Cette réduction est effectuée par le routeur (*MSS clamping*). Dans le RFC 4821 , les auteurs proposent une solution qui ne repose pas sur ICMP. L'idée consiste à ce que TCP relève la taille des segments perdus. Si ce sont les segments de grande taille, TCP diminue la MSS (*Maximum Segment Size*) de lui-même (et, par voie de conséquence, la valeur de la MTU).

Les problèmes de performance en termes de délai sont dus à l'utilisation de tunnels. La solution réside dans la sélection de points de sorties plus proches pour les tunnels. Au moment de la rédaction de ce document, le problème de délai n'a pas de solution (au niveau application) faisant l'objet d'une recommandation similaire à celle du RFC 8305 .

Conclusion

Le mécanisme de double pile permet de résoudre les craintes liées à la migration vers IPv6. Dès lors, il ne s'agit plus d'une migration mais d'une intégration de IPv6 dans le réseau existant. Le réseau IPv4 reste pleinement fonctionnel et l'intégration d'IPv6 ne risque pas de

compromettre le bon fonctionnement des services déployés. En effet, quand cela est possible, la communication se fait en utilisant la nouvelle version du protocole. Dès qu'un des éléments n'est pas compatible (réseau, système d'exploitation, application), le protocole IPv4 est utilisé. Le principal intérêt réside dans l'adaptation progressive de son système d'information et de son personnel à IPv6.

Notons que le déploiement double pile ne doit être que transitoire car il ne résout pas le problème de la pénurie d'adresses puisque chaque machine doit disposer d'une adresse IPv4 et d'une adresse IPv6. Cela complique aussi les mécanismes de configuration automatique et augmente la charge pour l'administrateur réseau. Lors de l'activation d'IPv6 pour un service existant en IPv4, il faut prendre des précautions afin que la qualité perçue par l'utilisateur ne soit pas dégradée.

Références bibliographiques

1. ↑ 1.0 1.1 Huston, G. (2008). The ISP Column. The Changing Foundation of the Internet: Confronting IPv4 Address Exhaustion
2. ↑ Bortzmeyer, S. IPv6 ou l'échec du marché
3. ↑ Wikipedia. Comparison of IPv6 support in operating systems
4. ↑ Linux Review. Free IPv4 to IPv6 Tunnel Brokers
5. ↑ Botzmeyer, S. (2006). Programmer pour IPv6 ou tout simplement programmer à un niveau supérieur?
6. ↑ Matthews, P. Kuarsingh, V. (2015). Internet-Draft. Some Design Choices for IPv6 Networks
7. ↑ Cisco. (2011). White paper. Solution Overview—Getting Started with IPv6
8. ↑ RIPE documents. (2012). Requirements for IPv6 in ICT Equipment
9. ↑ Marsan, C.D. (2010). Network World. U.S. military strong-arming IT industry on IPv6
10. ↑ Horley, E. (2013) IPv6 Unique Local Address or ULA - what are they and why you shouldn't use them
11. ↑ IANA. IPv6 Global Unicast Address Assignments
12. ↑ Rooney, T. (2013). Deploy 360 Programme. Internet Society. IPv6 Address Planning: Guidelines for IPv6 address allocation
13. ↑ Cisco. (2011); White paper. IPv6 and Applications
14. ↑ Bortzmeyer, S. (2011). Le bonheur des globes oculaires (IPv6 et IPv4)
15. ↑ 15.0 15.1 Huston, G. (2009). The ISP Column. A Tale of Two Protocols: IPv4, IPv6, MTUs and Fragmentation
16. ↑ Huston, G. (2012). The ISP Column. Bemused Eyeballs: Tailoring Dual Stack Applications for a CGN Environment

Pour aller plus loin

Scénarios de déploiement

- Guide de déploiement d'IPv6 par RIPE: Deploy IPv6 Now
- Deploying IPv6 in the Home and Small Office/Home Office (SOHO)

Sécurité

- Bortzmeyer, S. (2013). Exposé sur la sécurité d'IPv6 à l'ESGI
- Cisco White paper (2011). IPv6 Security Brief

Pour développer des applications compatibles avec IPv6:

- Livre blanc ARIN
- Cisco. White Paper. IPv6 and Applications
- Bortzmeyer, S. (2013) Lier une prise à IPv6 seulement ou bien aux deux familles, v4 et v6?

RFC et leur analyse par S. Bortzmeyer:

- RFC 1918 Address Allocation for Private Internets
- RFC 2851 Textual Conventions for Internet Network Addresses
- RFC 3315 Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Analyse
- RFC 3587 IPv6 Global Unicast Address Format
- RFC 3596 DNS Extensions to Support IP Version 6
- RFC 3736 Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6
- RFC 4038 Application Aspects of IPv6 Transition
- RFC 4057 IPv6 Enterprise Network Scenarios
- RFC 4193 Unique Local IPv6 Unicast Addresses Analyse
- RFC 4213 Basic Transition Mechanisms for IPv6 Hosts and Routers Analyse
- RFC 4459 MTU and Fragmentation Issues with In-the-Network Tunneling Analyse
- RFC 4472 Operational Considerations and Issues with IPv6 DNS Analyse
- RFC 4477 Dynamic Host Configuration Protocol (DHCP): IPv4 and IPv6 Dual-Stack Issues
- RFC 4821 Packetization Layer Path MTU Discovery Analyse
- RFC 4861 Neighbor Discovery for IP version 6 (IPv6) Analyse
- RFC 4862 IPv6 Stateless Address Autoconfiguration Analyse
- RFC 4941 Privacy Extensions for Stateless Address Autoconfiguration in IPv6 Analyse
- RFC 5211 An Internet Transition Plan Analyse
- RFC 5375 IPv6 Unicast Address Assignment Considerations Analyse
- RFC 5887 Renumbering Still Needs Work Analyse
- RFC 5902 IAB thoughts on IPv6 Network Address Translation Analyse
- RFC 6018 : IPv4 and IPv6 Greynets Analyse
- RFC 6092 Recommended Simple Security Capabilities in Customer Premises Equipment for Providing Residential IPv6 Internet Service Analyse

- RFC 6104 Rogue IPv6 Router Advertisement Problem Statement Analyse
- RFC 6106 IPv6 Router Advertisement Options for DNS Configuration
- RFC 6164 Using 127-Bit IPv6 Prefixes on Inter-Router Links Analyse
- RFC 6177 IPv6 Address Assignment to End Sites
- RFC 6180 Guidelines for Using IPv6 Transition Mechanisms during IPv6 Deployment Analyse
- RFC 6296 IPv6-to-IPv6 Network Prefix Translation
- RFC 6724 Default Address Selection for Internet Protocol version 6 (IPv6) Analyse
- RFC 7010 IPv6 Site Renumbering Gap Analysis Analyse
- RFC 7084 Basic Requirements for IPv6 Customer Edge Routers Analyse
- RFC 7113 Implementation Advice for IPv6 Router Advertisement Guard (RA-Guard) Analyse
- RFC 7123 Security Implications of IPv6 on IPv4 Networks Analyse
- RFC 7381 Enterprise IPv6 Deployment Guidelines Analyse
- RFC 7707 Network Reconnaissance in IPv6 Networks Analyse
- RFC 8305 Happy Eyeballs Version 2: Better Connectivity Using Concurrency Analyse

Présentations sur le déploiement d'IPv6

- Scott Hogg (2015) Keynote in Nanog. Successfully Deploying IPv6
- Leslie Nobile, Mark Kosters. (2015) Keynote in Nanog. Moving to IPv6
- Huston, G (2010) An Economic Perspective on the Transition to IPv6
- Cisco (2005) Enterprise IPv6 Deployment

Activité 43: Établir la connectivité IPv6

Problématique

Lorsqu'un réseau IPv6 veut joindre un autre réseau IPv6 séparé par un réseau en IPv4, le problème consiste à offrir une connectivité IPv6 entre ces 2 réseaux. La bonne solution serait de les interconnecter avec IPv6 uniquement, c'est-à-dire sans avoir recours à IPv4. Mais, quand cela n'est pas possible, la connectivité s'établit par des mécanismes de niveau réseau reposant sur le principe du tunnel. Ainsi, le tunnel est la solution pour utiliser une infrastructure IPv4 existante pour acheminer du trafic IPv6 [1] .

Les tunnels peuvent s'utiliser aussi bien pour la connectivité d'un site IPv6 avec l'Internet v6 (si le FAI n'offre pas encore nativement cette connectivité) que pour l'intérieur d'un site en IPv4 si celui-ci comporte des parties en IPv6 non connexes. Par la suite, nous allons décrire le fonctionnement d'un tunnel IPv6 sur IPv4 en montrant le principe du tunnel configuré et celui du tunnel automatique. De nombreuses techniques à base de tunnels existent, comme le rappelle le RFC 7059 . Nous retiendrons la technique adaptée à une simple connectivité avec l'Internet v6 et celle pour établir des tunnels automatiques à l'intérieur d'un site.

Principe du tunnel IPv6 sur IPv4

Le tunnel est un mécanisme bien connu dans le domaine des réseaux, qui consiste à faire qu'une unité de transfert d'un protocole (PDU Protocol Date Unit) d'une couche se trouve encapsulée dans la charge utile de l'unité de transfert (PDU) d'un autre protocole de la même couche. Ainsi, des protocoles «transportés» peuvent circuler dans un réseau construit sur un protocole encapsulant. Dans le cas d'IPv6, cette technique a été définie dans le RFC 4213 et porte le nom de *6in4* . L'encapsulation du paquet IPv6 dans le paquet IPv4 s'effectue comme illustré par la figure 1. Le paquet IPv6 occupe le champ `données` du paquet IPv4. Le champ `protocol` de l'en-tête IPv4 prend alors la valeur 41 pour indiquer IPv6. Les extrémités du tunnel peuvent être des hôtes ou des routeurs. Les noeuds, aux extrémités du tunnel, sont appelés des tunneliers (*tunnel end point*) et peuvent être configurés manuellement ou avoir une configuration dynamique. Dans ce dernier cas, on parle aussi de tunnel automatique.

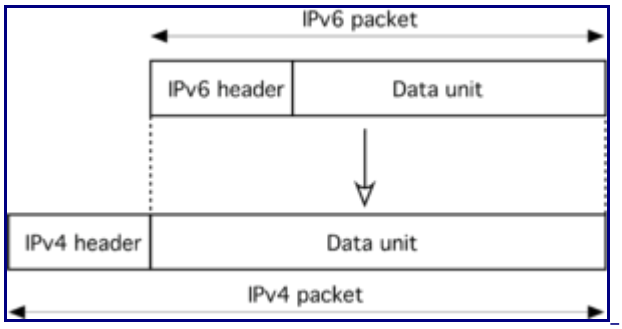


Figure 1: Encapsulation pour un tunnel.

Le notion de tunnel équivaut à un câble virtuel bidirectionnel permettant d'assurer une liaison

point à point entre deux nœuds IPv6 ou entre deux réseaux IPv6 et fournir ainsi une connectivité comme l'illustre la figure 2.

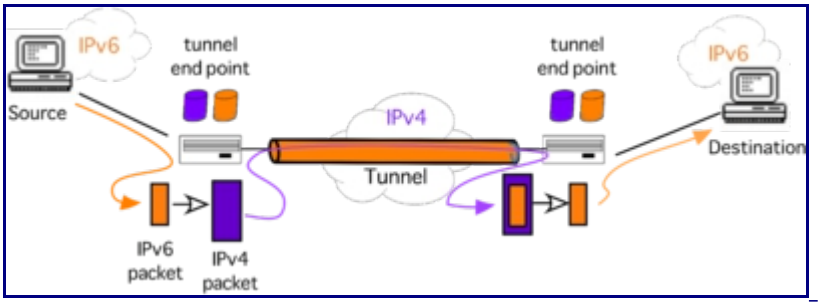


Figure 2: Tunnel entre des réseaux IPv6.

Les tunneliers sont, dans cet exemple, des routeurs en double pile. L'architecture de protocoles peut se représenter par la figure 3. Cette figure montre la réception d'un paquet en IPv6 natif et son émission dans le tunnel. La réception d'un paquet IPv6 du tunnel et son émission en natif empruntent le même chemin, mais en sens opposés. Le routeur tunnelier est un noeud qui, comme tous les routeurs, possède au moins 2 interfaces, une sur le réseau IPv4 et une sur le réseau IPv6,(cela peut être deux interfaces physiques distinctes, ou deux interfaces virtuelles sur la même interface physique). Il convient à ce stade de rappeler que les systèmes de transmission comme Ethernet ou Wi-Fi sont multi-protocoles: ils sont capables de transmettre des trames contenant des paquets IPv4 comme IPv6.

La particularité d'un tunnelier est qu'il dispose en plus d'une interface logique interne, extrémité du tunnel sur laquelle s'opère l'encapsulation/décapsulation des paquets IPv6 dans le champ données des paquets IPv4. Cette interface dispose d'une adresse IPv4 et d'une adresse IPv6 (GUA, ULA, ou d'une adresse, à préfixe nul " *IPv4 compatible* " ou " *IPv4 mapped* " étant donné qu'il s'agit d'une interface logique interne au routeur). Cette adresse IP sera l'adresse de « prochain saut » pour les routes vers les préfixes IPv6 à atteindre à l'autre extrémité du tunnel. Cela peut, également, être la route par défaut s'il s'agit d'un tunnel reliant un îlot IPv6 à l'Internet v6.

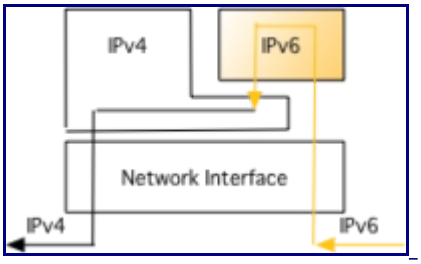


Figure 3: Architecture d'un routeur tunnelier.

La différence avec un câble réel porte sur la taille de la MTU. En raison de l'encapsulation dans IPv4, un tunnel diminue la MTU effective d'une vingtaine d'octets. Normalement, la fragmentation et la découverte de la MTU du chemin servent à adapter la taille des paquets IPv6 à la MTU du tunnel. En pratique, des routeurs mal configurés peuvent filtrer les messages ICMP, dont le type utilisé pour la découverte de la MTU (message ICMP *Packet Too Big*). Ceci a pour effet d'empêcher la détermination de la MTU, et donc rend la fragmentation IPv6

inopérante. Cela génère des erreurs de transmission, comme un client qui parvient a communiquer avec un serveur tant qu'il envoie des petits paquets mais qui ne reçoit rien quand il demande un fichier, c'est-à-dire quand les paquets de taille importante sont émis. Pour rappel, les paquets IPv6, lorsqu'il ne peuvent être transmis par un routeur à cause de leur taille, sont supprimés par celui-ci. Conjointement à la destruction du paquet, le message ICMP *Packet Too Big* est envoyé à la source pour que celle-ci ajuste la taille du paquet.

Tunnel configuré

La configuration d'un tunnel consiste à créer une interface réseau représentant l'extrémité du tunnel, indiquer les adresses IPv4 des extrémités, allouer un préfixe IPv6 pour ce lien point à point virtuel, et spécifier les routes pour suivre ce tunnel. Dans le cas d'un tunnel configuré, les informations de la réalisation du tunnel sont indiquées par un administrateur.

Pour illustrer la configuration d'un tunnel, la figure 4 montre le cas d'un tunnel reliant un hôte sous Linux avec un routeur. Dans cette situation, le fichier de configuration (ci-dessous) indique la création de l'interface nommée *6in4* . Le point de sortie du tunnel est indiqué. Comme les paquets IPv6 sont encapsulés dans un paquet IPv4, celui-ci doit avoir une adresse "source" et une adresse "destination". L'adresse "destination" est celle du point de sortie du tunnel, ici `192.0.3.1` . L'interface *6in4* est configurée par l'adresse et le préfixe `2001:db8:caf:1::2/64` . L'adresse IPv6 de l'interface *6in4* du coté du routeur ainsi que la route par défaut sont ensuite précisées. Ces instructions sont décrites dans la documentation en ligne de Linux dans la section 5 de *interfaces* .

```
auto 6in4
iface 6in4 inet6 v4tunnel
address 2001:db8:caf:1::2
netmask 64
endpoint 192.0.3.1
gateway 2001:db8:caf:1::1

post-up route -A inet6 add::/0 dev 6in4
```

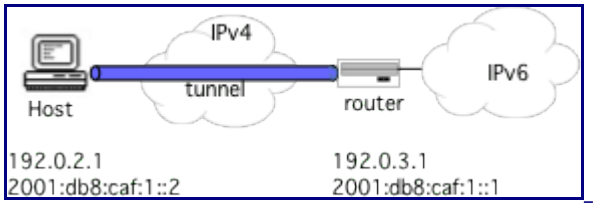


Figure 4: Cas d'un tunnel configuré.

Les performances d'un tunnel vont dépendre de sa longueur. Pour éviter d'avoir des délais trop importants, il convient de configurer un tunnel vers le point IPv6 le plus proche.

Tunnel automatique

Un tunnel configuré demande un travail de configuration, ce qui peut être vu comme un inconvénient. Des solutions d'automatisation ont été étudiées, qui ont comme principe de contenir l'adresse IPv4 du tunnelier de destination dans l'adresse IPv6. La technique de transition *6to4* décrite par le RFC 3056 suit ce principe. Elle vise à interconnecter entre eux des

sites IPv6 isolés en créant des tunnels automatiques IPv6 dans IPv4 en fonction du destinataire des données. La figure 5 montre 2 réseaux IPv6 communiquant entre eux via un tunnel automatique *6to4* . Le point fort du mécanisme présenté ici est l'automatisation, où l'intervention de l'administrateur est réduite à une phase de "configuration/initialisation" du service, et non à une phase de configuration des tunnels.

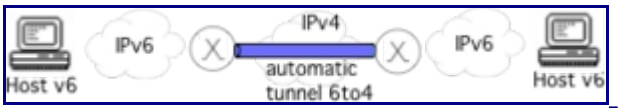


Figure 5: *6to4* .

Comme *6in4* , l'encapsulation des paquets IPv6 s'effectue directement dans les paquets IPv4. *6to4* bénéficie d'un préfixe IPv6 réservé: **`2002::/16`** du plan d'adressage agrégé (*adressage public*) RFC 3587 . Le préfixe de l'adresse IPv6 d'un tunnelier est composé automatiquement en concaténant le préfixe réservé et l'adresse IPV4 "unicast globale" de ce tunnelier, comme montré par la figure 6. Ainsi, un préfixe IPv6 de longueur 48 bits peut être aisément construit en utilisant l'adresse IPv4 du noeud en bordure des réseaux IPv4 et IPv6. Ce préfixe peut identifier un site IPv6. De cette manière, *6to4* se suffit à lui-même pour créer un préfixe IPv6 pour un site en toute autonomie. On peut remarquer que ce plan d'adressage est conforme au plan d'adressage global actuellement en vigueur puisqu'il réserve 16 bits pour numéroter les réseaux du site (noté SID) et 64 bits pour les identifiants d'interfaces (noté IID).

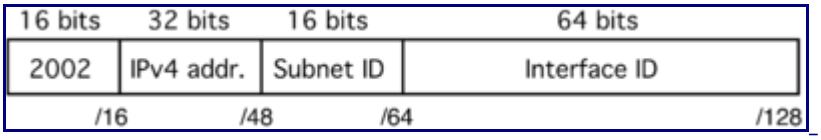


Figure 6: Format d'une adresse *6to4* .

Par exemple, la figure 7 illustre le mécanisme de construction d'un préfixe. Le routeur *6to4* se trouve en bordure du réseau. Il est connecté à la fois à l'Internet v4 et à un site IPv6. C'est un noeud en double pile; il possède obligatoirement une adresse IPv4, `192.0.2.1` dans l'exemple. Il va s'en servir pour construire le préfixe `2002:c000:201::/48` (0xc0 = 192). Ce préfixe de 48 bits va être utilisé par l'ensemble des noeuds IPv6 du site.

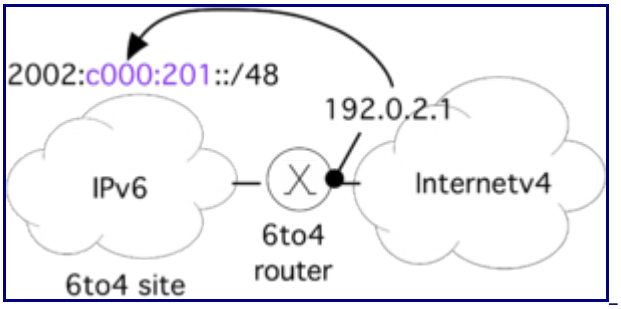


Figure 7: Construction d'un préfixe *6to4* .

Au niveau du routage, la figure 8 présente l'envoi d'un paquet IPv6 de l'hôte A vers l'hôte B. Il est important de noter ici que A et B sont des hôtes ayant une adresse IPv6 prise dans le plan d'adressage *6to4* . Dans un premier temps, A interroge le DNS pour connaître l'adresse IPv6 de B. Dans notre exemple, la réponse est `2002:c000:301:1::8051` . Dans un second temps, l'hôte A émet le paquet vers cette destination. Ce paquet IPv6, dont l'adresse de destination

commence par le préfixe `2002::/16` , doit passer par un tunnel *6to4* . C'est au routeur *6to4* du site de A qu'il revient d'effectuer cette opération. Ainsi, le paquet doit suivre la route IPv6 `2002::/16` pour atteindre ce routeur *6to4* . Ce dernier analyse l'adresse IPv6 de destination et trouve l'adresse IPv4 de l'autre extrémité du tunnel (`192.0.3.1` dans l'exemple). Il pourra alors effectuer la transmission en encapsulant le paquet IPv6 dans un paquet IPv4. C'est cette encapsulation qui forme le tunnel. Le routeur *6to4* du coté de B désencapsule le paquet IPv6 et le route normalement vers sa destination finale B en utilisant le routage interne.

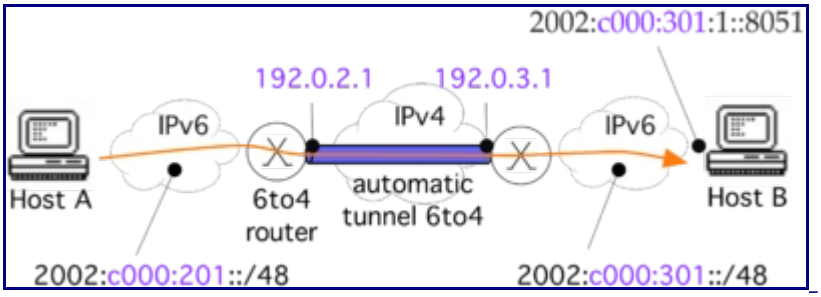


Figure 8: Acheminement d'un paquet IPv6 en *6to4* .

Si *6to4* est une technique intéressante pour relier deux nuages IPv6 à travers un nuage IPv4, elle se complique et n'est pas optimale lorsqu'il s'agit de communiquer avec une machine dont l'adresse est issue d'un plan de numérotation globale. Car, un site isolé utilisant *6to4* n'est pas directement connecté à l'Internet v6. Dans ce cas, le site *6to4* doit passer par un relais (ou routeur passerelle) qui est connecté à la fois à l'Internet v6 et à l'Internet v4. Dans le RFC 3068 , il a été proposé d'utiliser une adresse anycast qui soit commune à tous ces relais à travers le monde. Une adresse anycast est définie pour chaque version du protocole IP. N'importe quel relais *6to4* de l'Internet est joignable ainsi à cette adresse. Aussi, cette adresse est annoncée par les relais *6to4* et donc, existe de multiples fois sur l'Internet. Il est du ressort du routage (selon le principe du *routage anycast*) d'identifier le relais le plus proche de l'émetteur pour lui acheminer ses paquets.

Dans le contexte dérégulé de l'Internet actuel, aucun FAI n'a intérêt à offrir un tel service mutualisé, car il verra le trafic IPv6 des clients de ses concurrents charger son infrastructure au détriment de la qualité de service de ses propres clients. De plus, le relais pose aussi le problème de sa résistance au facteur d'échelle et de la qualité de sa connectivité. Comme il n'est pas possible de choisir le relais, et que leur qualité varie fortement, ceci rend la communication passant par *6to4* très imprévisible. Enfin, le routage n'est pas optimal et presque assurément asymétrique:

- le site *6to4* peut avoir choisi un routeur passerelle loin du destinataire;
- le site ayant un plan d'adressage global envoie les paquets vers le routeur passerelle le plus proche au sens du routage IPv6.

Cette asymétrie des trajets aller et retour vers l'Internet v6 peut se voir sur la figure 9. Le problème de l'asymétrie est qu'elle complique la tâche de recherche d'erreurs en cas de dysfonctionnement et, surtout, qu'elle introduit des délais de propagation élevés, dus à la longueur des tunnels.

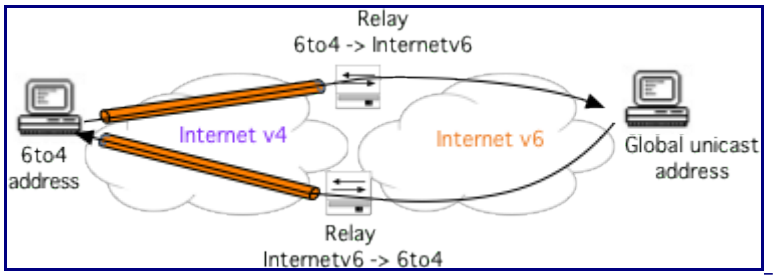


Figure 9: Routage asymétrique.

L'analyse du service de connectivité en *6to4* montre une mauvaise qualité [2] . En effet, le taux de panne des communications passant par *6to4* est significativement élevé. Ceci a eu pour effet de ralentir le déploiement d'IPv6. Bien que l'idée de tunnel automatique développée par *6to4* soit intéressante, sa mise en oeuvre est problématique. En mai 2015, par la publication du RFC 7526 , l'IETF prône la dépréciation de l'annonce du préfixe anycast réservé aux relais *6to4* ; ce qui, de fait, officialise l'abandon de cette technique. Nous verrons par la suite que *6to4* est finalement supplanté par la technique de tunnel connue sous le nom de *6rd* (RFC 5969).

Connectivité d'un site isolé: *Tunnel Broker*

La croissance du réseau IPv6 a commencé en s'appuyant sur l'infrastructure de communication de IPv4. Les premiers tunnels étaient configurés manuellement et pouvaient être très longs (et donc peu performants). La longueur d'un tunnel s'apprécie par le nombre de sauts IPv4 et/ou la distance qui sépare les 2 extrémités du tunnel. Pour des personnes non qualifiées, ceci reste complexe tant du point de vue technique que du point de vue du choix du point de sortie du tunnel. La constitution d'un tunnel a été simplifiée par l'introduction du *Tunnel Broker* [RFC 3053]. Les *Tunnel Brokers* représentent une méthode pour connecter un réseau IPv6 à l'Internet v6. L'idée du *Tunnel Broker* consiste à mettre en oeuvre une interaction de type "client/serveur". La partie cliente est localisée côté utilisateur tandis que la partie serveur traite les demandes de tunnels. Le modèle du *Tunnel Broker* est représenté par la figure 10.

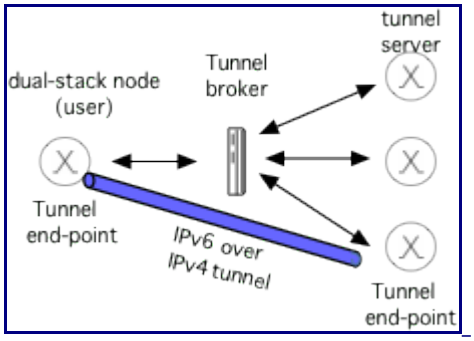


Figure 10: Modèle du *Tunnel Broker* .

La création d'un tunnel à l'aide d'un *Tunnel Broker* fonctionne de la manière indiquée par la figure 11; à savoir:

1. Une machine "double pile" du réseau IPv6 (typiquement un routeur) négocie avec le *Tunnel Broker* afin de s'authentifier et d'obtenir les informations de configuration du tunnel ainsi qu'un préfixe délégué.
2. Le *Tunnel Broker* configure le serveur de tunnel retenu.

3. Le *Tunnel Broker* envoie le script de configuration à la machine "double pile" coté utilisateur.
4. Cette dernière, en exécutant le script reçu, crée le tunnel. Elle va ensuite encapsuler ses paquets IPv6 dans des paquets IPv4 à destination du serveur de tunnels, qui sert également de routeur. Ainsi, une communication en IPv6 peut s'effectuer entre des noeuds d'un réseau IPv6 isolé avec des noeuds de l'Internet v6.

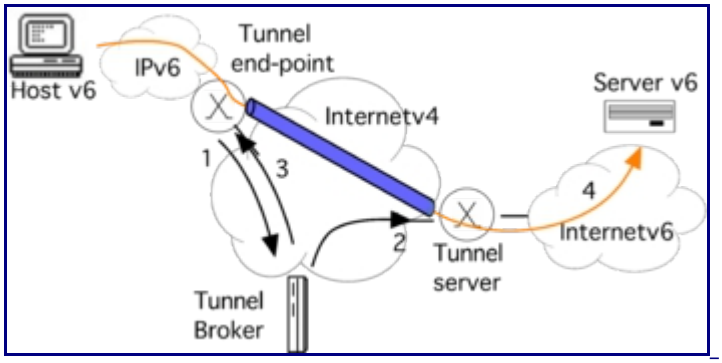


Figure 11: Configuration d'un *Tunnel Broker* avec TSP.

La négociation est opérée à l'aide du protocole TSP (*Tunnel Set Up Protocol*) [RFC 5572]. En l'absence de TSP, la demande de connexion au *Tunnel Broker* est réalisée par une interface web dont l'URL est connue à l'avance. Par cette interface, les paramètres nécessaires à l'établissement du tunnel entre le noeud de l'utilisateur et le serveur de tunnels sont récupérés. Le protocole de négociation TSP automatise cet échange. Plus précisément, TSP traite les paramètres suivants:

- l'authentification de l'utilisateur;
- le type de tunnel:
  - tunnel IPv6 sur IPv4 [RFC 4213],
  - tunnel IPv4 sur IPv6 [RFC 2473],
  - tunnel IPv6 sur UDP-IPv4 pour la traversée de NAT;
- les adresses IPv4 pour les deux extrémités du tunnel;
- l'adresse IPv6 assignée lorsque le client TSP est un terminal;
- le préfixe IPv6 alloué lorsque le client TSP est un routeur.

TSP s'appuie sur l'échange de simples messages XML dont un exemple est donné ci-dessous. Cet exemple correspond à la demande de création d'un tunnel simple par un client TSP:

```
-- Successful TCP Connection --
C:VERSION=2.0.0 CR LF
S:CAPABILITY TUNNEL=V6V4 AUTH=ANONYMOUS CR LF
C:AUTHENTICATE ANONYMOUS CR LF
S:200 Authentication successful CR LF
C:Content-length: 123 CR LF
tunnel action="create" type="v6v4"
client
address type="ipv4"1.1.1.1/address
/client
/tunnel CR LF
S: Content-length: 234 CR LF
200 OK CR LF
```

```
tunnel action="info" type="v6v4" lifetime="1440"
server
address type="ipv4"206.123.31.114/address
address type= "ipv6"3ffe:b00:c18:ffff:0000:0000:0000:0000/address
/server
client
address type="ipv4"1.1.1.1/address
address type= "ipv6"3ffe:b00:c18:ffff::0000:0000:0000:0001/address
address type="dn"userid.domain/address
/client
/tunnel CR LF
C: Content-length: 35 CR LF
tunnel action="accept"/tunnel CR LF
```

La connectivité offerte par les *Tunnel Brokers* est en général fournie à titre provisoire (soit en attendant que l'offre des FAI soit disponible, soit pour faire des tests de validation, par exemple). Elle peut aussi être une première étape pour un prestataire de services pour procurer de la connectivité IPv6 à ses usagers. Afin de promouvoir le passage à IPv6, les *Tunnel Brokers* sont souvent gratuits [3] . Lorsque le *Tunnel Broker* a une faible répartition géographique de ses serveurs de tunnels, pour certains utilisateurs, la longueur des tunnels reste un problème.

Connectivité sur une infrastructure IPv4: *6rd*

Le mécanisme *6rd* (*IPv6 Rapid Deployment*), proposé par le RFC 5569 après son déploiement par Free, a été étendu pour devenir un standard par le RFC 5969 . *6rd* reprend le principe des tunnels automatiques du *6to4* mais apporte des modifications pour éviter les défauts de performances et de fiabilité observés sur *6to4* . *6rd* est destiné à un opérateur pour offrir une connectivité IPv6 alors que son infrastructure repose sur IPv4. Cet opérateur peut être aussi bien public, comme un FAI, ou privé, comme une entreprise ou une administration.

6rd est une variante de *6to4* comme cela a été précisé. La différence majeure se situe sur l'utilisation du préfixe IPv6 propre à l'opérateur plutôt que le préfixe commun à tous, employé par *6to4* (`2002::/16`). Il s'ensuit que l'opérateur doit installer ses propres relais pour offrir la connectivité avec l'Internet v6. Le relais est un routeur de bordure équipé en "double pile". Dans la figure 12, qui schématise l'architecture de *6rd* , le routeur de bordure est noté, selon la terminologie du RFC 5969 , " *6rd* BR"(*Border Relays*). Le préfixe IPv6 propre à cet opérateur est noté "pref6rd". En contrepartie de l'installation des relais, l'opérateur contrôle les tunnels. Il peut ainsi garantir que la voie "retour" est symétrique à la voie "aller". Autre conséquence, les tunnels sont plus courts: ils servent à passer la section IPv4 de l'opérateur. En contrôlant les tunnels, les principaux défauts du déploiement de *6to4* , comme des délais importants ou l'asymétrie, sont corrigés. Avec *6rd* , on se retrouve dans le cas classique où les routeurs internes (dont les relais) traitent le trafic des noeuds internes. Ainsi, ces relais ne servent que les clients de l'opérateur (contrairement à *6to4* où les relais étaient mutualisés et publics). Comme *6to4* , *6rd* est sans état, et les routeurs de bordures peuvent utiliser une même adresse IPv4 (que l'on qualifie d'anycast). En somme, l'idée de *6rd* est de restreindre la technique *6to4* à un usage interne et local.

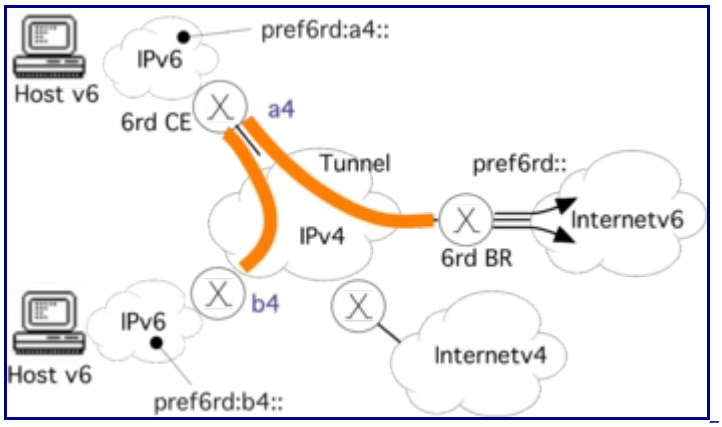


Figure 12: Architecture de *6rd* .

Le format de l'adresse IPv6 dérive d'un préfixe `2000::/3` pris dans le plan d'adressage global unicast. Il utilise le préfixe alloué au FAI par son registre régional (RIR) et non pas le préfixe réservé 6to4 (`2002::/16`), partagé entre tous FAI.

Le préfixe 6rd est automatiquement calculé par l'extrémité client (CE, Customer Edge, la box fournie par le FAI) en concaténant le préfixe 6rd du FAI et tout ou partie de l'adresse IPv4 allouée par ce FAI sur l'interface WAN IPv4 du CE (la box).

```
|     n bits     |     i bits     |   s bits  |  128 - n - i - s bits  |
+----------------+----------------+-----------+------------------------+
|  6rd prefix    | Ipv4 address | subnet ID |     interface ID       |
+----------------+----------------+-----------+------------------------+
|--- 6rd delegated prefix ---|
```

- 6rdDelegatedPrefix: le préfixe IPv6 alloué au client,
- 6rdDelegatedPrefixLen: longueur du préfixe alloué au client inférieur ou égal à 64 (n + o sur le schéma),
- 6rdPrefix: le préfixe 6rd retenu par l'opérateur pour un domaine 6rd
- 6rdPrefixLen: longeur du 6rdPrefix (n sur le schéma)
- IPv4MaskLen: longueur du masque de l'adresse Ipv4, c'est à dire, le nombre de bits de poids fort de l'adresse Ipv4 communs à tous les CE pour le domaine 6rd. Ces bits pourront être omis pour offrir un préfixe IPv6 délégué plus court au client et permettre de conserver m bits pour que le client puisse numéroter ses sous réseaux (*cf activité 14 de cette séquence 1*).

Il devient alors difficile de différencier un trafic sortant d'un réseau *6rd* d'un trafic IPv6 natif. Le préfixe IPv6 du domaine de l'opérateur est complété par tout ou partie de l'adresse IPv4 du routeur en "double pile" (appelé " *6rd* CE" (*Customer Edge*), le routeur du client à l'autre extrémité du tunnel dans la figure 12) du réseau IPv6 à connecter, pour former le préfixe IPv6 de ce réseau. L'adresse IPv4 du routeur " *6rd* CE" est normalement publique, mais ce n'est pas obligatoire. L'organisation de l'adresse IPv6 est décrite par la figure 13. À noter que, au sein d'un même opérateur, si les adresses IPv4 s'agrègent sur un préfixe commun (*IPv4MaskLen*), il n'est pas nécessaire d'encoder la totalité des 32 bits de l'adresse IPv4 dans le préfixe IPv6; ce qui libère des bits pour laisser une numérotation des liens internes (SID) au réseau IPv6 à connecter. Il est laissé le soin à chaque opérateur de définir le nombre de bits de l'adresse IPv4

à conserver. La seule contrainte est que le préfixe réseau ne doit pas dépasser 64 bits.

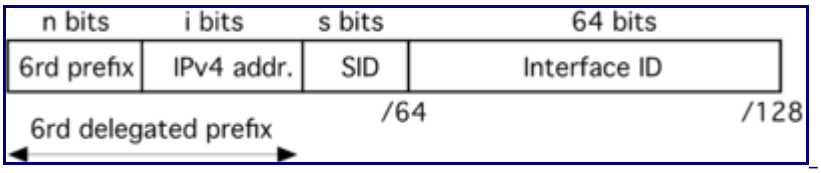


Figure 13: Format d'une adresse *6rd* .

Pour illustrer la figure 13, considérons tout d'abord que l'adresse IPv4 `192.0.2.129` (c000:2f1 en hexadécimal) a été attribuée à l'interface d'un " *6rd* CE". L'opérateur dispose du préfixe IPv6 `2001:db8::/32` pour son domaine *6rd* . Les adresses de tous les " *6rd* CE" s'agrègent sur le préfixe `192.0.0.0/8` . L'opérateur peut garder 24 bits comme partie significative. Les 24 bits de poids faible de l'adresse IPv4 suffisent, en effet, à distinguer chacun des " *6rd* CE" de son réseau. Les 8 bits du préfixe IPv4 (valeur décimale 192 dans notre exemple) peuvent être omis. Le préfixe IPv6 de chaque " *6rd* CE" aura donc une longueur de 56 bits, correspondant à l'addition du préfixe du domaine avec la partie significative de l'adresse IPv4. Dans notre exemple, le préfixe IPv6 pour ce " *6rd* CE" sera **`2001:db8:2:f100::/56`** . Il restera alors 8 bits, au titre du SID (*Subnet Identifier*), pour la numérotation des sous-réseaux internes du site connecté par le " *6rd* CE". À l'extérieur du site, ces adresses apparaîtront comme des adresses IPv6 natives, mais au travers d'un tunnel entre les routeurs de bordures de l'opérateur et le " *6rd* CE".

Le transfert avec la technique *6rd* s'organise selon 3 cas:

- Transfert inter-site. La figure 12 illustre ce cas lorsque les 2 hôtes souhaitent communiquer. La source de préfixe "pref6rd:a4" envoie un paquet IPv6 à destination de l'hôte de préfixe "pref6rd:b4". Le paquet IPv6 arrive en mode natif au " *6rd* CE" de la source. Si l'adresse IPv6 de destination est incluse dans le préfixe du domaine *6rd* configuré localement, il sera transmis directement à l'autre " *6rd* CE" comme c'est le cas ici. Les adresses IPv4 des " *6rd* CE" sont extraites des adresses IPv6 pour constituer le tunnel. Le paquet IPv4, d'adresse source "a4" et d'adresse destination "b4", encapsule le paquet IPv6. Ce paquet IPv4 est acheminé au " *6rd* CE" de destination par l'infrastructure IPv4 de l'opérateur. Le routeur " *6rd* CE" de destination reçoit le paquet IPv4. Il vérifie, par mesure de sécurité, que l'adresse source de l'en-tête IPv4 correspond à celle intégrée dans l'adresse IPv6 source. Il désencapsule le paquet IPv6 et le transmet sur le lien local pour son acheminement à la destination IPv6.
- Transfert du site vers l'Internet v6. Le trafic IPv6 est reçu en mode natif sur le " *6rd* CE". L'adresse de destination IPv6 ne correspond pas à un préfixe IPv6 du domaine de l'opérateur, ce qui signifie que la destination est extérieure au domaine de *6rd* local. Dans ce cas, le paquet IPv6 doit être transmis à un routeur de bordure *6rd* . Comme dans le cas du transfert inter-site, le paquet IPv6 est encapsulé dans un paquet IPv4. Cependant, la différence est que l'adresse IPv4 du routeur de bordure est obtenue dans la table de routage du " *6rd* CE". Le routeur de bordure reçoit le paquet IPv4 et supprime l'encapsulation IPv4. Après le contrôle de sécurité, le paquet IPv6 est transmis sur l'Internet v6.

- Transfert de l'Internet v6 vers le site. Si un routeur de bordure reçoit un paquet IPv6 à destination d'une adresse IPv4 incluse dans le préfixe *6rd* du domaine, il transmet le paquet au routeur " *6rd* CE" correspondant en utilisant le même principe que le cas précédent. Dans le cas du trafic retour, d'un flux initialisé par une machine 6rd. Comme l'adresse de destination est issue du préfixe global de l'opérateur la voie retour passera par le même relais. Ainsi la communication s'effectuera en empruntant la même route à l'aller et au retour.

La technique *6rd* est adaptée à une mise en œuvre locale d'IPv6 pour un opérateur dont l'infrastructure interne fonctionne encore en IPv4 [4] . Cette technique de tunnel répond à des questions de fiabilité et de délai dus au routage asymétrique de *6to4* . À la différence de *6to4* , *6rd* ne peut pas être déployé par un utilisateur seul. Comme le relais avec l'Internet v6 est administré par, et pour, l'opérateur lui-même, le service de connectivité est de meilleure qualité. En cas de défaillance, la responsabilité de l'opérateur est directement engagée.

Conclusion

Dans la démarche d'intégration d'IPv6, la meilleure solution est d'utiliser IPv6 nativement, comme IPv4. La complexité supplémentaire induite par les tunnels, ainsi que la réduction de la MTU qu'ils imposent (entraînant des problèmes de connectivité "épisodique") sont épargnées. Mais il n'est pas toujours possible de maintenir la connectivité IPv6 ou de trouver un opérateur offrant la connectivité IPv6. Alors, dans ces situations, il faut se résoudre à utiliser des tunnels. Le RFC 7059 effectue un inventaire des techniques d'intégration reposant sur des tunnels. Toutes les techniques ne se valent pas du point de vue des performances et de la fiabilité. Les meilleures techniques sont celles qui établissent des tunnels locaux ou de courte distance et pour lesquelles les extrémités du tunnel sont gérées et offrent un service contractuel. Le choix d'une technique de tunnel doit se faire en fonction des besoins de connectivité du réseau dans lequel IPv6 doit être intégré.

Nous avons présenté, dans cette activité, les techniques les plus intéressantes pour établir une connectivité IPv6. Le *tunnel broker* représente une méthode pour tirer un simple tunnel entre un réseau IPv6 isolé et un point d'entrée de l'Internet v6. Les techniques *6to4* et *6rd* utilisent des tunnels automatiques au sein du réseau IPv4 d'une organisation. Si le principe de tunnel automatique de *6to4* est pertinent, sa mise en oeuvre a été problématique. La dépréciation récente du préfixe anycast réservé à son usage entraîne, de fait, son déclin. La variante *6rd* , en corrigeant les défauts de *6to4* , se positionne comme une alternative.

6to4 et *6rd* reposent tous deux sur l'encapsulation directe: le paquet IPv6 est placé directement dans un paquet IPv4. Ce mode d'encapsulation ne traverse pas les NAT, car les NAT ont, pour la plupart, la capacité de traiter uniquement les protocoles de transport TCP et UDP. La technique de tunnel Teredo [RFC 4380] traite ce problème en encapsulant les paquets IPv6 dans UDP puis dans IPv4. Il a été reporté par l'article [5] des performances et une fiabilité du service de connectivité qui était pire que ce qui est rendu par *6to4* .

Pour conclure, nous rappelons la règle de connectivité d'IPv6 qui dit: *Dual stack where you can; tunnel where you must* .

Références bibliographiques

1. ↑ Cui Y., Dong J., Wu P., et al. (2012) IEEE Internet Computing. April. Tunnel-based IPv6 Transition.
2. ↑ Huston, G.(2010). The ISP Column. Flailing IPv6
3. ↑ Linux Review. Free IPv4 to IPv6 Tunnel Brokers
4. ↑ Cisco. (2011). White paper. IPv6 Rapid Deployment: Provide IPv6 Access to Customers over an IPv4-Only Network
5. ↑ Huston, G. (2011). The ISP Column. Testing Teredo

Pour aller plus loin

RFC et leur analyse par S. Bortzmeyer:

- RFC 2473 Generic Packet Tunneling in IPv6 Specification
- RFC 3053 IPv6 Tunnel Broker
- RFC 3056 Connection of IPv6 Domains via IPv4 Clouds
- RFC 3068 An Anycast Prefix for 6to4 Relay Routers
- RFC 4213 Basic IPv6 Transition Mechanisms Analyse
- RFC 4380 Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)
- RFC 5569 IPv6 Rapid Deployment on IPv4 Infrastructures (6rd) Analyse
- RFC 5572 IPv6 Tunnel Broker with the Tunnel Setup Protocol (TSP) Analyse
- RFC 5969 IPv6 Rapid Deployment on IPv4 Infrastructures (6rd) Analyse
- RFC 6180 Guidelines for Using IPv6 Transition Mechanisms during IPv6 Deployment Analyse
- RFC 6343 Advisory Guidelines for 6to4 Deployment
- RFC 6782 Wireline Incremental IPv6 Analyse
- RFC 7059 A Comparison of IPv6 over IPv4 Tunnel Mechanisms Analyse
- RFC 7381 Enterprise IPv6 Deployment Guidelines Analyse
- RFC 7526 Deprecating Anycast Prefix for 6to4 Relay Routers Analyse

Autres présentations

- Présentation de *6rd*

Activité 44: Interopérer les applications par traduction

Contexte d'utilisation de la traduction

Le besoin de traduction d'un protocole vers un autre apparaît si l'on souhaite faire communiquer deux machines ne parlant chacune qu'un seul de ces 2 protocoles, le traducteur jouant alors un rôle d'intermédiaire (ou relais) dans la communication. Ce besoin de traduction est la conséquence de l'échec du plan de migration envisagé au début et reposant sur la double pile. Les nouveaux noeuds ne peuvent plus avoir à la fois une adresse IPv4 et une adresse IPv6, du fait de l'épuisement des adresses IPv4. Cet état de fait conduit à l'apparition de noeuds avec IPv6 uniquement. Comme il y a des noeuds qui sont toujours en IPv4 uniquement car ils n'ont pas commencé à migrer, se pose le problème de la communication entre les noeuds uniquement IPv6 avec ceux uniquement IPv4. La traduction est la solution à ce problème et constitue le composant essentiel du nouveau plan de migration, qui peut se décrire de manière synthétique suivante: "tout le monde en IPv4" - "certains réseaux en IPv4 seul et certains en IPv6 seul" - "tout le monde en IPv6".

Afin de respecter les modèles d'architectures en couches (OSI, TCP/IP), la traduction n'intervient qu'entre protocoles d'un même niveau. On pourra donc distinguer la traduction de niveau applicatif, de niveau transport, et de niveau réseau. Dans le cas du protocole IP (niveau réseau), il s'agit bien sûr de faire communiquer 2 machines, chacune n'utilisant qu'une version du protocole, IPv4 ou IPv6. Dans le cadre d'une communication "client vers serveur", il y aura donc 2 cas:

1. le client ne parle qu'IPv6 et le serveur ne parle qu'IPv4,
2. le client ne parle qu'IPv4 et le serveur ne parle qu'IPv6.

Aujourd'hui, le cas le plus fréquent est le premier; les serveurs gardant majoritairement une connectivité IPv4. Il s'agit donc de mettre en place un dispositif pour offrir une connectivité IPv4 aux clients IPv6. Ainsi, ils pourront accéder à des serveurs qui n'ont toujours pas IPv6. Un moyen, pour offrir cette connectivité, est de traduire automatiquement les paquets IPv6 du client en IPv4 pour les envoyer au serveur, et de faire la traduction inverse au retour. Un tel dispositif devra naturellement se situer en coupure des communications entre le client et le serveur, afin d'en intercepter les paquets pour les traduire, et les réémettre sur le réseau du destinataire. Ce dispositif est comparable au traditionnel NAT (*Network Address Translator*) utilisé entre les réseaux IPv4 privés et publics. Mais, dans notre cas, ce dispositif n'effectue pas une simple **translation** d'un espace d'adressage à un autre, mais une véritable **traduction** de l'en-tête IP. Le traducteur assurant le relais entre un réseau IPv6 (coté client) et un réseau IPv4 (coté serveur) est appelé NAT64. La figure 1 représente la topologie d'utilisation du NAT64. Les spécifications pour cette traduction ont été publiées par le groupe de travail Behave [1] de l'IETF qui avait déjà publié des travaux pour le NAT44.

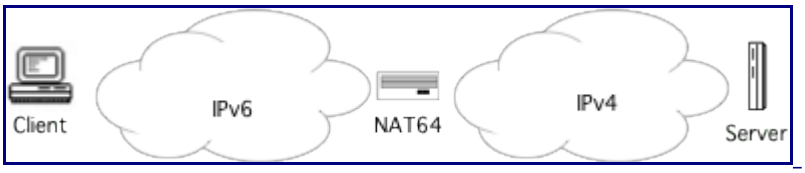


Figure 1: Topologie d'utilisation du NAT64.

Le RFC 6144 détaille les cas d'utilisation de la traduction entre IPv6 et IPv4 en distinguant l'Internet et un réseau. Ainsi, un réseau dont le plan d'adressage est administrable est distingué de celui qui ne l'est pas. Le RFC indique notamment que le cas du client IPv4 accédant à un serveur de l'Internet IPv6 n'est pas d'actualité et d'autres solutions que la traduction IP seront à envisager. Les cas d'utilisation communs de la traduction sont: soit un client d'un réseau IPv6 accédant à un serveur de l'Internet v4, soit des clients de l'Internet v6 accédant à un serveur d'un réseau IPv4. Dans le premier cas, le traducteur est du coté du client IPv6 pour le rendre capable d'accéder à des contenus disponibles uniquement sur l'Internet IPv4. Dans le RFC 7269 , ce type de NAT64 est appelé NAT64-CGN (*Carrier-Grade NAT*). Dans le second cas, le traducteur est du coté du serveur IPv4 pour rendre le service accessible aux clients de l'Internet IPv6. Le RFC 7269 qualifie ce NAT64 de NAT64-FE (*Front End*) dans la mesure où le NAT64 est devant les serveurs au sein d'un *data center* . Quelque soit le cas, la traduction reste une solution temporaire et vise à faciliter le déploiement d'IPv6 dans l'Internet v4.

Un contexte, pour lequel ce type de solution est pertinent, est celui des réseaux mobiles 3GPP *3rd Generation Partnership Project*) [2] . En effet, dans la norme 3GPP, les sessions PDP (*Packet Data Protocol*) mises en place pour la transmission de données ne peuvent être "double pile" que depuis la *Release-9* . Pour avoir un support "double pile" sur ces réseaux, il est nécessaire d'ouvrir 2 contextes, ce qui peut être préjudiciable pour le dimensionnement des équipements. Une solution est alors de ne déployer qu'une version du protocole sur le réseau mobile. Les équipements mobiles seront donc connectés à un réseau IPv6 et la compatibilité avec les services IPv4 sera assurée par la traduction d'en-tête IP.

Principe de la traduction entre protocoles IP

La traduction entre protocoles IP comporte essentiellement 2 composants [3] : une transposition protocolaire et une traduction des adresses. Le premier composant transpose les champs de l'en-tête IP (à l'exception des adresses) en conservant la sémantique du champ original. Le second composant met en correspondance les adresses "source" et "destination" du paquet reçu dans une version du protocole IP, dans leur équivalent dans l'autre version du protocole IP.

Les traductions peuvent être faites sans état (*stateless*) RFC 7915 ou bien avec état (*stateful*) RFC 6146 . Dans le premier cas, le traducteur n'a aucune mémoire. Chaque paquet est traité isolément et contient toutes les informations nécessaires à la traduction. Avec la traduction sans état, les meilleures performances sont obtenues en terme de quantité de paquets traités et de passage à l'échelle. Dans le second cas, celui de la traduction avec état, le traducteur se souvient de la correspondance qu'il a effectué entre les 2 versions du protocole, par exemple parce que l'adresse IPv6 n'est pas en correspondance univoque (1:1) avec l'adresse IPv4. Nécessitant une table des correspondances en mémoire, la traduction avec état passe moins à

l'échelle. Mais, dans certains cas, elle est la seule réaliste, puisqu'on ne peut pas stocker toutes les informations dans une seule adresse, surtout si elle est IPv4. Si le composant de la transposition des champs de l'en-tête s'effectue sans état, le composant de traduction des adresses fonctionne avec ou sans état.

Transposition protocolaire des champs de l'en-tête (RFC 7915)

Il faut ici bien situer le problème: le traducteur qui reçoit un paquet avec un en-tête IPvX doit créer un nouvelle en-tête IPvY à partir des informations à sa disposition: les données de l'en-tête IPvX et des données de configuration.

Si l'on observe les en-têtes IPv4 et IPv6, comme dans l'activité 21, on remarque qu'il y a un certain nombre de champs qui ont une sémantique très proche (TTL/Hop limit, DiffServ, Payload Length). Pour ces derniers, la transposition est évidente. Les tableaux 1 et 2 résument les informations qu'il faut utiliser pour renseigner les différents champs des en-têtes IPv4 ou IPv6 que doit créer le traducteur (Voir RFC 7915 section 4)

Champ de l'en-tête IPv4	Champ dans le nouvel en-tête IPv6	Valeur
Version	Version	6
IHL		Ignorer
Type Of Service	Traffic Class	Recopier
	Flow label	0
Packet Length	Payload Length	Packet Length - IHL (en-tête IPv4 + options) + 8 (si extension de fragmentation)
Ident./Flag/Offset	Extension Fragmentation	Créer une extension de fragmentation à partir des valeurs IPv4
TTL	Hop Limit	Décrémenter de 1
Protocol	Next Header	Recopier ou extension de fragmentation si besoin. ICMPv4 (1) devient ICMPv6 (58).
Checksum		Ignorer
Source Address	Source Address	Voir le paragraphe *Traduction des adresses*
Destination Address	Destination Address	Voir le paragraphe *Traduction des adresses*
Options IPv4		Les options IPv4 ne sont pas traduites.

Tableau 1: Création d'un en-tête IPv6 à partir d'un en-tête IPv4

Champ de l'en-tête IPv6	Champ dans le nouvel en-tête IPv4	Valeur
Version	Version	4
	IHL	5
Traffic Class	Type of Service	Recopie
IPv6 Flowlabel		Ignorer
Payload Length	Packet Length	Payload Length + IHL
	Ident./Flag/Offset	0
Hop Limit	TTL	Décrémenter de 1
Next Header	Protocol	Recopier. ICMPv6 (58) devient ICMPv4 (1)
	Checksum	Calculer une fois l'en-tête créé
Source Address	Source Address	Voir le paragraphe *Traduction des adresses*
Destination Address	Destination Address	Voir le paragraphe *Traduction des adresses*
Extensions IPv6		Les extensions d'en-tête IPv6 ne sont pas traduites.

Tableau 2: Création d'un en-tête IPv4 à partir d'un en-tête IPv6

Les adresses pour les traducteurs d'adresse NAT64, NAT46 (RFC 6052)

Le RFC 6052 décrit les différents formats d'adresse mis en oeuvre par les traducteurs IPv6 ↔ Ipv4. (NAT46 et NAT64 avec ou sans état).

tolérance de notation (rappel)

Lorsque l'adresse IPv4 occupe la partie basse de l'adresse IPv6, les 32 bits de poids faible (bits 97 à 128), la notation décimale pointée traditionnelle d'IPv4 est tolérée. Ainsi l'adresse `2001:db8:900d:cafe::` **`c0a8:a05`** peut être notée `2001:db8:900d:cafe::` **`192.168.10.5`** lors d'une saisie (configuration manuelle d'interface ou passage de paramètre en ligne de commande, ...). Cependant elle sera affichée sous sa forme canonique (RFC 5952) `2001:db8:900d:cafe::c0a8:a05` dans le journal de bord (log système) de la machine. Dans ce cas si la saisie peut nous sembler familière, la correspondance entre l'adresse IPv6 et l'adresse IPv4 embarquée est moins évidente à l'affichage.

Le RFC définit un préfixe réservé (*well-known prefix*) **64:ff9b::/96** ainsi que les règles pour embarquer une adresse IPv4 dans des préfixes IPv6 de 32, 40, 48, 56, 64 ou 96 bits.

```
+--+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|PL| 0-------------32--40--48--56--64--72--80--88--96--104---------|
+--+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|32|     prefix    |v4(32)         | u | suffix                    |
+--+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|40|     prefix        |v4(24)     | u |(8)| suffix                |
+--+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|48|     prefix            |v4(16) | u | (16)  | suffix            |
+--+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|56|     prefix                |(8)| u |  v4(24)   | suffix        |
+--+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|64|     prefix                    | u |   v4(32)      | suffix    |
+--+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|96|     prefix                                    |    v4(32)     |
+--+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

Les 8 bits aux positions 64 à 71 sont réservés et doivent être nuls. Cela entraîne que pour les préfixes de longueur 40, 48 et 56 l'adresse IPv4 est scindée en 2 parties.

Note: *Le préfixe réservé* ***64:ff9b::/96*** *est neutre vis à vis du calcul du checksum intégrant le pseudo entête (cf sequence 3)* .

Traduction des adresses

La traduction d'adresse d'un protocole à un autre suit le même principe que celui appliqué dans les passerelles NAT traduisant des adresses IPv4 privées vers des adresses IPv4 publiques (appelé aussi NAT44). Le traducteur reçoit un paquet avec des adresses "source" et "destination" chacune dans un des espaces d'adressage, et doit traduire ces adresses dans l'autre espace d'adressage pour pouvoir réémettre le paquet. Le traducteur doit donc mettre en correspondance une adresse de l'espace d'adressage IPv6 avec une adresse de l'espace d'adressage IPv4 et vice-et-versa à la fois pour l'adresse "source" et l'adresse "destination". Afin de faire cette correspondance, le NAT64 dispose d'un ensemble d'adresses IPv6 et d'un ensemble d'adresses IPv4, comme le montre la figure 2. L'ensemble d'adresses IPv6 du NAT64 (notée N6) va servir à représenter les adresses IPv4 (notée H4) dans le réseau IPv6. Et, de manière similaire, l'ensemble des adresses IPv4 du NAT64 (notée N4) va servir à représenter les adresses IPv6 (notée H6) dans le réseau IPv4.

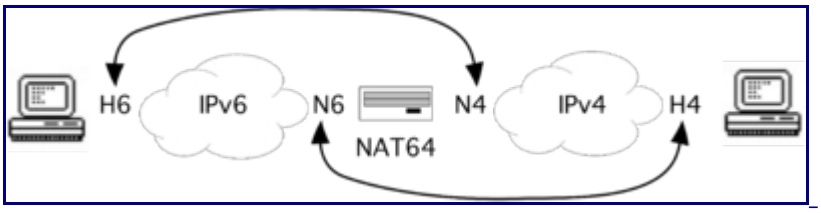


Figure 2: Les adresses utilisées pour la traduction.

La correspondance entre une adresse IPv4 avec une adresse IPv6 est évident lorsque l'adresse IPv6 comporte l'adresse IPv4. En effet, représenter une adresse IPv4 dans l'espace d'adressage IPv6 est simple car ce dernier est assez large pour contenir l'ensemble des adresses IPv4. Il est donc toujours possible de trouver une adresse IPv6 à faire correspondre

avec une adresse IPv4. Le RFC 6052 décrit la méthode pour créer une adresse IPv6 à partir d'une adresse IPv4. La méthode consiste à inclure les 32 bits l'adresse IPv4 à la suite d'un préfixe IPv6. Selon la longueur du préfixe IPv6, le mécanisme d'inclusion de l'adresse IPv4 est différent, comme précisé dans le RFC 6052 Section 2.2. Une adresse IPv6 embarquant une adresse IPv4 (*IPv4-embedded IPv6 address*) est qualifiée, soit de "traduisible en IPv4" (*IPv4-translatable IPv6 address*) si elle est unique globalement, routable et donc attribuée à un noeud IPv6, soit de "IPv4 convertible" (*IPv4-converted IPv6 address*) si elle ne fait que représenter un noeud IPv4 dans l'espace d'adressage IPv6. Selon le cas d'utilisation du NAT64, le préfixe d'une adresse IPv6 embarquant une adresse IPv4 (notée *pref64* dans la représentation ci-dessous) peut être le préfixe dit *Well-Known Prefix* (WKP) ou un préfixe pris dans le plan d'adressage de l'organisation déployant le NAT64 dit *"Network-Specific Prefix* (NSP). Le WKP se définit par `64:ff9b::/96` et sert uniquement à constituer des adresses IPv6 embarquant une adresse IPv4 convertible. Ce préfixe n'est pas routable sur l'Internet v6. Il doit être utilisé uniquement en routage interne à un réseau.

La traduction d'adresses utilisant une adresse IPv6 embarquant une adresse IPv4 est qualifiée de **sans état** . Le point essentiel dans ce mode de traduction est que le noeud IPv6 est identifié dans l'adressage IPv4 par une adresse IPv4. Il y a une correspondance de 1:1 entre l'adresse IPv6 et IPv4. Ainsi, les adresses peuvent être traduites indépendamment et de manière transparente pour l'utilisateur. La traduction peut se représenter de la manière suivante:

```
IPv6  ------- IPv4
pref64:H4        H4
```

où *pref64* représente un préfixe IPv6 pour constituer une adresse IPv6 embarquant une adresse IPv4 (notée ici H4). L'adresse IPv6 ainsi constituée est notée *pref64:H4* . Le préfixe IPv6 utilisé sera un préfixe routé vers le traducteur, afin que celui-ci assure son rôle de relais. L'adresse IPv6 ainsi créée permet d'identifier un noeud IPv4 dans l'espace d'adressage IPv6.

Lorsque l'adresse IPv6 n'embarque pas l'adresse IPv4 et que l'adresse IPv4 ne peut contenir une adresse IPv6, alors mettre en correspondance une adresse IPv6 avec une adresse IPv4 demande une traduction d'adresse **avec état** . La mise en correspondance est faite dynamiquement par le traducteur. Celui-ci utilise une adresse IPv4 libre, sélectionnée dans un ensemble (*pool*) d'adresses délégué au traducteur. Comme il peut ne pas y avoir assez d'adresses IPv4 pour les noeuds IPv6 (l'ensemble d'adresses IPv4 délégué au traducteur peut être moins fourni que le nombre de noeuds IPv6 pour lequel il assure la traduction), le traducteur peut être amené à utiliser le numéro de port de la couche de transport pour reconnaître les noeuds IPv6. La combinaison d'une adresse IP et d'un port est appelée adresse de transport. Le traducteur doit alors retenir cette association d'adresses (ou d'adresse de transport) entre IPv4 et IPv6 dans un état. Par exemple, dans le cas d'un traducteur entre un client IPv6 du réseau local et un serveur de l'Internet v4, le traducteur ne sait pas comment traduire l'adresse source du paquet IPv6: il doit utiliser une de ses propres adresses IPv4 pour définir une adresse de transport en IPv4. Le paquet "retour" contient alors cette adresse de transport comme destination. Le traducteur a bien besoin ici d'un état: la correspondance choisie pour le paquet "aller" entre l'adresse de transport "source" IPv6 et l'adresse de transport

"source" IPv4. La traduction est alors dite "à état" car elle fait intervenir cette information. La traduction peut se représenter de la manière suivante, avec H6 qui représente l'adresse IPv6, et H4, l'adresse IPv4:

```
IPv6  -------------- IPv4
  H6  (état  H6-H4)  H4
IPv6  -------------  IPv4
  H6  (état  H6-H4)  H4
```

La traduction **avec état** est similaire à celle que l'on trouve avec le NAT44. L'adresse de transport constituée par une adresse IPv6 et le numéro de port est convertie en une autre adresse de transport dans le réseau IPv4. On retiendra que dans ce mode de traduction, plusieurs noeuds IPv6 peuvent partager une adresse IPv4. Il y a alors une correspondance de N:1 entre l'adresse IPv6 et IPv4.

Mécanismes complémentaires

Traduction des paquets ICMP

Comme décrit dans l'activité 31, les messages ICMP servent au contrôle de la connectivité de bout en bout, ainsi qu'aux rapports d'erreurs d'acheminement des paquets. La présence d'un traducteur sur ce chemin ne doit pas perturber ce mécanisme, sous peine de grandement complexifier son fonctionnement. Celui-ci doit donc s'efforcer de traduire les messages ICMPv4 en messages ICMPv6, et inversement, pour être ainsi transparent dans ces échanges.

Le traducteur recevant un message ICMPv4 (resp. ICMPv6) doit donc interpréter le contenu de ce message pour créer un message ICMPv6 (resp. ICMPv4) à retransmettre. L'en-tête IP est traduit selon les mécanismes présentés plus haut. L'en-tête ICMPv4 (resp. ICMPv6) doit donc être transformé par le traducteur en en-tête ICMPv6 (resp. ICMPv4). Cette traduction est facilitée par le fait que les sémantiques des messages de ces 2 protocoles ne sont pas très éloignées: les fonctions supplémentaires de découverte de voisins intégrées dans ICMPv6 ne sont valides que sur le lien et ne seront pas traduites. De plus, les paquets ICMP n'ont pas besoin d'informations contextuelles pour être interprétés. La traduction des messages ICMP est dite **sans état** . Le RFC 7915 définit le mécanisme pour effectuer cette traduction.

Le champ ICMP `type` devra être ajusté dans certains cas lors de la traduction car les valeurs pour la même sémantique de messages peuvent être différentes entre les versions du protocole. Par exemple, les messages *Echo Request* et *Reply* sont identifiés par la valeur du champ ICMP `type` : 8 et 0 en ICMPv4, 128 et 129 en ICMPv6. Certains messages ICMPv4 ne seront pas traduits car leur sémantique (obsolète) n'a pas été transposée dans ICMPv6.

La traduction de l'en-tête ICMP modifie les en-têtes des niveaux réseau et transport. Elle impacte donc la somme de contrôle calculée pour ces en-têtes. Le champ `checksum` doit donc être recalculé suite à la traduction.

Relais-traducteur DNS auxiliaire (**RFC 6147**)

Les clients IPv6 ne pouvant pas initier une communication avec des serveurs n'ayant qu'une

adresse IPv4, il est nécessaire de les «leurrer» en fabriquant dynamiquement des adresses IPv6. Cette fabrication d'une adresse IPv6 pour le serveur IPv4 revient au relais DNS auxiliaire (*DNS Application Layer Gateway* : DNS-ALG). Celui-ci convertit l'adresse IPv4 obtenue par la résolution d'adresse en une adresse IPv6 imbriquant une adresse IPv4. En quelque sorte, le relais DNS auxiliaire ment en répondant au client par un enregistrement de type AAAA (adresse IPv6) à partir de l'enregistrement réel A (adresse IPv4) du serveur. Du point de vue du client, le relais DNS auxiliaire se comporte comme n'importe quel serveur DNS de rattachement. Il accepte les requêtes et les transfère au serveur DNS de rattachement, s'il ne dispose pas déjà de l'information dans son cache local. Mais ce DNS ment car il est capable de répondre positivement à la demande d'une ressource inexistante. Un relais DNS effectuant la résolution en IPv6 de nom de domaine enregistré uniquement en IPv4 est appelé **DNS64** .

La figure 3 montre un chronogramme des opérations de résolution d'adresse avec un DNS64. Le préfixe IPv6 utilisé dans cet exemple pour construire une adresse IPv6 "IPv4-convertible" est le WKP de longueur 96 bits (`64:ff9b::/96`). L'usage d'un préfixe spécifique de type NSP fonctionne selon le même principe. Les opérations sont les suivantes:

1. Lorsqu'un client IPv6 formule une requête de type AAAA pour résoudre le nom d'un serveur, le DNS64 la transfère au serveur DNS en charge du nom de domaine du serveur.
2. Si la réponse est vide, le DNS64 renvoie une requête de type A pour le même nom de serveur au serveur DNS.
3. Le DNS64 reçoit une réponse à sa requête de type A.
4. Le DNS64 applique alors la traduction de l'adresse IPv4 obtenue en adresse IPv6, comme spécifié dans le RFC 6052 . Il combine le préfixe IPv6 aux 32 bits de chacune des adresses obtenues comme résultats. L'adresse IPv6 obtenue sera transmise au client en réponse à sa requête AAAA.

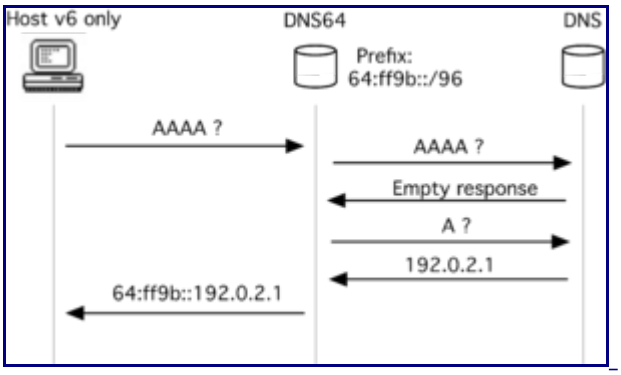


Figure 3: Opérations du DNS64.

Les versions récentes du logiciel serveur DNS BIND/Named peuvent assurer le rôle de DNS64. Le logiciel *Trick or Treat Deamon* (TOTD) peut également être utilisé pour cet usage.

Mécanisme de transition NAT64/DNS64

NAT64 et DNS64 constituent ensemble une technique de traduction de niveau réseau. Le fonctionnement du NAT64 fonctionne **sans état** ou **avec état** en fonction du mode de traduction

de l'adresse "source" et de l'adresse "destination" du paquet reçu par le traducteur [4] .

NAT64: traduction "sans état" RFC 7915

Le NAT64 "sans état" signifie que les adresses IPv6 du paquet sont traduites **chacune** "sans état", à l'aide de l'algorithme de correspondance défini dans le RFC 6052 . Comme indiqué précédemment, le point essentiel dans ce mode de traduction est que l'adresse IPv4 est comprise dans l'adresse IPv6. Aussi, un préfixe IPv6 spécifique est dédié pour représenter les systèmes IPv4 dans le monde IPv6. Dans le monde IPv4, tous les systèmes IPv6 ont une adresse IPv4. Ainsi, quel que soit le sens de la traduction, la correspondance d'adresse est unique: d'un coté il faut l'extraire de l'adresse IPv6, de l'autre coté il faut combiner l'adresse IPv4 avec le préfixe pour former une adresse IPv6. C'est grâce à cette correspondance directe qu'il n'est pas nécessaire de maintenir un état pour la traduction entre IPv6 et IPv4. Cependant, cela requiert que les systèmes IPv6 devant communiquer avec le monde IPv4 soient configurés, manuellement ou via DHCPv6, avec les adresses IPv6 embarquant une adresse IPv4 [RFC 6052]. On voit là apparaître la principale faiblesse de ce mode de traduction "sans état": il consomme une adresse IPv4, car les noeuds IPv6 ont besoin d'une adresse IPv4 qui leur soit propre (de manière similaire aux noeuds en double pile). La figure 4 représente le transfert d'un paquet du noeud IPv6 vers le noeud IPv4. Dans cette figure, H6 et H4 sont des adresses IPv4. Ces adresses trouvent leur correspondance dans l'espace d'adressage IPv6 en les préfixant par un préfixe IPv6 réservé à cet usage, noté "pref64". Du point du vue du routage, NAT64 annonce ce préfixe dans le réseau IPv6 pour recevoir le trafic à destination des noeuds IPv4. Il fait de même du coté IPv4 en annonçant une route pour les adresses IPv4 des noeuds IPv6.

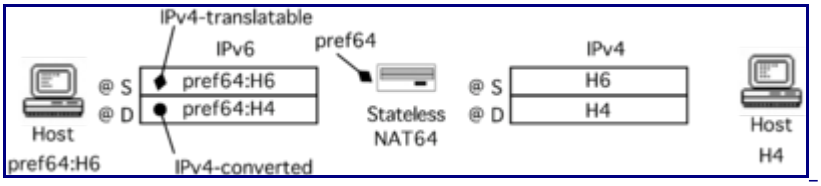


Figure 4: Type des adresses utilisées pour un NAT64 "sans état".

Du fait de son caractère "sans état", ce traducteur passe mieux à l'échelle et il n'introduit pas un point de faiblesse pour les communications en respectant l'indépendance du réseau vis-à-vis des hôtes. Lorsque le réseau est indépendant des hôtes, une panne dans le réseau n'entraîne pas la réinitialisation des communications en cours. C'est un principe pour assurer la robustesse du système. Dans notre cas, la robustesse de la traduction dans le réseau peut être elle-même renforcée si plusieurs NAT64 sont déployés en parallèle. Cependant, le manque d'adresses IPv4 disponibles le rend difficilement utilisable, voire inutile [5] . Comme il va être nécessaire d'agréger plusieurs noeuds IPv6 sur une simple adresse IPv4, la solution s'oriente alors vers le traducteur "avec état".

NAT64: traduction "avec état" RFC 6146

Décrit par le RFC 6146 , le NAT64 "avec état" possède une adresse IPv4 qu'il partage entre plusieurs systèmes IPv6. Il s'ensuit que l'algorithme de correspondance des adresses reposant sur une adresse IPv6 embarquant une adresse IPv4 défini dans le RFC 6052 n'est plus

applicable. À la place, un état est créé pour chaque flot de paquets pour mettre en correspondance cette adresse IPv4 avec des adresses IPv6. Comme pour le NAT44, le numéro de port est utilisé pour identifier les noeuds IPv6. La différence majeure avec le traducteur "sans état" porte sur une des adresses du paquet IPv6. Celle-ci n'est pas traduite en IPv4 par la méthode de traduction "sans état". Comme le décrit la figure 5, le NAT64 "avec état" utilise à la fois une traduction "avec état" et une traduction "sans état". Sur cette figure, l'hôte IPv6 d'adresse H6 émet un paquet à destination de l'hôte IPv4 d'adresse H4. N4 représente l'adresse IPv4 partagée que le traducteur utilise pour la représentation des adresses "source" IPv6 dans le monde IPv4. Le NAT64 annonce une route de préfixe pref64 pour recevoir le trafic IPv6 a destination du réseau IPv4.

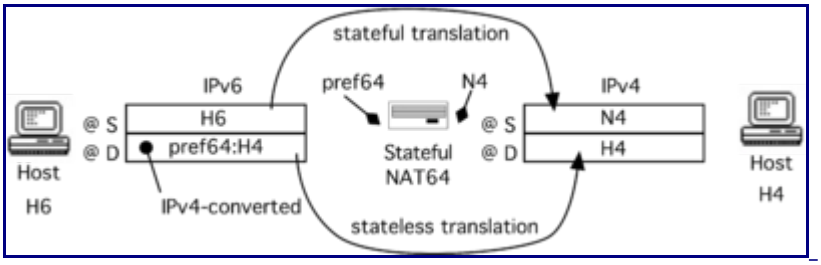


Figure 5: Type des adresses utilisées pour un NAT64 "avec état".

Pour illustrer le fonctionnement conjoint du NAT64 et du DNS64, nous allons prendre l'exemple du déploiement d'un NAT64 "à état" sur le réseau mobile. Comme décrit au début de l'activité, le déploiement d'un réseau "seulement IPv6" peut s'avérer intéressant dans le cadre d'un réseau mobile type UMTS (3G). L'interopérabilité avec les services IPv4 peut alors être réalisée en traduisant les paquets IPv6 en paquets IPv4 à travers un dispositif NAT64, couplé à un relais-traducteur DNS64. L'intérêt d'un tel dispositif est qu'il est relativement simple à configurer côté équipement client: il suffit que celui-ci utilise l'adresse du DNS64 en tant que serveur de résolution de nom. La figure 6 montre la structure du réseau du point de vue IP. Le client est un mobile, souvent un smartphone, noté ME (*Mobile Equipment*) connecté à un réseau sans fil interconnecté avec l'infrastructure IP au moyen d'un routeur noté GGSN (*Gateway GPRS Support Node*).

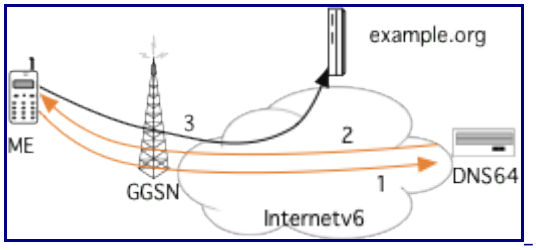


Figure 6: Accès à un serveur en IPv6.

Dans la figure 6, le client ME cherche ici à joindre le service hébergé sur le serveur IPv6 "example.org".

1. Pour en connaître l'adresse IP, il interroge le serveur de résolution de noms, en l'occurrence le dispositif DNS64. L'interrogation du client concerne les enregistrements IPv6 (AAAA) car ceux-ci sont les seuls qui seront utilisables depuis le client connecté sur un réseau IPv6 seul (étape 1).

2. Ce nom de domaine possède une résolution en IPv6 (il possède un enregistrement AAAA). Le dispositif DNS64 se comporte alors comme un "résolveur" de noms normal et transfère cet enregistrement au client en guise de réponse (étape 2).
3. Le client peut alors se connecter directement au service à partir de l'adresse IPv6 obtenue (étape 3).

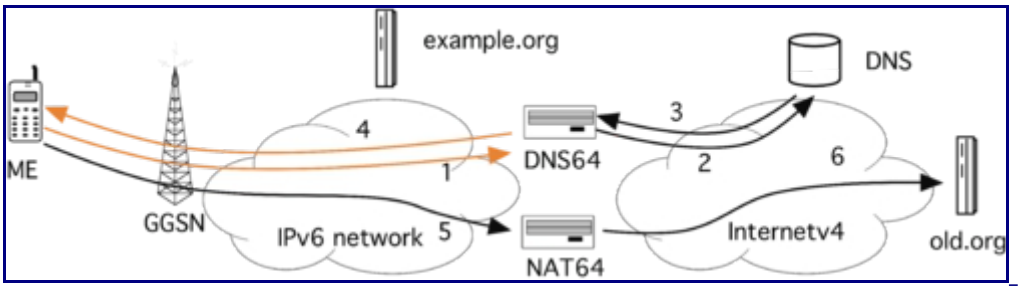


Figure 7: Accès à un serveur en IPv4.

Dans la figure 7, le client ME cherche maintenant à joindre un autre service, comme "old.org". Or, ce service ne possède pas de connectivité IPv6.

1. Comme précédemment, le client va interroger son "résolveur" de noms, le DNS64, sur la présence d'un enregistrement AAAA pour le service (étape 1).
2. Le DNS64 interroge le service DNS (étape 2) sur les différentes adresses disponibles.
3. Le DNS64 n'obtient que des adresses de type IPv4 (enregistrement A) (étape 3).
4. Ces enregistrements ne correspondent pas aux adresses attendues par le client. Le DNS64 va alors transformer les adresses IPv4 obtenues du service, en adresses IPv6 afin de satisfaire la demande du client. Cette traduction d'adresse se fait conformément au RFC 6052 . Dans notre exemple, le DNS64 complète le préfixe `64:ff9b::/96` avec l'adresse IPv4 obtenue (étape 4).
5. Le client utilise donc cette adresse IPv6 comme destinataire de la communication. Ici, le navigateur du client ouvre une connexion TCP à destination d'une adresse appartenant au préfixe `64:ff9b::/96` . Ce préfixe est routé dans l'infrastructure du réseau mobile vers le dispositif NAT64. Celui-ci reçoit donc les paquets en provenance du client et à destination de l'adresse transformée par le DNS64 (étape 5).
6. Le NAT64 doit maintenant traduire ces paquets IPv6 vers IPv4. Il crée donc un en-tête IPv4 à partir des champs de l'en-tête IPv6, comme spécifié dans le RFC 7915 . Pour l'adresse destination du paquet IPv4, le traducteur applique la transformation inverse de celle appliquée par le DNS64: il extrait l'adresse IPv4 en soustrayant de l'adresse destination du paquet IPv6 le préfixe utilisé pour la traduction d'adresse dans l'infrastructure mobile, en l'occurrence `64:ff9b::/96` . Ne pouvant représenter l'adresse IPv6 du client dans une adresse IPv4, le traducteur choisit, comme adresse "source", une adresse IPv4 de son jeu d'adresses (*pool* d'adresses) réservées à cet usage. Au sein de l'Internet v4, le routage sur cette adresse IPv4 conduira vers le traducteur afin qu'il puisse traduire les paquets "retour" à destination du client. Comme l'adresse IPv4 est partagée entre les clients IPv6, le traducteur va aussi utiliser le numéro de port "source" du coté du réseau IPv4 pour identifier la source sur le noeud IPv6. Afin de pouvoir effectuer la traduction inverse, ainsi que la traduction des paquets suivants de

ce flux, le traducteur conserve, comme informations contextuelles de la connexion, une trace, en mémoire, de cette correspondance entre l'adresse de transport IPv6 du client et l'adresse de transport IPv4 choisie (l'état de la connexion). Après avoir fait ces traitements, le NAT64 transmet des paquets IPv4 vers le service "old.org" fonctionnant encore avec le protocole archaïque (étape 6).

Selon les cas d'utilisation indiqués par le RFC 6144 , les détails de la configuration d'un réseau comportant un traducteur NAT64 sont décrits dans cet article [6] .

Conclusion

Le déploiement de réseaux seulement en IPv6 apporte la réponse au manque d'adresses IPv4 mais pose le problème de l'accès aux services restés en IPv4. La traduction de paquets comme opérée par NAT64 offre une alternative pour les applications qui sont indépendantes du format d'adresse IP au niveau de leur protocole applicatif (si celui-ci ne transporte pas d'adresses IP). Sous cette condition, le dispositif de traduction NAT64 s'utilise de façon quasi transparente. Aucune modification du client ou du serveur n'est requise. Tout est fait dans le traducteur. Cependant, ce dispositif souffre de certains inconvénients du NAT44, comme une faible capacité à passer à l'échelle pour les traducteurs "à état", ou du partage des adresses IPv4 [RFC 6269]. Il faut de plus noter, dans le cas d'un client IPv6, que les applications et les protocoles utilisés par ce client devront être compatibles avec IPv6. Lorsque cette compatibilité n'existe pas, le client ne pourra pas alors profiter de l'interopérabilité rendue possible par le NAT64. Il demandera d'autres solutions de transition reposant sur une adresse IPv4, telle que la double traduction 464xlat [RFC 6877].

Il peut paraitre contradictoire d'utiliser IPv6 pour se passer de la traduction ou de la double traduction d'IPv4 pour, en fait, retrouver des traducteurs dans les communications. Tout d'abord, il faut noter que cette solution se veut transitoire. Dans l'article [7] , les auteurs avancent que NAT64 doit se voir comme une évolution du NAT44 servant à éviter l'utilisation d'un étage de traduction (NAT444). De plus, le nombre de services accessibles uniquement par IPv4 va diminuer au fur et à mesure qu'IPv6 va se diffuser dans l'Internet. Cette évolution dans le temps va entraîner une diminution du trafic IPv4 au profit du trafic IPv6. Au contraire de se qui se passe aujourd'hui dans l'Internet avec IPv4, les dispositifs de traduction vont être de moins en moins sollicités.

Bien que NAT 64 ne soit pas une solution universelle [RFC 7269], il se développe de plus en plus car il devient intéressant aujourd'hui de pouvoir déployer des réseaux seulement IPv6 à la place de réseaux IPv4 privés, notamment quand l'espace d'adressage privé n'est plus suffisant pour adresser l'ensemble des noeuds. Certains opérateurs mobiles ont notamment fait ce choix pour leur réseau (comme T-Mobile aux USA). De plus, ce mécanisme constitue le composant essentiel pour la migration vers IPv6 dans la situation actuelle de l'Internet (épuisement effectif des adresses IPv4 disponibles et forte inertie pour la migration des noeuds IPv4). Les solutions de traduction comme NAT64 trouvent donc leur intérêt pour que des noeuds IPv6 accèdent aux contenus disponibles sur IPv4.

Références bibliographiques

1. ↑ Bortzmeyer, S. (2008). Le groupe de travail BEHAVE de l'IETF
2. ↑ 3GPP *3rd Generation Partnership Project* 3GPP
3. ↑ Bagnulo, M.; Garcia-Martinez, A. and Van Beijnum, I. (2012). IEEE Communications Magazine, Vol. 50, No. 7, July. The NAT64/DNS64 tool suite for IPv6 transition
4. ↑ Cisco. (2011). White paper. NAT64—Stateless versus Stateful
5. ↑ Pepelnjak, I. (2011). Blog IP space. Stateless NAT64 is useless
6. ↑ Cisco. (2012). White paper. NAT64 Technology: Connecting IPv6 and IPv4 Networks
7. ↑ Boucadair, M.; Binet, D. et Jacquenet, C. (2011). Techniques de l'ingénieur. Transition IPv6 - Outils et stratégies de migration

Pour aller plus loin

RFC et leur analyse par S. Bortzmeyer

- RFC 6052 IPv6 Addressing of IPv4/IPv6 Translators Analyse
- RFC 6144 Framework for IPv4/IPv6 Translation Analyse
- RFC 6146 Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers Analyse
- RFC 6147 DNS64: DNS extensions for Network Address Translation from IPv6 Clients to IPv4 Servers Analyse
- RFC 6269 Issues with IP Address Sharing Analyse
- RFC 6333 Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion Analyse
- RFC 6877 464XLAT: Combination of Stateful and Stateless Translation
- RFC 7051 Analysis of Solution Proposals for Hosts to Learn NAT64 Prefix
- RFC 7050 Discovery of the IPv6 Prefix Used for IPv6 Address Synthesis Analyse
- RFC 7269 NAT64 Deployment Options and Experience Analyse
- RFC 7757 Explicit Address Mappings for Stateless IP/ICMP Translation
- RFC 7915 IP/ICMP Translation Algorithm Analyse

Activité 45: Interopérer des applications par passerelles applicatives

Contexte d'utilisation des passerelles applicatives

Il n'existe pas une solution magique à tous les problèmes. Le déploiement bien trop lent d'IPv6 a laissé une situation peu satisfaisante face au manque d'adresses IPv4. La migration vers IPv6 ne pourra pas se faire sans la traduction. Comme nous l'avons vu, la traduction au niveau réseau à l'aide de NAT64 est un dispositif qui vise à faciliter le déploiement des clients IPv6, tout en étant aussi utilisable pour rendre les serveurs IPv4 accessibles à l'Internet v6. Si NAT64 est une solution fonctionnelle pour la communication avec des systèmes IPv4, le retour d'expérience rapporté par les RFC 6586 et RFC 7269 montre que certaines applications ne fonctionnent plus lorsque leurs communications passent par un NAT64. C'est par exemple le cas de la signalisation de la téléphonie: les adresses IP sont transmises dans la signalisation, et ne sont pas traduites par NAT64. Lorsque l'utilisation de NAT64 conduit à une situation d'échec, le recours à une passerelle applicative constitue une alternative pour les applications dont l'installation d'un relais intermédiaire est possible.

Outre la résolution de certains défauts de fonctionnement, la solution de la passerelle applicative offre une technique d'interopérabilité moins intrusive que NAT64 au niveau de l'infrastructure de communication. En effet, déployer NAT64 demande de modifier le routage et d'allouer des adresses. Le déploiement du NAT64 est transparent pour les hôtes mais nécessite des modifications au niveau de l'infrastructure de communication. Dans le cas du déploiement d'une passerelle applicative, nous sommes dans une situation inverse. Les modifications sont à apporter uniquement dans la configuration des hôtes: installation de la passerelle, mais aussi du client qui, dans certains cas, doit être configuré pour déléguer ses requêtes à la passerelle, à l'instar du navigateur web dont on configure la référence du proxy par exemple). Ainsi, il est possible, avec une passerelle applicative, d'avoir un déploiement progressif d'IPv6 dans le réseau, sans perturber les services en place. Dans le cadre d'une infrastructure de communication en production, cette caractéristique peut être appréciée.

Enfin, dans le cas d'un client IPv4 qui se connecte à des serveurs de l'Internet v6, la passerelle applicative est de nos jours la seule méthode d'interopérabilité. Mais il est vrai que ce scénario n'est pas encore d'actualité au vu de l'état du déploiement de l'Internet v6. Nous allons détailler, dans la suite de cette activité, les scénarios d'utilisation de ce dispositif dans le cas d'un client IPv6 avec un serveur IPv4.

Principe des passerelles applicatives

Les passerelles applicatives, ou ALG (*Application Level Gateway*), représentent le moyen le plus simple pour assurer une relation entre le monde IPv4 et le monde IPv6. Il s'agit de machines avec une double pile (cf. figure 1) configurées pour accéder aux deux versions du protocole. Les clients IPv6 émettent leurs requêtes vers la passerelle applicative comme s'ils s'adressaient directement au service. La passerelle interprète le contenu de ces requêtes pour

les retransmettre ensuite en IPv4 à destination du service concerné.

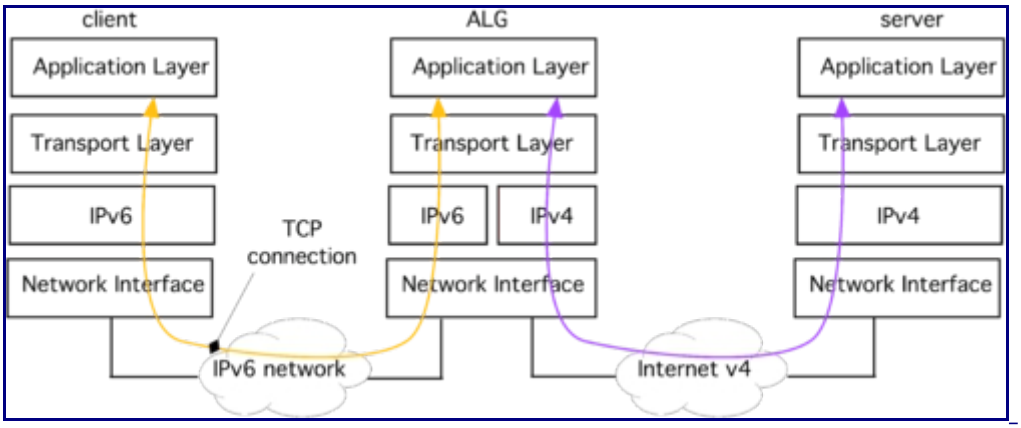


Figure 1: Communication par passerelle applicative.

Une ou plusieurs passerelles peuvent être installées en fonction des services rendus disponibles sur le réseau (par exemple: serveur d'impression, serveur de messagerie, web, etc.). Les machines clientes doivent être configurées pour adresser leurs requêtes applicatives à ces passerelles.

L'usage de ces techniques est très fréquent dans les réseaux privés pour communiquer avec l'extérieur. Tous les protocoles ne peuvent pas utiliser les passerelles applicatives. Certains protocoles ne sont pas prévus pour intégrer un relais intermédiaire (par exemple *telnet*). D'autres protocoles, par leur nature propriétaire, ne permettent pas le développement de passerelles par une tierce partie si celle-ci n'est pas disponible (comme par exemple *Skype*). Mais, comme la liste suivante l'indique, les ALG concernent des applications courantes qui représentent une proportion importante du trafic. Cela permet également d'alléger le travail d'autres mécanismes de transition qui sont plus complexes à mettre en œuvre. Les passerelles applicatives regroupent:

- les *proxies* et les caches web,
- les *spoolers* d'impression,
- les serveurs de courrier électronique,
- les serveurs DNS,
- ...

Cas du service Web

Il s'agit ici de faire communiquer des clients avec des services Web, client et serveur utilisant une version différente du protocole IP. La passerelle applicative utilisée dans ce cas est un relais HTTP qui va interpréter les requêtes des clients pour les retransmettre vers le serveur Web. Deux modèles de déploiement existent pour ce type de relais:

- le déploiement d'un serveur mandataire (*proxy*) dans le réseau des clients, leur permettant d'atteindre les serveurs extérieurs, dont ceux qui n'utilisent pas la même version du protocole IP,

- le déploiement d'un relais inverse (*reverse proxy*) dans le réseau du serveur, permettant d'accepter les requêtes des clients qui n'utilisent pas la même version du protocole IP que le serveur.

ALG placée du coté du client

Le relais HTTP est ici localisé dans le réseau des clients, généralement dans la DMZ du site ou sur le routeur domestique, comme le montre la figure 2. Les clients sont configurés pour utiliser cette passerelle en tant que serveur mandataire afin d'atteindre les services Web extérieurs. Ce type de déploiement est couramment utilisé pour sécuriser les clients d'accès Web vers des sites malveillants.

Afin de permettre l'interopérabilité entre les différentes versions du protocole IP, la passerelle est connectée et configurée sur un réseau "double pile". Si, par exemple, les clients sont sur un réseau seulement IPv6, l'adresse IPv6 de la passerelle leur est indiquée en tant que serveur mandataire. La passerelle recevra alors les requêtes HTTP de ces clients et les relaiera vers les services demandées en IPv4 ou en IPv6 selon le protocole utilisé par le serveur.

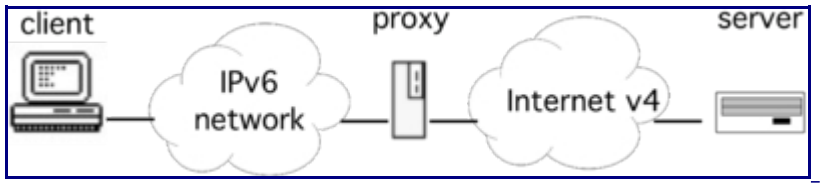


Figure 2: Exemple de passerelle applicative placée du coté client.

Le listing suivant donne un extrait de la configuration d'un serveur Apache pour que celui-ci serve de relais aux requêtes émises par des navigateurs. Aucune configuration n'est relative au protocole IPv6. Il suffit d'activer la fonction de proxy.

```
#cat /usr/local/etc/apache/httpd.conf
#
# Proxy Server directives. Uncomment the following lines to
# enable the proxy server:
#
<IfModule mod_proxy.c>
ProxyRequests On
<Directory proxy:*>
Order deny,allow
Allow from all
</Directory>
#
# Enable/disable the handling of HTTP/1.1 "Via:" headers.
# ("Full" adds the server ver.;"Block" removes all outgoing Via: headers)
# Set to one of: Off | On | Full | Block
#
ProxyVia On
</IfModule>
# End of proxy directives.
```

ALG placée du coté du service

La problématique ici à résoudre est de rendre un service Web accessible avec les 2 versions du

protocole IP alors que celui-ci n'en utilise qu'une seule. S'ajoute à cette problématique la contrainte opérationnelle du service: le fonctionnement du site Web sera-t-il perturbé par l'intégration d'IPv6? L'expérience utilisateur des visiteurs va-t-elle être impactée?

Pour rendre accessible un service Web en IPv6, la solution la plus simple consiste à activer la connectivité IPv6 sur le réseau où est connecté ce service, ainsi que sur la machine qui l'héberge. Mais cette solution pose un ensemble de problèmes opérationnels car l'infrastructure d'hébergement d'un site Web peut être assez complexe (système d'équilibrage de charge ou *load balancers* , cache, etc.). Une réelle étude du passage à IPv6 de cette infrastructure peut être nécessaire pour effectuer une transition pérenne. Le RFC 6589 s'intéresse à cette problématique et délivre un ensemble de conseils pour les hébergeurs qui veulent rendre leurs serveurs accessibles en IPv6.

Déploiement d'un relais inverse

Une solution moins coûteuse et plus rapide à mettre en oeuvre (mais avec bien sûr quelques limitations) consiste à déployer un relais inverse (*reverse-proxy*) proche du serveur, comme montré par la figure 3. Le rôle de ce relais est d'accepter les requêtes vers le service Web utilisant la version du protocole qui n'est pas encore déployée sur le serveur. Les clients envoient leur requête au relais de manière transparente, comme s'il s'agissait du service. Le relais se charge, pour le client, de transférer les requêtes vers le serveur et de recevoir sa réponse, en utilisant le protocole IP déployé sur le serveur.

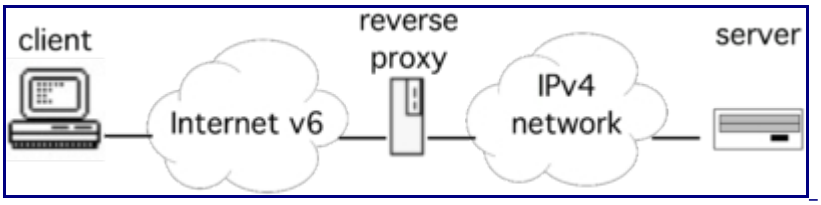


Figure 3: Exemple de passerelle applicative placée du coté serveur.

Dans la mise en oeuvre du relais inverse, une étape importante consiste en la configuration du DNS. En effet, l'adresse du relais doit être renseignée comme l'un des enregistrements pour le service concerné. Ainsi, par exemple, pour un service seulement accessible en IPv4, l'adresse IPv6 du relais sera renseignée comme enregistrement AAAA au même niveau que l'enregistrement A de l'adresse du serveur.

Le listing suivant donne un extrait de la configuration d'un relais inverse opéré par le logiciel nginx . La configuration consiste à indiquer le renvoi des requêtes Web reçues en IPv6 vers le serveur resté joignable en IPv4.

```
#cat  /etc/nginx/sites-available/default
...
location / {
        proxy_pass http://192.0.2.1/;
}
```

Dans le contexte initial, le service Web n'est accessible qu'en IPv4. L'adresse IPv4 du service (notée S4) est enregistrée dans le DNS. Celle-ci est récupérée par les clients à partir du nom du

service afin d'initier une connexion directe vers le serveur, comme montrée dans la figure 4.

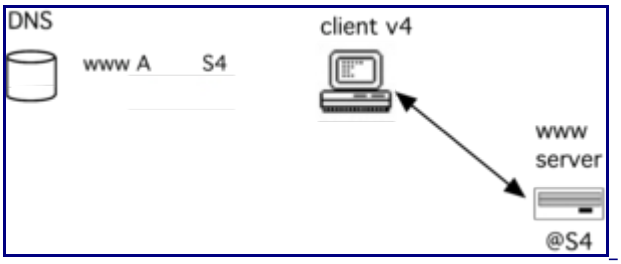


Figure 4: Accès direct pour les clients IPv4.

Le scénario d'intégration d'IPv6 par un relais inverse pour un service Web passe par 2 actions, comme représenté par la figure 5:

- la mise en place d'un relais inverse dans l'infrastructure du service, sur un réseau "double pile";
- l'enregistrement de l'adresse IPv6 du relais (notée S6) comme l'adresse IPv6 officielle du serveur.

Un client possédant une connectivité IPv6 et souhaitant consulter le service va résoudre le nom du service en 2 adresses: une IPv4 et une IPv6. La préférence à IPv6 du navigateur lui fera utiliser en priorité cette adresse. Sa requête se fera alors de manière transparente à destination du *reverse proxy* comme indiqué par la figure 5.

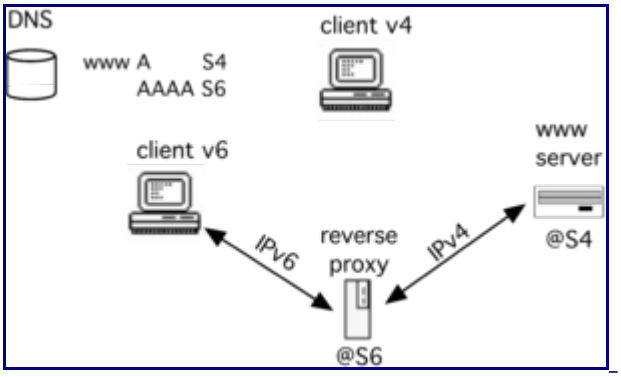


Figure 5: Accès par le relais inverse pour les clients IPv6.

Le relais inverse propose donc une solution simple pour assurer une interopérabilité de son service Web avec IPv6. Cependant, elle n'est pas adaptée à des sites à large audience. Même largement dimensionné, un unique relais ne pourrait pas absorber la portion IPv6 des requêtes, même si celle-ci est encore en dessous des 10%. De plus, le relais constitue un point de faiblesse unique (SPOF, *Single Point of Failure*) pouvant compromettre l'accès au service.

Utilisation d'un service d'hébergement ou de distribution des contenus

Pour ces sites à large audience, plusieurs solutions peuvent être envisagées pour permettre l'interopérabilité avec IPv6 [RFC 6883]:

- migrer son infrastructure d'hébergement en "double pile" (comme mentionné plus haut, cette solution est la plus complexe);

- faire appel à un service d'hébergement offrant une connectivité "double pile";
- continuer à héberger son service en IPv4, mais utiliser un réseau de distribution de contenus (CDN, *Content Delivery Network*) "double pile".

Les 2 dernières solutions permettent au responsable du service de déléguer la complexité de l'intégration et de la gestion d'IPv6 à un prestataire extérieur. Ces services sont aujourd'hui assez répandus. Les hébergeurs de sites Web offrent maintenant couramment un accès "double pile" aux services hébergés, que ce soit sur des offres de serveurs mutualisés ou dédiés. Toutes les prestations d'hébergement des acteurs majeurs en France que sont OVH, Gandi ou Online, intègrent IPv6 dans leurs offres.

Les réseaux de distribution de contenus (ou CDN) ont pour objectif de répliquer le contenu du service en différents points stratégiques du réseau, permettant aux utilisateurs d'accéder plus rapidement au service et à l'infrastructure du service d'être soulagée d'une partie du trafic. Les CDN peuvent, de plus, permettre l'interopérabilité avec IPv6 en jouant le même rôle que le relais inverse vu précédemment, avec bien sûr une infrastructure plus robuste. Des services de CDN comme Akamai, CloudFlare ou Cedexis permettent ainsi d'offrir des contenus en IPv6 alors que ceux-ci sont hébergés sur des services seulement IPv4.

Conclusion

Les passerelles applicatives offrent un moyen simple d'interopération entre des clients et des serveurs qui n'utilisent pas la même version du protocole IP. Parce qu'elles interprètent le contenu du paquet dans la couche d'application, elles sont transparentes pour l'infrastructure de communications (routeurs). Elles ne demandent pas de modifications au niveau du réseau. Cependant, les passerelles applicatives posent des contraintes qui limitent leur usage [1] , telles que:

- introduction d'un délai pour le traitement des paquets,
- difficultés à passer le facteur d'échelle, et possibilité de congestion,
- applications non conçues pour fonctionner avec un relais intermédiaire.

Passage à l'échelle

Le passage à l'échelle, dans ce contexte, signifie une croissance de la taille, soit en nombre de clients du service applicatif, soit en terme de volume de flux. La mise en place d'une passerelle applicative ajoute un relais protocolaire dans la chaîne de communication entre le client et le serveur applicatif. Bien que ce relais puisse être fonctionnel et transparent, la montée en charge peut poser problème. La capacité du relais étant finie et limitée, il peut introduire des défauts à partir d'un certain nombre de clients ou d'une certaine quantité de trafic.

En effet, des protocoles propriétaires, ainsi que certains protocoles assurant la confidentialité des communications peuvent rendre impossible la mise en oeuvre d'un tel dispositif (pour un protocole de sécurité, une telle passerelle pourrait s'apparenter à un "homme au milieu"). De plus, selon le protocole utilisé, la mise en oeuvre d'une telle passerelle peut s'avérer complexe. Par exemple, le protocole SIP nécessite une interprétation de l'ensemble de la signalisation. Enfin, une passerelle applicative n'est pas forcément le meilleur choix si le protocole applicatif

embarque des adresses IP.

Références bibliographiques

1. ↑ IPv6.com. (2008). Tech spotlight. ALG - Application Level Gateway

Pour aller plus loin

RFC et leur analyse par S. Bortzmeyer

- RFC 6144 Framework for IPv4/IPv6 Translation Analyse
- RFC 6384 An FTP Application Layer Gateway (ALG) for IPv6-to-IPv4 Translation
- RFC 6586 Experiences from an IPv6-Only Network Analyse
- RFC 6589 Considerations for Transitioning Content to IPv6 Analyse
- RFC 6883 IPv6 Guidance for Internet Content Providers and Application Service Providers Analyse
- RFC 7269 NAT64 Deployment Options and Experience Analyse

Conclusion

La croissance de l'Internet a rendu IPv4 obsolète. Le nouveau protocole IPv6 vise à retrouver le principe de "bout en bout". Ce principe fondateur de l'Internet a assuré son succès. C'est par ce principe que l'Internet est devenu une source d'innovation et le support de l'économie du numérique. La migration d'IPv4 vers IPv6 est bien plus qu'un simple changement de tuyau. C'est tout l'écosystème qui est appelé à évoluer. Aussi, la sensibilisation de tous les acteurs à la problématique de la migration est cruciale. Le déploiement d'IPv6 se conduit, comme un projet, avec une planification. Il touche tous les métiers du système d'information.

Le déploiement d'IPv6 doit se faire en tenant compte de l'existant et de manière progressive. IPv6 est appelé à coexister avec IPv4. Autrement dit, il est une évolution d'IPv4 et non le moyen de faire un Internet parallèle et disjoint de l'existant. Pour maintenir cette connectivité globale, IPv6 comporte des mécanismes transitoires pour qu'il puisse interopérer avec IPv4. Ces mécanismes sont maintenant connus. Ils sont responsables en grande partie de l'image de complexité que peut dégager le passage à IPv6. Cependant, ils ne sont pas tous à utiliser: il faut retenir celui qui permet de faire interopérer IPv6 avec son système de communication. Au cours de cette séquence, nous avons présenté 3 techniques d'intégration:

- la double pile, qui est la solution par excellence du déploiement progressif;
- le tunnel, pour interconnecter des îlots IPv6 par des liens virtuels en IPv6, établis sur des liaisons réelles en IPv4;
- la traduction, lorsque la double pile ne peut plus être utilisée du fait du manque d'adresses IPv4, ou pour rendre des services accessibles à IPv6 sans avoir à mettre à jour le serveur.

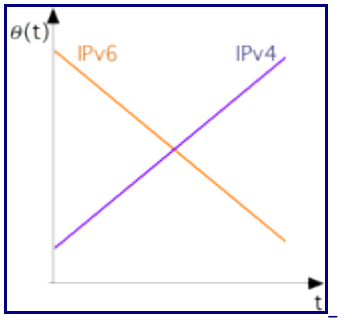


Figure 1: Évolution du coût opérationnel

L'usage de ces techniques est appelé à diminuer au fur et à mesure de l'extinction d'IPv4. Contrairement à IPv4, qui était partie d'une table rase, IPv6 doit tenir compte de l'existant, ce qui particularise et complexifie son déploiement initial. Mais, contrairement à IPv4, la connectivité IPv6 va devenir de plus en plus simple. L'évolution du coût opérationnel, autrement dit de la complexité, pour chacune des versions du protocole IP, peut se schématiser comme indiqué par la figure 1.

Bien qu'IPv6 existe depuis longtemps, le déploiement s'est accéléré ces dernières années en même temps que la pénurie d'adresses IPv4 est devenue plus marquée du fait de l'épuisement des adresses IPv4 disponibles. Aussi, IPv6 est devenu inévitable à court terme. Ce n'est pas une expérience de laboratoire et s'en préoccuper tardivement ne fait qu'augmenter la

complexité et le coût de son déploiement. L'objectif final du déploiement d'IPv6, c'est d'avoir IPv6 partout dans l'Internet et ainsi d'avoir des potentialités de croissance et d'innovation.

Comme, aujourd'hui, les réseaux IPv6, seuls ou déployés conjointement avec IPv4, deviennent de plus en plus courant, il est important d'avoir les bonnes pratiques de déploiement et d'administration qui émergent progressivement. Il est donc important de se tenir informé, de partager et d'adapter ses propres pratiques en fonction des expériences de chacun.

Pour en savoir plus

Des vidéos sur la transition:

- Transition mechanisms by RIPE
- Comment assurer une transition heureuse par S. Bortzmeyer (2011)
- 6DEPLOY-2 e-Learning and IPv6 in 5 minutes

Remerciements

Les auteurs souhaitent remercier Stéphane Bortzmeyer pour ses analyses de RFC sur IPv6 (http://www.bortzmeyer.org/) dont des extraits ont été utilisés pour ce cours.